Conditional Modeling and Conditional Inference

by

Lo-Bin Chang

B.A., National Taiwan University, Taiwan, 2000

M.S., National Taiwan University, Taiwan, 2004,

M.S., Brown University, RI, 2009

A Dissertation Submitted in Partial Fulfillment of the
Requirements for the Degree of Doctor of Philosophy
in the Division of Applied Mathematics at Brown University

Providence, Rhode Island

May 2010

This dissertation by Lo-Bin Chang is accepted in its present form
by the Division of Applied Mathematics as satisfying the
dissertation requirement for the degree of Doctor of Philosophy.

Date________________ ____
Stuart Geman, Ph.D., Advisor

Recommended to the Graduate Council

Date________________ ____
Elie Bienenstock, Ph.D., Reader

Date________________ ____
Basilis Gidas, Ph.D., Reader

Approved by the Graduate Council

Date________________ ____
Sheila Bonde, Dean of the Graduate School

The Vita of Lo-Bin Chang

Lo-Bin Chang was born in Taipei, Taiwan, on April 26, 1978.

In 1996, he started his undergraduate study at National Taiwan University, and he received his Bachelor of Science degree in Mathematics in 2000. After completing his undergraduate study, he served in the army for twenty-two months. Then, he entered graduate school at National Taiwan University, and he received his Master of Science degree in Applied Mathematics in 2004. After graduating from National Taiwan University, he worked for Polaris Financial Group and Academia Sinica for two years.

Lo-Bin Chang came to the United States in 2006 and studied at the School of Mathematics, Georgia Institute of Technology for one year. He then transferred to Brown University and attended the Ph.D. program in the Division of Applied Mathematics. He received a Master of Science degree in Applied Mathematics in 2009 and received his Doctorate of Philosophy in May 2010. This dissertation was defended on May 3, 2010.

Acknowledgments

I would like to thank all of the people who have helped me to complete my thesis.

Most of all, I want to extend my deepest gratitude to my advisor, Professor Stuart Geman, for his invaluable guidance, support, and encouragement throughout my graduate studies. His inspiring ideas and valuable suggestions shaped my thought and guided my career. This thesis would not have been possible without his persistent help.

I would also like to thank my other committee members, Professor Elie Bienenstock and Professor Basilis Gidas, for reading my dissertation and for giving me valuable comments.

I wish to acknowledge Prof. Chii-Ruey Hwang, Prof. Charles Lawrence, Prof. Matt Harrison, Doc. Wei Zhang and Luan Lin, who have been very helpful during my research and during my life in Brown. Also, I want to thank Deqing Sun, Oren Freifeld, Sergey Kushnarev and all of my friends in the Division of Applied Mathematics and in Brown Taiwanese Student Association for helping me in every aspect.

Last but certainly not least, I owe my thanks to my family. Thanks to my parents and my brother, who support me spiritually. Without their encouragement and understanding it would have been impossible for me to finish this work.

Contents

List of Figures

Part I

Introduction and Overview

Increasingly, scientists are faced with the task of modeling and analyzing extremely high-dimensional data, such as financial time series, multi-channel neurophysiological recordings, and large amounts of text and imagery available through the Internet. Despite the availability of large data sets developing tractable statistical models for complex data in high dimensions remains a challenge and oftentimes a bottleneck to successful applications. Traditional parametric methods are often inappropriate, either because the class of reasonable models is too big or because our understanding of the source is still too limited. Fully nonparametric methods are useful but often limited by sparseness and dimensionality. Here we propose to study a collection of semi-parametric approaches within the frameworks broadly known as conditional modeling and conditional inference ([53]).

Overview

In general, let X be a high-dimensional random variable and $S(x)$ be an arbitrary function on the range of X. The statistic S may be low-dimensional, e.g. a component of x or an expected value relative to its empirical distribution, or high-dimensional, e.g. the empirical distribution itself. Assuming (for illustration) that x is discrete, we factor its distribution, p_X, as $p_X(x) = p_X(x|S = s)p_S(s)$, where S refers to the random variable $S(X)$, p_S is its distribution, and the value of s is $S(x)$. The probabilities, $p_X(x|S = s)$ and $p_S(s)$, may or may not be specified in a parametric form ($p_X(x|S = s) = p_X(x|S = s; \theta)$, or $p_S(s) = p_S(s; \phi)$ in which case S is sufficient for ϕ). When S is low dimensional, our focus is on modeling or inference about p_S; when S is high dimensional, our focus is on $p_X(x|S = s)$. In some applications the goal is to "estimate S" itself, as in Part III.

As an example, consider the sequence of five-minute returns on a stock trading at price P_t, where $t = 0, 1, ...n$ indexes time in units of five minutes. The return at time t is $R_t = log(P_t/P_{t-1})$. Writing R for the sequence of n returns, and r for a generic value of R, let $S(r)$ be the empirical distribution of the observed returns. Under the Black-Scholes model, and its usual extensions to heavy-tailed distributions, $p_R(r|S = s)$ is exchangeable (invariant to permutations of the components of r). This opens the door to permutation tests for the validity of the model (see Chapter 2), and further, by focusing on *local* permutations (permutations ρ of $\{1, 2, \ldots, n\}$ for which $|\rho(k) - k|$ is small), to an exploration of the time scale of departures from the Black-Scholes model. A number of theories about market dynamics can be critically examined, and subjected to exact hypothesis tests, by introducing a function of r (test statistic) and comparing its distribution under permutation to its value at the observed sequence of returns. The distribution of S itself is much studied and much debated, but irrelevant to questions about the exchangeability (as in Black-Scholes) or *local exchangeability* (as in "stochastic volatility" models) of R. Thus the idea is to sidestep difficult modeling assumptions by focusing on a relevant conditional distribution. See Chapter 2 for some statistics that focus on large returns ("rare events") and their peculiar temporal structure.

As a second, quite different, example, consider the following problem from computer vision in which S is low-dimensional and the focus is on p_S rather than the conditional distribution given S. Let Y be a 20 by 50 (say) array of pixel intensities. As a step towards a fully generative model for images, the task is to develop a model for the distribution of Y given that it comes from a particular class of objects, say the right eye of a human face. Real images have complex dependency structures that are extremely hard to model, even for a modest-sized image patch with only 1,000 pixels. But for the purposes of recognition and interpretation, as well as image generation, a conditional model can be entirely adequate. Let T be a 20 by 50 template of an eye — a prototype image, perhaps of an individual's eye or a "representative" eye from an ethnic population. Later we will introduce mixtures over multiple templates, but for now consider the function $S(y) = corr(T, y)$, the normalized correlation of the image patch y with the template T. (So S is a statistic and T is a parameter.) One way to finesse the complex task of modeling the distribution for the appearance of right eyes, $p_Y(y)$, is to factor it into the distribution of S, which presumably favors high correlations, and the distribution of everything else about Y, which presumably includes characteristics of the texture and lighting of generic patches of real images.

Let p_Y^o be a "null" or "background" distribution of 20 by 50 image patches. Think of the distribution of a randomly chosen patch from a collection of images, such as the collection of images available on the Internet. There are many special properties of p_Y^o having to do with neighborhood relationships, spacial scaling, the appearance of shadows and specular reflections, and so on. We can "borrow" these special properties without actually modeling them by adopting a conditional eye-patch model: $p_Y(y) = p_Y^o(y|S = s)p_S(s)$, where p_S is either a parametric form (e.g. $p_S(s) \propto \exp(-\lambda(1 - s))$ on $s \in [-1, 1]$) or, being one-dimensional, is inferred from a training set of right eyes. It is easy to estimate λ, for which S is sufficient. But what about T, and therefore S itself? It turns out that the maximum likelihood estimator for λ *and for the template* T depends on p_Y^o only through p_S^o, the marginal background distribution of S, which is straightforward to estimate for any given T. The upshot is that T and λ can be estimated from data, and the resulting maximum-likelihood templates have proven to be extremely discriminating for detection and classification problems. See Chapter 6 for generalizations, in several directions, and related modeling and computational challenges.

This thesis will explore a variety of problems in conditional modeling and conditional inference, as well as two target application areas: market dynamics(Part II), computer vision(Part III).

Market Dynamics: Stock Prices and the Statistics of their Returns

Part II is about the statistical analysis of stock returns. We will introduce new tools for exploring the temporal dependency between returns and we will give strong statistical evidence against standard models and many of their variations. We will examine stochastic volatility and other modifications of Black-Scholes, and find that these models would require very frequent high-amplitude departures from homogeneity to fit the data. We will focus on the temporal structure of large excursions in the return process, but the methods are suitable for studying many aspects of the market and its models, including "momentum," "mean reversion," regime changes and change points, and the viability of models that would purport to explain volatility clustering. In general, we will advocate for permutation and other combinatorial statistical approaches that robustly and efficiently exploit symmetries shared by large classes of models, supporting exact hypothesis tests as well as exploratory data analysis. These so-called nonparametric methods complement the more typical approaches founded on parameter estimation, as might be used to fit returns within the class of infinitely divisible laws, or to fit volatility trajectories with multiple regression and auto-regression parameters.

Our approach was motivated by the recent experiments of C.-R. Hwang and colleagues (Chang et al. [17]), revealing a surprising invariant in the timing of large excursions of returns in stock prices. Consider the sequence of thirty-minute returns derived from a year of transactions of IBM stock. Define an excursion ("rare event") as a return that falls either below the tenth percentile or above the ninetieth percentile of this population of thirty-minute returns. (There is nothing particularly special about the tenth and ninetieth percentiles, and a similar picture emerges from experiments with other thresholds.) If we break ties by small random perturbations, then exactly twenty percent of the returns are "rare." The corresponding sequence of zeros (for returns in the middle eighty percentiles) and ones (for returns in either of the two ten-percentile tails) has eighty percent zeros and twenty percent ones and, among other things, we could study the waiting-time distribution between the excursions, meaning the distribution on the number of zeros between two ones. It is perhaps not surprising that this distribution departs from the distribution predicted by the Black-Scholes model, including the usual variants that replace the increments of Brownian motion by infinitely divisible laws (heavy-tailed processes, jump diffusions, and so-on). But it is surprising that this distribution is nearly invariant to time scale (e.g. two thousand thirty-minute returns versus two thousand thirty-second returns), to year (e.g. 2000 versus 2005), and to stock (e.g. IBM versus Citigroup).

In Chapter 1 we will review the discoveries of Hwang et al., and discuss their connections to the classical geometric Brownian motion and related models. In Chapter 2 we will introduce combinatorial methods for exact hypothesis tests of a large class of pricing models. We will conclude, as others have, that were we to stay within the framework of random walks then these "standard models" would fail to account for evident fluctuations in the volatility process. We will examine fractional Brownian motion as an alternative to the independent-increment processes, but find that the sample paths are unconvincing, at least in the necessary parameter range. We will give evidence for a long "memory" among large excursions, as measured by the order of a Markov process constructed to fit the data. In Chapter 3 we will take a closer look at stochastic volatility models and explore the time scale of volatility fluctuations. Using a sequence of thirty-second returns from four days of IBM data (in 2005), we will give evidence that a volatility process working within the random-walk framework would require rapid and large-amplitude fluctuations, possibly as big as a 40% average change in σ every four minutes. We will search, unsuccessfully, for likely sources and correlates of stochastic volatility and volatility clustering, including stochastic time changes pegged to trading volume or other measures of market activity ("market time"). We will conclude, in Chapter 4, with a summary, a discussion, and some challenges.

Computer Vision

Generative part-based models have become more popular as an approach to detection and recognition in computer vision. The core idea is to decompose the target object into simpler parts, and build a Bayesian generative model. Thus, the models are composed of two components: the prior distribution on the hierarchical image interpretations which tells us how to combine parts appropriately, and the conditional distribution on image pixel intensities given an interpretation which is connected to our image data.

Difficult tasks of detection and recognition in images and video may require more a priori architecture than is traditionally used to support machine-vision applications. By almost any measure, a substantial gap persists between biological and machine performance, despite the existence of image and video training sets that are far larger than any human being sees in a lifetime and the existence of theorems from mathematical statistics that ensure optimal classification rates in the large-sample-size limit.

We and others have argued that scalable unconstrained vision systems will require specific architectures that accommodate detailed models of part/whole hierarchies (cf. [6], [26], [36], [11], [64], [31]). There is plenty of evidence that these organizational principles are in force in the mammalian visual system (e.g. [62]), and a growing number of vision scientists are achieving state-of-the-art performance with hierarchical architectures (e.g. [5], [52], [70], [41], [69]). Within the hierarchical framework, we have been studying fully generative models, meaning probabilistic models that are specified to the pixel level. These models provide a foundation for a Bayesian approach to image interpretation.

In Chapter 5, we will introduce a probabilistic framework for modeling hierarchy, reusability, and conditional data models. We will focus on the first component of the Bayesian composition model: a prior probability distribution on the hierarchical image interpretation. This will be a non-Markovian distribution ("context-sensitive grammar") on hierarchical structures. A sufficient condition for the existence of non-Markovian distributions will be provided, and the convergence of an iterative perturbation scheme for achieving these desired distributions will be proven. In addition, we will investigate coordinate systems including absolute coordinate distributions and relative coordinate distributions. We will prove some invariance properties of the

"r-cube law," which provides an appropriate distribution for scales. A closed form for the joint distribution of relative coordinates will be obtained by an approximation method. Chapter 6 will focus on the second component of the Bayesian composition model: a conditional distribution on pixel intensities given an interpretation of a hierarchical structure. A conditional modeling trick will be studied in order to finesse the complexity of the high dimension data. We will propose an approximate sampling method of the generative model, based on the choices of background image patches. Chapter 7 will propose a probabilistic combinatorial formulation of the image analysis problem and examine optimal recognition performance from the Neymann-Pearson point of view. Through an asymptotic analysis and basic large-deviation theory, we will argue that essentially optimal performance can be attained through a computationally feasible sequential decision analysis. Chapter 8 will include experiments with X-ray image classification using composition system. The conclusion and possible future directions will be in Chapter 9.

Part II

Stock Prices and the Statistics of their Returns

Chapter 1

Waiting times between large excursions: a surprising invariant

Returns that follow the Black-Scholes model are exchangeable, whether or not the increments are normal. In this chapter, we will prove a limiting property of a exchangeable sequence, and we will reveal a surprising invariant in the timing of large excursions of returns in stock prices that is a cowork with C.-R. Hwang and colleagues at Academia Sinica, in the summer of 2007. Consider the sequence of thirty-minute returns derived from a year of transactions of IBM stock. Define an excursion ("rare event") as a return that falls either below the tenth percentile or above the ninetieth percentile of this population of thirty-minute returns. (There is nothing particularly special about the tenth and ninetieth percentiles, and a similar picture emerges from experiments with other thresholds.) If we break ties by small random perturbations, then exactly twenty percent of the returns are "rare." The corresponding sequence of zeros (for returns in the middle eighty percentiles) and ones (for returns in either of the two ten-percentile tails) has eighty percent zeros and twenty percent ones and, among other things, we could study the waiting-time distribution between the excursions, meaning the distribution on the number of zeros between two ones. It is perhaps not surprising that this distribution departs from the distribution predicted by the Black-Scholes model, including the usual variants that replace the increments of Brownian motion by infinitely divisible laws (heavy-tailed processes, jump diffusions, and so-on). But it is surprising that this distribution is nearly invariant to time scale (e.g. two thousand thirty-minute returns versus two thousand thirty-second returns), to year (e.g. 2000 versus 2005), and to stock (e.g. IBM versus Citigroup).

1.1 Waiting Times Between Excursions

There were 252 trading days in 2005. The traded prices of IBM stock (s_n, $n = 1, 2, \ldots, 3024$) at every 30-minute interval from 10:00AM to 3:30PM (twelve prices each day), throughout the 252 days, are plotted in Figure 1.1, Panel A.[1] The opening (9:30) and closing (16:00) prices are not included. The corresponding intra-day returns, $r_n \doteq \log \frac{s_{n+1}}{s_n}$, $n = 1, 2, \ldots, 2772$ (eleven returns per day) are plotted in Panel B.

We declare a return "rare" if it is rare relative to the interval of study, in this case the calender year 2005. We might, for instance, choose to study the largest and smallest returns in the interval, or the largest 10% and smallest 10%. Panel C shows the 2005 intra-day returns with the tenth and ninetieth percentiles superimposed. More generally, and precisely, given any fractions $f, g \in [0, 1]$ (e.g. .1 and .9), define

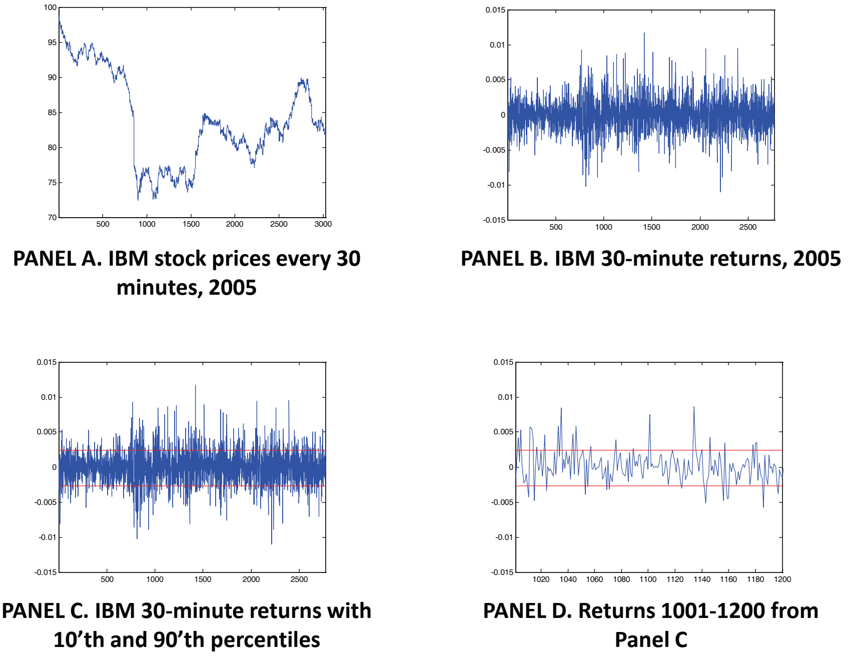
$$l_f \quad = l_f(r_1, \ldots, r_N) \quad = \inf\{r : \#\{n : r_n \leq r, 1 \leq n \leq N\} \geq f \cdot N\} \tag{1.1}$$

$$u_g \quad = u_g(r_1, \ldots, r_N) \quad = \sup\{r : \#\{n : r_n \geq r, 1 \leq n \leq N\} \geq (1 - g) \cdot N\} \tag{1.2}$$

where, presently, N =2,772. The lower and upper lines in Panel C are $l_{.1}$ and $u_{.9}$, respectively. Panel D is a magnified view, covering $r_{1001}, \ldots, r_{1200}$, but with $l_{.1}$ and $u_{.9}$ still figured as in equations (1.1) and (1.2) from the entire set of 2,772 returns.[2]

[1] The price at a specified time is defined to be the price at the most recent trade.

[2] To help eliminate discrete-price artifacts, and possible confounding effects from "micro-



PANEL A. IBM stock prices every 30 minutes, 2005

PANEL B. IBM 30-minute returns, 2005

PANEL C. IBM 30-minute returns with 10'th and 90'th percentiles

PANEL D. Returns 1001-1200 from Panel C

Figure 1.1: **Returns, percentiles, and the excursion process. A.** IBM stock prices, every 30 minutes, during the 252 trading days in 2005. The opening (9:30) and closing (16:00) prices are excluded, leaving 12 prices per day (10:00,10:30,. . .,15:30). **B.** Intra-day 30-minute returns for the prices displayed in A. There are 252×11=2,772 data points. **C.** Returns, with the 10'th and 90'th percentiles superimposed. **D.** Zoomed portion of C with 200 returns. The "excursion process" is the discrete time zero-one process that signals (with ones) returns above or below the selected percentiles.

The *excursion process* is the zero-one process that signals large returns, meaning returns that either fall below l_f or above u_g:

$$z_n = 1_{r_n \leq l_f \text{ or } r_n \geq u_g}$$

In the situation depicted in Figure 1.1C, $f = .1$ and $g = .9$ and hence $z_n = 1$ for at least 20% of $n \in \{1, 2, \ldots, 2772\}$. Obviously, many generalizations are possible, involving indicators of single-tale excursions (e.g. $f = 0$, $g = .9$ or $f = .1$, $g = 1$) or many-valued excursion processes (e.g. z_n is one if $r_n \leq l_f$, two if $r_n \geq u_g$, and

structure," prices are first perturbed, independently, by a random amount chosen uniformly from between ±\$.005.

zero otherwise). Or we could be more selective by choosing a smaller fraction f and a larger fraction g, and thereby move in the direction of truly rare events. (Though there is likely to be a tradeoff between the magnitude of the excursions and the statistical power of the methods that we will introduce for studying their timing.) We will stick to the special case $f = .1$ and $g = .9$, but a similar exploration could be made of these other excursion processes.

1.2 The Role of the Geometric Distribution Excursions

Suppose, for the time being, that stock prices are a geometric Brownian motion, as in the "standard model." If w_t is Brownian motion and S_t is the stock price at time t, then

$$dS_t = \mu S_t dt + \sigma S_t dw_t \tag{1.3}$$

$t \in [0, \infty)$. One implication is that for any unit of time δ the returns

$$R_n \doteq \log \frac{S_{n\delta}}{S_{(n-1)\delta}} \qquad n = 1, 2, \ldots \tag{1.4}$$

(or, alternatively, $R_n \doteq \frac{S_{n\delta} - S_{(n-1)\delta}}{S_{(n-1)\delta}}$) would be independent and identically distributed (iid). It is often observed that the tails of the empirical distribution of returns of actual stocks are too heavy to be consistent with the increments of Brownian motion, as in (1.3), suggesting variations on the standard model in which w_t is replaced, more generally, by a Lévy process (e.g. "jump diffusions" and other heavy-tailed increment processes). These variations (henceforth the "standard models") also produce iid returns.

Any iid sequence of random variables, $R_1, \ldots, R_N$, is *exchangeable*, meaning that the distribution of $R_1, \ldots, R_N$ is invariant to permutation: if $\mathcal{L}(\cdot)$ denotes probability distribution (or "law"), then $\mathcal{L}(R_{\rho(1)}, R_{\rho(2)}, \ldots, R_{\rho(N)}) = \mathcal{L}(R_1, R_2, \ldots, R_N)$ for any permutation ρ of $1, 2, \ldots, N$. Since $l_f(r_{\rho(1)}, \ldots, r_{\rho(N)}) = l_f(r_1, \ldots, r_N)$ and $u_g(r_{\rho(1)}, \ldots, r_{\rho(N)}) = u_g(r_1, \ldots, r_N)$, as is evident from equations (1.1) and (1.2), it follows that the excursion process

$$Z_n = 1_{R_n \leq l_f(R_1,\ldots,R_N) \text{ or } R_n \geq u_g(R_1,\ldots,R_N)} \tag{1.5}$$

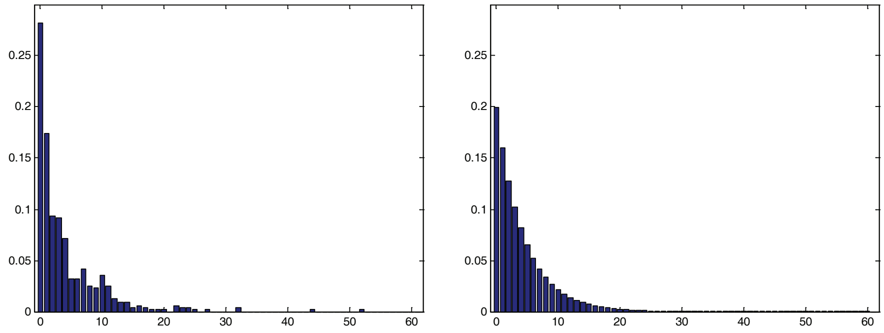
is also exchangeable (though not iid). Equivalently, if $N_1 = \#\{n : Z_n = 1\}$, then under the standard models each of the $\binom{N}{N_1} = \frac{N!}{N_1!(N-N_1)!}$ locations for excursions is equally likely. What's more, these remarks are equally valid for any time unit δ (30 seconds, 60 seconds, 30 minutes, etc.) and any stock, S.

One way to examine this class of pricing models is to examine how closely the empirical excursion process resembles an exchangeable sequence. Consider the particular case $f = .1$ and $g = .9$, and consider that there are then very nearly $N_1 = .2 \cdot N$ ones in the sequence $Z_1, \ldots, Z_N$. Since all arrangements of the N_1 ones are assumed to be equally likely, the probability that a one (an excursion) is immediately followed by another one (a second excursion) should be $\frac{N_1-1}{N-1}$, or very nearly .2 when N is large. But examination of stock data invariably points to a substantially higher probability, typically in the neighborhood of .28. This "volatility clustering" is remarkably constant across both time scale and stock.

Taking this further, we studied the empirical distribution on waiting times between excursions (between successive ones) in the excursion process $Z_1, \ldots, Z_N$. Figure 1.2, Panel A, shows the relative frequencies of the numbers of zeros between successive ones in the particular excursion process $z_1, \ldots, z_N$ computed for the 2005 30-minute IBM returns. Thus about 28% of the excursions were followed immediately by another excursion, about 17% were followed by a single zero before the next excursion, about 9% were followed by two zeros before the next excursion, and so on.[3]

For comparison, the waiting-time distribution for a Bernoulli, $p = .2$, process is shown in Panel B. In the case of the Bernoulli process the probability of w zeros

[3] For computational convenience, the sequence $z_1, \ldots, z_N$ is concatenated on the left and right by a single excursion (single one), so there are $N_1 + 1$ waiting times that contribute to the empirical distribution. In all of the studies N_1 is large and the padding is of no consequence.



PANEL A. Distribution of waiting times between excursions of IBM 30-minutre returns, 2005

PANEL B. Geometric waiting-time distribution

Figure 1.2: **Waiting times between excursions. A.** Empirical distribution of the number of zeros between two consecutive ones of the process $z_1, \ldots, z_{2772}$. **B.** Exact waiting time distribution for the Bernoulli process with parameter .2. If returns were exchangeable (as in the "standard model"), then A would approximate B.

between two successive ones is $.2(.8)^w$, i.e. the geometric, $p = .2$, distribution. It turns out that any exchangeable sequence of zeros and ones, with $.2 \cdot N$ ones, has a waiting-time distribution that approaches the geometric, $p = .2$, distribution as $N \to \infty$ (see the Proposition bellow). Hence under the standard models the histogram in Figure 1.2, Panel A, would approach the distribution in Panel B, as $N \to \infty$, regardless of the time interval, δ, or the particular stock, S.

Now, Let us prove the above property for standard model: as the size of the data set gets large, the empirical distribution of waiting time converges to the geometric distribution whenever the observations are a realization of any Exchangeable joint distribution. A theorem by Diaconis and Freedman [21] makes for an easy proof of the convergence. See Chang et al. [18] for more on this topic, including other forms of convergence. Following the notation in the text, let $p = f + (1 - g)$ be the fraction of ones in the excursion process, and let $\hat{P}_W(w)$, $w = 0, 1, \ldots$, be the empirical ("waiting-time") distribution of the number of zeros between two ones. Let

P be the uniform probability measure on $\{(z_1, \ldots, z_N) \in \{0,1\}^N : \sum_{k=1}^N z_k = \lfloor pN \rfloor\}$, where $\lfloor x \rfloor$ is the largest integer less than or equal to x. ($P = P_N$ uniform because $Z_1, \ldots, Z_N$ exchangeable.)

Proposition. Under P,

$$\sup_{w \geq 0} |\hat{P}_W(w) - p(1-p)^w| \to 0$$

in probability, as $N \to \infty$.

Proof. Since, for every N, $\hat{P}_W(w)$ is a probability measure, it is enough to show that $E_P|\hat{P}_W(w) - p(1-p)^w| \to 0$ for every $w \in \{0, 1, \ldots\}$, where E_P is expectation with respect to P.

Partition $z_1, \ldots, z_N$ into successive sequences of length $M_N \doteq \lfloor \sqrt{N} \rfloor$:

$$I_k = (z_{M_N(k-1)+1}, \ldots, z_{M_N k}) \quad k = 1, 2, \ldots, M_N$$

(The left-over z's, $(z_{M_N^2+1}, \ldots, z_N)$, will be of no consequence.) Let $\hat{P}_W^{I_k}(w)$ be the empirical waiting-time distribution for the sequence $z_{M_N(k-1)+1}, \ldots, z_{M_N k}$ (the $k'th$ interval), and let N_k be the number of ones in the $k'th$ interval. Then

$$\begin{aligned} \hat{P}_W(w) &= \frac{1}{pN} \sum_{k=1}^{M_N} \hat{P}_W^{I_k}(w) N_k + O(\frac{1}{M_N}) \\ &= \frac{1}{M_N} \sum_{k=1}^{M_N} \hat{P}_W^{I_k}(w) \frac{N_k}{pM_N} + O(\frac{1}{M_N}) \end{aligned}$$

Fix $w \in \{0, 1, \ldots\}$.

$$\begin{aligned} E_P|\hat{P}_W(w) - p(1-p)^w| &\leq \frac{1}{M_N} \sum_{k=1}^{M_N} E_P|\hat{P}_W^{I_k}(w) \frac{N_k}{pM_N} - p(1-p)^w| + O(\frac{1}{M_N}) \\ &= E_P|\hat{P}_W^{I_k}(w) \frac{N_1}{pM_N} - p(1-p)^w| + O(\frac{1}{M_N}) \end{aligned}$$

Assume $p > 0$ (the result is trivial if $p = 0$). Note that $\frac{N_1}{pM_N} \leq \frac{1}{p}$ (since $N_1 \leq M_N$) and, of course, $|\hat{P}_W^{I_k}(w)| \leq 1$. Consequently, it is enough to show that $\hat{P}_W^{I_k}(w) \to p(1-p)^w$ and $\frac{N_1}{M_N} \to p$, both in probability (wrt P).

Let Q be the Bernoulli measure on binary sequences, with parameter $\tilde{p} = \frac{\lfloor pN \rfloor}{N}$ (i.e. $Q\{z_k = 1\} = \tilde{p}$). Obviously, $\tilde{p} \to p$. For $n \leq N$, let P_n and Q_n be the marginal measures on $(z_1, \ldots, z_n)$, corresponding to P and Q, respectively. By Theorem 4 of Diaconis & Freedman (1980),

$$||P_n - Q_n||_{var} \leq \frac{4n}{N} \tag{1.6}$$

where $|| \cdot ||_{var}$ is the variational distance. If $A_N = \{(z_1, \ldots, z_{M_N}) : |\frac{N_1}{M_N} - p| > \epsilon\}$, then $Q_{M_N}(A_N) \to 0$ and hence, in light of (1.6), $P_{M_N}(A_N) \to 0$. As for $\hat{P}_W^{I_1}(w)$, the reasoning is the same but with A_N replaced by $B_N = \{(z_1, \ldots, z_{M_N}) : |\hat{P}_W^{I_1}(w) - p(1-p)^w| > \epsilon\}$.

1.3 Invariance in the Distribution of Waiting Times Between Excursions

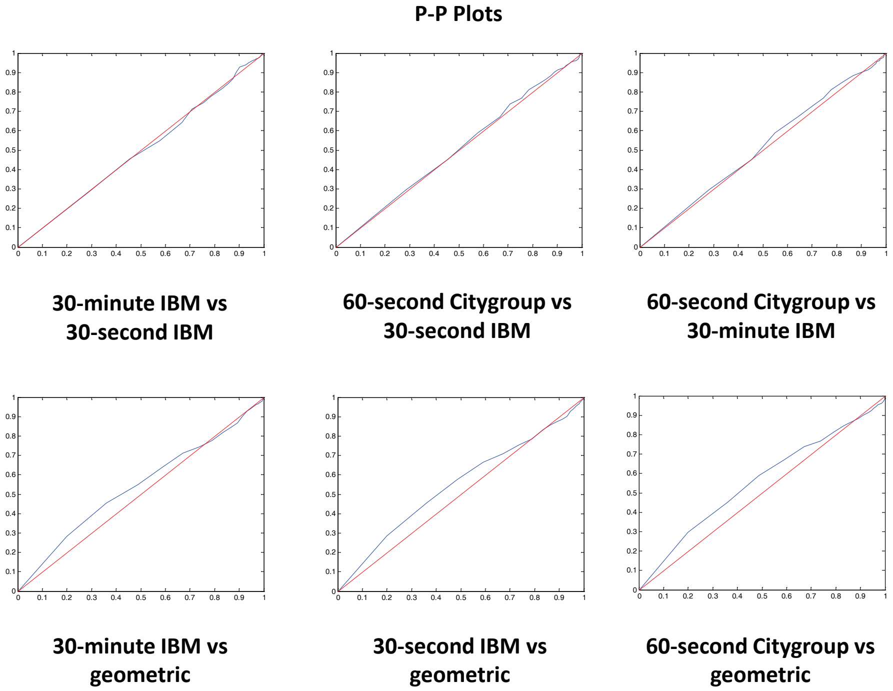
The standard models therefore imply an invariance in the distribution of waiting times between excursions: for all intervals and all stocks the waiting-time distribution approaches the geometric distribution as the number of returns grows. The parameter of the geometric is determined by the excursion thresholds as defined by equations (1.1) and (1.2) through the fractions f and g; thus p is simply $f + (1 - g)$. But the geometric is a reliably poor fit, as already observed by comparing the empirical probability of zero waiting time between excursions (about .28) to the predicted (geometric) probability (.2) in the case of 30-minute IBM returns. Still, this does not preclude the possibility that some other distribution plays the role of the geometric distribution as a universal limit of empirical waiting times. The notion may seem far fetched, but the evidence for an approximate invariant of this kind is pretty strong. Hwang et al. have studied the data systematically and extensively. Figure 1.3 is a snapshot, involving two stocks and two time scales. The P-P plots in the top row demonstrate that the waiting-time distributions for excursions of the 2005

IBM 30-minute returns, four days of 2005 IBM 30-second returns (3,040 returns[4]), and eight days of early 2008 Citigroup 60-second returns (also 3,040 returns[5]) are all quite similar. By contrast, as seen in the bottom-row plots, the geometric distribution is a poor fit. The actual histograms (as in Figure 1.2) show that, as compared to the geometric distribution, excursions in real stocks come in clusters. There are too many short waits and therefore (by virtue of the definition of excursions) also too many long waits. The clustering of volatility is not surprising and not new. But the near invariance to scale, stock, and even era (Hwang et al. have studied decades of returns) of distributions of waiting times between large excursions calls for some investigation.[6]

[4] Computed from prices at every 30 seconds, beginning each day with 9:35 AM EST and ending with 3:55 PM EST, over the four trading days January 4 through January 7

[5] Also computed from daily prices starting at 9:35 AM and ending at 3:55 PM, but for eight days in 2008—the week of March 24 and the first three days, March 31 through April 2, of the following week.

[6] Hwang (personal communication) reports that the similarities across scale and stock were well maintained in the second half of 2008 and the early months of 2009 but, interestingly, these returns established a somewhat different waiting-time invariant.



30-minute IBM vs 30-second IBM

60-second Citygroup vs 30-second IBM

60-second Citygroup vs 30-minute IBM

30-minute IBM vs geometric

30-second IBM vs geometric

60-second Citygroup vs geometric

Figure 1.3: **Invariance.** P-P plots of pairs of excursion waiting-time distributions. **Top row:** Comparison of the empirical distributions of waiting times between excursions for IBM 30-minute returns versus 30-second returns, Citigroup 60-second returns versus IBM 30-second returns, Citigroup 60-second returns versus IBM 30-minute returns. **Bottom row:** Comparison of each of the three empirical waiting-time distributions (IBM 30 minute, IBM 30 second, Citigroup 60 second) to the geometric distribution. Waiting-time distributions for excursions of stock returns are surprisingly invariant to the particular stock and the particular time scale.

Chapter 2

The peculiar statistics of large returns

We turn now to the statistical study of the timing of excursions in the returns of stock prices. We will develop exact hypothesis tests of the standard models by exploiting the presumed exchangeability of the returns. These tests are of the combinatorial type, which are often highly efficient in the sense that they sacrifice little statistical power while making a minimum of statistical assumptions (cf. Lehmann [47], Hollander & Wolfe [38]). Given the evidence already presented, it can not come as a surprise that we will reject the standard models at high significance. But these methods serve a second purpose, possibly more useful. Applied as tools for exploratory analysis, we will examine questions related to the structure of real stock returns, such as the adequacy of models based on fractional Brownian motion, the apparent window of dependency among returns, the time scale of the evident fluctuations in volatility, and the relationship, if any, to fluctuations in the number and volume of trades.

2.1 Conditional Inference

The idea behind conditional inference is simple enough: if $R_1, \ldots, R_N$ is an exchangeable sequence of random variables (in our case, returns as in equation (1.4)), then $R_1, \ldots, R_N$ should be indistinguishable from a random permutation, say $R_{\rho(1)}, \ldots, R_{\rho(N)}$, where ρ is chosen from the uniform distribution on the set of all $N!$ permutations. In fact, given M such permutations, $\{\rho_m\}$ $1 \leq m \leq M$, the entire set of $M+1$ sequences, $R_1, \ldots, R_N$ together with $\{(R_{\rho_m(1)}, \ldots, R_{\rho_m(N)})\}$, $m = 1, 2, \ldots, M$, should be indistinguishable. This can be tested, using a single observed sequence of returns $R_1 = r_1, \ldots, R_N = r_N$ and any scalar function of the data ("statistic") $H = H(r_1, \ldots, r_N)$, by examining the corresponding set of $M+1$ values $H_0 \doteq H(r_1, \ldots, r_N)$ and $H_m \doteq H(r_{\rho_m(1)}, \ldots, r_{\rho_m(N)})$, $1 \leq m \leq M$. H_0 should look like the rest of the population, and in fact the placement of H_0 among these $M+1$ values (say smaller than the vast majority) turns immediately into a p-value for the "null hypothesis" of exchangeability.

There are many variations on the theme. One is to focus on permutations that exchange returns locally, rather than arbitrarily, in order to explore the time scale of volatility clustering; see Chapter 3. Another is to examine Markov models, of some order (memory), by restricting permutations to preserve certain marginal statistics; see Section 2.4. In these examples it is sometimes useful to think of the permutation of returns in terms of the underlying sequence of stock prices, $s_0, \ldots, s_N$. Since the sum of all returns is obviously preserved under permutation, the process of sampling permutations of returns is equivalent to the process of sampling stock trajectories, all of which start at s_0 and end at s_N, but take their own paths via their own orderings of the intervening steps. The proposed approach to inference, then, amounts to examining a stock's trajectory (through a statistic, H) and comparing it to a population of surrogate trajectories presumed, by hypothesis, to have the same likelihood. In balance, the surrogates might look more concave or more convex, more like momentum or more mean-reverting, or have more or less volatility clustering, than the original.

Our focus will be on the waiting-time distribution between excursions of the return process, as defined in Chapter 1, and this will inform our choice of statistic H. But for now we will stay with the more general picture outlined above, and fill in some of the formal details.

Given any sequence of returns, $v_1, \ldots, v_N$, let $\mathcal{S}(v_1, \ldots, v_N)$ be the set of all return sequences that arise out of permutations of the original sequence:

$$\mathcal{S}(v_1, \ldots, v_N) = \{(v_{\rho(1)}, \ldots, v_{\rho(N)}) : \rho \text{ is a permutation of } 1, 2, \ldots, N\}$$

If, for some class of models (e.g. the standard models), the return process is exchangeable, then conditioned on $(R_1, \ldots, R_N) \in \mathcal{S}(v_1, \ldots, v_N)$ any sequence of returns in $\mathcal{S}$ is equally likely—i.e. $(R_1, \ldots, R_N)$ is uniformly distributed on $\mathcal{S}$. Now consider M permutations, chosen, as above, to be uniform on the set of all permutations and independent of $(R_1, \ldots, R_N)$. Observe that, while still conditioning on $(R_1, \ldots, R_N)$ $\in \mathcal{S}(v_1, \ldots, v_N)$, the $M + 1$ sequences

$$(R_1, \ldots, R_N) \cup \{(R_{\rho_m(1)}, \ldots, R_{\rho_m(N)})\}_{m=1}^{M}$$

are independent and identically distributed (specifically, iid with uniform distributions on $\mathcal{S}$). If $H_0 \doteq H(R_1, \ldots, R_N)$ and $H_m \doteq H(R_{\rho_m(1)}, \ldots, R_{\rho_m(N)})$, $1 \leq m \leq M$, then the $M + 1$ random variables, $H_0, H_1, \ldots, H_M$, are also (conditional) iid.

The idea behind conditional inference is to study the placement of H_0 among the $M + 1$ values $H_0, H_1, \ldots, H_M$. With this in mind, define

$$\mathcal{O} = \mathcal{O}(H_0; H_1, \ldots, H_M) = \#\{k \in \{0, 1, \ldots, M\} : H_k \leq H_0\}$$

Then, in light of the remarks above,

$$Pr\{\mathcal{O} \leq m | (R_1, \ldots, R_N) \in \mathcal{S}(v_1, \ldots, v_N)\} \leq \frac{m+1}{M+1}$$

(Think of ordering, smallest to largest, the $M+1$ numbers $H_0, H_1, \ldots, H_M$. H_0 could equally well occupy any of the $M + 1$ positions. If it were not for the possibility of ties—inevitable with only a finite number of permutations—then the probability would be exactly $\frac{m+1}{M+1}$.)

The last step in the argument is to recognize that the bound, $\frac{m+1}{M+1}$, is independent of $v_1, \ldots, v_N$ and hence still holds at $v_k = R_k$, $k = 1, 2, \ldots, N$. We have, therefore, a left-tail test of exchangeability with p-value $\frac{\mathcal{O}+1}{M+1}$, derived by simply ordering $H(r_1,\ldots,r_N)$, $H(r_{\rho_1(1)},\ldots,r_{\rho_1(N)})$, ..., $H(r_{\rho_M(1)},\ldots,r_{\rho_M(N)})$ and computing $\mathcal{O}$. As usual, depending on H, and on the alternative hypothesis being entertained, the right tail ($\mathcal{O} \geq m$) or both tails, might be more appropriate.

Before moving on, specifically, to the timing of excursions, we illustrate the approach with a simple example: Consider the 3,040 30-second returns derived from four-days of IBM prices in 2005 (see Section 1.3). One measure of correlation between successive returns is the (normalized) total variation of the return process:

$$H(r_1,\ldots,r_N) = \frac{1}{3039}\sum_{k=2}^{3040} |r_k - r_{k-1}|$$

which turns out to be, approximately, $3.98 \cdot 10^{-4}$ for this data set. Small values of H go with stock trajectories that have sustained trends; large values belong to trajectories that have more high-frequency fluctuations. Figure 2.1 shows the histogram of values of $H(r_{\rho_m(1)},\ldots,r_{\rho_m(N)})$ derived from 5,000 permutations ($m = 1,\ldots,5000$) of $\{1,\ldots,3040\}$, with the original value superimposed. It is very unlikely that the observed total variation is a member of the population represented by the histogram; we reject the hypothesis that the returns *from this sequence of prices* are exchangeable, in favor of the hypothesis that one return predicts the next return, at p-value of approximately .0004. Apparently, returns tend to cluster into similar values, at least slightly. This observation has nothing to do with the starting and ending prices of the stock, per se, since these are shared by all of the surrogate trajectories. A trader might chalk this up to “momentum,” perhaps correctly. It would be interesting to explore the phenomena more closely, to uncover the timescale of these correlations in returns. We will not make the analysis here, but the methods introduced in Chapter 3, in which the permutation group is replaced by a subgroup of “local” permutations, might make a sensible starting point.

Of course 3,040 is a substantial number of returns and the usual cautions about distinguishing statistical significance from practical significance are worth repeating. Since almost nobody believes that returns are iid (and hence exchangeable), finding a statistic that distinguishes the given ordering from a random ordering is not a surprise, and the low p-value says as much about the large sample size as it does about the magnitude of the effect. Our emphasis is on exploring the data, especially the time-scale of departures from the standard models, rather than testing hypotheses per se.

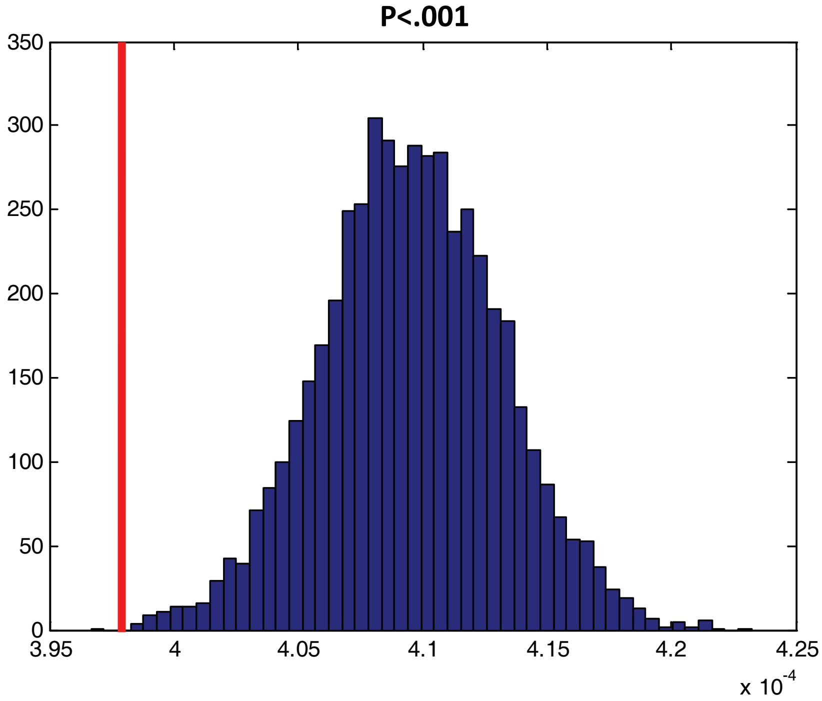


Figure 2.1: **Total Variation.** The average absolute change between successive pairs of 30-second returns of IBM stock over four days in 2005 was computed, and indicated by the vertical line segment in the figure. The corresponding average was also computed for each of 5,000 random permutations of the returns, and displayed as a histogram. The average for the original data falls in the left tail and yields a highly significant one-tailed p-value (about .0004) against the null hypothesis that stock returns are exchangeable.

2.2 The Timing of Large Returns

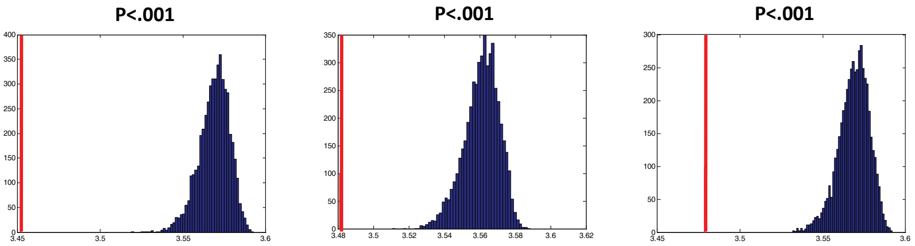
Returning now to the excursion process $\{Z_n\}$ of equation (1.5), and in particular the waiting-time between excursions (between ones), observe that any scalar function of waiting times is a scalar function of $R_1, \ldots, R_N$ and therefore qualifies as a statistic, H, for conditional inference in the sense of Section 2.1. We have already remarked on the prominent clustering of excursions, as is readily seen by comparing an empirical waiting-time distribution to the geometric distribution, as in Figure 1.2. The data suggests that the waiting-time distribution of actual excursions is more regular than would be expected from a random placement of an equal number of excursions among an equal number of observations (20% ones for the values of f and g used here—see Chapter 1). Entropy is a common measure of randomness in a distribution, suggesting the statistic

$$H(r_1, \ldots, r_N) = -\sum_{w=0}^{\infty} \hat{P}_W(w) \log \hat{P}_W(w)$$

where log is typically base 2 (H in units of bits) and where $\hat{P}_W(w)$ is the empirical probability of finding w zeros between a one and the next ensuing one.

Figure 2.2 shows the results after testing exchangeability for each of the three sets of returns (Citigroup one minute for 2008, IBM 30 minute for 2005, and IBM 30 second for 2005) explored in Figure 1.2. In these three examples, as in almost every example of every stock at every time scale and in every era, provided that a substantial number (say 3,000) of returns are examined, the observed waiting time entropy is significantly too small to be consistent with the standard models. (It would be incorrect to interpret the results in Figure 2.2 as implying that the significance can be made arbitrarily small by choosing an arbitrarily large number of permutations. Without a doubt there are permutations, in "the neighborhood" of the identity, which would yield smaller entropies than the observed sequences of returns. What is more, eventually there will be ties, e.g. the identity permutation will reproduce the observed entropy. In fact the asymptotic p-value, as the number of permutations increases, has a lower bound which is small but not zero: $1/(N! + 1)$.)

We close this section with a remark about the interpretation of the entropies recorded in our experiments. We will continue to use entropy as a gauge of randomness, and departures from randomness, in the distributions of waiting times between excursions. When an empirical entropy is smaller than each of the 5,000 entropies


P<.001

P<.001

P<.001

Figure 2.2: **Waiting Times Between Excursions Have Low Entropies.** Each panel shows the entropy of the empirical distribution of waiting times between ones in the excursion process (vertical bar), and the distribution of corresponding entropies for 5,000 random permutations of the returns. Left Panel: eight days of one-minute Citigroup returns in early 2008. Middle Panel: 250 days of 30-minute IBM returns in 2005. Right Panel: four days of 30-second IBM returns in 2005. Each plot is annotated with the p-value from a test of the exchangeability hypothesis. In all three the observed entropy is the smallest entropy, and hence the p-value is simply 1/5001.

derived from independent permutations of returns (equivalently, derived from a random placement of 20% ones and 80% zeros), the result is obviously statistically significant. The actual numbers, the entropies themselves, are less easy to interpret. In our experiments, a set of about 3,000 real returns typically produced a waiting-time entropy in the approximate range of 3.45 to 3.48 bits. The average entropy of a permuted sequence is about 3.57 bits. This small differences in entropies can obscure a substantial difference in waiting times. We have already noted that the probability of a large return (i.e. an excursion) being followed directly by another large return is about 33% higher for real returns (approximately .28) than for permuted returns (very nearly .2).

2.3 Fractional Brownian Motion

The evidence for invariance in the waiting-time distribution between excursions suggests looking for mathematical models of stock prices that produce invariant distributions on the excursion process itself. The standard models do the trick since the returns are iid, which in and of itself is enough to completely determine $\mathcal{L}\{Z_n\}$, the distribution of the excursion process. In particular, the distribution on Z_n is independent of the return interval, δ (30 seconds, 60 seconds, 30 minutes, etc.). But it is abundantly clear that the waiting-time distribution for an iid process, albeit invariant, does not match the (nearly) invariant distribution for real stock returns—refer again to Figure 1.3. In fact the entropy of the waiting times between excursions for stock returns (say, 3,000 returns) is typically in the range of 3.45 to 3.49, whereas the corresponding entropies for an iid process is in the range of 3.52 to 3.59. Real excursions are more predicatable than the modeled excursions.

Researchers have studied fractional Brownian motion (FBM) as a modeling tool for capturing the correlation structure of stock returns (e.g. Elliott & Hoek [22], Hu & Oksendal [39], Bjork & Hult [13]). In the most straightforward implementation the solution to the standard model (equation (1.3))

$$S_t = S_0 e^{at+bw_t}$$

for suitable a and b, is modified by replacing the Brownian motion w_t with FBM w_t^H:

$$S_t = S_0 e^{at+bw_t^H} \tag{2.1}$$

The correlations are controlled by H, the "Hurst index," which lies in $[0,1]$. The special case $H = .5$ recovers the usual Brownian motion, whereas $H > .5$ promotes positively correlated returns (as in "momentum") and $H < .5$ promotes negatively correlated returns (as in "mean reversion"). For any $H \in [0,1]$, w_t^H is "self similar" in that

$$\mathcal{L}\{W_{\delta t}^H, t \geq 0\} = \mathcal{L}\{\delta^H W_t^H, t \geq 0\}$$

which is sufficient to guarantee that the distribution of the excursion process, Z_n, is invariant to the return interval δ. This follows from the observation that Z_n, defined in equation (1.5), is invariant to monotone transformations of the return process, R_n, along with the following argument (where we write $R_n^{(\delta)}$ and $Z_n^{(\delta)}$ in place of R_n

and Z_n, respectively, to explicitly designate the return interval):

$$\begin{aligned} R_n^{(\delta)} &= \log \frac{S_{n\delta}}{S_{(n-1)\delta}} \\ &= (an\delta + bw_{n\delta}^H) - (a(n-1)\delta + bw_{(n-1)\delta}^H) \\ &= a\delta + b(w_{n\delta}^H - w_{(n-1)\delta}^H) \end{aligned}$$

Hence

$$\begin{aligned} \mathcal{L}\{R_n^{(\delta)}\} &= \mathcal{L}\{a\delta + b\delta^H(w_n^H - w_{(n-1)}^H)\} \\ &= \mathcal{L}\{G(R_n^{(1)})\} \end{aligned}$$

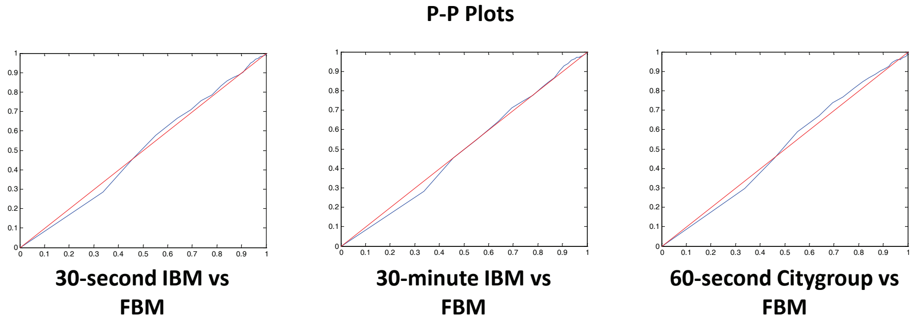
where $G(\cdot)$ is the monotone function

$$G(x) = \delta^H(x-a) + a\delta$$

So $\mathcal{L}\{Z_n^{(\delta)}\} = \mathcal{L}\{Z_n^{(1)}\}\ \forall\delta$, as might have been expected starting with a self-similar process.

Recall that $H = .5$ brings us back to geometric Brownian motion, with its high waiting-time entropy in the range of 3.52 to 3.59. As H increases towards one, positive correlations in returns enforce regularity in the excursion process, decreasing the entropy of waiting times between excursions. Fractional Brownian motion is easy to sample; we find that the waiting-time entropy is in the right empirical neighborhood of 3.45 to 3.49, when $H = .8$ and the number of returns is about 3,000, thus similar to stock data (Chapter 1) with similar numbers of returns. (Keep in mind that the waiting-time entropy is itself a random variable, with a variance that will depend on the number of returns in the sequence.) The waiting-time entropy again decreases, as H is made *smaller* from *below* $H = .5$. But there does not seem to be any $H < .5$ that has entropy as low as the empirical entropy when looking at returns from 3,000 real stock prices.

By setting $H = .8$ and sampling repeatedly from the FBM, we can again use the entropy of the waiting-time distribution to test the geometric-FBM model of equation (2.1). An ensemble of FBM processes (of length, say, 3,000) produces a corresponding ensemble of entropies, in which the empirical entropy of real stocks is typically neither in the left nor the right tail; we can not reject the model based on this statistic. (Unsurprising, given that H was chosen to match the entropy of the model to the entropy of stocks.) But not rejecting is not the same as accepting. Aside from matching the entropy, how does the geometric FBM ($H = .8$) waiting-time distribution, itself, match that of real stocks? Better than the standard models but not particularly well, as can be seen by comparing Figure 2.3 to Figure 1.3. The FBM model systematically over-emphasizes short waits between excursions.



30-second IBM vs FBM

30-minute IBM vs FBM

60-second Citygroup vs FBM

Figure 2.3: **Fractional Brownian Motion.** P-P plots compare the distribution on waiting times for stock returns to the corresponding distribution of a "geometric fractional Brownian motion" with Hurst index $H = .8$, chosen by matching the empirical entropies of the distributions for real stocks to those of the model. Left, middle, and right panels compare IBM 30-second returns, IBM 30-minute returns, and Citigroup 60-second returns, respectively, to the returns of the fractional Brownian motion model. Geometric Brownian motion ($H = .5$) under-estimates short waiting times between excursions (see Figure 1.3), whereas geometric fractional Brownian motion, with $H = .8$, somewhat overestimates short waiting times.

2.4 Memory

Geometric fractional Brownian motion introduces memory into the excursion process, Z_n, by way of correlation in the return process, R_n. This can be approached more directly by *starting with* a model of the excursion process itself. To explore the influence of memory on large returns, we can think of the process Z_n as a Markov process of some order, say k, so that, *a priori*,

$$P(Z_n = z_n|Z_{n-1} = z_{n-1}, \ldots, Z_1 = z_1) = P(Z_n = z_n|Z_{n-1} = z_{n-1}, \ldots, Z_{n-k} = z_{n-k}) \tag{2.2}$$

(*Every* process is almost a Markov process, of sufficiently high order, so the issue here is to discover how high an order is required to explain the time scale, and hence Markov order, of the correlation effects highlighted by the permutation test.) The special case of zero-order Markov ($k = 0$) refers to the iid model. When $k = 0$ the conditional process, conditioned on $\#\{n : Z_n = 1\} = .2N$, is exchangeable. Hence the permutation test introduced in Section 2.1 can be thought of as a test of the hypothesis $k = 0$, rather than a test of a class of models, such as the standard models, for stock prices. Either way, permutation leaves the distribution $\mathcal{L}\{Z_n\}$ unchanged. What, if any, invariants could be exploited to test the k-order Markov property when $k > 0$?

Take the case $k = 1$: we model $\{Z_n\}$ as first-order Markov and observe $\{z_n\}$. Then any sequence that starts at z_1 and has the same numbers of transitions as $\{z_n\}$ of each of the four types ($0 \to 0$, $0 \to 1$, $1 \to 0$, $1 \to 1$) is equally likely, and in fact has $.2N$ ones and ends at z_N. Formally, given any binary (zero-one) sequence $v_1, \ldots, v_N$, let

$$a_1 = a_1(l_1, l_2; v_1, \ldots, v_N) = \#\{n \in \{1, 2, \ldots, N-1\} : v_n = l_1, v_{n+1} = l_2\} \quad l_1, l_2 \in \{0, 1\}$$

(so a_1 counts the transitions of each type) and let

$$\begin{aligned} \mathcal{S}_1(v_1, \ldots, v_N) \;=\; & \{(v_{\rho(1)}, \ldots, v_{\rho(N)}) : \rho \text{ is a permutation of } 1, 2, \ldots, N,\ \rho(1) = 1 \text{ and} \\ & a_1(l_1, l_2; v_{\rho(1)}, \ldots, v_{\rho(N)}) = a_1(l_1, l_2; v_1, \ldots, v_N),\ l_1, l_2 \in \{0, 1\}\} \end{aligned}$$

(so $\mathcal{S}_1$ is the set of sequences possessing the observed number of transitions of each type). Then $Pr\{(Z_1, \ldots, Z_N)|(Z_1, \ldots, Z_N) \in \mathcal{S}_1(z_1, \ldots, z_N)\}$ is uniform on $\mathcal{S}_1(z_1, \ldots, z_N)$. This means that the hypothesis "excursion process first-order Markov" can be tested using any statistic $H(Z_1,\dots,Z_N)$ and comparing $H_0 \doteq H(z_1,\dots,z_N)$ to the ensemble of values H_1,H_2, …,H_M

$$H_m \doteq H(z_{\rho_m(1)},\dots,z_{\rho_m(N)}), \quad m = 1\dots,M$$

where ρ_m is a random permutation chosen from the uniform distribution on permutations satisfying $(z_{\rho_m(1)},\dots,z_{\rho_m(N)}) \in \mathcal{S}_1(z_1,\dots,z_N)$. In other words, the same procedure as in §II.A, except that the random permutations are chosen uniformly from a specific subgroup of the permutation group.

The generalization to higher order, $k = 2, 3, \dots$, is straightforward:

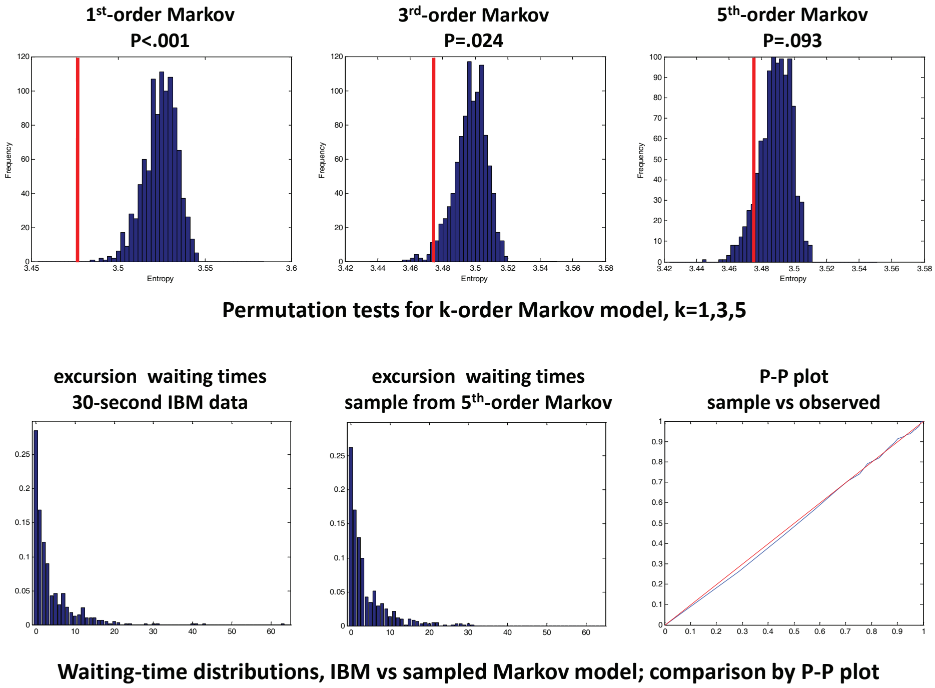
$$\begin{aligned} a_k &= a_k(l_1, l_2, \dots, l_{k+1}; v_1, \dots, v_N) \\ &= \#\{n \in \{1, 2, \dots, N-k\} : v_n = l_1, \dots, v_{n+k} = l_{k+1}\} \quad l_1, \dots, l_{k+1} \in \{0,1\} \end{aligned}$$

and

$$\begin{aligned} \mathcal{S}_k(v_1,\dots,v_N) \;=\; & \{(v_{\rho(1)},\dots,v_{\rho(N)}) : \rho \text{ is a permutation of } 1,2,\dots,N, \\ & \rho(i) = i,\ i = 1,\dots,k, \text{ and } a_k(l_1,\dots,l_{k+1}; v_{\rho(1)},\dots,v_{\rho(N)}) \\ & = a_k(l_1,\dots,l_{k+1}; v_1,\dots,v_N),\ l_1,\dots,l_{k+1} \in \{0,1\}\} \end{aligned}$$

What is less straightforward is efficiently sampling from the uniform distribution on $\mathcal{S}_1(z_1,\dots,z_N)$ (equivalently, sampling from the uniform distribution on the subgroup of the permutation group that leaves the first k elements, $1, 2, \dots, k$, as well as all transition counts, a_k, unchanged). Suffice it to say that efficient algorithms exist (for completeness, one is described in Appendix A) and, therefore, the approach introduced in Section 2.1 is suitable for exploring the "Markov order" of the process $Z_1,\dots,Z_N$.

As the Markov order, k, increases, the sequences in $\mathcal{S}_k$ are increasingly constrained to look like $z_1,\dots,z_N$. Inevitably, H_0 will resemble the sampled statistics $H_1,\dots,H_M$. At what order, k, does the entropy of the excursion waiting-time distribution of $z_1,\dots,z_N$ resemble that of the ensemble of sampled sequences? At zero order, H_0 is too small and the result is highly significant, as demonstrated earlier—see Figure 2.2. Focusing on the IBM 30-second data, Figure 2.4, top row, shows the corresponding experiments for $k = 1,\ 3$, and 5, using 1,000 samples from $\mathcal{S}_1$, $\mathcal{S}_3$, and $\mathcal{S}_5$, respectively. Naturally, p-values increase (significance decreases) at higher orders. The surprise, if any, is that the third-order process is still a substantial misfit. The fifth-order process is a good fit, as can be seen by using the empirical transition matrix (which is the maximum-likelihood estimate) to produce a single sample (length 3,040) from the fifth-order Markov process and comparing it to the original 3,040-length IBM excursion sequence. See Figure 2.4, bottom row.



Permutation tests for k-order Markov model, k=1,3,5

Waiting-time distributions, IBM vs sampled Markov model; comparison by P-P plot

Figure 2.4: **Memory in the Excursion Process. Top row:** The Markov order of the 30-second 2005 IBM data was explored. The entropy of the excursion waiting-time distribution was used to test the hypothesis that the data comes from (left-to-right) a first, third, or fifth-order Markov process. An ensemble of entropies is produced by random sub-group permutations that preserve, respectively, the second, fourth, and sixth-order (consecutive) marginal distributions of the observed excursion process. **Bottom row:** The maximum-likelihood transition matrix for the fifth-order Markov model was used to produce a single sample, of length 3,040 (matching the IBM data), of the excursion process. The waiting-time distribution for the IBM data is on the left, the waiting time distribution for the sample is in the middle, and the P-P plot comparing the sampled and observed distributions is on the right.

Chapter 3

Time Scale and Stochastic Volatility Models

Not all orderings of returns are equally likely. In particular, large returns (positive or negative) are followed by more large returns, and small returns by more small returns, more often than would be expected from an exchangeable process. We see these effects when comparing empirical waiting-time distributions between excursions to the geometric distribution, as discussed in Chapter 2. These observations are consistent with the well-accepted stochastic, or at least time-varying, nature of volatility (σ) in the standard model (equation (1.3)). If we take the point of view that the standard model is itself evolving in time (e.g. $\sigma = \sigma(t)$), as in the various stochastic volatility models, then it might be useful to postpone the specification of specific theories in order to first explore the time-scale of change. One way to do this, from a more-or-less model-free viewpoint, is to modify the permutation test introduced in Section 2.1 so as to restrict the exchanges of returns to be local rather than global.[1] If the geometric Brownian motion (or one of its standard extensions, e.g. through Lévy processes) is to be taken seriously as the starting point for model building, then presumably the dynamics of stock fluctuations are at least *locally* consistent with equation (1.3).

Consider again the IBM 30-second data taken from four days in 2005. (Other

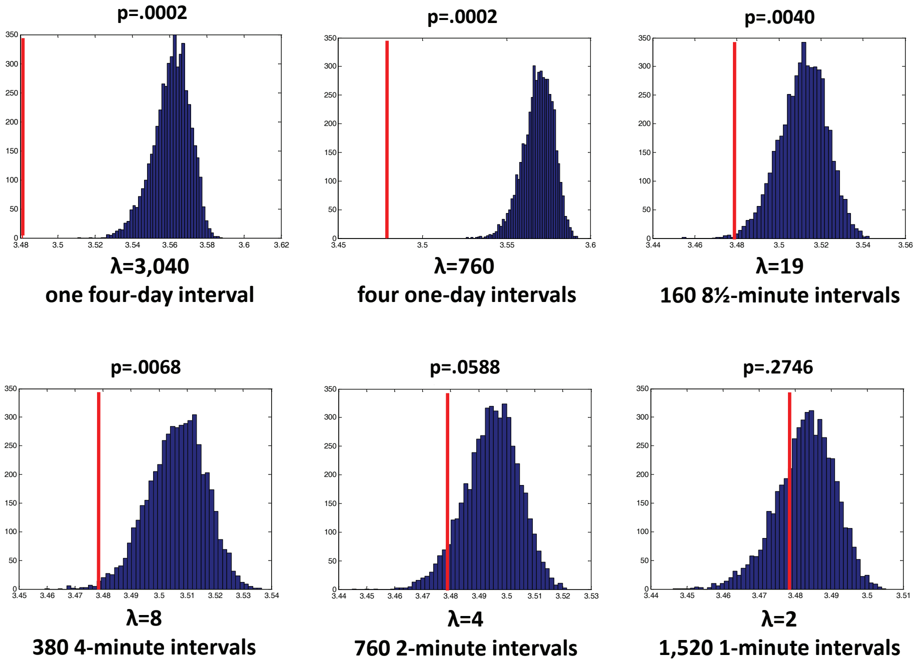
[1] A similar form of "conditional inference" has been used for the analysis of neurophysiological data (Hatsopoulos et al. [35], Harrison & Geman [34]).

intervals, other stocks, and other eras lead to similar results and conclusions.) We can not treat the entire set of 3,040 returns as exchangeable. But are they, at least approximately, exchangeable within each of the four days, or perhaps within one-hour or one-half-hour intervals? A convenient way to explore these questions is to partition the index set $\{1, 2, \ldots, 3040\}$ into disjoint intervals of length λ, where $\lambda \cdot \delta$ represents a time span over which the returns are presumed to be (essentially) exchangeable. For the 30-second data, we would use $\lambda = 760$ to test for exchangeability within single days (recall that the first and last five minutes of each day of prices are excluded), and $\lambda = 19$, 8, 4, and 2, respectively, to explore exchangeability in eight-and-a-half, four, two, and one-minute intervals. Let I_k be the set of indices in the k^{th} interval of length λ, $I_k = \{(k-1)\lambda + 1, \ldots, k\lambda\}$, and let $\mathcal{S}_\lambda(r_1, \ldots, r_N)$ be the set of re-arranged returns that preserve the collections $\{r_i : i \in I_k\}$:

$$\begin{aligned}\mathcal{S}_\lambda(r_1, \ldots, r_N) \quad &= \quad \{(r_{\rho(1)}, \ldots, r_{\rho(N)}) : \rho \text{ is a permutation of } 1, 2, \ldots, N, \\ &\qquad \text{and } i \in I_k \Rightarrow \rho(i) \in I_k, \ \forall k\}\end{aligned}$$

The hypothesis to be tested is that $Pr\{(R_1, \ldots, R_N) | (R_1, \ldots, R_N) \in \mathcal{S}_\lambda(r_1, \ldots, r_N)\}$ is uniform on $\mathcal{S}_\lambda(r_1, \ldots, r_N)$, which can be done using the entropy of the waiting-times between excursions and, as usual, comparing $H_0(r_1, \ldots, r_N)$ to $H_m \doteq H(r_{\rho_m(1)}, \ldots, r_{\rho_m(N)})$, $1 \le m \le M$. In this case, ρ_m is chosen by randomly and independently permuting, via the uniform distribution, the indices in each set I_k.

Figure 3.1 shows the results of hypothesis tests of local exchangeability for the 30-second IBM data, at various λ. Five-thousand permutations were used for each test. The upper-left panel, identical to the right-most panel in Figure 2.2, is the result of testing for unrestricted exchangeability. It is included for comparison. The remaining panels test exchangeability over one-day, 8.5-minute, 4-minute, 2-minute, and 1-minute intervals. Rejection of the null hypothesis (e.g. exchangeability of returns within 4-minute intervals) per se is not remarkable. Indeed, it would be surprising if the 30-second returns really were exactly exchangeable, even over short intervals. It is perhaps a little surprising that the evidence is strong enough to show up in just four days of returns. But regardless of the interpretation, our purpose here is more exploratory. In particular, we propose that time-scale experiments such as these offer a useful benchmark for examining observations and theories about the correlative structure of returns in general, and the time-varying nature of volatility in particular. As illustration, we turn now to some examples.



p=.0002

λ=3,040
one four-day interval

p=.0002

λ=760
four one-day intervals

p=.0040

λ=19
160 8½-minute intervals

p=.0068

λ=8
380 4-minute intervals

p=.0588

λ=4
760 2-minute intervals

p=.2746

λ=2
1,520 1-minute intervals

Figure 3.1: **Time Scale.** The entropy of the waiting-time distribution for the excursion process derived from the 2005 four-day IBM data (3,040 returns) is about 3.48. The panels show distributions of waiting-time entropies generated by 5,000 restricted permutations of the IBM returns, respecting the specified intervals. For comparison, the observed value (3.48) is highlighted in each panel by a vertical bar. In the upper left panel, which is identical to right-most panel of Figure 2.2, the entire data set is treated as a single interval. Four days of 30-second returns (lower-left panel)is enough to reject the hypothesis of local (four-minute-interval) exchangeability with high significance ($p \approx .007$).

3.1 Implied Volatility

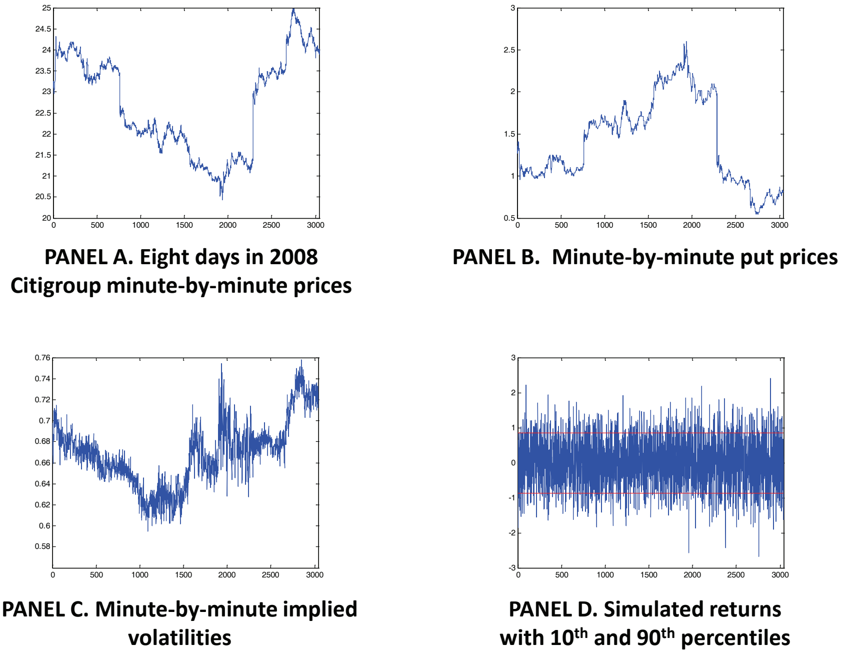
One place to look for a volatility process that might explain the breakdown of exchangeability, as measured by the waiting-time distribution between excursions, is the volatility implied by the pricing of options. Do the fluctuations of the implied volatility, taken as a model for $\sigma = \sigma_t$ in (1.3), have sufficient amplitude, speed, or at least correlation, to support the lack of (global *and* local) exchangeability in the return process?

We experimented with the Citigroup data (eight days of one-minute returns from 2008—Chapter 1), and derived an implied volatility process from the April 19, 2008, put option with strike price 22.5.[2] Figure 3.2 shows the stock prices, sampled at every minute, the corresponding returns, the corresponding implied volatilities, and a sample from the return process generated by the geometric Brownian motion (equation (1.3)) with σ replaced by implied volatilities. The percentiles, which define the excursion process, as well as the waiting-time distribution and its entropy, are superimposed on the simulated returns (Panel D). How do the simulated returns compare to the observed returns with respect to exchangeability and time scale? Figure 3.3, which shows the results of the permutation test for the entire interval, as well as intervals of length one-half day and thirty-eight minutes, demonstrates a sharp contrast in the behaviors of the simulated and actual return processes. In particular, the real stock data has a substantially lower entropy of the excursion waiting-time distribution (about 3.45 versus about 3.57), and, in contrast to the simulated returns, there is strong evidence against exchangeability of the entire set of returns as well as the half-day subsets of returns.

Different derivatives (puts and calls with different strike prices and expirations) imply different volatilities. It is possible that other derivatives would give a substantially better match, but unlikely given the extent of the mismatch revealed in

[2] Computed numerically from the CRR pricing model (Cox et al. [20]).

Figure 3.3. It is our impression that the changes in volatility implied by the Black-Scholes formula are too small and too slow to explain the excursion behavior of real stocks. Perhaps this should have been expected, in that implied volatility is said to be "forward looking," meaning it reflects an investor's belief about future volatility rather than today's volatility. Possibly, the fluctuations in these sentiments are substantially less rapid and less extreme than an "actual" volatility σ_t, under which the geometric Brownian motion model would presumably provide a better fit.



PANEL A. Eight days in 2008 Citigroup minute-by-minute prices

PANEL B. Minute-by-minute put prices

PANEL C. Minute-by-minute implied volatilities

PANEL D. Simulated returns with 10th and 90th percentiles

Figure 3.2: **Implied Volatility. A.** Citigroup price every minute for eight days in 2008. **B.** Put prices (strike price 22.5, maturing on April 19, 2008) sampled at the same one-minute intervals. **C.** Implied volatilities (Black-Scholes model). **D.** Simulated, minute-by-minute, returns, generated from a geometric Brownian motion with volatility function ($\sigma = \sigma_t$) equal to the implied volatility. Percentiles (10'th and 90'th), defining the excursion process through equation (1.5), are superimposed.

3.2 Experiments with Artificial Volatility Processes

Can we *invent* a volatility process which generates excursions that resemble the excursions in the returns of real stocks? We did some experiments in curve fitting,

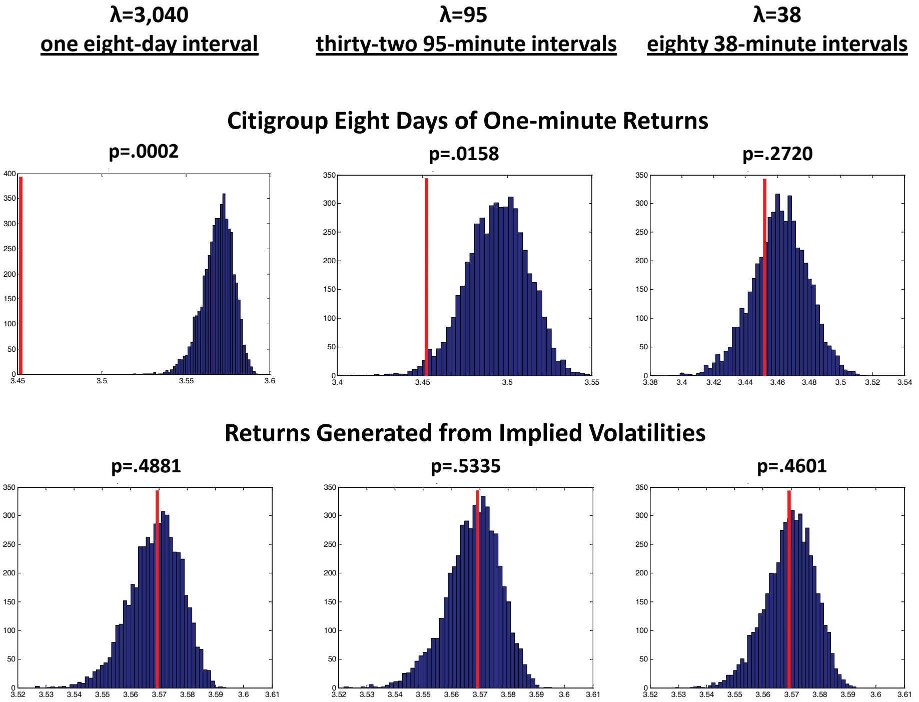


Figure 3.3: **Implied Volatility and the Time Scale of Exchangeability.** Comparison of the exchangeability of the simulated (implied volatility) and actual Citigroup returns. **Top row:** permutation tests using the actual returns. **Bottom row:** permutation tests using the simulated returns (see Figure 3.2). With 3,040 samples (eight days of one-minute returns), exchangeability of the actual returns within thirty-eight minute intervals can not be rejected, but exchangeability over half days is rejected ($p < .02$), and exchangeability over the entire eight days is rejected at very high significance. The same test based on the same statistic produces no evidence for lack of exchangeability on any time scale in the simulated returns.

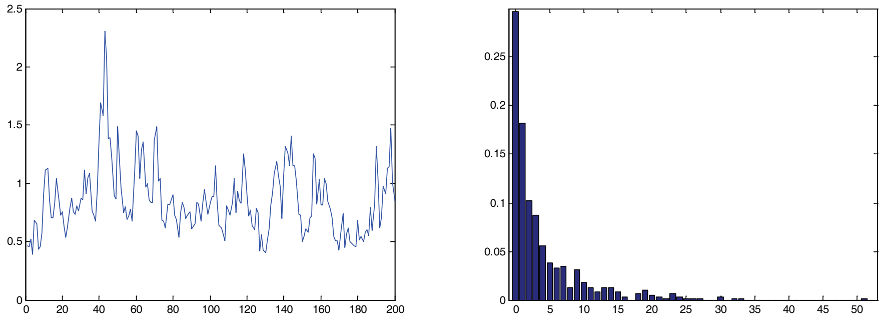
meaning that we tried generating a stochastic volatility process that gave a similar profile of the relationship of time scale to exchangeability as was seen in the four-day IBM data (Figure 3.1). The idea was to get a sense for the amplitude and frequency of fluctuations in σ_t that might be necessary to match the time-scale results of the real data. To generate an artificial sequence of returns, we explored a two-parameter family of Ornstein-Uhlenbeck processes

$$d\alpha_t = -\theta\alpha_t dt + \eta d\tilde{w}_t$$

where $\tilde{w}_t$ is a standard Brownian motion, and then used $\sigma = \sigma_t = e^{\alpha_t}$ in equation (1.3) (with w_t independent of $\tilde{w}_t$) to simulate prices and returns. The model is a special case of the model of Hull and White [40], and similar to the model of Heston [37].

With an eye on the IBM data, we took the units of t to be minutes and generated 3,040 artificial returns. In the absence of $\tilde{w}_t$, α_t mean-reverts (passes through the fraction $\frac{1}{e}$ of its starting value) in $\frac{1}{\theta}$ minutes. The transition in p-value between the four and two-minute interval permutations (from about .002 to about .07) suggests looking at values of θ with $2 \leq \frac{1}{\theta} \leq 4$; we chose $\theta = \frac{1}{3}$. At a given θ, the mean amplitude of fluctuations in α_t, and hence in σ_t, over a specified time interval, is determined by η. For example, when $\eta \approx 0.2$ the mean four-minute fluctuation in σ_t, as measured by the mean of the fraction $|\frac{\sigma_{t+4}-\sigma_t}{\sigma_t}|$, is about 18%. With these parameters, the permutation test for exchangeability, applied to the simulated process, was not significant, even with $\lambda = 3,040$. We could not reject unrestricted exchangeability of the returns. The empirical waiting-time entropy was 3.56, which is well within the range of typical waiting-time entropies for 3,040 fully exchangeable returns (e.g., refer to the upper-left panel in Figure 3.1).

How far do we have to go, in terms of the amplitude of fluctuations in σ_t, in order to reject exchangeability at the various time scales revealed by the IBM data? Figure 3.4 (left panel) shows 200 samples of σ_t from the (exponentiated) Ornstein-Uhlenbeck



200 samples from a stochastic volatility model

excursion waiting-time distribution for simulated returns

Figure 3.4: **Stochastic Volatility: Fitting the Data.** Volatility was modeled as the exponential of an Ornstein-Uhlenbeck process, with parameters chosen to approximately match the exchangeability results observed in the IBM 30-second data (Figure 3.1). **Left Panel:** A sample of length 200 from the volatility process, σ_t. The process changes an average of 39% every eight time units. **Right Panel:** Empirical excursion waiting-time distribution from 3,040 returns generated by geometric Brownian motion with stochastic volatility σ_t.

process with η adjusted upward (to 0.4) until the entropy of the waiting-time distribution of excursions in the corresponding (simulated) return process matched the IBM entropy (3.47 for the simulated process versus 3.48 for IBM).[3] The waiting-time distribution itself is shown in the right panel.[4] It is quite similar to the waiting-time distribution for the actual return—see bottom row, left panel, Figure 2.4. Figure 3.5 compares the time-scale of exchangeability between the two sequences: IBM in the top row and the geometric Brownian motion driven by the simulated stochastic volatility process in the bottom row. In qualitative terms, there is a good match, but it is accompanied by a surprising 39% average four-minute fluctuation in σ_t.

There are many stochastic volatility models to choose from (cf. Shephard [58]).

[3] The entropy fluctuates, from simulation to simulation, in the neighborhood of 3.48. We *selected* the particular sample shown for its close match in entropy.

[4] The distribution of the excursion process is invariant to the scale of σ_t, i.e. $c \cdot \sigma_t$ produces the same excursion process for every $c \neq 0$. Hence the vertical scale in the left-hand panel is arbitrary.

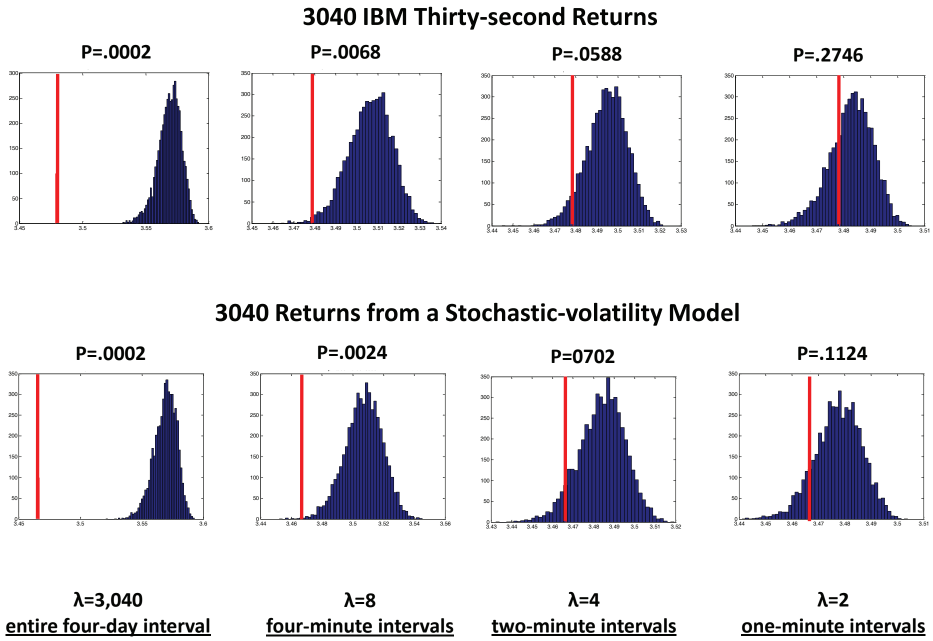


Figure 3.5: **Stochastic Volatility: Comparing time Scales.** Comparison of p-values for IBM 30-second data against p-values for returns generated by a stochastic-volatility process (exponentiated Ornstein-Uhlenbeck process—see Figure 3.4). Exchangeability was tested at each of four time scales (four days, four minutes, two minutes, and one minute). For this model, a good match of time-scale dependence, like the one shown here, appears to require rapid and high-amplitude fluctuations of the volatility process, e.g. 39% every four minutes. (The top four panels are a selection from the eight time-scale experiments illustrated in Figure 3.1.)

There can be little doubt that other models, in addition to the Ornstein-Uhlenbeck type model explored here, could be made to reproduce the temporal relationships revealed by the permutation experiments. It is possible, though, that any volatility model that matches the time scale of excursions in the return process will require very high-frequency, high-amplitude, fluctuations, raising the question of mechanisms that might underlie such a drastic departure from the random walk (geometric Brownian motion) model.

3.3 Volume, Trades, and Stochastic Time Change

There is no reason to believe that a good model for the logarithm of stock prices should be homogeneous in time. To the contrary, the random walk model suggests that the variance of a return should depend on the number or volume of transactions (the number of "steps") rather than the number of seconds. The compelling idea that "market time" is measured by accumulated trading activity rather than the time on the clock was first suggested by Clark [19], and has been re-visited in several influential papers since then (see the discussion by Shephard [58], General Introduction, for an excellent review and additional references).

Following Ané and H. Geman [7], we studied various time changes, $t \to \tau(t)$, from clock time to market time, where the function $\tau(t)$ is non-decreasing. The idea is to model stock prices as time-changed geometric Brownian motions, so that $S_t = U_{\tau(t)}$, where

$$dU_s = \mu U_s ds + \sigma U_s dw_s$$

(Discontinuities in $\tau(t)$ accommodate discontinuities in prices.) To keep things simple, we assumed that $\{w_t\}_{t>=0}$ is independent of $\tau(\cdot) = \{\tau(t)\}_{t>=0}$, which is consistent with Clark's original analysis, as well as the formulation of Ané and H. Geman. Conditioning on $\tau(\cdot)$, the returns, $R_1, \ldots R_N$, are still independent, but no longer identically distributed. Specifically,

$$R_n = \log \frac{S_{n\delta}}{S_{(n-1)\delta}} = \log \frac{U_{\tau(n\delta)}}{U_{\tau((n-1)\delta)}} \sim N(\eta\delta_n, \sigma^2\delta_n) \tag{3.1}$$

where $\eta = \mu - \sigma^2/2$, and $\delta_n = \tau(n\delta) - \tau((n-1)\delta)$ is the $n'th$ increment of transformed time. Continuing to condition on $\tau(\cdot)$, the "corrected" returns

$$\tilde{R}_n = \frac{R_n - \eta\delta_n}{\sigma\sqrt{\delta_n}} \tag{3.2}$$

are again iid.

Various schemes for modeling τ and estimating η and σ have been introduced. We note here that, given τ, the maximum-likelihood estimators of η and σ are available through

$$(\hat{\eta}, \hat{\sigma}) = \arg\max_{\tilde{\eta},\tilde{\sigma}} \prod_{n=1}^{N} \frac{1}{\sqrt{2\pi\tilde{\sigma}^2}} e^{\frac{(R_n - \tilde{\eta}\delta_n)^2}{\tilde{\sigma}^2\delta_n}}$$

leading to

$$\hat{\eta} = \frac{\sum_{n=1}^{N} R_n}{\sum_{n=1}^{N} \delta_n}$$

$$\hat{\sigma} = \frac{1}{N} \sum_{n=1}^{N} \frac{(R_n - \hat{\eta}\delta_n)^2}{\delta_n}$$

Return sequences often include extreme outliers, which argues for a more robust estimator for σ. A simple alternative, that we employed in place of the maximum-likelihood estimator, is a two-sigma based quantile estimator: Let F denote the cumulative distribution of a standard normal random variable and let $\hat{F}$ be the empirical cumulative distribution of the re-centered and re-scaled returns:

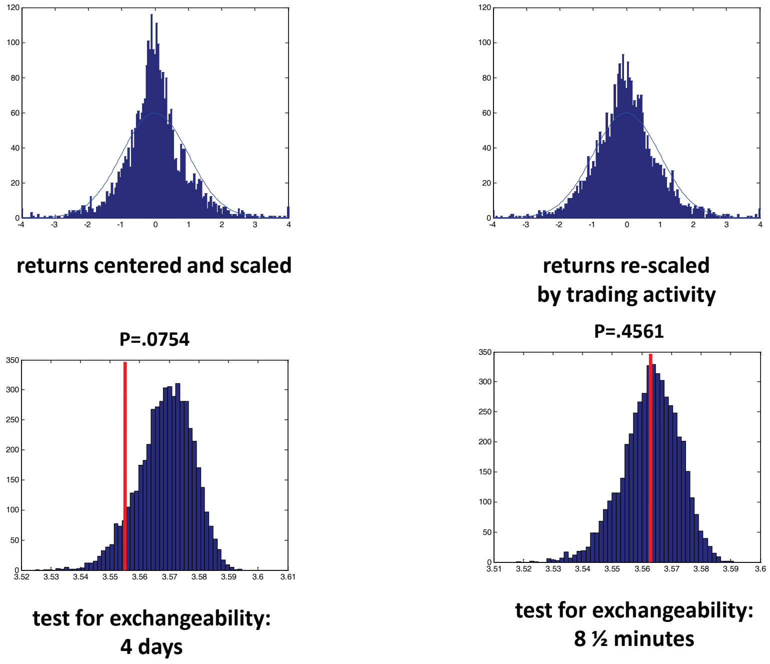
$$\hat{F}(r) \doteq \frac{\#\{n : (R_n - \hat{\eta}\delta_n)/\sqrt{\delta_n} \le r\}}{N}$$

Then $\hat{\sigma} = [\hat{F}^{-1}(F(2)) - \hat{F}^{-1}(F(-2))]/4$ is consistent for σ and unchanged by a large fraction of outliers in the returns.

Consider the accumulated numbers of transactions, T_t (as opposed to the accumulated volume, V_t, which includes the shares per transaction), as a first model for market time. Referring to Figure 3.6, we examined the return sequence, $r_1, \ldots, r_{3040}$, from the four days of 30-second 2005 IBM data. The upper-left panel shows the corrected returns, as defined in equation (3.2), but with η and σ estimated under the standard model, i.e. without time change ($\tau(t) = t,\ \delta_n = \delta$). In comparison to the standard normal distribution (superimposed), the returns are more populated both near zero and in the tails. This is a typical example of the leptokurtosis found in stock returns. Using the recorded trades for the four days of IBM data, we next set $\tau(t) = T_t$, re-computed $\hat{\eta}$ and $\hat{\sigma}$ with $\delta_n = T_{n\delta} - T_{(n-1)\delta}$, and again compared the corrected returns to a standard normal distribution. The fit is substantially better (upper-right panel, Figure 3.6). Continuing with the time change $\tau(t) = T_t$, we then simulated 3,040 returns $R_1, \ldots, R_{3040}$ according to (3.1). The results are summarized in the lower two panels. Permutation tests for exchangeability, over intervals of length 8.5 minutes, and even at the coarsest scale (*full* exchangeability), were not significant ($p \approx .46$ and $p \approx .075$, respectively). In particular, the entropy of the excursion waiting-time distribution, over several simulations, was consistently too large to match the empirical entropy of the real returns.

A more direct test of the time change $t \to T_t$ is through the observed returns for equal increments of accumulated activity, T_t. For the four days of IBM data, there were, on average, about 10 trades per 30-second increment. Ignoring clock time, we collected the sequence of returns, $\tilde{R}_1, \ldots, \tilde{R}_{\tilde{N}}$, defined by successive intervals of 10 trades over these four days, where $\tilde{N}$ turned out to be 3,094. In other words, stock prices, $\tilde{S}_k$ were recorded after $k{\cdot}10$ accumulated trades, for each $k = 0, 1, \ldots, 3094$, with the first price ($\tilde{S}_0$) taken at approximately 9:35 on the first day and the last price ($\tilde{S}_{3094}$) at 15:55 on the fourth day. Under the model $\tau(t) = T_t$, these prices define "equal-market-time" returns $\tilde{R}_k = \log \frac{\tilde{S}_n}{\tilde{S}_{n-1}}$, $k = 1, 2, \ldots, 3094$. The empirical distribution of these returns closely resembles the empirical distribution of the returns standardized by (3.2) with $\delta_n = T_{n\delta} - T_{(n-1)\delta}$, i.e. the distribution displayed in the upper-right panel of Figure 3.6.

Of course market-time returns could fail to be normal but still be independent and



returns centered and scaled

returns re-scaled by trading activity

test for exchangeability: 4 days

test for exchangeability: 8 ½ minutes

Figure 3.6: **Transactions and Volatility. Upper Left:** Standardized returns from the 30-second 2005 IBM data. The high peak near zero and heavy tails (leptokurtosis), relative to the standard normal (superimposed curve), are typical. **Upper Right:** The fit, though still imperfect, is much improved when returns are re-scaled by the square-root of the number of transactions. **Lower Left:** Returns were simulated under the assumption that "market time," and hence volatility, is measured by the number of transactions rather than the number of seconds. In contrast to the actual returns (compare to Figure 3.1), the simulated returns are not incompatible with full exchangeability. **Lower Right:** Failure to reject local (8.5 minute) exchangeability is also in contrast to actual returns.

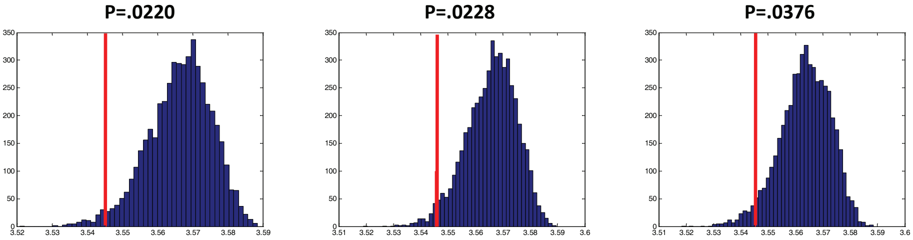


Figure 3.7: **Market-time Returns.** 3,094 returns were collected from four days of 2005 IBM stock data, with each return spanning an equal number (i.e. 10) trades. Full exchangeability was tested, and rejected (left panel, $p \approx .02$), as were exchangeability of sequences of 773 returns (about one sequence for each of the four days, middle panel, $p \approx .02$) and 16 returns (about one sequence for every eight minutes, right panel, $p \approx .04$).

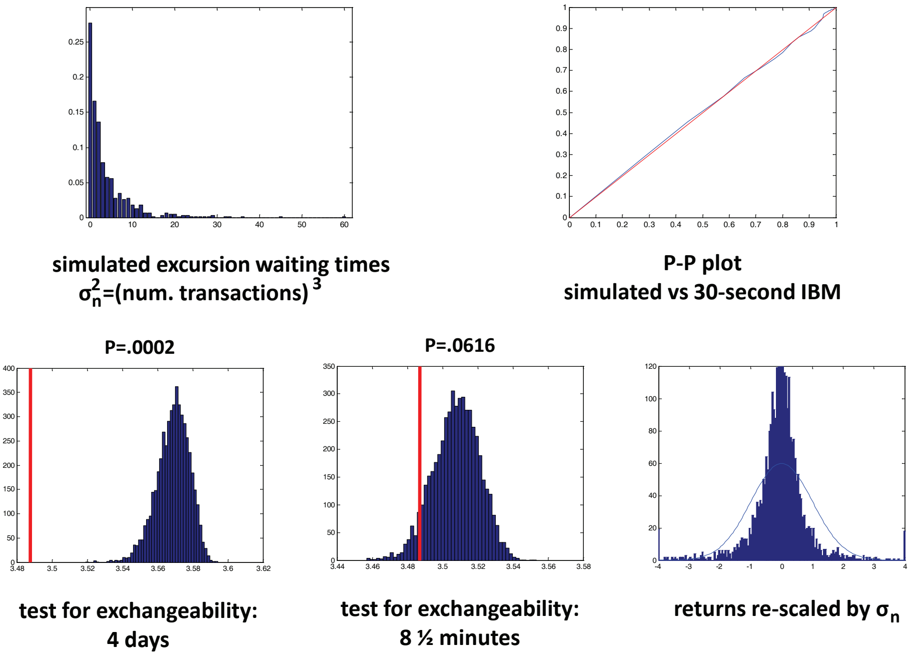
identically distributed increments of a random walk. But in that case the returns would be exchangeable, which they are not, judging from the results of various interval permutation tests shown in Figure 3.7.

If we are to stay with the geometric Brownian motion model, albeit with a stochastic time change and possibly non-Gaussian increments, then the evidence is that transaction numbers vary too slowly to be the basis for volatility clustering. The results from experiments with trading volume ($\tau(t) = V_t$) were similar.

Clark did not hypothesize a simple relationship between market activity and market time. In fact Clark gave evidence that a highly convex function of activity would be required to standardize returns, at least for the futures markets examined in his original (1973) paper. Consistent with these observations, we found that $\delta_n = (T_{n\delta} - T_{(n-1)\delta})^p$, with $p \approx 3$ rather than $p = 1$, was necessary in order to get a match between the entropy of the waiting-time distributions of simulated returns (iid with $\sigma_n^2 \propto \delta_n$) and real returns. In fact, overall, the waiting time distributions of the IBM data and the simulated data are well matched at $p = 3$, as can be seen from the top panels in Figure 3.8. What's more, the permutation test for exchangeability of the return process, as a function of time scale, gives reasonably similar results for the simulated and real returns (compare lower-left and lower-middle panels in Figure 3.8

to corresponding intervals in Figure 3.1). However, $\delta \to \delta_n = (T_{n\delta} - T_{(n-1)\delta})^3$ is not actually a time change, and, besides, returns standardized by δ_n (i.e. $\tilde{R}_n = R_n/\sqrt{\delta_n}$) are far from normal, as can be seen from the lower-right panel in Figure 3.8.

All in all, we think it is unlikely that any simple function of transaction number or volume will successfully serve both purposes of standardizing returns, at least to the normal distribution, and simultaneously providing an adequate explanation for the peculiar excursion behavior of those returns.



simulated excursion waiting times
σ_n^2**=(num. transactions)** ^{3}

P-P plot
simulated vs 30-second IBM

test for exchangeability:
4 days

test for exchangeability:
8 ½ minutes

returns re-scaled by σ_n

Figure 3.8: **Simulated Returns.** An effort was made to match the exchangeability results of real returns by standardizing with a power of the observed number of transactions. The random-walk model suggests $\sigma_n^2 \propto$ (num. transactions) (exponent $p = 1$), but then simulated returns appear to be nearly exchangeable (Figure 3.6), in contrast to real returns. A good match to the 30-second IBM data required, instead, exponent $p = 3$. **Upper Left:** Distribution of waiting times between excursions of 3,040 simulated returns. **Upper Right:** P-P plot of simulated waiting-time distribution against observed waiting-time distribution. **Lower Left:** Entropy-based test for exchangeability of the simulated returns over the entire four-day interval. **Lower Middle:** Test for exchangeability over 8.5-minute intervals. In summary, test results for interval exchangeability of simulated returns are comparable to the results with real returns (see Figure 3.1). **Lower Right:** But exponent $p = 3$ is not compatible with a time-changed Brownian motion, as can be seen by the poor fit to Gaussian of the re-scaled returns.

Chapter 4

Summary and Concluding Remarks

We have introduced a collection of statistical tools for exploring the temporal characteristics of stock returns. These tools are of the nonparametric type (Lehmann [47], Hollander & Wolfe [38]), meaning that they are valid independent of detailed distributional assumptions. Nonparametric hypothesis tests can be nearly as powerful (in the sense of small type-II error) as parametric tests (e.g. the t-test, the F-test, and the χ^2-test), even when the data is drawn from the assumed distribution (e.g. normal). Furthermore, nonparametric tests are still valid, and in fact exact, for essentially arbitrary distributions, whereas parametric tests can be very misleading when their distributional assumptions are violated. The nonparametric approach seems particularly well suited for studying the behavior of stock returns, since return distributions are an object of much discussion and debate and, in any case, rather far from normal.

We have focused on the time scale of dependencies among returns. By permuting returns, either locally (on short time scales) or over an entire interval of study, we can assess the extent and the strength of correlations. The uniform distribution on the permutation group, together with the uniform distribution on various subgroups that permute within intervals, provides an essentially inexhaustible collection of surrogate return processes. Functions of the observed return process (i.e. statistics) will match the corresponding functions of the surrogates to the extent that log prices can be modeled locally as random walks (i.e. as processes with independent and identically distributed increments). We have paid special attention to excursions of the return process ("rare events"), because of their evident interest to the markets and because of the surprising nature of the clustering of large excursions discovered by C.-R. Hwang and his colleagues.

Many generalizations are possible. Other percentile-based processes could be studied, possibly including three or more values, each indicating a different percentile range. A variety of statistics (beyond the entropy of excursion waiting times) could be introduced, tailored to explore different aspects of temporal variability. Possibly, perturbations from the uniform distribution on permutations and subgroups could be helpful in exploring, more finely, the departures from local exchangeability. And possibly there are more general invariants that could be exploited to test for larger classes of models, including the hidden-Markov models, of which stochastic-volatility models are a special case.

Non-normality of returns and correlations among returns have been the focus of study and speculation for many decades. Stochastic volatility models preserve the random-walk foundation by introducing time-dependent variance into the random steps. Some of these models are quite elaborate. There is no question that stochastic volatility models can be crafted to accommodate many of the salient features of stock-price returns, including the correlative structure revealed by excursion processes and permutation tests. But in and of itself, a good fit does not make an informative model. We simulated volatility and return processes at various parameter settings. These provided a good match to the observed distribution of excursion waiting times, but required extreme volatility fluctuations on very short time scales.

These observations raise the question of *mechanism*, as opposed to model, per se. What sorts of interplay between bids and asks, volume and volatility, trade books and human psychology or computer algorithms, can explain a persistent and substantial deviation from simple random-walk dynamics and, at the same time, a surprisingly repeatable waiting-time distribution between large excursions? An appealing approach due to Clark [19], based on ideas about information flow and its effects on market activity, is to make a time change from the wall clock to a market clock, proportional to accumulated trading volume. A rich (and quite likely adequate) class of inhomogeneous stochastic processes can be represented as simple random walks through a stochastic time change (see H. Geman et al. [30]). Such a mechanism could explain the near invariance of waiting-time distributions across stocks through the strong correlations in market activity across stocks (though not the invariance to time increment or era). The focus of empirical studies has been on the demonstration of near normality of returns under suitable time changes based on market activity. But we found that functions of market activity that standardize returns to a nearly normal distribution fail to fit the time scales of exchangeability of real returns. It is possible that these time scales, by providing additional constraints, will be useful in the further study of mechanisms and their formulation as models.

Appendix A

Monte Carlo Sampling of Sequences with Constrained Transition Counts. Given a set of states $S = \{1, \ldots, M\}$ and a sequence of observations $o_1, \ldots, o_n \in S$, and assuming that $o_1, \ldots, o_n$ is generated by a first-order, but otherwise unknown, Markov process P, the general problem is to produce samples from P that have the same likelihood as $o_1, \ldots, o_n$. If $n_{st} = \#\{k \in \{2, \ldots, n\} : o_{k-1} = s, o_k = t\}$, then any sequence $\tilde{o}_1, \ldots, \tilde{o}_n$ that starts at o_1 ($\tilde{o}_1 = o_1$) and has n_{st} transitions from s to t for each $s, t \in S$ (henceforth, any "conforming sequence"), has the same likelihood as $o_1, \ldots, o_n$ (and, incidentally, ends at $\tilde{o}_n = o_n$). Consequently, iid samples from P conditioned on $\{n_{st}\}_{s,t \in S}$ can be generated by sampling from the uniform distribution on conforming sequences. The connection to $k'th$-order binary Markov processes (§II.C) $Z_1, \ldots, Z_N$ is made by taking $M = 2^k$, and coding successive sequences $(Z_{i+1}, \ldots, Z_{i+k})$ as elements of S.

The problem of producing samples from the uniform distribution on conforming sequences has been studied in computational biology, discrete mathematics, and statistics. Altschul and Erickson [3] provided an algorithm that is suitable for small state spaces, involving an acceptance/rejection step which quickly becomes inefficient for large M. Based on results by Aldous [1] and Broder [16], Kandel et al. [43] (see also Besag and Mondal [10]) introduced a modification of the Altschul and Erickson scheme that is fast and involves no rejections. The method we chose is very simple and involves no rejections, but is a little less statistically efficient in that it samples from the subset of conforming sequences that share the final transition ($\tilde{o}_{n-1} = o_{n-1}$ and $\tilde{o}_n = o_n$):

Step 1 Generate a table of transitions, with one row for each state $s \in S$. The first entry in row $s \in S$ is the pair (s, t) corresponding to the transition $(s \to t)$ from the first visit by $o_1, \ldots, o_n$ to s. The second entry is the transition from the second visit to s, and so-on.

Step 2 Permute all but the last entry of row s, randomly from the uniform distribution on permutations, for every $s \neq o_n$. Permute all of the entries for the row $s = o_n$.

Step 3 Beginning in state o_1 ($\tilde{o}_1 = o_1$), read off the sequence of states defined by the permuted transition table.

The sample, $\tilde{o}_1, \ldots, \tilde{o}_n$, is from the uniform distribution on conforming sequences that share with $o_1, \ldots, o_n$ the last transition out of each state $s \neq o_n$.

Part III
Computer Vision

Chapter 5

On the formula of distributions on hierarchical models

When talking about object detection or recognition in computer vision, we can characterize most approaches as either generative or discriminative according to whether or not a probability distribution of the image features is modeled.

The idea of the discriminative method is to compute or minimize a cost function, which consists of penalty functions, regularization and kernel functions. The cost function may come from our understanding of the object including the features of the object (eg. edges, corners, contours, gradients, etc). Many discriminative model and classification machines have been applied to computer vision. These include for example: Support Vector Machine, Neural Network and Boosting(see [12]).

In contrast, the generative approach is based on a probability model $P(\vec{x}, Y)$, to compute a likelihood function $P(\vec{x}|Y)$ given an image Y, where $\vec{x}$ is in terms of interpretations or classes. The main difference between the discriminative model and generative model is that the generative model can generate image Y, but a discriminative model can not.

Many generative models, like Gaussian mixture models, hidden Markov models or Markov random fields, have been used in many areas. In particular, Gaussian mixture models are commonly used in the computer vision field (e.g. [44, 28, 29]) because of their simple structure. However, as the object becomes more and more complicated, the traditional generative models are not adequate to portray the architecture of the object. In recent decades, researchers began to build models hierarchically due to their rich representations and reuseable properties. For instance, we can expand the representation of current object models by allowing for structure variation using "and-or" graphs(see [71]) and by using context-sensitive relations among the parts(e.g. [41], [71]). In addition, building the "reusable" parts among many categories is feasible in a hierarchical setting(e.g [24]), and it might be key to scaling up to the modeling of large numbers(e.g. thousands) of object categories. From a computational standpoint, coarse-to-fine structures for efficient detection can be built in hierarchical models, and they provide a tractable framework for combining bottom-up and top-down computation(e.g. [60], [45]). A growing number of hierarchical models achieve state-of-the-art performance in a growing number of applications. Some are biologically motivated (e.g. [57]), and others are computationally motivated (e.g. [25]). Some involve learned hierarchies (e.g. [52]); others are hand-designed (e.g. [41], [42]). Some are deep hierarchies (e.g. [66, 49] and [68]), some are mid-level hierarchies ([15]), and others are shallow (e.g. POP model [5], [2], Constellation model [65, 23]).

I will focus on the category of hierarchical generative models due to my belief in Baysian modeling. The hierarchical generative model is composed of two components: the prior distribution $p(\vec{x})$ on image "interpretations" or "parses", and the conditional data distribution (or the conditional likelihood function) $P(Y|\vec{x})$ on the image given its interpretation. In this chapter, we will focus on the first component, prior distribution, and then explore conditional modeling in the next chapter.

We plan to apply this model to do Bayesian scene analysis through a prior distribution on scene "parses" (interpretations). Parses are represented in a graphical model. The components of a parse are low-to-high-level(bottom-up) abstract variables such as "edge," "eye," "face," "person," "people," or "crowd." To emphasize the reusability of these "parts", the vertices of the graph are called bricks (as in Lego bricks). The specific assignment is application dependent. For example, in the application of reading license plates [41], [42], the semantic bricks represent different meanings as shown in Figure 5.1.

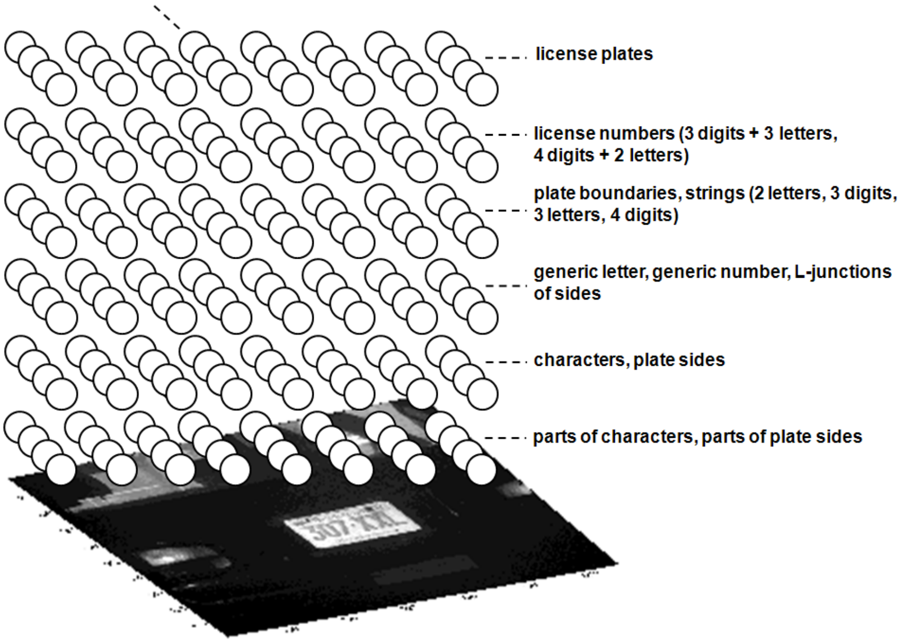

license plates
license numbers (3 digits + 3 letters, 4 digits + 2 letters)
plate boundaries, strings (2 letters, 3 digits, 3 letters, 4 digits)
generic letter, generic number, L-junctions of sides
characters, plate sides
parts of characters, parts of plate sides
307·XX

Figure 5.1: Semantic hierarchy for plate-reading application

Let us first introduce some notations we are going to use in this thesis. The following table is a reference of the notations. We will explain the notations by a simple composition system in the Figure 5.2.

Notation

$\mathcal{B}$	finite set of bricks
$\mathcal{T} \subseteq \mathcal{B}$	terminal bricks
$x^\alpha \in \{0, 1, \ldots, n^\alpha\},\ n^\alpha \geq 1$	states of the α brick, $\alpha \in \mathcal{B}$ (particular interpretations of the α brick)
$\vec{x} = \{x^\beta : \beta \in \mathcal{B}\}$	state of all bricks (interpretation)
$\{\epsilon_i^\alpha\}_{i=0}^{n^\alpha},\ 0 \leq \epsilon_i^\alpha \leq 1,\ \sum_{i=0}^{n^\alpha} \epsilon_i^\alpha = 1$	state probabilities, $\alpha \in \mathcal{B}$
$C_i^\alpha \subseteq \mathcal{B},\ \alpha \in \mathcal{B} \setminus \mathcal{T}$	i'th set of children of α, $1 \leq i \leq n^\alpha$, ($C_i^\alpha \neq C_j^\alpha$ when $i \neq j$)
$D^\alpha \subseteq \mathcal{B},\ \alpha \in \mathcal{B} \setminus \mathcal{T}$	the set of all possible descendant bricks of α,
$\vec{x}^{D^\alpha} = \{x^\beta : \beta \in D^\alpha\}$	state vector of the descendant bricks of α

Interpretation

Consider a directed acyclic graph (DAG) $\mathcal{G}$ defined by

- A vertex for every brick $\beta \in \mathcal{B}$
- A directed edge from α to β if $\beta \in C_i^\alpha$ for some $i \in \{1, 2, \ldots, n^\alpha\}$

An "interpretation" $\vec{x}$ is defined as an assignment of states to $\{x^\beta\}_{\beta \in \mathcal{B}}$ such that $\alpha \in \mathcal{B} \setminus \mathcal{T}$ and $x^\alpha > 0 \Rightarrow x^\beta > 0\ \forall \beta \in C_{x^\alpha}^\alpha$. Let $\mathcal{I}$ be the set of interpretations. If we declare a brick α "on" when $x^\alpha > 0$, and if we call $C_{x^\alpha}^\alpha$ the chosen children of brick α in state $x^\alpha > 0$, then an interpretation is a state vector $\vec{x}$ in which the chosen children of every non-terminal "on" brick are themselves "on".

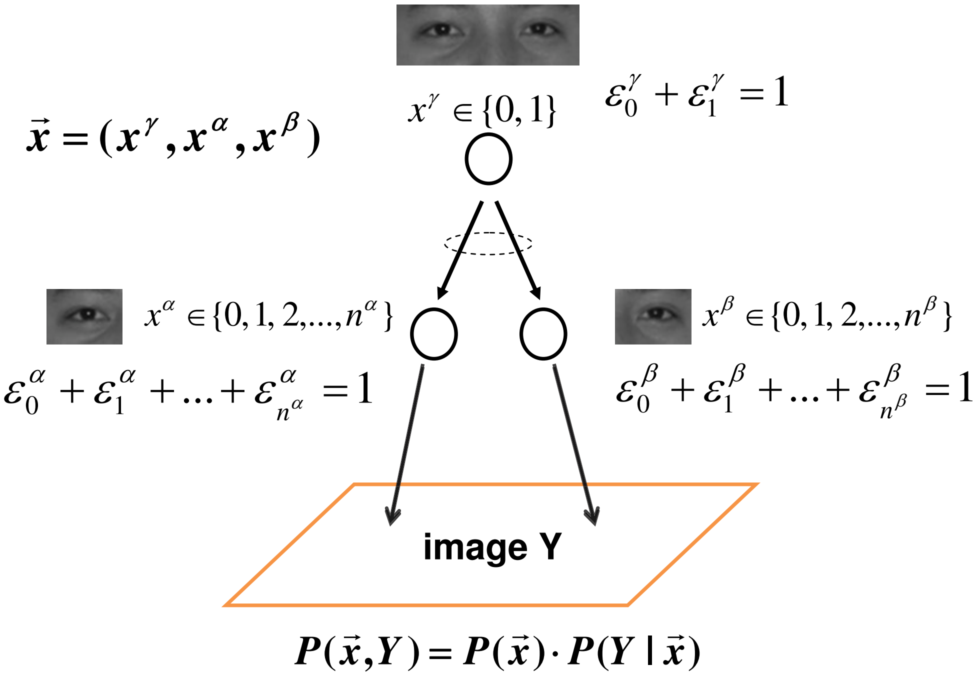

$\vec{x} = (x^\gamma, x^\alpha, x^\beta)$
$x^\gamma \in \{0, 1\}$
$\varepsilon_0^\gamma + \varepsilon_1^\gamma = 1$
$x^\alpha \in \{0, 1, 2, ..., n^\alpha\}$
$\varepsilon_0^\alpha + \varepsilon_1^\alpha + ... + \varepsilon_{n^\alpha}^\alpha = 1$
$x^\beta \in \{0, 1, 2, ..., n^\beta\}$
$\varepsilon_0^\beta + \varepsilon_1^\beta + ... + \varepsilon_{n^\beta}^\beta = 1$
image Y
$P(\vec{x}, Y) = P(\vec{x}) \cdot P(Y \mid \vec{x})$

Figure 5.2: a DAG of a simple hierarchical model

Let us use a simple example, a pair of eyes, and its directed acyclic graph to illustrate it. In Figure 5.2, we have three bricks, γ, α, β, which correspond to "pair of eyes," "right eye," "left eye". The γ brick "on" ($x^\gamma > 0$) means that the pair of eyes is present, and "off" means that is not. To be an interpretation for $\vec{x}$, its children bricks, α brick and β brick have to be "on" if their parent brick γ is "on". It is natural that we declare that both the right eye and the left eye are present if we say that the pair of eyes is present. Furthermore, when α brick is "on", x^α could be any integer from 1 to n^α. Each integer represent a different way or state to interpret the right eye or a category of right eyes, for instance: closed eye, open eye, angry eye, American eye, Chinese eye, different poses of the eye, etc.. Therefore, n^α could be a huge number of all the interpretations for the right eye. Similarly, we may have different pair of eyes; n^γ can also be a big number(in Figure 5.2 n^γ=1).

More generally, we can consider a more complicated composition structure. For example, a human face can be composed of many parts, like the nose, mouth, a pair of ears and a pair of eyes, and, for each part, we can build its own composition system, like in the previous example. We have built the hierarchy of a pair of eyes composed of its right eye and left eye. Moreover, we can look at the right eye or the left eye in more detail, as for example the right or left corner of the eye, the lower eyelid, the upper eyelid, the pupil, the iris etc. Furthermore, all the parts and details of every level could be shared in a different parent or ancestor brick state. For instance, Chinese face and Korean face could have similar eyes, and Indian eye and Chinese eye may have similar pupils. Thus, they are organized based on hierarchy and reusability, and these models are full of rich representations in the hierarchical system as shown in Figure 5.3.

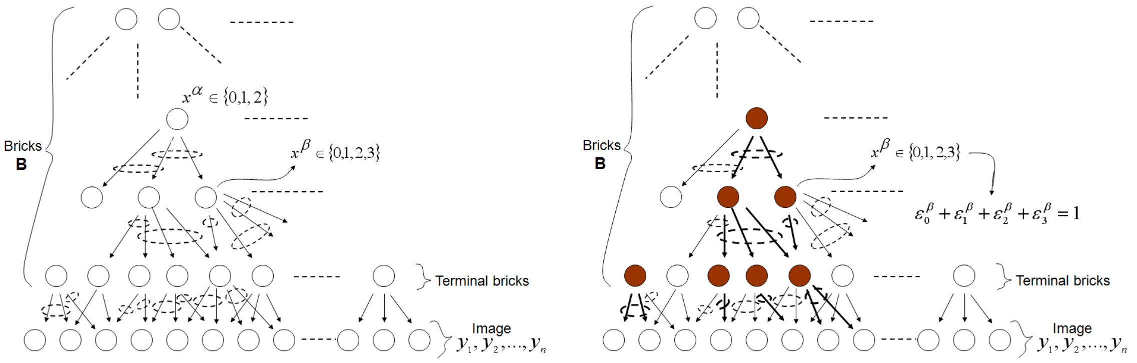

$x^{\alpha} \in \{0,1,2\}$
Bricks
B
$x^{\beta} \in \{0,1,2,3\}$
Terminal bricks
Image
$y_1, y_2, \ldots, y_n$
Bricks
B
$x^{\beta} \in \{0,1,2,3\}$
$\varepsilon_0^{\beta} + \varepsilon_1^{\beta} + \varepsilon_2^{\beta} + \varepsilon_3^{\beta} = 1$
Terminal bricks
Image
$y_1, y_2, \ldots, y_n$

Figure 5.3: Architecture. Left. A hierarchy of "bricks," each representing a disjunction of conjunctions. Bottom row is the image (pixel) data and the row above it is the set of terminal bricks. The state of a brick signals a chosen set of children. Right. An "interpretation," which is an assignment of states such that the chosen children of any "on" brick are also on. There can be multiple roots and shared subtrees. Filled circles represent on bricks (non-zero states), and highlighted edges represent chosen children.

After building the hierarchical structure, the next step is to put a probability distribution on it[i.e. to construct the prior distribution $P(\vec{x})$]. One of the easier models is the Markov Random Field with respect to the directed acyclic graph. We usually call this kind of Markov structure "Markov Backbone" or "Context-free Grammar."

5.1 Probabilities on Markov Backbone

For $\vec{x} \in \mathcal{I}$, we define the *below set* $B = B(\vec{x})$ by

$$B = \{\beta \in \mathcal{B} : \beta \in C^{\alpha}_{x^{\alpha}}, \text{ for some } \alpha \in \mathcal{B} \setminus \mathcal{T} \text{ with } x^{\alpha} > 0\}$$

The Markov ("context-free") probability of an interpretation $\vec{x} \in \mathcal{I}$ is defined as

$$P_0(\vec{x}) = \frac{\prod_{\beta \in \mathcal{B}} (\epsilon^{\beta}_{x^{\beta}})}{\prod_{\beta \in B(\vec{x})} (1 - \epsilon^{\beta}_0)} \tag{5.1}$$

Remarks:

1. $\sum_{\vec{x} \in \mathcal{I}} P_0(\vec{x}) = 1$, *as can be seen by ordering $\mathcal{G}$ by generations, starting with the roots, and then generating a random $\vec{x}$ in the same order, according to ϵ^{α}_i, $i \in \{0, 1, \ldots, n^{\alpha}\}$, for any brick not chosen by a parent, and $\frac{\epsilon^{\alpha}_i}{1-\epsilon^{\alpha}_0}$, $i \in \{1, 2, \ldots, n^{\alpha}\}$, otherwise.*

2. $P_0(\vec{x})$ *is a 'Bayes Net' with respect to the DAG $\mathcal{G}$, and hence Markov with respect to the undirected 'moral' graph derived from $\mathcal{G}$.*

3. *There is an obvious connection to probabilistic context-free grammars: think of $\alpha \to \{\beta : \beta \in C^{\alpha}_i\}$ as a production, chosen with probability $\frac{\epsilon^{\alpha}_i}{1-\epsilon^{\alpha}_0}$. But keep in mind that there is no unique "start" symbol, that an interpretation can include many trees, that trees can share parts (instantiations overlap), and that there is a fixed topology (hence no recursion).*

In the license application [41], [42], Jin and Geman sampled from the Markov backbone, given the semantic assignment of bricks (in Figure 5.1) and the manually hardwired children sets. The left panel of Figure 5.4 shows a 4-digit sample under the Markov backbone. As seen from the figure, although the parts of each digit are present and they are in roughly the correct locations, neither the parts nor the digits are properly situated.

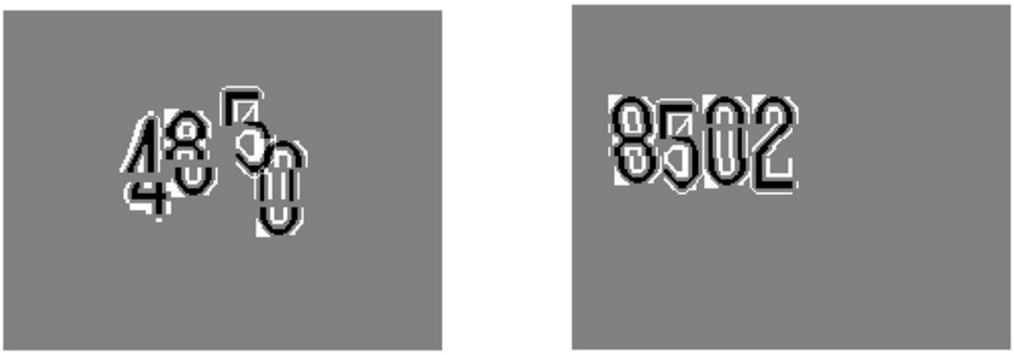


Figure 5.4: Samples from Markov backbone (left panel, '4850') and compositional distribution (right panel, '8502').

5.2 Content Sensitivity and Non-Markovian Perturbations

Most of the proposed generative models in the literature share the Markov property ([65], [61], [27], [32], [54, 55]) due to its computational advantage. But this context-free (Markov) property is problematic. Constituents, in vision and language, are composed with a likelihood that depends not just on their "labels," (stroke, letter, noun phrase, verb phrase, etc.), but also on the details of their instantiations (position, font, gender, tense, etc.). Biological-level ROC performance of an image-analysis system will almost certainly need to be content sensitive. This raises the difficult question of constructing useful non-Markovian probability distributions on hierarchical models. One approach, beginning with coding and description length, was explored in [33]. A different approach, through perturbations, is explored here.

Imagine that we have associated with every brick $\beta \in \mathcal{B}$ an attribute function (scalar valued or vector valued), $a^\beta(\vec{x})$. A prototypical example is the set of pose coordinates (or *relational* pose coordinates) of the chosen children or descendants of β (see Section 5.5). Depending on the depth of the instantiation of the children, $a^\beta(\vec{x})$ may depend on the states of the bricks that are several generations removed from β itself (grandchildren, great grandchildren, etc.). Thus, we usually assume

$a^\beta(\vec{x})$ is a function of $\vec{x}^{D^\beta}$.

Start with the Markov probability P_0, as defined in (5.1), and fix a particular brick $\gamma \in \mathcal{B}$. Under (5.1), a^γ has *some* distribution, $P_0(a^\gamma|x^\gamma)$, for every state, $x^\gamma \in \{0, 1, \dots, n^\gamma\}$. If, say, $a^\gamma(\vec{x}^{D^\gamma})$ is the vector of poses of the chosen children or descendants of γ, then it is hopeless that P_0 corresponds to the *empirical* (or "real-world") distribution on the positions of the parts of γ. After all, (5.1) is context free and, in particular, the instantiations of the chosen children of γ are independent (Markov property).

Let $P_c^\gamma(a^\gamma|x^\gamma)$ (as opposed to $P_0(a^\gamma|x^\gamma)$) be the correct conditional distribution on the attribute a^γ and call it a *conditional constraint.* We may have several conditional constraints. Could we find a probability distribution P satisfying these constraints? This raises the following two questions:

1. Given $\{P_c^\gamma(a^\gamma|x^\gamma) : x^\gamma > 0\}_{\gamma \in \mathcal{B} \setminus \mathcal{T}}$, does there exist a distribution P on $\vec{x} \in \mathcal{I}$ such that $P(a^\gamma|x^\gamma) = P_c^\gamma(a^\gamma|x^\gamma)$, $\forall x^\gamma > 0$?

2. If there exists such a P, how can it be constructed?

The following two sections address these two questions in more depth.

5.3 Question1: Does There Exist a Probability Distribution Satisfying the Conditional Constraints

To be concrete, the attribute function $a^\gamma = a^\gamma(\vec{x})$ is a function of $\vec{x}$. Its domain is $\mathcal{I}$ and its range is

$$R^\gamma = \{a^\gamma(\vec{x}) | \vec{x} \in \mathcal{I}\}.$$

We can regard a^γ as a random variable taking values from R^γ. Let

$$R_j^\gamma = \{a^\gamma(\vec{x}) | x^\gamma = j, \vec{x} \in \mathcal{I}\}.$$

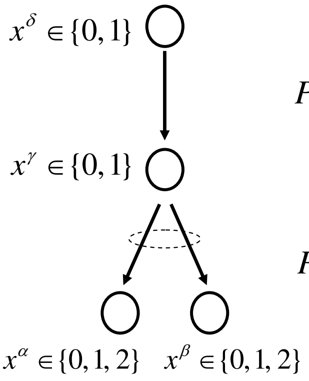
If $P_c^\gamma(\cdot|x^\gamma = j)$ is a proper probability, then two consequences of these definitions are

$$\begin{array}{ll} 1. & \sum_{a^\gamma \in R_j^\gamma} P_c^\gamma(a^\gamma|x^\gamma = j) = 1,\ \gamma \in \mathcal{B} \setminus \mathcal{T}, \forall j = 1, 2, ..., n^\gamma \\ 2. & P_c^\gamma(a^\gamma|x^\gamma = j) = 0,\ \forall a^\gamma \in R^\gamma \setminus R_j^\gamma,\ \gamma \in \mathcal{B} \setminus \mathcal{T}, \forall j = 1, 2, ..., n^\gamma. \end{array} \tag{5.2}$$

Now, given these conditional constraints, the question can be specified as to whether or not we can find a *non-trivial* probability distribution P consistent with these constraints, where "non-trivial" means

$$P(x^\gamma = i) > 0,\ \gamma \in \mathcal{B}, \forall i \in 1, ..., n^\gamma \tag{5.3}$$

(i.e. the probability of the set $\{x^\gamma = i\}$ that we conditioned on is positive). However, under (5.2) above, the answer is no or at least not always. Let us first see the counter example in Figure 5.5.



$$P_c^\delta((x^\alpha, x^\beta) \mid x^\delta = 1) = \begin{cases} 0.5 & \text{if } x^\alpha = x^\beta \neq 0 \\ 0 & \text{else} \end{cases}$$

$$P_c^\gamma((x^\alpha, x^\beta) \mid x^\gamma = 1) = \begin{cases} 0.5 & \text{if } (x^\alpha, x^\beta) = (1,2) \text{ or } (2,1) \\ 0 & \text{else} \end{cases}$$

Figure 5.5: The counter example for question 1

In the counter example, if P is the probability distribution satisfying the two constraints as well as (5.2), we have

$$\begin{aligned} 0 < {} & P(x^\alpha = 1, x^\beta = 1|x^\delta = 1)P(x^\delta = 1) \\ = {} & P(x^\alpha = 1, x^\beta = 1, x^\delta = 1) \\ \leq {} & P(x^\alpha = 1, x^\beta = 1, x^\gamma = 1) \\ = {} & P(x^\alpha = 1, x^\beta = 1|x^\gamma = 1)P(x^\gamma = 1) = 0 \end{aligned}$$

and we thus get a contradiction. Therefore, the question remains : when does the P exist, and what is the condition under which it exists?

Mathematically, for every attribute function a^{γ} and its conditioned set $\{x^{\gamma} = j\}$, we can find the σ-algebra $\sigma(a^{\gamma}, \{x^{\gamma} = j\})$ generated by a^{γ} and $\{x^{\gamma} = j\}$. If we have n constraints, we will have n σ-algebras, say $F_1, F_2, ..., F_n$. Thus, we need to check whether or not there is a conflict in the $\sigma-$algebra $F = \sigma(F_1, F_2, ..., F_n)$. For the example in Figure 5.5, the two constraints conflict with the monotone property: $\forall A, B \in F, A \subseteq B \Longrightarrow 0 \leq Pro(A) \leq Pro(B) \leq 1 = Pro(\Omega)$. Under the given constraints, the probability of $A = \{x^{\alpha} = 1, x^{\beta} = 1, x^{\delta} = 1\}$ has to be greater than the probability of $B = \{x^{\alpha} = 1, x^{\beta} = 1, x^{\gamma} = 1\}$, but set A is contained in set B.

However, it is difficult in practice to check whether there is a conflict or not. Thus, we will provide a *sufficient condition* which is easy to verify in practice. Let us first introduce some definitions and notations about the "*level*" in a composition system(see the Figure 5.6):

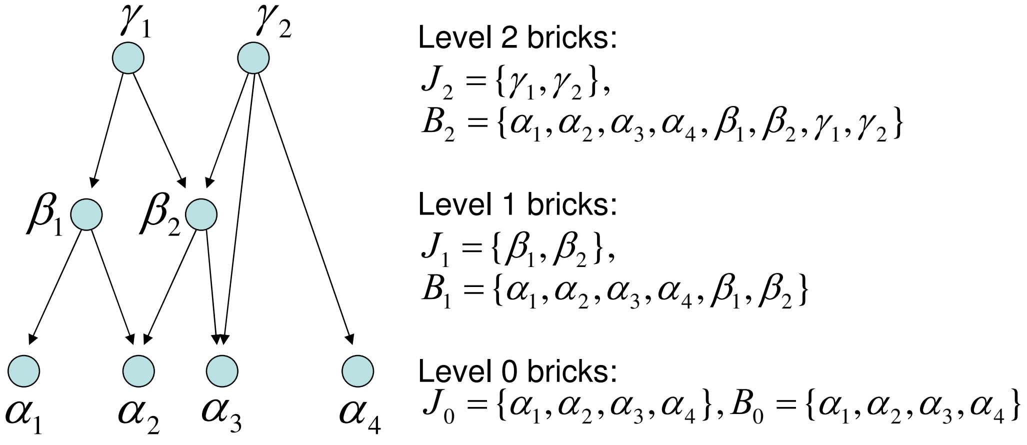

γ_1
γ_2
β_1
β_2
α_1
α_2
α_3
α_4
Level 2 bricks:
$J_2 = \{\gamma_1, \gamma_2\}$,
$B_2 = \{\alpha_1, \alpha_2, \alpha_3, \alpha_4, \beta_1, \beta_2, \gamma_1, \gamma_2\}$
Level 1 bricks:
$J_1 = \{\beta_1, \beta_2\}$,
$B_1 = \{\alpha_1, \alpha_2, \alpha_3, \alpha_4, \beta_1, \beta_2\}$
Level 0 bricks:
$J_0 = \{\alpha_1, \alpha_2, \alpha_3, \alpha_4\}, B_0 = \{\alpha_1, \alpha_2, \alpha_3, \alpha_4\}$

Figure 5.6: The figure illustration and notations for each level

- *level 0 brick*: we call a brick a level 0 brick if it is a terminal brick.
- *level h brick*: we call a brick a level h brick if the brick has at least one children brick in level $h-1$, and if all of its child bricks are among level 1 to level $h-1$.

- J_h: the set of all *level h* bricks.
- $\mathcal{B}_h$: the set of all bricks from level 0 to level h.
- $\mathcal{G}_{\mathcal{B}_h}$: the sub-graph of $\mathcal{G}$ whose vertex set is $\mathcal{B}_h$.
- $\mathcal{I}_{\mathcal{B}_h} = \{\vec{x}^{\mathcal{B}_h} | \vec{x} \in \mathcal{I}\}$: the set of sub-interpretations of $\mathcal{I}$ for the sub-graph $\mathcal{G}_{\mathcal{B}_h}$.
- $\mathcal{I}^+_{\mathcal{B}_h} = \{\vec{x}^{\mathcal{B}_h} | \vec{x}^{\mathcal{B}_h} \in \mathcal{I}_{\mathcal{B}_h}, \vec{x}^{\mathcal{B}_h} > 0\}$.

In general, if the attribute function $a^\gamma(\vec{x})$ only depends on the descendant bricks $\vec{x}^{D^\gamma}$, then the constraints restrict the conditional probabilities on their corresponding descendant bricks. In the following theorem, we consider the *stricter* constraints whose attribute function is $a^\gamma(\vec{x}) = \vec{x}^{\mathcal{B}_{l-1}}$ where l is the level of brick γ, and we get a sufficient condition for the existence of P. Notice that if we can find a probability distribution in this *severe* circumstance, we certainly can create a probability distribution in general circumstances.

Theorem 1. *For the directed acyclic graph $\mathcal{G}$, given any set of conditional constraints $\{P^\gamma_c(\vec{x}^{\mathcal{B}_{l-1}} | x^\gamma = j) : \ j = 1, 2, ..., n^\gamma, \ \gamma \in J_l, \ l = 1, 2, ..., L\}$ with*

$$\{\vec{x}_{\mathcal{B}_{l-1}} | P^\gamma_c(\vec{x}^{\mathcal{B}_{l-1}} | x^\gamma = j) > 0\} = \mathcal{I}^+_{\mathcal{B}_{l-1}},$$

for all $j = 1, 2, ..., n^\gamma$, $\gamma \in J_l$, and $l = 1, ..., L$, then there exists at least one distribution P on $\mathcal{I}$ such that

$$P(x^\gamma = i) > 0, \ \gamma \in \mathcal{B}, \ \forall i \in 1, ..., n^\gamma$$

and

$$P(\vec{x}^{\mathcal{B}_{l-1}} | x^\gamma = j) = P^\gamma_c(\vec{x}^{\mathcal{B}_{l-1}} | x^\gamma = j)$$

for all $j = 1, 2, ..., n^\gamma$, $\gamma \in J_l$, and $l = 1, ..., L$.

Proof. We will prove the theorem hierarchically by looking for the probability distribution P on $\mathcal{B}_l$, where l moves from the first level to the top level. Let us start from the first level.

Step1

Let $\mathcal{B}_0 = \{\alpha_1, \alpha_2, \alpha_3, ..., \alpha_n\}$ and $J_1 = \{\gamma_1, \gamma_2, \gamma_3, ..., \gamma_m\}$, so that

$$\mathcal{B}_1 = \{\alpha_1, \alpha_2, \alpha_3, ..., \alpha_n, \gamma_1, \gamma_2, \gamma_3, ..., \gamma_m\}.$$

Then, let $\{z_1, ..., z_N\} = \mathcal{I}^+_{\mathcal{B}_0}$ and $\{z_1, ..., z_N, z_{N+1}, ..., z_{\hat{N}}\} = \mathcal{I}_{\mathcal{B}_0}$, so that $N = |\mathcal{I}^+_{\mathcal{B}_0}|$ and $\hat{N} = |\mathcal{I}_{\mathcal{B}_0}|$. For simplicity, define

$$q^k_{i,j} = P^{\gamma_i}_c(\vec{x}^{\mathcal{B}_0} = z_k | x^{\gamma_i} = j)$$

for $k = 1, ..., N$ and define

$$P^k_{j_1,j_2,...,j_m} = P(\vec{x}^{\mathcal{B}_0} = z_k, x^{\gamma_1} = j_1, x^{\gamma_2} = j_2, ..., x^{\gamma_m} = j_m)$$

for $k = 1, ..., \hat{N}$. Now we want to construct the probability distribution P on $\mathcal{I}_{\mathcal{B}_1}$. We first set $P(x^{\gamma_i} = j) = t$ for all $i = 1, ..., m$ and $j = 1, ..., n^{\gamma_i}$. We then let

$$P^k_{x^{\gamma_1},...,x^{\gamma_m}} = \epsilon t \tag{5.4}$$

for all $(x^{\gamma_1}, ..., x^{\gamma_m})$ among which at least two of the x^{γ_i}'s are nonzero, and for $k \in \{1, ..., N\}$. Then, we have

$$\begin{aligned} q^k_{i,j}t &= P^k_{0,..,0,j,0,...,0} + \textstyle\sum_{(x^{\gamma_1},..,x^{\gamma_i-1},x^{\gamma_i+1},...x^{\gamma_m})\neq(0,0,....0)} P^k_{x^{\gamma_1},..,x^{\gamma_i-1},j,x^{\gamma_i+1},...x^{\gamma_m}} \\ &= P^k_{0,..,0,j,0,...,0} + (\textstyle\prod_{s\neq i}(1 + n^{\gamma_s}) - 1)\epsilon t. \end{aligned}$$

Therefore, we can choose the ϵ small enough so that

$$P^k_{0,...,0,j,0,...,0} = (q^k_{i,j} - (\prod_{s\neq i}(1 + n^{\gamma_s}) - 1)\epsilon)t > 0 \tag{5.5}$$

is well defined. Next, we sum up all of the probabilities assigned by the equation 5.4 and the equation 5.5, and call it θt. We can make t small enough such that $\theta t < 1$, and then we can set the remainder of the probabilities as

$$P^k_{0,0,...,0} = \frac{1 - \theta t}{\hat{N} + 1}$$

for all $k \in \{0, 1, ..., \hat{N}\}$, completing step1.

Step2

In this step, we assume that we have set up the probability, call it $\hat{P}$ up to level h. The ideal is to preserve $\hat{p}$ as a marginal probability distribution on $\mathcal{I}_{\mathcal{B}_h}$. Let us still assume that $\mathcal{B}_h = \{\alpha_1, \alpha_2, \alpha_3, ..., \alpha_n\}$ and $J_{h+1} = \{\gamma_1, \gamma_2, \gamma_3, ..., \gamma_m\}$. Then we let $\{z_1, ..., z_N\} = \mathcal{I}^+_{\mathcal{B}_h}$ and $\{z_1, ..., z_N, z_{N+1}, ..., z_{\hat{N}}\} = \mathcal{I}_{\mathcal{B}_h}$. Now, using the same notations and following the same procedure of Step1, we can create P by letting $P(x^{\gamma_i} = j) = t$ for all $i = 1, ..., m$ and $j = 1, ..., n^{\gamma_i}$, and by letting

$$P^k_{x^{\gamma_1},...,x^{\gamma_m}} = \epsilon t$$

for $(x^{\gamma_1}, ..., x^{\gamma_m})$ among which at least two of the x^{γ_i}'s are nonzero, and for $k \in \{1, ..., N\}$. Then we can make ϵ small enough to get

$$P^k_{0,...,0,j,0,...,0} = (q^k_{i,j} - (\prod_{s\neq i}(1 + n^{\gamma_s}) - 1)\epsilon)t > 0.$$

Now, we can establish the equation for $\hat{P}(z_k)$ for all $k \in \{1, 2, ..., N\}$:

$$\hat{P}(z_k) = P^k_{0,0,...,0} + \sum_{(x^{\gamma_1},...x^{\gamma_m}) \neq (0,...,0)} P^k_{x^{\gamma_1},...x^{\gamma_m}} \equiv P^k_{0,0,...,0} + C_k t.$$

Thus, we can let t be small enough such that for all $k \in \{1, 2, ..., N\}$,

$$P^k_{0,0,...,0} = p^k - C_k t > 0.$$

Finally, we set $P^k_{0,0,...,0} = \hat{P}(z_k)$ for all $k \in \{N + 1, ..., \hat{N}\}$, finishing Step2. We continue this process until we reach the top level. Then, the proof is complete. □

Under the condition

$$\{\vec{x}^{\mathcal{B}_{l-1}} | P^\gamma_c(\vec{x}^{\mathcal{B}_{l-1}} | x^\gamma = j) > 0\} = \mathcal{I}^+_{\mathcal{B}_{l-1}},$$

the above theorem indicates that the probability distribution P exists consistent with the given constraints. However, in practice, the constraints are usually not as strict as those of the theorem. The attribute function $a^\gamma(\vec{x})$ only depends on the set of the descendant bricks of γ. Specifying $\{P^\gamma_c(\vec{x}^{\mathcal{B}_{l-1}} | x^\gamma = j)\}$ is more than specifying $\{P^\gamma_c(a^\gamma(\vec{x}^{D^\gamma}) | x^\gamma = j)\}$. In particular, any specification of $\{P^\gamma_c(a^\gamma(\vec{x}^{D^\gamma}) | x^\gamma = j)\}$ can be re-written as at least one specification of $\{P^\gamma_c(\vec{x}^{\mathcal{B}_{l-1}} | x^\gamma = j)\}$(for example, dividing $\{P^\gamma_c(\vec{x}^{\mathcal{B}_{l-1}} | x^\gamma = j)\}$ equally among $\{P^\gamma_c(a^\gamma(\vec{x}^{D^\gamma}) | x^\gamma = j)\}$). Hence, the theorem above implies the following corollary which provides a sufficient condition for the existence of the common constraint probabilities.

Corollary: *For the directed acyclic graph $\mathcal{G}$ and the instantiation $\mathcal{I}$, given any set of conditional constraints $\{P^\gamma_c(a^\gamma | x^\gamma = j) : \ j = 1, 2, ..., n^\gamma, \ \gamma \in \mathcal{B} \setminus \mathcal{T}\}$ where the attribute function $a^\gamma = a^\gamma(\vec{x}) = a^\gamma(\vec{x}^{D^\gamma})$ only depends on the descendant bricks of γ, if*

$$\{a^\gamma | P^\gamma_c(a^\gamma | x^\gamma = j) > 0\} = R^\gamma_j \ \big(= \{a^\gamma(\vec{x}) | x^\gamma = j, \vec{x} \in \mathcal{I}\}\big) \tag{5.6}$$

for $j = 1, 2, ..., n^\gamma$ and $\gamma \in \mathcal{B} \setminus \mathcal{T}$, then there exists at least one distribution P on $\mathcal{I}$ such that

$$P(x^\gamma = i) > 0, \ \gamma \in \mathcal{B}, \ \forall i \in 1, ..., n^\gamma$$

and

$$P(a^\gamma | x^\gamma = j) = P^\gamma_c(a^\gamma | x^\gamma = j)$$

for all $j = 1, 2, ..., n^\gamma$, $\gamma \in \mathcal{B} \setminus \mathcal{T}$.

5.4 Question2: How to Achieve the Probability Distribution Satisfying the Conditional Constraints

We have proved in the previous section the existence of a distribution P on $\mathcal{I}$ that satisfies all of the conditional constraints $\{P_c(a^\alpha|x^\alpha) : x^\alpha > 0\}_{\alpha \in \mathcal{B} \setminus \mathcal{T}}$, if the conditional constraints satisfy (5.6). Now, given the existence of one distribution P satisfying the desired constraints, there may be many such distributions satisfying the desired constraints. But how do we construct such a probability distribution? Let us start with Markov probability P_0 defined in (5.1). Now, one way to achieve $P_c^\alpha(a^\alpha|x^\alpha)$ is to "perturb" P_0, so as to correct the conditional a^α distributions by choosing the new distribution P^* which is closest to P_0, subject to the constraint that $P^*(a^\alpha|x^\alpha) = P_c^\alpha(a^\alpha|x^\alpha)$ for all a^α and all x^α. If, by "closest," we mean that the relative entropy $D(P_0||P^*)$ (Kullback-Leibler divergence) is minimized, then it is an easy calculation to show that

$$P^*(\vec{x}) = P_0(\vec{x}) \frac{P_c^\alpha(a^\alpha(\vec{x})|x^\alpha)}{P_0(a^\alpha(\vec{x})|x^\alpha)}.$$

Remarks:

1. *The particular distribution $P_c^\alpha(a^\alpha|x^\alpha = 0)$ is largely irrelevant to the problem of modeling an object α. This would be very hard to measure, and in any case can be taken as $P_0(a^\alpha|x^\alpha = 0)$ so that there is no perturbation at all unless the α brick is "on".*

2. *Bearing in mind the considerations of the previous remark, P^* is a "perturbation" in the sense that P is only altered in the event of $x^\alpha > 0$ (α "on"), which is presumably quite rare for most bricks, as they represent particular parts, objects, or collections of objects.*

3. *In general P^* is no longer Markov, but it is still normalized.*

4. *In most cases of interest, $a^\alpha(\vec{x})$ would be a function only of x^α and its possible progeny, meaning that every brick could appear in its instantiations.*

Evidently, the process can be repeated at other bricks, enforcing a brick-dependent attribute distribution at each step. For any "brick visitation schedule," $\alpha_1, \alpha_2, \ldots, \alpha_{|\mathcal{B}|}$, with $\{\alpha_1, \alpha_2, \dots, \alpha_{|\mathcal{B}|}\} = \mathcal{B}$, we end up with a distribution

$$\begin{aligned} P^*(\vec{x}) &= P_0(\vec{x}) \prod_{v=1}^{|\mathcal{B}|} \frac{P_c^{\alpha_v}(a^{\alpha_v}(\vec{x})|x^{\alpha_v})}{\tilde{P}^{\alpha_v}(a^{\alpha_v}(\vec{x})|x^{\alpha_v})} \\ &= \frac{\prod_{\beta \in \mathcal{B}}(\epsilon^{\beta}_{x^\beta})}{\prod_{\beta \in B(\vec{x})}(1-\epsilon^{\beta}_{0})} \prod_{v=1}^{|\mathcal{B}|} \frac{P_c^{\alpha_v}(a^{\alpha_v}(\vec{x})|x^{\alpha_v})}{\tilde{P}^{\alpha_v}(a^{\alpha_v}(\vec{x})|x^{\alpha_v})}, \end{aligned} \tag{5.7}$$

where $\tilde{P}^{\alpha_v}(a^{\alpha_v}(\vec{x})|x^{\alpha_v})$ is the distribution on a^{α_v} given x^{α_v} *at the time of the visit to the α_v brick*, and where $\tilde{P}^{\alpha_v}(a^{\alpha_v}(\vec{x})|x^{\alpha_v}) = P_0(a^{\alpha_v}(\vec{x})|x^{\alpha_v})$ when $v = 1$. The result is unsatisfactory in two regards:

1. The distribution turns out to be different for different visitation schedules.

2. Each perturbation, while establishing a desired conditional distribution $P_c^{\alpha}(a^{\alpha}|x^{\alpha})$, perturbs the previously established distributions, so that the already-visited bricks no longer have, precisely, the desired attribute distributions. (This applies to the epsilon probabilities as well.)

The study of specific examples suggests that the attribute functions $\{a^{\alpha}(\vec{x})\}_{\alpha \in \mathcal{B}}$ together with the attribute (conditional) distributions $\{P_c^{\alpha}(a^{\alpha}|x^{\alpha})\}_{\alpha \in \mathcal{B}}$ will usually under-determine the distribution on $\mathcal{I}$. There are typically many distributions with the desired constraints. This raises the related questions about convergence: is it true that an iterative procedure that visits every site infinitely often converges to a distribution with the desired attribute probabilities? In this section, we will prove that we can iteratively perturb the Markovian distribution P_0 such that, under a hypothesis of "non triviality," it will converge to an asymptotic distribution P^* on $\mathcal{I}$, and that P^* will satisfy the conditional constraints as well. "Non trivial" means that $\forall \alpha \in \mathcal{B} \setminus \mathcal{T}$, $\forall x^{\alpha} > 0$,

$$P_c(a^{\alpha}(\vec{x})|x^{\alpha}) \text{ has the same support as } P_0(a^{\alpha}(\vec{x})|x^{\alpha}), \tag{5.8}$$

where $P_0(a^{\alpha}(\vec{x})|x^{\alpha})$ is the marginal conditional distribution of $a^{\alpha}(\vec{x})$ under P_0.

Let N be the number of bricks in $\mathcal{B}$, and denote $\mathcal{B}$ as $\mathcal{B} = \{\alpha_1, \alpha_2, \dots, \alpha_N\}$. Let

$M_i \triangleq n^{\alpha_i}$, i.e., $x^{\alpha_i} \in \{0, 1, \ldots, M_i\}$, $\forall i \in \{1, \ldots, N\}$. Let P_k be the distribution after k steps of perturbations. We define an infinite sequence of perturbations as follows:

$$\begin{aligned}
P_1(\vec{x}) &= P_0(\vec{x}) \cdot \left(\frac{P_c(a^{\alpha_1}(\vec{x})|x^{\alpha_1} = 1)}{P_0(a^{\alpha_1}(\vec{x})|x^{\alpha_1} = 1)} \right)^{1_{\{x^{\alpha_1}=1\}}} \\
P_2(\vec{x}) &= P_1(\vec{x}) \cdot \left(\frac{P_c(a^{\alpha_1}(\vec{x})|x^{\alpha_1} = 2)}{P_1(a^{\alpha_1}(\vec{x})|x^{\alpha_1} = 2)} \right)^{1_{\{x^{\alpha_1}=2\}}} \\
&\vdots \\
P_{n^{\alpha_1}+1}(\vec{x}) &= P_{n^{\alpha_1}}(\vec{x}) \cdot \left(\frac{P_c(a^{\alpha_2}(\vec{x})|x^{\alpha_2} = 1)}{P_{n^{\alpha_1}}(a^{\alpha_2}(\vec{x})|x^{\alpha_2} = 1)} \right)^{1_{\{x^{\alpha_2}=1\}}} \\
&\vdots
\end{aligned}$$

In general, let $M_s = \sum_i^N M_i$. With this notation, $\forall m \in \{0, 1, 2, \ldots\}$, $\forall t \in \{1, 2, \ldots, M_s\}$, and $\forall l \in \{1, 2, \ldots, N\}$, if $t \in [\sum_{i=1}^{l-1} M_i + 1, \sum_{i=1}^{l} M_i]$, we have a general perturbation formula as follows:

$$P_{mM_s+t}(\vec{x}) = P_{mM_s+t-1}(\vec{x}) \cdot \left(\frac{P_c(a^{\alpha_l}(\vec{x})|x^{\alpha_l} = t - \sum_{i=1}^{l-1} M_i)}{P_{mM_s+t-1}(a^{\alpha_l}(\vec{x})|x^{\alpha_l} = t - \sum_{i=1}^{l-1} M_i)} \right)^{1_{\{x^{\alpha_l}=t-\sum_{i=1}^{l-1} M_i\}}} \tag{5.9}$$

with the exception that $P_{mM_s+t}(\vec{x}) = P_{mM_s+t-1}(\vec{x})$, if $\alpha_l \in \mathcal{T}$. We stop whenever $P_k(a^\alpha(\vec{x})|x^\alpha) = P_c(a^\alpha(\vec{x})|x^\alpha)$, $\forall \alpha \in \mathcal{B} \setminus \mathcal{T}$, $\forall x^\alpha > 0$, where $k = mM_s + t$. Otherwise we continue . (The "non-triviality" condition defined in (5.8) guarantees that the denominator of the ratio in each perturbation defined above is non-zero if its corresponding numerator is non-zero.) To ease the notation, we will simply use the form in (5.9) for all of the bricks $\alpha_l \in \mathcal{B}$, while assuming that

$$\left(\frac{P_c(a^{\alpha_l}(\vec{x})|x^{\alpha_l} = t - \sum_{i=1}^{l-1} M_i)}{P_{mM_s+t-1}(a^{\alpha_l}(\vec{x})|x^{\alpha_l} = t - \sum_{i=1}^{l-1} M_i)} \right)^{1_{\{x^{\alpha_l}=t-\sum_{i=1}^{l-1} M_i\}}} = 1, \quad \text{if } \alpha_l \in \mathcal{T}.$$

Theorem 2. *Under the non-triviality condition, if $\mathbb{P}_0$ is non-empty (i.e there exists a distribution $P(\vec{x})$ on the interpretations $\mathcal{I}$ s.t. $\forall \alpha \in \mathcal{B} \setminus \mathcal{T}$, $\forall x^\alpha > 0$, $P(a^\alpha(\vec{x})|x^\alpha) = P_c(a^\alpha(\vec{x})|x^\alpha)$), then the sequence of perturbations $\{P_k\}_k$ defined above gives us a pointwise convergent distribution P^* on $\mathcal{I}$, i.e. $P_k(\vec{x}) \overset{k\to\infty}{\longrightarrow} P^*(\vec{x})$, $\forall \vec{x} \in \mathcal{I}$. The asymptotic distribution P^* satisfies: $\forall \alpha \in \mathcal{B} \setminus \mathcal{T}$, $\forall x^\alpha > 0$, $P^*(a^\alpha(\vec{x})|x^\alpha) = P_c(a^\alpha(\vec{x})|x^\alpha)$.*

Proof. Let

$$\{\gamma_k\}_{k=1}^{\infty} = \{\underbrace{\alpha_1, \alpha_1, ..., \alpha_1}_{n^{\alpha_1}}, \underbrace{\alpha_2, \alpha_2, ..., \alpha_2}_{n^{\alpha_2}}, ..., \underbrace{\alpha_N, \alpha_N ..., \alpha_N}_{n^{\alpha_N}}, \underbrace{\alpha_1, \alpha_1, ...}_{...}\}$$

and $\{B_k\}_{k=1}^{\infty} = \{\{\vec{x} : x^{\alpha_1} = 1\}, \{\vec{x} : x^{\alpha_1} = 2\}, ..., \{\vec{x} : x^{\alpha_1} = n^{\alpha_1}\}, \{\vec{x} : x^{\alpha_2} = 1\}, \{\vec{x} : x^{\alpha_2} = 2\}, ..., \{\vec{x} : x^{\alpha_2} = n^{\alpha_2}\}, ..., \{\vec{x} : x^{\alpha_N} = 1\}, \{\vec{x} : x^{\alpha_N} = 2\}, ..., \{\vec{x} : x^{\alpha_N} = n^{\alpha_N}\}, \{\vec{x} : x^{\alpha_1} = 1\}, \{\vec{x} : x^{\alpha_1} = 2\}, ...\}$.

Step 1. $\forall k \in \{0, 1, \ldots\}$,

$$P_{k+1}(\vec{x}) = P_k(\vec{x}) \cdot \left(\frac{P(a^{\gamma_k}(\vec{x})|B_k)}{P_k(a^{\gamma_k}(\vec{x})|B_k)} \right)^{1_{B_k}}$$

$$\begin{aligned}
D(P\|P_k) - D(P\|P_{k+1}) &= \int_{\vec{x}} P(\vec{x}) \log \left(\frac{P(\vec{x})}{P_k(\vec{x})} \frac{P_{k+1}(\vec{x})}{P(\vec{x})} \right) \\
&= \int_{B_k} P(\vec{x}) \log \frac{P(a^{\gamma_k}(\vec{x})|B_k)}{P_k(a^{\gamma_k}(\vec{x})|B_k)} \\
&= \int_{a^\gamma} \int_{\{\vec{x} \in B_k :, a^{\gamma_k}(\vec{x}) = a^\gamma\}} P(\vec{x}) \log \frac{P(a^{\gamma_k}(\vec{x}) = a^\gamma | B_k)}{P_k(a^{\gamma_k}(\vec{x}) = a^\gamma | B_k)} \\
&= \int_{a^\gamma} P(B_k, a^{\gamma_k}(\vec{x}) = a^\gamma) \log \frac{P(a^{\gamma_k}(\vec{x}) = a^\gamma | B_k)}{P_k(a^{\gamma_k}(\vec{x}) = a^\gamma | B_k)} \\
&= P(B_k) D(P(a^{\gamma_k}(\vec{x})|B_k) \| P_k(a^{\gamma_k}(\vec{x})|B_k))
\end{aligned}$$

, so $D(P\|P_k)$ is decreasing. Since $D(P\|P_k)$ is positive, $D(P\|P_k)$ has a limit and $D(P\|P_k) - D(P\|P_{k+1})$ tends to zero as k goes to ∞. Similarly,

$$\begin{aligned}
D(P_{k+1}\|P_k) &= \int_{\vec{x}} P_{k+1}(\vec{x}) \log \frac{P_{k+1}(\vec{x})}{P_k(\vec{x})} \\
&= \int_{a^\gamma} \int_{\{\vec{x} \in B_k : a^{\gamma_k}(\vec{x}) = a^\gamma\}} P_k(\vec{x}) \frac{P(a^{\gamma_k}(\vec{x}) = a^\gamma | B_k)}{P_k(a^{\gamma_k}(\vec{x}) = a^\gamma | B_k)} \log \frac{P(a^{\gamma_k}(\vec{x}) = a^\gamma | B_k)}{P_k(a^{\gamma_k}(\vec{x}) = a^\gamma | B_k)} \\
&= \int_{a^\gamma} P_k(B_k, a^{\gamma_k}(\vec{x}) = a^\gamma) \frac{P(a^{\gamma_k}(\vec{x}) = a^\gamma | B_k)}{P_k(a^{\gamma_k}(\vec{x}) = a^\gamma | B_k)} \log \frac{P(a^{\gamma_k}(\vec{x}) = a^\gamma | B_k)}{P_k(a^{\gamma_k}(\vec{x}) = a^\gamma | B_k)} \\
&= P_k(B_k) D(P(a^{\gamma_k}(\vec{x})|B_k) \| P_k(a^{\gamma_k}(\vec{x})|B_k)) \\
&= \frac{P_k(B_k)}{P(B_k)} (D(P\|P_k) - D(P\|P_{k+1})).
\end{aligned}$$

Therefore, $D(P_{k+1}\|P_k)$ tends to zero as k goes to ∞.

On the other hand, $\vec{x}$ has a finite domain. Hence, there exists a subsequence $\{P_{k_l}\}_{l=1}^{\infty}$ and a limit probability distribution P^* s.t. $\forall \vec{x}$,

$$P_{k_l}(\vec{x}) \to P^*(\vec{x}), \quad \text{as } l \to \infty.$$

Now, let $[q]$ be the closest integer smaller than q, and define $R_l = k_l - N\left[\frac{k_l}{N}\right]$. Then, we have $R_l \in \{1, 2, 3, .., N\}$ for all l. Thus, there exists an integer $m \in \{1, 2, 3, ..., N\}$ such that $R_l = m\ i.o$, so we can pick up a further-subsequence $\{k_{l_i}\}$ such that $R_{l_i} = m$ for all i. For simplification, we still call the further-subsequence $\{k_l\}$. Therefore, we now have a subsequence $\{P_{k_l}\}$ converging to P^* and $R_l = m$ for all l. Next, since $D(P_{k+1}\|P_k)$ tends to zero, P_{k_l+j} converges to P^* for all $j = 0, 1, 2, ..., N-1$. Therefore, the sequence

$$P_{k_1}, P_{k_1+1}, ..., P_{k_1+N-1}, P_{k_2}, P_{k_2+1}, ..., P_{k_2+N-1}, ...$$

converges to P^*. Since

$$D(P\|P_k) - D(P\|P_{k+1}) = P(B_k)D(P(a^{\gamma_k}(\vec{x})|B_k)\|P_k(a^{\gamma_k}(\vec{x})|B_k)) \longrightarrow 0,$$

we have

$$D(P(a^{\gamma_{k_l+j}}(\vec{x})|B_{k_l+j})\|P_{k_l+j}(a^{\gamma_{k_l+j}}(\vec{x})|B_{k_l+j})) \longrightarrow 0 \text{ as } l \longrightarrow \infty, \forall j = 0, 1, 2.., N-1.$$

Therefore, $D(P(a^{\gamma_i}(\vec{x})|B_i)\|P^*(a^{\gamma_i}(\vec{x})|B_i)) = 0$ for $i = 1, 2, 3, ..., N$. This implies $P^* \in \mathbb{P}_0$, so we can replace P by P^* and repeat step 1. We can get the same P_k, and $D(P^*\|P_k)$ is a decreasing sequence. Furthermore, $D(P^*\|P_{k_l}) \longrightarrow 0$, so

$$D(P^*\|P_k) \longrightarrow 0.$$

□

In practice, it is painful to construct P by searching the limit above, so we usually use an approximation instead of an exact solution. The license-plate application explored in [41], [42] used a simple approximation. Each pre-perturbation probability, $\tilde{P}^\alpha(a^\alpha|x^\alpha))$ in (5.7), was assumed to be close to, and was therefore replaced by, the corresponding conditional probability under the Markov distribution (5.1), which is denoted by $p_0(a^\alpha|x^\alpha)$. The right panel of Figure 5.4 shows a compositional 4-digit sample generated by Jin and Geman [41], [42] from this non-Markovian model. As we can see, dramatic improvement is achieved as compared to the sampling result from the Markov backbone (in the left panel of Figure 5.4). Although the dynamic programming machinery is no longer available for non-Markovian models, certain coarse-to-fine computational engines are available.

The Markov distribution is easy to work with and estimates or even exact values can be derived for the conditional attribute probabilities. Since the target distributions, $\{P^\alpha_c(a^\alpha|x^\alpha)\}_{\alpha\in\mathcal{B}}$, are fixed (by hand or inference) and the "null" probabilities $\{p_0(a^\alpha|x^\alpha)\}_{\alpha\in\mathcal{B}}$, all derive from the Markov distribution (5.1), there is no dependence on order. These considerations lead to the useful (and order-independent) approximation:

$$P^*(\vec{x}) \propto \frac{\prod_{\beta \in \mathcal{B}} (\epsilon^{\beta}_{x^\beta})}{\prod_{\beta \in B(\vec{x})} (1 - \epsilon^{\beta}_{0})} \prod_{\beta \in \mathcal{B}} \frac{P^{\beta}(a^{\beta}(\vec{x}) | x^{\beta})}{p_0(a^{\beta}(\vec{x}) | x^{\beta})}.$$

A price is paid in that the normalization is no longer exact. On the other hand, the parameters in the Markov "backbone" (5.1) as well as the null probabilities under the Markov distribution can be estimated by more-or-less standard approaches, and the remaining terms, the brick-conditioned attribute probabilities, are in principle available from examples of the objects of interest.

5.5 The distribution on absolute coordinate system and relative coordinate system

We have roughly built the hierarchical model for the prior, by first assuming the Markov backbone model and then by perturbing it to satisfy given conditional constraints. Yet, we have not specified what the exact probability model on each brick is, and what those conditional constraints are. We will focus on how to choose or build those probability distributions.

The probability distribution on a brick is problem dependent. We may have many interpretations on a brick. For the eye example, we may want to consider different types of eyes: male eye, female eye, American eye, Asian eye, etc. And almost for every problem, we need to consider the distribution of the poses. Moreover, when talking about conditional constraints on descendants, many people may ask how to set up an appropriate constraints. The constraints usually come from the relative poses, or so-called relative coordinates. For example, if a right eye is in the middle of an image, then its left eye should be in the right side of the image. Therefore, in this section, we will hone in on the distribution on poses including absolute poses (coordinates) and relative poses (coordinates).

A (absolute) coordinate system in 2-D image analysis is composed of three parts:

the location, orientation and scale. Thus, it is a four dimensional random variable, (x, y, θ, s), where θ is the angle of the orientation, and $s = log(r)$, where r is the scale for the image. For example, we can define an absolute coordinate of an image patch in an image such as Figure 5.7.

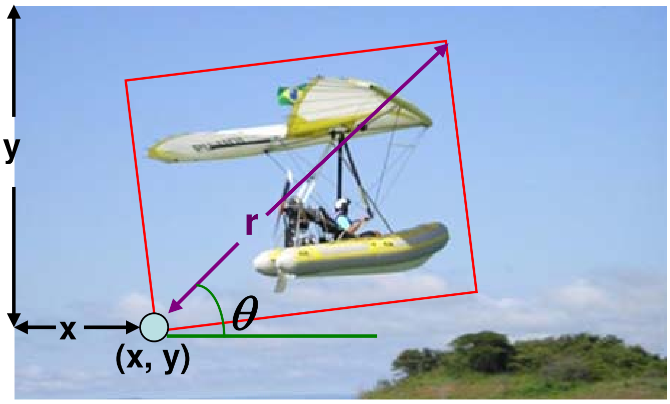


Figure 5.7: Absolute coordinate $(x, y, \theta, s = log(r))$.

The relative coordinate $R(X_2; X_1)$ of $X_2 = (x_2, y_2, \theta_2, s_2)$ relative to $X_1 = (x_1, y_1, \theta_1, s_1)$ is defined as follows:

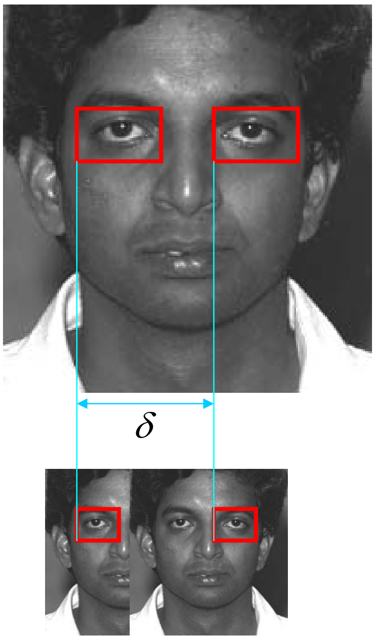
$$R(X_2; X_1) = \begin{pmatrix} u \\ v \\ \varphi \\ t \end{pmatrix} = \begin{pmatrix} e^{-s_1} R_{-\theta_1} \begin{pmatrix} x_2 - x_1 \\ y_2 - y_1 \end{pmatrix} \\ \theta_2 - \theta_1 \\ s_2 - s_1 \end{pmatrix}.$$

The definition can also be extended to n-body version

$$R(X_2, X_3, .., X_n; X_1) = (R(X_2; X_1), R(X_3; X_1), ..., R(X_n; X_1)).$$

Notice that some people may ask why do we not simply define $R(X_2; X_1) = X_2 - X_1$? However, it is not enough to fully specify their relation, since there are some ambiguities. For example, in the Figure 5.8, while the top two image patches can be a pair of eyes, the two bottom images patches can not, but they have the same

$X_2 - X_1$.



$$X_2 - X_1 = \begin{pmatrix} x_2 - x_1 \\ y_2 - y_1 \\ \theta_2 - \theta_1 \\ s_2 - s_1 \end{pmatrix} = \begin{pmatrix} \delta \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

$$\tilde{X}_2 - \tilde{X}_1 = \begin{pmatrix} \tilde{x}_2 - \tilde{x}_1 \\ \tilde{y}_2 - \tilde{y}_1 \\ \tilde{\theta}_2 - \tilde{\theta}_1 \\ \tilde{s}_2 - \tilde{s}_1 \end{pmatrix} = \begin{pmatrix} \delta \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

Figure 5.8: X_1,X_2 are the absolute coordinates of two image patches on the top image. $\tilde{X}_1$, $\tilde{X}_2$ are the absolute coordinates of two images patches on the bottom images.

Before, we go on, let us summarize some properties of our defined relative coordinate.

Properties

1 Identity element: if $X_1 = (0, 0, 0, 0)$, then $R(X_2; X_1) = X_2$.

2 Linear: $\alpha R(X_1; X) + \beta R(X_2, X) = R(\alpha X_1 + \beta X_2; X)$.

3 Transition: $R(R(X_n, ..., X_2; X); R(X_1; X)) = R(X_n, ..., X_2; X_1)$.

Now, let us start with a small or local system like Figure 5.9, which will be easy to extend to more a complicated hierarchical system. We can think of this system like the face example. Assume X to be the absolute coordinate of the face and $X_1, X_2, ..., X_N$ to be the absolute coordinates of its N parts(like mouth, nose, right eye, left eye,...). Then, we factor $P(X, X_1, ..., X_N)$ into $P(X)$ and $P(X_1, ..., X_N|X)$,

$$P(X, X_1, ..., X_N) = P(X)P(X_1, ..., X_N|X).$$

In the first part of this section, we will discuss how to set up a common probability distribution of X, and how to give a universal or common law on scales which we call r-cube law. In the second part of this section, we will set up the conditional probability model $P(X_1, ..., X_N|X)$ (conditional constraints) through relative coordinate distributions, and then we will propose an approximation method to obtain the joint probability of parts $X_{i_1}, X_{i_2}, ..., X_{i_k}$, $P(X_{i_1}, ..., X_{i_k})$, which is computationally feasible.

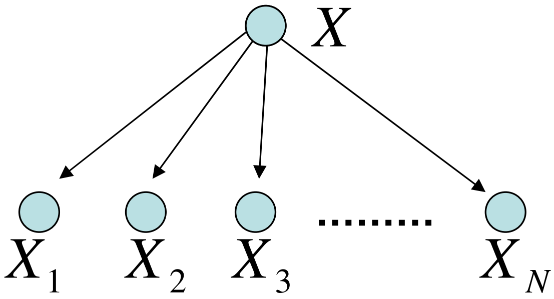


Figure 5.9: Small composition system

5.5.1 r-cube law

In this section, we will discuss how to set up a probability distribution on the poses or the absolute coordinate. Usually, if we do not have a preference regarding locations or orientations, we could assume that they are uniformly distributed. We will maintain this assumption in this section. But what about scale r? Big images could contain more objects, so the distribution on the scale should not be uniform. Many researchers have studied the properties about scale and its models from different points of view. In particular, one of the most striking properties about scale is the scale invariance in natural images. In Ruderman [56], they discussed the scaling of second-order statistics, and of dead-leaves models with disks. They concluded that scaling is related to a power-law size distribution of statistically independent regions. In Alvarez [4], they used the area approach to analyze the scale of the images, and they propose the "area law" for image objects. In Lee [46], they used the probability measure of the poisson process to show, under the scale invariance assumption, "the cubic law of sizes." We will discuss some different aspects about the scale probability distribution in theorem 3.

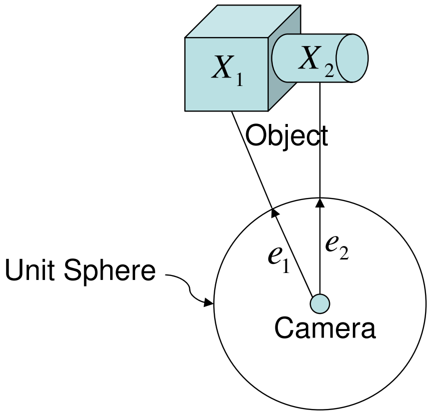


Figure 5.10: the unit vectors e_1 and e_2 represent the locations of X_1 and X_2 in the new coordinate system.

Let us consider a 2-body object (only having two parts). In order to avoid boundary problems, we derive the "r-cube law" using a different coordinate system described as follows: for the two parts X_1 and X_2 of an object, $X_1 = (e_1, \theta_1, s_1)$ and $X_2 = (e_2, \theta_2, s_2)$, where e_1 and e_2 are three-dimensional vectors with $||e_1||_2 = ||e_2||_2 = 1$. We can think of the object as being in three-dimensional space, and the observer, or the camera, is in the origin of the coordinate system(see Figure 5.10). We assume that the object is equally likely to be in every direction(i.e. we assume e_1 and e_2 are uniformly distributed on the surface of the unit sphere with the center

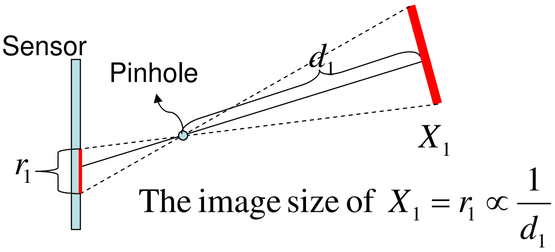


Figure 5.11: the relationship between r_1 and d_1, where r_1 is the image size of X_1 and d_1 is the distance between X_1 and the camera.

$(0, 0, 0))$. Imagining that we take a picture of the object, the size of X_1(or X_2) r_1(or r_2) would be proportional to the distance d_1(or d_2) as showed in Figure 5.11. We also assume that θ_1 and θ_2 are the rotation angles rotating X_1 and X_2 around vectors e_1 and e_2 respectively, and we assume that θ_1 and θ_2 are uniform on $[0, 2\pi]$. Therefore, the only difference between the new coordinate system and the old coordinate system is that we use the unit 3-dimensional vectors e_1 and e_2 for location instead of (x_1, y_1) and (x_2, y_2). The relative coordinate $R(X_2; X_1)$ for the new coordinate system is defined as follows:

$$R(X_2; X_1) = \begin{pmatrix} \vec{w} \\ \varphi \\ t \end{pmatrix} = \begin{pmatrix} e^{-s_1} R_{-\theta_1}(e_2 - e_1) \\ \theta_2 - \theta_1 \\ s_2 - s_1 \end{pmatrix},$$

where $\vec{w}$ is in a 3-dimensional space, and R_{θ_1} is a 3×3 rotation matrix. In the next theorem, we will assume e_1, θ_1 and e_2, θ_2 are uniformly distributed and independent of s_1 and s_2 respectively. In other words, for $i = 1, 2$

$$dP(X_i) = dP(e_i) \times \frac{1}{2\pi} d\theta_i \times dP(r_i)$$

where $dP(e_i)$ is a uniform measure on the unit sphere. If we parameterize e_1 to be $(cos(u_i)sin(v_i), sin(u_i)sin(v_i), cos(v_i))$, then

$$dP(e_i) = \frac{1}{4\pi} sin(v_i) du_i dv_i.$$

In addition, if we let $R_2 = R(X_2; X_1)$, then the Radon-Nikodym derivative $\frac{d(X_1,R_2)}{d(X_1,X_2)} = e^{-2s_1}$, which is the same as the old coordinate system.

Now, we are going to derive the "r-cube law" under this coordinate system. Let us first introduce some notations and give some definitions:

1. **Object:** If we say $P(X_1, X_2)$ defines an object, then $R(X_2; X_1)$ is independent of X_1 as well as independent of X_2.

2. **Orbit:** An "orbit" is the set of $(G(X_1), R(X_2; X_1))$, where $G(X_1)$ is the group of translations, scales and rotations of X_1, and $R(X_2; X_1)$ is fixed.

3. **True model:** Assume X_1 and X_2 have a joint probability distribution $P(X_1, X_2)$, called "true model", and that
$$P_1(X_1) = \int P(X_1, X_2)dX_2,\ P_2(X_2) = \int P(X_1, X_2)dX_1.$$

4. **Un-composed model:** Like the Markov backbone model, X_1 and X_2 are independent given that their parent brick is "on."
$$P^{(2)}(X_1, X_2) = P_1(X_1)P_2(X_2).$$

5. **Compositional model:** From the previous section, we can perturb the above model to create a compositional model:
$$P^c(X_1, X_2) = P^{(2)}(X_1, X_2)\frac{P(R(X_2; X_1))}{P^{(2)}(R(X_2; X_1))} = P^{(2)}(X_1, X_2|R(X_2; X_1))P(R(X_2; X_1))$$
so that it has the right relative coordinate distribution, $P(R(X_2; X_1))$.

Now we will demonstrate the "r-cube" law from several directions in the following theorem:

Theorem 3. **(r-cube Law)** *Assume that e_i and θ_i are uniformly distributed and are independent of s_i under P_i for $i = 1, 2$, and s_i has a density function. Then for the following four statements,*

1. $P^c(X_1, X_2)$ defines an object,

2. *$P_i^{(c)} = P_i$ for $i = 1, 2$,*

3. *$P_i(r) \sim \frac{1}{r^3}$ for $i = 1, 2$,*

4. *$\frac{P(X_1,X_2)}{P^{(2)}(X_1,X_2)}$ is independent of X_i on an orbit $(G(X_i), R(X_j; X_i) = R_j)$ where $(i, j) = (1, 2)$ or $(2, 1)$,*

we have

$$1 \Leftrightarrow 3 \Rightarrow 2.$$

Moreover, if P defines an object, then we have

$$3 \Leftrightarrow 4.$$

Proof. For simplification, let $R_2 \equiv R(X_2; X_1)$ and $R_1 \equiv R(X_1; X_2)$, and the notations $P, P^{(2)}, P^c$ could be thought of as density functions. Since we assume that the location and the rotation are uniformly distributed, the distribution of X_i, $P_i(X_i)$, can be regarded as $c_i P_i(r_i)$, for $i = 1, 2$, where c_i's are constants. Remember that the last component in the coordinate system that we defined is $s = log(r)$. Thus, proving the r-cube law $P(r) \sim \frac{1}{r^3}$ is equivalent to proving $P(s) \sim e^{-2s}$.

Proof of 1⇒2: P^c defines an object $\Rightarrow P^c(X_1, R_2) = P(R_2)P^{(2)}(X_1|R_2) = P(R_2)P_1^c(X_1)$ and $P^c(X_2, R_1) = P(R_1)P^{(2)}(X_2|R_1) = P(R_1)P_2^c(X_2) \Rightarrow P^{(2)}(X_1|R_2) = P_1^c(X_1)$ and $P^{(2)}(X_2|R_1) = P_2^c(X_2) \Rightarrow P_i^{(c)} = P_i$ for $i = 1, 2$.

Proof of 1⇔3: By definition of $P^{(c)}$ and $P^{(2)}$,

$$\begin{aligned} P^c(X_1, R_2) &= P^c(X_1, X_2(X_1, R_2))e^{2s_1} \\ &= P^{(2)}(X_1, X_2(X_1, R_2))\tfrac{P(R_2)}{P^{(2)}(R_2)}e^{2s_1} \\ &= P_1(X_1)P_2(X_2(X_1, R_2))\tfrac{P(R_2)}{P^{(2)}(R_2)}e^{2s_1} \\ &= P_1(X_1)P_2(t_2 + s_1)c_2\tfrac{P(R_2)}{P^{(2)}(R_2)}e^{2s_1}, \end{aligned} \tag{5.10}$$

where t_2 is the last component of R_2. Since $1 \Rightarrow 2$, $1 \Leftrightarrow P^c(X_1, R_2) = P_1(X_1)P(R_2)$ and $P^c(X_2, R_1) = P_2(X_2)P(R_1)$. According to equation 5.10 above, $P^c(X_1, R_2) = P_1(X_1)P(R_2) \Leftrightarrow c_2P_2(t_2 + s_1)\frac{P(R_2)}{P^{(2)}(R_2)}e^{2s_1} = P(R_2)$ does not depend on $X_1 \Leftrightarrow P_2(t_2 + s_1)e^{2s_1}$ is the function of t_2 and $P_2(s) \sim e^{-2s}$. Similarly, $P^c(X_2, R_1) = P_2(X_2)P(R_1) \Leftrightarrow P_2(s) \sim e^{-2s}$.

Proof of 3⇔4: By definition,

$$\begin{aligned} P(X_1, X_2) &= P_1(X_1)P(X_2|X_1) \\ &= P_1(X_1)P(R(X_2; X_1)|X_1)e^{-2s_1} \\ &= P_1(X_1)P(R(X_2; X_1))e^{-2s_1}. \end{aligned}$$

Therefore,

$$\frac{P(X_1,X_2)}{P^{(2)}(X_1,X_2)}=\frac{P(R(X_2;X_1))e^{-2s_1}}{P_2(X_2)}=\frac{P(R(X_2;X_1))e^{-2s_1}}{c_2P_2(s_2)}.$$

If $R(X_2;X_1)$ is fixed on the orbit $(G(X_1),R(X_2;X_1)=R_2)$, then we have

$$\frac{P(X_1,X_2)}{P^{(2)}(X_1,X_2)}=\frac{P(R_2)e^{-2s_1}}{c_2P_2(s_1+t_2)}$$

where t_2 is the last component of R_2. Hence, $\frac{P(X_1,X_2)}{P^{(2)}(X_1,X_2)}$ is independent of X_1 on the orbit $(G(X_1),R(X_2;X_1)=R_2)$ if and only if $P_2(s)\sim e^{-2s}$. Similarly, $\frac{P(X_1,X_2)}{P^{(2)}(X_1,X_2)}$ is independent of X_2 on the orbit $(G(X_2),R(X_2;X_1)=R_1)$ if and only if $P_1(s)\sim e^{-2s}$.

□

Remarks:

1. For simplification, we only considered the two-body object, which has only two parts, but this theorem can be extended to the general compositional object(the object can be represented hierarchically in a very complicated way). In practice, we build a model hierarchically by first defining it as the Markov Backbone model and then by perturbing it to be the context sensitive model. The composed model P^c in the theorem is exactly the small version or the small part of our model. Therefore, "1 $\Leftrightarrow$ 3" gives us a reliable distribution of the scale for our model.

2. From a statistical point of view, the fourth statement in the theorem explores the likelihood ratio of the true model P and the un-composed model $P^{(2)}$. The un-composed model $P^{(2)}$ is obtained by assuming that X_1 and X_2 are independent. Thus, under $P^{(2)}$, it is coincident that X_1 and X_2 slip on the orbit, so we can think of the likelihood as "Composition VS Coincidence." Therefore, the fourth statement states that "Composition VS Coincidence" is independent of the scale. This is exactly what we expected since we do not want the likelihood ratio to be dependent on the scale of the object when we perform the ratio test.

5.5.2 A joint probability distribution of parts

In many applications of detection and recognition, the target may have many parts. We need to consider their relationship between each other, or more technically speaking, their relative coordinate distribution. It is usually not enough to assume that, given the pose of the target, the poses of its parts are conditionally independent. This means that most of time it is not a Markov backbone composition structure,

$$P(X_1, ..., X_N|X) \neq P(X_1|X)P(X_2|X)...P(X_N|X),$$

and thus we need to find the correct conditional probability or conditional constraint, $P(X_1, ..., X_N|X)$ in order to perturb the Markov structure or context-free grammar to creating non-Markov structure(see the Section 5.2). One way to go about this is to consider their relative coordinate distribution $P(R(X_1; X), R(X_2; X), ..., R(X_N; X))$, which is, by definition of "Object", independent of the pose X of the object as in the following assumption.

Assumption :

$$P(R(X_1; X), R(X_2; X), ..., R(X_N; X)|X) = P(R(X_1; X), R(X_2; X), ..., R(X_N; X))$$

Then, we only need to work on the joint distribution of these relative coordinates. It is always natural to assume that they are joint Gaussian distributed. Let Σ and μ be the covariance matrix and the mean of the $4N$-dimensional Gaussian distribution, so that the density function

$$f_N(R_1, R_2, ..., R_N) = \frac{1}{(2\pi)^{2N}|\Sigma|^{\frac{1}{2}}} e^{-\frac{1}{2}((R_1,R_2,...,R_N)-\mu)\Sigma^{-1}((R_1,R_2,...,R_N)-\mu)^T}$$

where

$$R_i = R(X_i; X) = \begin{pmatrix} e^{-s}R_{-\theta}\begin{pmatrix} x_i - x \\ y_i - y \end{pmatrix} \\ \theta_i - \theta \\ s_i - s \end{pmatrix}$$

and

$$X = \begin{pmatrix} x \\ y \\ \theta \\ s \end{pmatrix}, \ X_i = \begin{pmatrix} x_i \\ y_i \\ \theta_i \\ s_i \end{pmatrix}$$

for $i = 1, ..., N$. In addition to the r-cube law we just proposed, we have the joint density function of $X, R(X_1; X), ..., R(X_N; X)$ as follows:

$$\begin{aligned} \tilde{f}(X, R_1, ..., R_N) &= f_X(X) f_N(R_1, ..., R_N) \\ &= \frac{c}{(2\pi)^{2N}|\Sigma|^{\frac{1}{2}}} e^{-\frac{1}{2}((R_1,R_2,...,R_N)-\mu)\Sigma^{-1}((R_1,R_2,...,R_N)-\mu)^T} e^{-2s}. \end{aligned}$$

Next, we change the variable $X, R_1, R_2, ..., R_N$ to $X, X_1, ..., X_N$ and get the following density function of $X, X_1, ..., X_N$ by multiplying the Jacobian determinate e^{-2Ns}:

$$f(X, X_1, ..., X_N) = \frac{c}{(2\pi)^{2N}|\Sigma|^{\frac{1}{2}}} e^{-\frac{1}{2}((R_1(X_1,X),...,R_N(X_N,X))-\mu)\Sigma^{-1}((R_1,R_2,...,R_N)-\mu)^T} e^{-2s} e^{-2Ns}.$$

Now, in order to get the marginal distribution of the $X_1, ..., X_N$ we need to integrate out X as follows

$$\int \frac{c}{(2\pi)^{2N}|\Sigma|^{\frac{1}{2}}} e^{-\frac{1}{2}((R_1(X_1,X),...,R_N(X_N,X))-\mu)\Sigma^{-1}((R_1,R_2,...,R_N)-\mu)^T} e^{-2s} e^{-2Ns} dx dy d\theta ds.$$

Notice that we do not have a closed form of the quaternion integral. This causes a big problem in the detection or recognition process because we need to do numerical integrals many times. In practice, we do not need to do it exactly. We could use an approximation to get a closed form. Here, we are going to propose the following steps to achieve it:

step 1: We integrate out x and y, which we can do exactly since the power of the exponential is quadratic of x and y, given t and s.

step 2: After step 1, we write the integral as the following integral form up to a constant:

$$\int e^{-\frac{1}{2}Q(\theta,s,e^{-s}cos(\theta),e^{-s}sin(\theta))}d\theta ds,$$

where $Q(\theta, s, e^{-s}cos(\theta), e^{-s}sin(\theta))$ is a quadratic function of $\theta, s, e^{-s}cos(\theta), e^{-s}sin(\theta)$. Then, we can obtain

$$Q(\theta, s, e^{-s}cos(\theta), e^{-s}sin(\theta)) = (a(\theta-\alpha)^2 + b(\theta-\alpha)(s-\beta) + (s-\beta)^2) + G(\theta, s)$$

by completing square only on the terms, $\theta^2, \theta s, s^2, \theta, s$.

step 3: Now, we can get an approximated closed form by approximating $g(\theta, s) = e^{-\frac{1}{2}G(\theta,s)}$ by

$$g(\alpha, \beta) + \frac{\partial g}{\partial \theta}(\alpha, \beta)(\theta - \alpha) + \frac{\partial g}{\partial s}(\alpha, \beta)(s - \beta)$$

and then by integrating over θ and s.

Hence, we can get any approximated closed form for any subset of all parts by the same procedure above. This process is very useful when we want or when we allow to look at only few parts, for example in the occlusion problem. In addition, working on sequential tests,(see Chapter 7) we will need this kind of clean closed form for parts and we will use it often. Furthermore, I want to underline some ideal approach that people may think of but it may not work through. Commonly, people may ask why we do not just assume that the relative coordinates are Gaussian distributed directly instead of doing this type of approximated integral. The following items are the problems and the reasons why we do not proceed in this manners.

1. If we assume the relative coordinate $R(X_2; X_1)$ is four dimensional Gaussian distributed, then in general $R(X_1; X_2)$ is not Gaussian anymore. That means a relative coordinate does not have the symmetric property under the Gaussian assumption.

2. We can not simply assume the relative coordinates, $R(X_2; X_1)$, $R(X_3; X_1)$, $R(X_3; X_2)$ of the three parts X_1, X_2, X_3 are joint Gaussian, since $R(X_3; X_2)$ is

a function of $R(X_2; X_1)$ and $R(X_3; X_1)$ by property 3 that does not preserve the Gaussian property. That means if you assume $R(X_2; X_1)$ and $R(X_3; X_1)$ are joint Gaussian, $R(X_3; X_2)$ will not be Gaussian distributed in general. It is true for the special case that assumes that θ and r are constant and fixed, since, in this case, $R(X_3; X_2)$ is a linear function of $R(X_3; X_1)$ and $R(X_2; X_1)$. But in general, we need to, again, deal with the following integral to obtain the density of $R(X_3; X_2)$:

$$\int \frac{1}{(2\pi)^4|\Sigma|^{1/2}} e^{-\frac{1}{2}\left(\left(\begin{array}{c} (A_\eta^{-1}R+\eta) \\ \eta \end{array}\right)-\mu\right)^T \Sigma^{-1}\left(\left(\begin{array}{c} (A_\eta^{-1}R+\eta) \\ \eta \end{array}\right)-\mu\right)} e^{2t} d\eta$$

where μ and Σ^{-1} are the mean and covariance matrix of the 8-dimensional distribution,

$$\eta = \left(\begin{array}{c} u \\ v \\ \varphi \\ t \end{array}\right),$$

and

$$A_\eta = \left(\begin{array}{cc} e^{-t}R_{-\varphi} & 0 \\ 0 & I_{2\times 2} \end{array}\right),$$

and where $R = R(X_3; X_2)$.

3. As opposed to a Gaussian distribution, it seems to be difficult to discover a neat distribution that can preserve its own property under the relative coordinate operations.

Chapter 6

Learning features and pixel-level data models

In this chapter, we will focus on conditional models on the observable images given a particular interpretation under the prior. This brings us to a distribution on the data. In particular, we look at some kinds of features extracted. The way we are approaching this seems to be ambitious because we are trying to model the pixel intensities as supposed to trying to model a distribution of features that are extracted from images. We will point out the difference between modeling the distribution of features and modeling the distribution of the data itself, for example the actual pixel intensities. That forces us to face a certain formulation in order to make it work, which requires knowing a certain conditional probability that we can not possibly know. However, there is a conditional modeling trick that allows us to make real progress, and turns out to be the key to accomplish this task. We will provide some examples of this model of data, and we will take samplings from the distribution. Finally, we will look at some applications to classifications and then compare them to K-means method and Gaussian generative model.

Bayesian generative model starts with the prior $P(\vec{x})$, a probability distribution on the analysis, its possible interpretations, or the parses of the images, which we have described in the previous chapter. In this chapter, we will study another side of building a Bayesian model that is the "data model": a probability distribution $P(Y|\vec{x})$ on images given an interpretation, $\vec{x} \in \mathcal{I}$, which is the part of the model that generates data where Y is the image pixel intensity vector. Then, the Bayesian (generative) framework is completed, and we will have the full Bayesian setup. In practice, we explore it by taking the posterior distribution $P(\vec{x}|Y)$, the distribution on interpretation given an actual image that is proportional to the prior times the data likelihood.

$$P(\vec{x}|Y) \propto P(Y|\vec{x})P(\vec{x})$$

Let us first look at some notations and assumptions in this framework.

Notation

$\mathcal{R}$	index set (pixels) of the "image"
$Y = \{y_j : j \in \mathcal{R}\}$	image (pixel grey levels)
$Y_D = \{y_j : j \in D\}$	image values at locations $j \in D$, for any $D \subseteq \mathcal{R}$
$\mathcal{R}_i^\tau \subseteq \mathcal{R},\ \tau \in \mathcal{T}$	image locations in the support of terminal brick $\tau \in \mathcal{T}$ when $x^\tau = i > 0$
$\bigcup_{i=1}^{n^\tau} \mathcal{R}_i^\tau$	"receptive field" of brick $\tau \in \mathcal{T}$

Given an interpretation $\vec{x} \in \mathcal{I}$, define $D = D(\vec{x}) = \{\tau : x^\tau > 0\}$. The support of an interpretation $\vec{x} \in \mathcal{I}$ is defined as

$$\mathcal{R}_D = \mathcal{R}_D(\vec{x}) = \bigcup_{\substack{\tau \in \mathcal{T} \\ x^\tau > 0}} \mathcal{R}_{x^\tau}^\tau$$

The support is the set of pixels directly addressed by an interpretation.

Independence Assumptions

These are assumptions about the conditional distribution on pixel intensities given an interpretation. They are not unreasonable, as approximations, and they make data modeling much easier. Use $x^{\mathcal{T}}$ to indicate the configuration of the terminal bricks, $\{x^{\tau} : \tau \in \mathcal{T}\}$.

A1. $P(Y|\vec{x}) = P(Y|x^{\mathcal{T}})$ the conditional distribution on image data depends only on the states of the terminal bricks

Let $\vec{x}_0 \in \mathcal{I}$ be the "zero" interpretation: $\vec{x}_0 = \{x_0^{\beta}\}_{\beta \in \mathcal{B}}$ where $x_0^{\beta} = 0 \; \forall \beta \in \mathcal{B}$.

A2. $\frac{P(Y|x^{\mathcal{T}})}{P(Y|x_0^{\mathcal{T}})} = \frac{P(Y_{\mathcal{R}_{\mathcal{D}}}|x^{\mathcal{T}})}{P(Y_{\mathcal{R}_{\mathcal{D}}}|x_0^{\mathcal{T}})}$ the (data) likelihood ratio of interpretation $\vec{x}$ to the "zero" interpretation, $\vec{x}_0$, depends only on the data in the support of $\vec{x}$

Remark: *A2 holds if, for example, the image data that is not supported is i.i.d. from a fixed "null" distribution.*

The support of an interpretation $\vec{x}$ is covered by the supports of the active ("on") terminal bricks. These define connected components of pixels (connected by overlapping supports), and if the independence assumptions are expanded to connected components, then the task of data modeling is the task of data modeling a set of overlapping supports, conditioned on states of the corresponding terminal bricks. These models can be built from individual *templates* – one for each support $\mathcal{R}_i^{\tau}$, $i \in \{1, 2, \ldots, n^{\tau}\}$. The following sections will focus on building a reasonable model for $y_{\mathcal{R}_i^{\tau}}$, given that $x^{\tau} = i$.

6.1 Feature distributions and data distributions

Let us first consider local image patches, for example, a 20 by 20 eye image patch like Figure 6.1. In this case, the vector y of pixel intensities is 400-dimensional. Note the notation "y" is simplified from $y_{\mathcal{R}_i^{\tau}}$ both here and in the rest of this chapter.

Now, perhaps we suspect there is an eye, or the interpretation indicates that it should be an eye. What is the distribution of the patch pixel intensities y? In other words, I want to find the distribution on what we can see in the image given that it

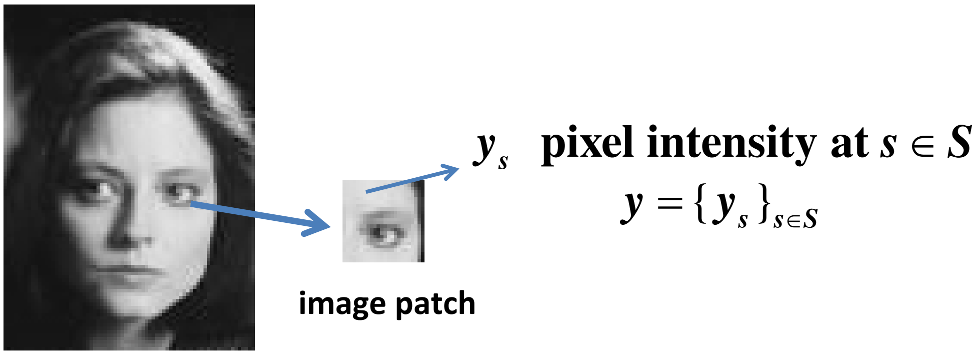


Figure 6.1: local eye image patch

is an eye. Typically we model either the feature vector or we model the distribution on the data through some kind of features. Let

$$\begin{aligned} c(y) &\ : \text{ feature function of } y. \\ P_C(c) &\ : \text{ probability density function of feature.} \end{aligned}$$

In general, this procedure can give us a tool for modeling a general class of image patches.

There are many kinds of features that have been used in image analysis: variance of patch, histogram of gradients, sift features, normalized template correlation, and so on. Here we will use normalized template correlation as the feature. Given a template T(“prototypical eye” for example),

$$c(y) = \text{normalized correlation(T,y)} = \frac{\sum_s (T_s - \bar{T})(y_s - \bar{y})}{\sqrt{\sum_s (T_s - \bar{T})^2}\sqrt{\sum_s (y_s - \bar{y})^2}}$$

and we assume

$$P_C(c) = \alpha_\lambda e^{-\lambda(1-c)}$$

where the λ indicates the reliability of the template. [1]

Note the reason that we use the normalized correlation is because it is a one-

[1] In the real experiments, we may use some kind of bounded density function like Figure 6.2 to make the learning program stable in the learning algorithm.

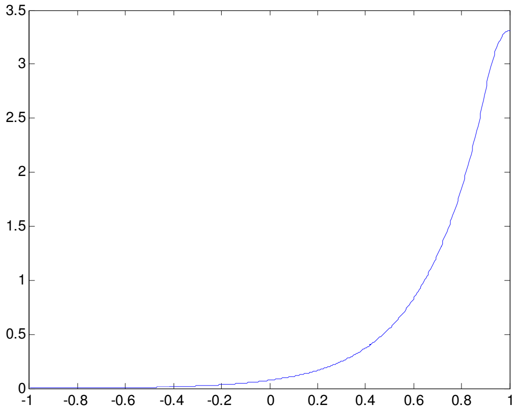


Figure 6.2: feature density function $P_C(c)$

dimensional feature which is easy to model and to learn. Also, it has scale invariance and therefore the light in the environment would have a smaller effect. More specifically, for any linear transformation $ay + b$ where a, b are constants and a is nonzero, $c(ay + b) = c(y)$.

Now, in the above setup, we have two parameters, λ and T. Thus given N image patches, $y_1, y_2, ...y_N$, how do we learn λ and T? Usually, we would use maximum likelihood, but what is the likelihood? Typically, people may be tempted to think that the data is the set of correlations, $c(y_1), ..., c(y_N)$, rather than the set of image patches, so the likelihood (feature likelihood)

$$L(\lambda, T) = \prod_{k=1}^{N} P_C(c(y_k)) = \prod_{k=1}^{N} \alpha_\lambda e^{-\lambda(1-c(y_k))}.$$

However, the data is $y_1, ..., y_N$ and

$$P_Y(y) = P_C(c(y))P_Y(y|C = c(y))$$

so the data likelihood should be as follows:

$$L(\lambda, T) = \prod_{k=1}^{N} \alpha_\lambda e^{-\lambda(1-c(y_k))} P_Y(y_k|C = c(y_k)).$$

Yet, is the term $P_Y(y|C = c(y))$ important? Both methods are consistent for estimating λ but not for estimating T, because the term $P_Y(y|C = c(y))$ depends critically on T. Remember that the correlation is a low dimensional statement about what the patch looks like and there are a lot of image patches consistent with one particular correlation, and thus this term might be important. From a mathematical point of view, assuming that the image patches are living in a standardized space, [2] , given a correlation, y is then in an $n-3$ dimensional subspace, since the correlation is a one-dimensional constraint. Therefore, there would be a lot of action in that term. In particular, if you consider the number of ways that you can obtain this correlation, for some correlations there will be numerous ways to reach them, but for others, there will be only a few. That can make a huge difference in weighting the likelihood and the consequence templates.

Moreover, people might try to avoid modeling the high dimensional distribution $P_Y(y|C = c(y))$ by simply re-normalizing the exponential function as follows:

$$P_Y(y) = \frac{1}{Z_{\lambda,T}} e^{-\lambda(1-c(y))}.$$

It is, of course, a data model now (not a feature model), but the marginal on the correlation will no longer be exponential. You will not get the distribution, $P_C(c) = \alpha_\lambda e^{-\lambda(1-c)}$, for the correlation anymore.

6.2 Conditional modeling

This section will focus on building the conditional probability distribution $P_{Y|\vec{X}}(y|\vec{x})$ of the data given an interpretation $\vec{x}$(we will simply write the distribution as $P_Y(y)$ as in the previous section). From the previous section, we can always write the data distribution in the following form:

$$P_Y(y) = P_C(c(y))P_Y(y|C = c(y)).$$

[2] The vector in the space has a mean of 0 and a variance of 1. Thus, the standardized space is an $n-1$ dimensional unit sphere surface.

To model $P_C(c)$ is not difficult due to the low dimensionality of feature c. For example, exponential density function $P_C(c) = \alpha_\lambda e^{-\lambda(1-c)}$ for the normalized correlation feature. However, it is difficult to model the high-dimensional distribution $P_Y(y|C=c)$. Therefore, we propose the following principle as our conditional modeling ideal to in order to solve this problem.

Principle : Start with a "null" or "background" distribution $P_Y^0(y)$, and choose $P_Y(y)$

1. consistent with $P_C(c)$, and
2. otherwise "as close as possible" to $P_Y^0(y)$.

Remark :

1. In order to understand the ideal, we can think of the background distribution $P_Y^0(y)$ simply as i.i.d. Gaussian or i.i.d. Uniform. However, the background never looks like i.i.d. and there is no shadow or regularity for the i.i.d model. Later, We will discuss this in more depth.
2. The "as close as possible" means that under some distance measure of two distributions, we minimize the distance over all of the probability distributions consistent with $P_C(c)$.

Now, if we choose K-L divergence for measuring the distance between two distributions, we will get the following clean formula by some simple calculation,

$$P_Y(y) = \text{argmin}_{\mathbb{S}} D(P_Y^0||P_Y) = P_C(c(y))P_Y^0(y|C=c(y))$$

where $\mathbb{S} = \{P_Y : C(Y) \text{ has distribution } P_C(c)\}$ and

$$D(P_Y^0||P_Y) = \int P_Y^0(y) log \frac{P_Y^0(y)}{P_Y(y)} dy.$$

Now, that we have completely specified the model, next we need to learn the parameters of the model. For the rest of this section, we will assume that our feature is the normalized correlation, and that the distribution of the feature is $P_C(c) = \alpha_\lambda e^{-\lambda(1-c)}$, but this procedure could be extended to other general features and their distributions. Therefore, the parameters are T and λ, and the data likelihood will be a function of T and λ,

$$L(\lambda, T) = \prod_{k=1}^{N} P_C(c(y_k)) P_Y^0(y_k | C = c(y_k)).$$

Then we want to propose a trick to overcome the dimensionality problem of the second term,$P_Y^0(y_k|C = c(y_k))$. We can divide the likelihood by a constant $\prod_{k=1}^{N} P_Y^0(y_k)$, and each term in the product can be written as the product of the background probability on the feature and the background probability of the data given the feature as follows:

$$\begin{aligned} L(\lambda, T) &\propto \textstyle\prod_{k=1}^{N} \frac{P_C(c(y_k))P_Y^0(y_k|C=c(y_k))}{P_Y^0(y_k)} \\ &= \textstyle\prod_{k=1}^{N} \frac{P_C(c(y_k))P_Y^0(y_k|C=c(y_k))}{P_C^0(c(y_k))P_Y^0(y_k|C=c(y_k))} \\ &= \textstyle\prod_{k=1}^{N} \frac{P_C(c(y_k))}{P_C^0(c(y_k))}. \end{aligned}$$

Then we end up with the product of the likelihood ratios above. Notice that the term $P_Y^0(y)$ in the denominator does not depend on λ and does not depend on T. Instead, it is simply a constant, so maximizing the likelihood is the same as maximizing this product of ratios. Therefore, we only need to model or estimate the background probability of the feature, which is much easier for low-dimension features.

Next, instead of considering one template T, we may want to have more templates $T_1, T_2, ..., T_M$ as our representatives. Thus, we want to extend what we have done above to mixture models.

$$P_Y(y) = \sum_{m=1}^{M} \epsilon_m P_{C_m}(c_m(y)) P_Y^0(y|C_m = c_m(y))$$

where the ϵ_ms are mixing probabilities and $c_m(y)$ is the normalized correlation of y and T_m for $m = 1, .., M$. For the eye example, we want to think of every possible patch that could be an eye from some big mixture of many possible eyes (more open, partially closed, of different ethnicities, and so on). Then, the same trick can be applied to the likelihood of the mixture model.

$$\begin{aligned} L(\epsilon_1, ..., \epsilon_M, \lambda_1, ..\lambda_M, T_1, .., T_M) &= \prod_{k=1}^{N} \sum_{m=1}^{M} \epsilon_m P_{C_m}(c_m(y_k)) P_Y^0(y_k | C_m = c_m(y_k)) \\ &\propto \prod_{k=1}^{N} \sum_{m=1}^{M} \epsilon_m \frac{P_{C_m}(c_m(y_k)) P_Y^0(y_k | C_m = c_m(y_k))}{P_Y^0(y_k)} \\ &= \prod_{k=1}^{N} \sum_{m=1}^{M} \epsilon_m \frac{P_{C_m}(c_m(y_k)) P_Y^0(y_k | C_m = c_m(y_k))}{P_{C_m}^0(c_m(y_k)) P_Y^0(y_k | C_m = c_m(y_k))} \\ &= \prod_{k=1}^{N} \sum_{m=1}^{M} \epsilon_m \frac{P_{C_m}(c_m(y_k))}{P_{C_m}^0(c_m(y_k))} \\ &= \prod_{k=1}^{N} \sum_{m=1}^{M} \epsilon_m \frac{\alpha_{\lambda_m} e^{-\lambda_m(1-c_m(y_k))}}{P_{C_m}^0(c_m(y_k))}. \end{aligned}$$

Remember $c_m(y_k)$ is the normalized correlation between y_k and T_m, and sometimes we denote it as $cor(y_k, T_m)$). Now we need the term $P_{C_m}^0(c_m(y_k))$ to fully specify the likelihood model. Although we do not know what the background model is, because C_m is one dimensional we can learn its distribution from a huge data set of background image patches. Assuming we have the background distribution of the feature, $P_{C_m}^0$ (we will come back to this term soon), we can learn λ_ms and T_ms through EM algorithm.

Expectation Step. $\forall k$,

$$\begin{aligned} \hat{P}_m^{(k)} &= P(Z_k = m | y_k, \vec{\theta}^{(c)}) \\ &= \frac{r_m^{(c)}(y_k) \cdot \epsilon_m^{(c)}}{\sum_{m=1}^{M} r_m^{(c)}(y_k) \cdot \epsilon_m^{(c)}}, \end{aligned}$$

where $Z_k = m$ means that the m^{th} template T_m generated y_k, $\theta = (\lambda_1, .., \lambda_M, T_1, ..., T_M)$; the form $\theta^{(c)}$ stands for the "current" guess of θ and

$$r_m^{(c)}(y_k) = \frac{\frac{1}{1-e^{-2(\lambda_m)^{(c)}}} \, (\lambda_t)^{(c)} \; e^{-(\lambda_m)^{(c)} \; (1-cor(y_k, T_m^{(c)}))}}{P_{C_m}^0(cor(y_k, T_m^{(c)}))}.$$

Maximization Step. Maximize

$$\begin{aligned} B &= \sum_{m=1}^{M}\sum_{k=1}^{N}\hat{P}_m^{(k)} \cdot \log\left[\epsilon_m \cdot \frac{P(cor(y_k,T_m)|Z_k=m,\vec{\theta})}{P^0(cor(y_k,T_m)|Z_k=m,\vec{\theta})}\right] \\ &= \sum_{m=1}^{M}\sum_{k=1}^{N}\hat{P}_m^{(k)} \cdot \log(\epsilon_m) + \sum_{m=1}^{M}\sum_{k=1}^{N}\hat{P}_m^{(k)} \cdot \log\left(\frac{\frac{1}{1-e^{-2\lambda_m}}\ \lambda_m\ e^{-\lambda_m\ (1-cor(y_k,T_m))}}{P^0_{C_m}(cor(y_k,T_m))}\right), \end{aligned}$$

over $\{\epsilon_m\}_m, \{\lambda_m\}_m$, and $\{T_m\}_m$ subject to:

$$\sum_{m=1}^{M}\epsilon_m = 1.$$

It is straightforward to solve for $\{\lambda_t\}$ and $\{\epsilon_t\}$,

$$\begin{aligned} \epsilon_m &= \frac{1}{N}\sum_{k=1}^{N}\hat{P}_m^{(k)} \\ \lambda_m &= \frac{2\lambda_m e^{-2\lambda_m} + e^{-2\lambda_m} - 1}{e^{-2\lambda_m} - 1} \cdot \frac{\sum_{k=1}^{N}\hat{P}_m^{(k)}}{\sum_{k=1}^{N}\hat{P}_m^{(k)} \cdot (1 - cor(y_k,T_m))}, \end{aligned}$$

where λ_m can be identified by a simple numerical method, for example, Newton's method or the binary search. To solve for templates T_ms, we calculate the gradient of B with respect to T_m,

$$\nabla B = \sum_{m=1}^{M}\sum_{k=1}^{N}\hat{P}_m^{(k)}(\lambda_m - \frac{{P^0_{C_m}}'(cor(y_k,T_m))}{P^0_{C_m}(cor(y_k,T_m))})(\frac{y_k-\bar{y}_k}{||y_k-\bar{y}_k||||T_m-\bar{T}_m||} + cor(y_k,T_m)\frac{T_m-\bar{T}_m}{||T_m-\bar{T}_m||^2})$$

where $\bar{y}_k$ and $\bar{T}_m$ are the means of y_k and T_m and $||\cdot||$ is the L_2 norm. Then we can use the gradient ascent method to get T_ms.

But what is the background model $P^0_{C_m}$ of the feature? Let us first look at the artificial background model, the i.i.d background model (i.i.d Uniform, i.i.d Gaussian, etc). Then without estimating or learning $P^0_{C_m}$, we can use a generalized Central Limit Theorem to approximate it(see Appendix).

$$P^0_C(y) \approx \frac{\sqrt{n}}{\sqrt{2\pi}}e^{-\frac{nc(y)^2}{2}}$$

where $n = |S|$ is the number of pixels. Therefore the likelihood

$$L(\epsilon_1, ..., \epsilon_M, \lambda_1, ..\lambda_M, T_1, .., T_M) \propto \prod_{k=1}^{N} \sum_{m=1}^{M} \epsilon_m \frac{\alpha_{\lambda_m} e^{-\lambda_m (1 - c_m(y_k))}}{\frac{\sqrt{n}}{\sqrt{2\pi}} e^{-\frac{nc(y)^2}{2}}}.$$

Then the model is fully specified under the background model i.i.d. [3]

Now, we can learn the templates and λs assuming the i.i.d. background. However, the background images of the real world are full of structures and thus are never like i.i.d. Depending upon the problems that we are working on, background probabilities could be very different from i.i.d. For example, outdoor image patches (Figure 6.3) are quite smooth. Indoor image patches (Figure 6.4) are full of edges and structures. As another example, X-ray image patches (Figure 6.5) seem to be very different from both indoor and outdoor images.

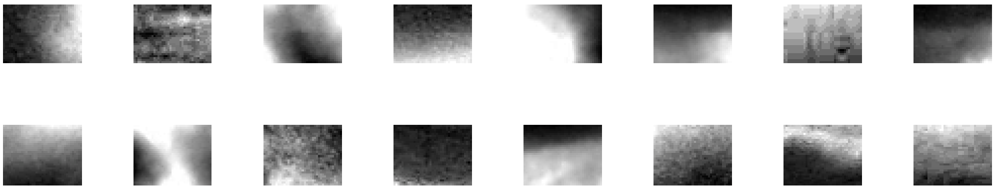


Figure 6.3: Natural outdoor image patches

Therefore, we need to learn the feature distribution $P^0_{C_m}$ in order to get a better model. However, the feature distribution depends on templates, so we will first learn the templates and λs using the i.i.d background model as we have described. Then we can use the learned templates to learn the feature distribution $P^0_{C_m}$. Once we get $P^0_{C_m}$, we can update the $P^0_{C_m}$ in the algorithm and then repeat this process over

[3] Zhang [67] has studied in the particular case of the i.i.d Uniform, and she derived the likelihood

$$L(\epsilon_1, ..., \epsilon_M, \lambda_1, ..\lambda_M, T_1, .., T_M) = \prod_{k=1}^{N} \sum_{m=1}^{M} 256^n \epsilon_m \frac{\alpha_{\lambda_m} e^{-\lambda_m (1 - c_m(y_k))}}{\frac{\sqrt{n}}{\sqrt{2\pi}} e^{-\frac{nc(y)^2}{2}}},$$

which is consistent with the equation obtained from our trick.

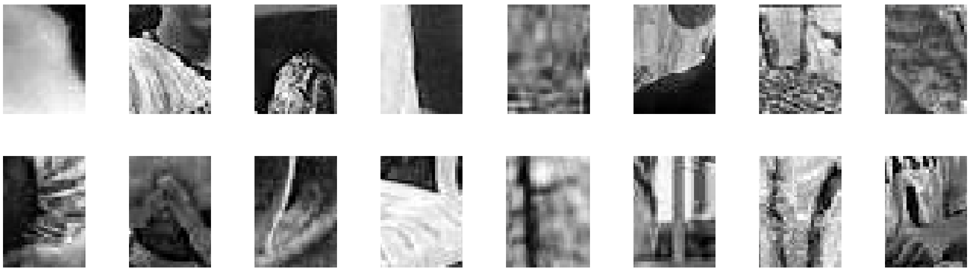


Figure 6.4: Indoor image patches

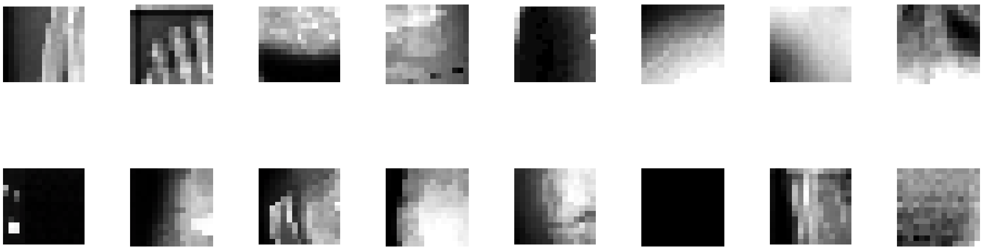


Figure 6.5: Medical image patches(X-ray image patches)

again both forwards and backwards until it converges.

In the general setup with *any* feature, we have the decomposition that comes from minimizing the K-L divergence in a mixture model:

$$P_Y(y) = \sum_{m=1}^{M} \epsilon_m P_{C_m}(c_m(y)) P_Y^0(y|C_m = c_m(y)).$$

Notice the m above can index a lot of things. For example, m could index alternative models, such as classes of eyes. In each class of eyes, it could index transformations of scale, rotation, and so on. Thus, not only do we have the mixture over different templates, but we can also have many other extensions of mixtures: mixtures over different features, mixtures over different resolutions, and most importantly, mixtures over poses which allow us to learn templates even in a poorly-registered training image set(see the next section). More specifically, suppose each template T_m is associated with N_s scales and N_r rotations. Let $Q_{s,r}$ be the set of possible discrete spatial shifts of a template under scale s and rotation r within the image patch Y. And let

$$g_{s,r,l}(T_m) = M_{s,r,l} \cdot T_m$$

be the projection of T_m on to the pixel coordinate of Y, under scale s, rotation r, and spatial shift l, where $M_{s,r,l}$ stands for the projection matrix. Let ϵ_m be the mixing probability associated with each template T_m, δ_s^m be the chance that scale s is selected for T_m, η_r^m be the chance that rotation r is selected for T_m, and $\xi_l^{m,s,r}$ be the chance that spatial shift l is selected for T_m, scale s and rotation r. Hence, $\sum_{m=1}^{M} \epsilon_m = 1$, and $\forall m, \sum_{s=1}^{N_s} \delta_s^m = 1$, $\sum_{r=1}^{N_r} \eta_r^m = 1$, and $\sum_{l \in Q_{s,r}} \xi_l^{m,s,r} = 1$. Let $Y^{s,r,l}$ be the sub-region of Y covered by $g_{s,r,l}(T_m)$. Let n be the total number of pixels in Y, while $n_{s,r,l}$ is the number of pixels in $Y^{s,r,l}$. Then,

$$\frac{P_Y(y)}{P_Y^0(y)} = \sum_{m=1}^{M} \sum_{s=1}^{N_s} \sum_{r=1}^{N_r} \sum_{l \in Q_{s,r}} \epsilon_m \ \delta_s^m \ \eta_r^m \ \xi_l^{m,s,r} \frac{P(S_{m,s,r,l}(y^{s,r,l}))}{P^0(S_{m,s,r,l}(y^{s,r,l}))} \tag{6.1}$$

where $S_{m,s,r,l}(y^{s,r,l}) = cor(y^{s,r,l}, M_{s,r,l}T_m)$. Without changing $S_{m,s,r,l}(y^{s,r,l})$, $M_{s,r,l}$ can be adjusted such that the mean of $M_{s,r,l} \cdot T_t$ is equal to zero for simplification. For example, $M_{s,r,l}(i,j)$ can be replaced with ($M_{s,r,l}(i,j)-$ the mean of the j^{th}column of $M_{s,r,l}$). Therefore, we can again use the same trick to get the easy version of likelihood.

$$\begin{aligned} & L(\{\epsilon_m, \lambda_m, T_m, \delta_s^m, \eta_r^m, \xi_l^{m,s,r}\}) \\ \propto & \prod_{k=1}^{N} \frac{P_Y(y_k)}{P_Y^0(y_k)} = \prod_{k=1}^{N} \sum_{m,s,r,l} \epsilon_m \delta_s^m \ \eta_r^m \ \xi_l^{m,s,r} \ \frac{\frac{1}{1-e^{-2\lambda_m}} \ \lambda_m \ e^{-\lambda_m \ (1-S_{m,s,r,l}(y_k^{s,r,l}))}}{P^0(S_{m,s,r,l}(y^{s,r,l}))}. \end{aligned}$$

Similarly we can learn the templates, the mixing probabilities and λs by the iterative EM algorithm.

Expectation Step. $\forall k$,

$$\begin{aligned}\hat{P}^{(k)}_{(m,s,r,l)} &= P(X_k = (m,s,r,l)|y_k, \vec{\theta}^{(c)}) \\ &= \frac{\epsilon_m^{(c)}(\delta_s^m)^{(c)}(\eta_r^m)^{(c)}(\xi_l^{m,s,r})^{(c)} \cdot \gamma^{(c)}_{(m,s,r,l)}(y_k)}{\sum_{m,s,r,l} \epsilon_m^{(c)}(\delta_s^m)^{(c)}(\eta_r^m)^{(c)}(\xi_l^{m,s,r})^{(c)} \cdot \gamma^{(c)}_{(m,s,r,l)}(y_k)},\end{aligned} \tag{6.2}$$

where $\vec{\theta}$ stands for all the unknown parameters, and

$$\gamma^{(c)}_{(m,s,r,l)}(y_k) = P(y_k|X_k = (m,s,r,l), \vec{\theta}^{(c)}\ ) = \frac{\frac{1}{1-e^{-2\lambda_m^{(c)}}}\ \lambda_m^{(c)}\ e^{-\lambda_m^{(c)}\ (1-S_{m,s,r,l}(y_k^{s,r,l}))}}{P^0(S_{m,s,r,l}(y^{s,r,l}))}.$$

Maximization Step. Maximize

$$\begin{aligned} B &= \sum_{m,s,r,l}\sum_k \hat{P}^{(k)}_{(m,s,r,l)} \cdot \log\left[\epsilon_m\ \delta_s^m\ \eta_r^m \xi_l^{m,s,r} \cdot P(y_k|X_k = (m,s,r,l), \vec{\theta}\ )\right] \\ &= \sum_{m,s,r,l}\sum_k \hat{P}^{(k)}_{(m,s,r,l)} \cdot \log(\epsilon_m\ \delta_s^m\ \eta_r^m\ \xi_l^{m,s,r}) \\ &\quad + \sum_{m,s,r,l}\sum_k \hat{P}^{(k)}_{(m,s,r,l)} \cdot \log\left(\frac{\frac{1}{1-e^{-2\lambda_m}}\ \lambda_m\ e^{-\lambda_m\ (1-S_{m,s,r,l}(y_k^{s,r,l}))}}{P^0(S_{m,s,r,l}(y^{s,r,l}))}\right),\end{aligned}$$

over $\{\{T_m\}_m, \{\epsilon_m\}_m, \{\lambda_m\}_m, \{\delta_s^m\}_{s,m}, \{\eta_r^m\}_{r,m}\}$ subject to:

$$\sum_{m=1}^{M} \epsilon_m = 1;\ \sum_{s=1}^{N_s} \delta_s^m = 1,\ \sum_{r=1}^{N_r} \eta_r^m = 1,\ \forall m;\ \sum_{l\in Q_{s,r}} \xi_l^{m,s,r} = 1\ \forall m,s,r.$$

It is straightforward to solve out $\{\lambda_m\}$, $\{\epsilon_m\}$, $\{\delta_s^m\}$, $\{\eta_r^m\}$and $\{\xi_l^{m,s,r}\}$ [4] ,

$$\begin{aligned}
\epsilon_m &= \frac{\sum_{k,s,r,l} \hat{P}^{(k)}_{(m,s,r,l)}}{\sum_{k,m,s,r,l} \hat{P}^{(k)}_{(m,s,r,l)}} = \frac{1}{N} \sum_{k,s,r,l} \hat{P}^{(k)}_{(m,s,r,l)}, \\
\delta_s^m &= \frac{\sum_{k,r,l} \hat{P}^{(k)}_{(m,s,r,l)}}{\sum_{k,s,r,l} \hat{P}^{(k)}_{(m,s,r,l)}}, \\
\eta_r^m &= \frac{\sum_{k,s,l} \hat{P}^{(k)}_{(m,s,r,l)}}{\sum_{k,s,r,l} \hat{P}^{(k)}_{(m,s,r,l)}}, \\
\xi_l^{m,s,r} &= \frac{\sum_{k} \hat{P}^{(k)}_{(m,s,r,l)}}{\sum_{k,l} \hat{P}^{(k)}_{(m,s,r,l)}}, \\
\lambda_m &= \frac{2\lambda_m e^{-2\lambda_m} + e^{-2\lambda_m} - 1}{e^{-2\lambda_m} - 1} \cdot \frac{\sum_{k,s,r,l} \hat{P}^{(k)}_{(m,s,r,l)}}{\sum_{k,s,r,l} \hat{P}^{(k)}_{(m,s,r,l)} \cdot (1 - S_{m,s,r,l}(y_k^{s,r,l}))},
\end{aligned}$$

where λ_m can be identified by a numerical searching method, e.g. Newton's method or binary search. For solving templates T_m, we can calculate the gradient of B with respect to T_m,

$$\nabla B = \sum_{k,s,r,l} \hat{P}^{(k)}_{(m,s,r,l)} (\lambda_m - \frac{P^{0\prime}(S_{m,s,r,l}(y^{s,r,l}))}{P^0(S_{m,s,r,l}(y^{s,r,l}))}) \nabla S_{m,s,r,l}(y_k^{s,r,l}),$$

where

$$\nabla S_{m,s,r,l}(y_k^{s,r,l}) = \frac{M_{s,r,l}^\top}{\sqrt{T_m^\top M_{s,r,l}^\top M_{s,r,l} T_m}} \cdot \frac{y_k^{s,r,l} - \bar{y}_k^{s,r,l}}{||y_k^{s,r,l} - \bar{y}_k^{s,r,l}||} - \frac{M_{s,r,l}^\top M_{s,r,l} T_m}{T_m^\top M_{s,r,l}^\top M_{s,r,l} T_m} \cdot S_{m,s,r,l}(y_k i^{s,r,l}).$$

where $\bar{y}_k^{s,r,l}$ is the mean of $y_k^{s,r,l}$, and $||\cdot||$ is L_2 norm. Then, we can use the gradient ascent method to get T_ms.

[4] Usually, considering an object is almost equally likely to appear anywhere in an image patch, we assume $\xi_l^{m,s,r} = \frac{1}{||Q_{s,r}||}$, where $||Q_{s,r}||$ is the counting measure of set $Q_{s,r}$. For the experiments in this thesis, we assumed that it is uniform.

6.3 Experiments on Learning Templates and Applications on Ethnicity Classification

In this section, we will implement different versions of the maximum-likelihood template model on the Feret Face database. This database is composed of 499 gray-scale face images, each 215×214 pixels. Each face image has fifteen facial landmarks manually labeled in advance. Figure 6.6 shows twelve face images from this dataset and the corresponding landmarks. With the help of the landmarks, we cropped out different groups of facial parts (left eyes, right eyes, noses and mouths) and scaled them down for training.

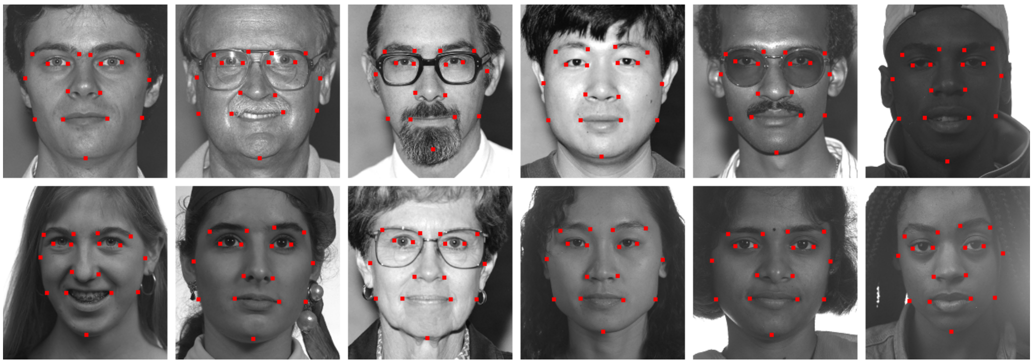


Figure 6.6: 12 face images from Feret Face database, each with 17 landmarks labeled manually.

6.3.1 Right Eye Template Learning

Let us first see the example of learning eye templates as in Figure 6.7. The first part is 24 eye image patches with size 30×40 from the training sets of eyes. The second part is the EM learning for 8 templates with size 30×40. We started from i.i.d white noise in the first row of the second part, using the i.i.d background model. After six iterations, we updated the background distribution $P^0_{C_m}$ by fitting $cor(T_m, y^B_j)$ $j = 1, 2, ...L$ with a Gaussian distribution, where T_ms are learned templates from the six iterations, and y_j^Bs are random background patches.[5] Then, we do the six iterations again using the new background distribution, and then do it forward and backward three times. The final row in the figure is the final iteration with the final $P^0_{C_m}$. As we can see, they do not belong to anybody, but somehow in the likelihood space, they are representatives.

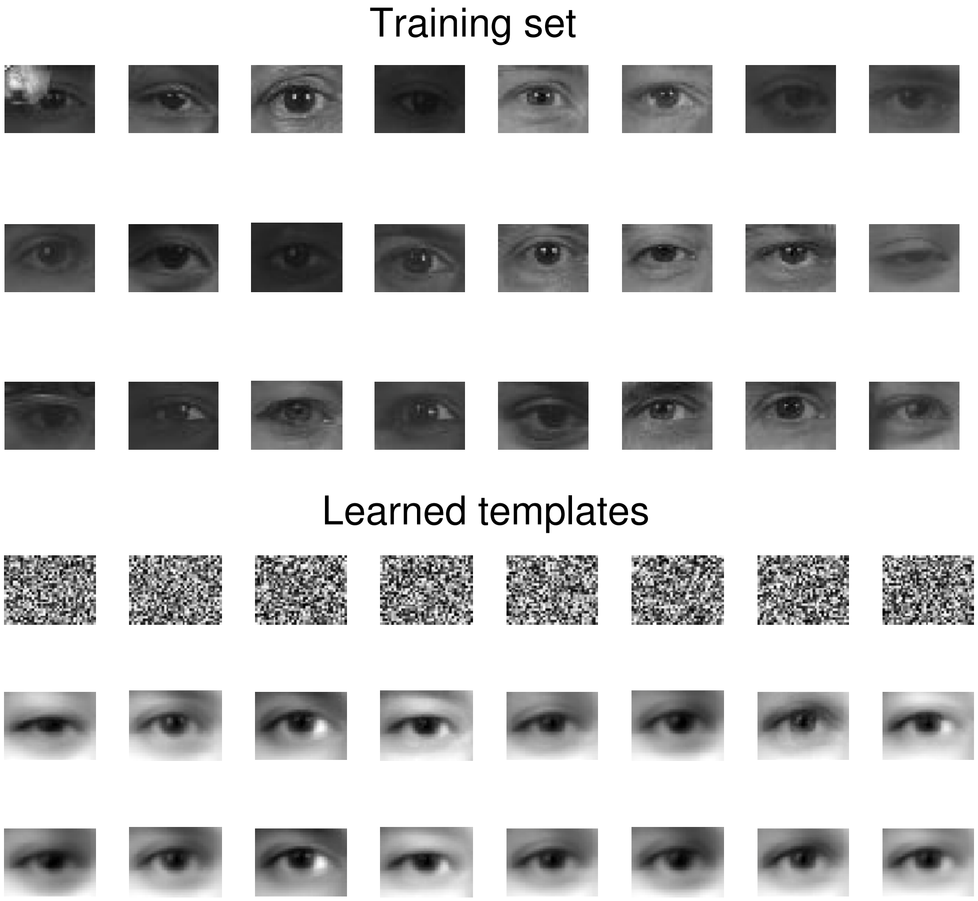


Figure 6.7: Right eye template learning: the top panel shows 24 training image patches. The bottom panel shows the learning of 8 templates with size 30×40

In the next experiment, we want to show what would happen if we used the

[5] The choice of background images depends upon the application. In this chapter, we chose them from natural outdoor images.

feature likelihood to learn templates instead of using the data likelihood.(i.e. learn templates based on

$$\prod_{k=1}^{N}\sum_{m=1}^{M}\epsilon_m\alpha_{\lambda_m}e^{-\lambda_m(1-c_m(y_k))}$$

instead of

$$\prod_{k=1}^{N}\sum_{m=1}^{M}\epsilon_m\frac{\alpha_{\lambda_m}e^{-\lambda_m(1-c_m(y_k))}}{P^0_{C_m}(c_m(y_k))}\ ).$$

We can likewise run the EM algorithm and we will get identical templates as in Figure 6.8.

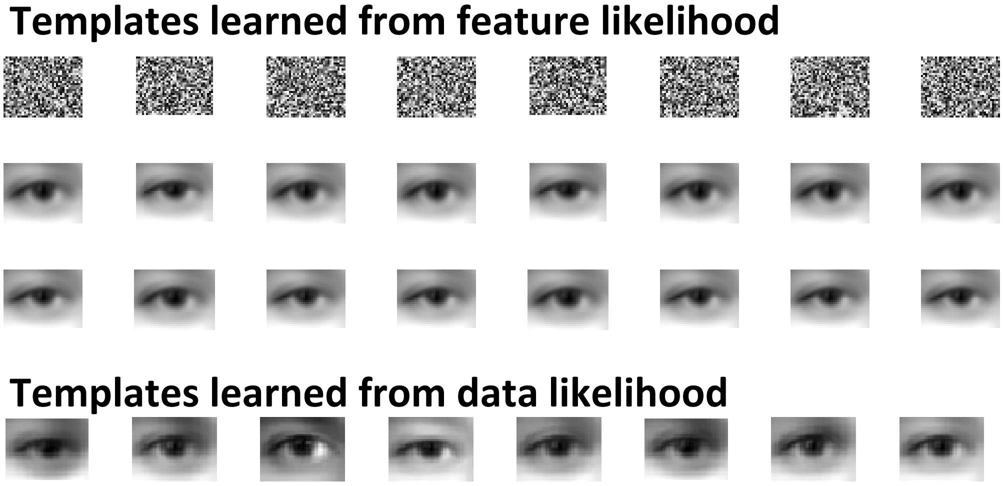

Templates learned from feature likelihood
Templates learned from data likelihood

Figure 6.8: Feature likelihood training

As you can see, they all come together into one eye, which is because we are not maximizing the likelihood of the data and this causes them to lose their separating properties.

6.3.2 Mixture over Poses and Comparison with K-means Cluster

It is always important to have a good training set for learning templates. Most of time we need to do supervised learning. That means we need to landmark where the object is, and, ideally, we want to create a training set with the same scale, the same orientation or the same pose for all training images. However, the registrations of images are never absolutely precise because those landmarks are always human-made. Therefore, our model should have some mixtures over transformations and should allow the templates to move around a little bit and to have different scales and orientations. Let us first look at an experiment in [67].

The dataset was composed of 499 nose images cropped from 499 face images that had been randomly rotated (the rotation angel $\in [-10^o,\ 10^o]$) and scaled (the scaling factor $\in [0.3, 0.5]$). Hence the nose in each image patch was tilted and was not always in the center; the size of each image patch ranged from 16×18 to 30×33. The model parameters were set as follows: 16 templates, each with size 15×18, three discrete scales $\{1.17,\ 1,\ 0.83\}$, and three discrete spatial shifts $\{-6.7^o,\ 0^o,\ 6.7^o\}$. Figure 6.9 shows 120 training images and the 16 learned templates obtained by the EM algorithm.

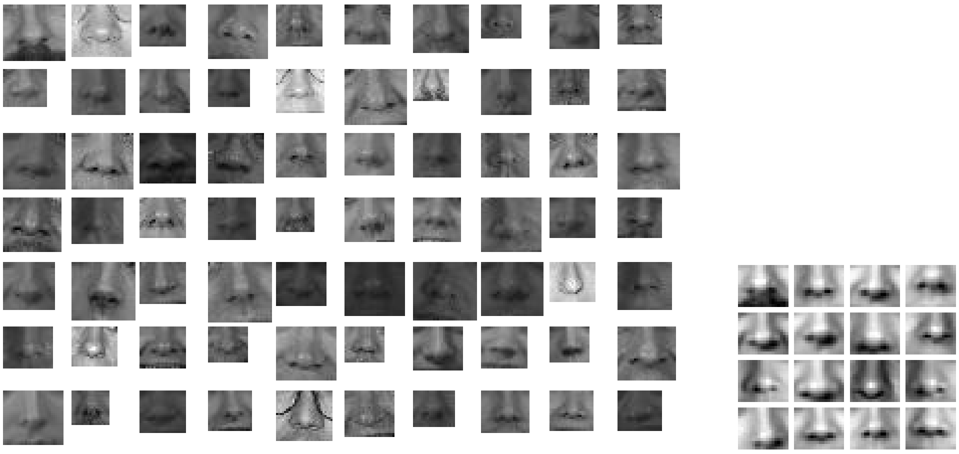


Figure 6.9: The left panel shows 120 training image patches. The right panel shows the 16 learned templates, each with size 15 × 18, from the fully generalized model, with mixtures over multiple templates, spatial shifts, scales, and rotations.

Another way to think of template learning is to regard it as a clustering method. We can think of every template as a cluster, and think of it as a representative of a subgroup. When talking about clustering, people may first think of the most common clustering method, K-means clustering. However, the performance of K-means clustering depends on the initial condition, and the outlier will certainly influence the result as showed in the Figure 6.10. The first run of K-means was affected by an outlier that you can see at the top-left corner of the training set in Figure 6.7. The second run was better, but both of them are not as smooth as the templates learned by the Maximum likelihood estimator that we proposed.

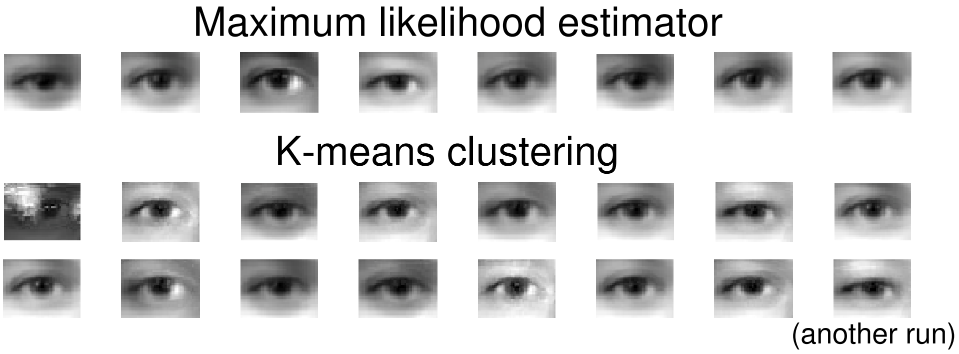


Figure 6.10: The two runs of K-means Clustering show both in-stability and the initial dependence of K-means clustering as well as the outlier affect.

Furthermore, if our training images are poorly registered as in Figure 6.11, K-means clustering get lousy results. Of course, we could create another version of K-means clustering that allows movements or other transformations, and we may get better result. However, it is still only a clustering method, not a probability model. It is limited for many applications: detection, classification, recognition, etc. Instead, our maximum likelihood method is based on a concrete probability model, which can easily tell us how likely the image patch is the target object in those real applications.

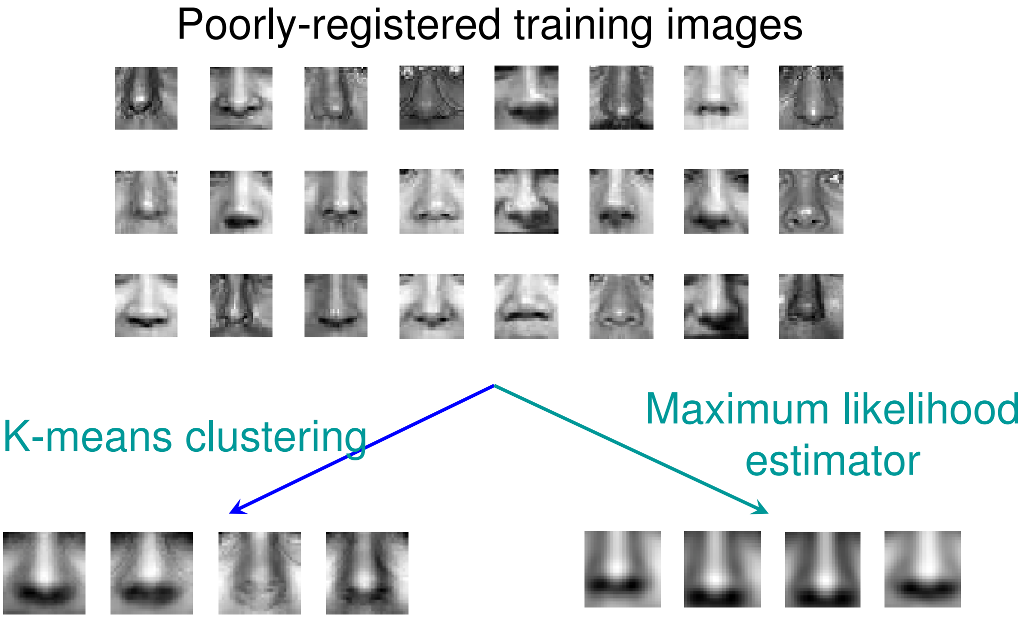

Poorly-registered training images
K-means clustering
Maximum likelihood estimator

Figure 6.11: The comparison of maximum likelihood estimator and K-means clustering for poorly registered training images.

6.3.3 Classification and Comparison with Gaussian Model and K-means Clustering

We downloaded 152 images of South Asian people and 218 images of East Asian people from the internet. The task was to distinguish East Asian faces from South Asian faces. Certainly a complete face model could be built for each ethnic group and then used for classification. But, we fulfilled this face classification task exclusively by examining the region around the eyes – i.e. East Asian eyes indicated an East Asian face while South Asian eyes indicated a South Asian face. In other words, we classified the eye image patches cropped from the face images first, and then used the eye classification result to make a decision about the original face images. The classification of eye image patches was based on our eye model involving templates.

We designed the experiment as follows. First the region of the pair of eyes was cropped out of each face and scaled to have the same size 22 × 43 images. Now we had two sets of image patches, 169 East Asian eyes (call it set A_e) and 124 South Asian eyes (call it set A_i). We used half (selected randomly) of the image patches from A_e as training data for the East Asian group, and we repeated this process for the South Asian group. The other half from A_e and the other half from A_i were merged together, and they played the role of testing data. We implemented the model with mixtures over 8 templates (each with size 21×41), and over 6 spatial shifts.

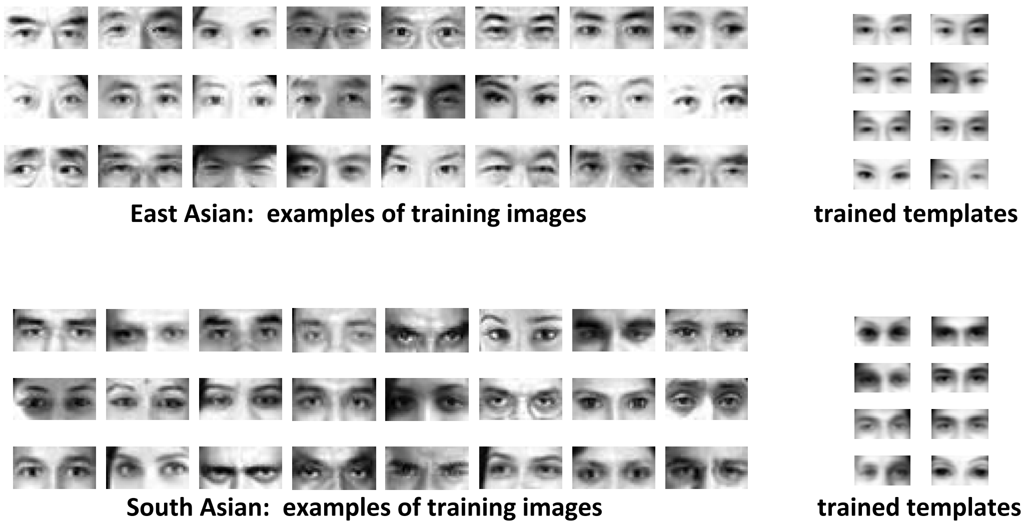

East Asian: examples of training images
trained templates
South Asian: examples of training images
trained templates

Figure 6.12: The East Asian and South Asian eye image sets and templates learned by maximum likelihood estimator.

Figure 6.12 shows the training images of East Asian pairs of eyes and South Asian pairs of eyes and 41 by 21 templates learned by our mixture model mixing over 6 shifts and 8 templates. Each template has a mixing probability and λ specified in the definition of the feature distribution. In other words , two models, $P(y|\text{East Asian eyes})$ and $P(y|\text{South Asian eyes})$, were learned where y represented an image patch.

As for testing the performance of image classification, for y from the testing data, we classified y and the associated face image as East Asian if

$$P(y|\text{East Asian eyes}) \geq P(y|\text{South Asian eyes})$$

, and as South Asian if

$$P(y|\text{East Asian eyes}) < P(y|\text{South Asian eyes}).$$

As shown in the figure, East Asian eye templates and South Asian eye templates looked very different from each other, and they accurately captured the facial features associated with the corresponding ethnic group.

Since we only had a few face images, in order to achieve a less biased result, we performed 20 cross-validations. Within each cross validation, we repeated the training and testing procedure described above. Each cross validation gave a correct classification rate for the East Asian group, and a correct classification rate for the South Asian group. These two rates were averaged and recorded as R_i, $i \in \{1, \ldots, 20\}$. Finally, after 20 cross-validations were finished, we averaged all the Rs (R_1 through R_{20}), giving us a classification rate of 84 percent.

We want to mention that the feature that we are using is correlation. Since correlation is invariant to the scale, we are not looking at skin tones. Of course, it may be helpful to work on some features involving skin tones, but we are not considering it presently in this model. There is a popular generative model, the Gaussian mixture model defined below:

$$P_Y(y) = \sum_{m=1}^{M} \frac{1}{\sqrt{2\pi|\Sigma_m^{-1}|}} e^{-\frac{1}{2}(y-\mu_m)\Sigma_m^{-1}(y-\mu_m)^T}$$

which is skin tone dependent. And the K-means clustering method is also skin tone dependent. Figure 6.13 shows the templates learned by the three models: our model, the Gaussian Model, and the K-means clustering method. The m in the Gaussian mixture model indexes the same mixture components, 6 shifts and 8 templates, and the covariance matrix Σ is assumed to be diagonal for our experiment.[6] Then we can

[6] There are many variations about the covariance matrix, but this is the most common and the

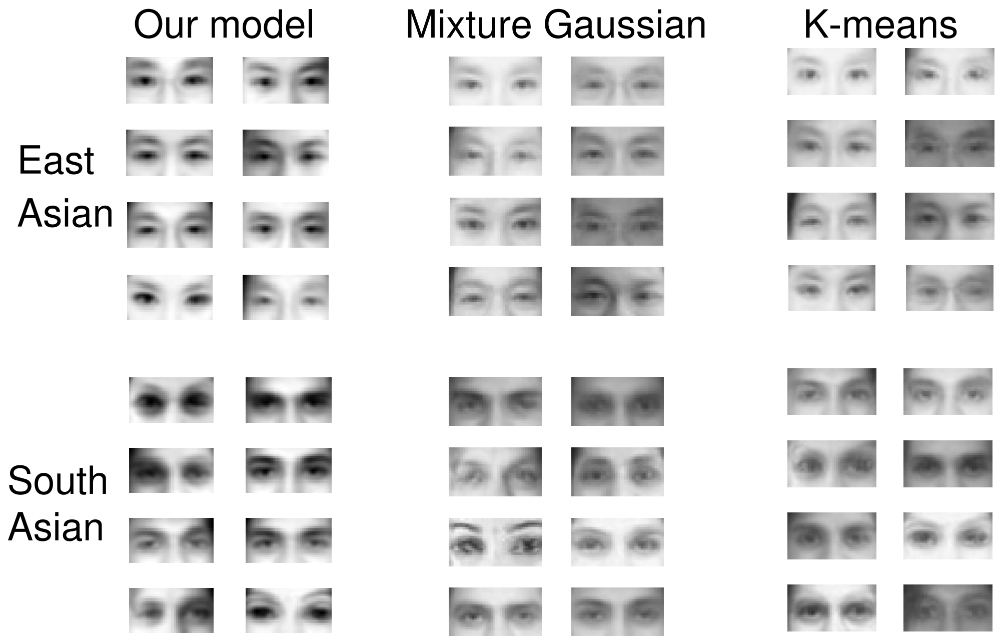


Figure 6.13: Templates learned by our model, Gaussian model and K-means Clustering.

do classification experiments for the Gaussian generative model in the same way that we did before, looking at the posterior probabilities. We get an 80% classification rate. However, for the K-means clustering method, we do not have a posterior probability distribution to do classification, so we need to consider other classification methods, like nearest neighbor, Support Vector Machine, etc. However, choosing the classification methods usually depends on the applications, and therefore we do not support the K-means clustering method. By implementing nearest neighbor method, we get a 79% classification rate in the classification experiment. As you can see our generative model has better performance, and it seems that the Generative Models are more reliable.

easiest assumption.

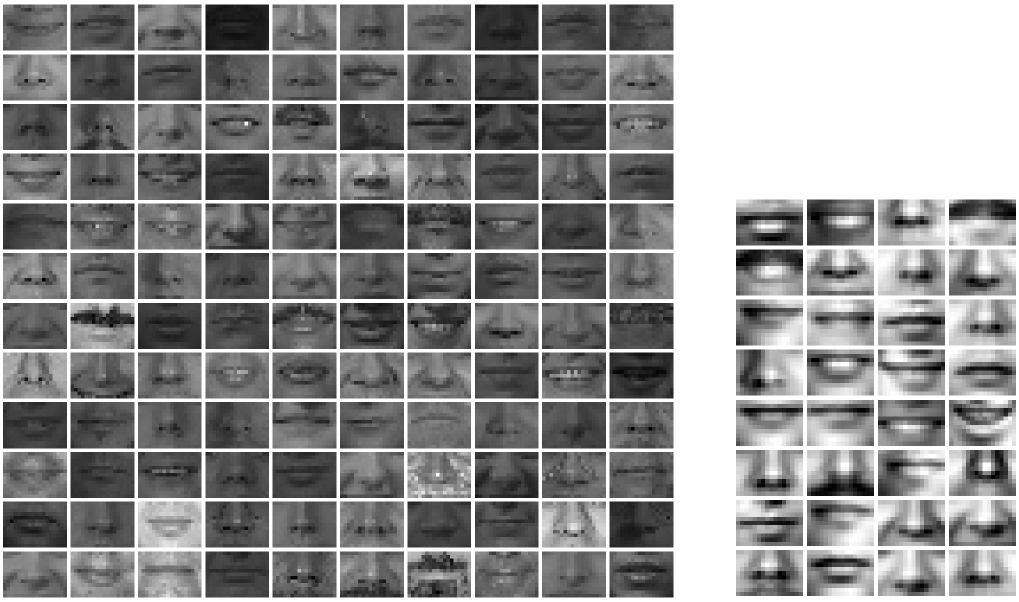


Figure 6.14: The left panel shows 70 training 13×18 nose or mouth image patches. The right panel shows the 16 learned templates each with size 11×16.

6.3.4 Other experiments

Another way to verify the effectiveness of the model is to do experiments on a training data set composed of two different types of images, and then to see whether it can get the two different types of templates associated with the two types of training images, and then to see whether it can get a similar proportion of the weights on the two populations. Let us first see an experiment in [67]. The training data set was composed of 499 nose images and 499 mouth images, each with size 13×18. The EM algorithm learned 32 templates after 15 runs, each with size 11×16. Figure 6.14 shows 120 training image patches and the 32 learned templates. Besides the apparent difference between a nose and a mouth, there was a big variation of facial features and expressions among the training image patches – for example, with or without a moustache, and smiling or not smiling. Hence, as expected, the learned templates from the EM algorithm revealed this variation of facial features and expressions, aside from distinguishing noses from mouths. In addition, both the summation of the mixing probabilities associated with nose templates and the summation of the mixing probabilities associated with mouth templates were very close to 0.5, which indicated that our model was properly weighted.

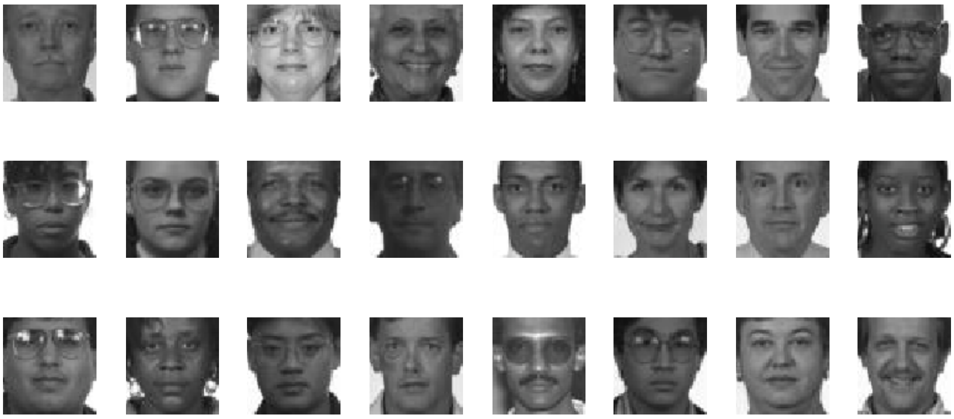


Figure 6.15: 24 images from a set of 58 faces with size 52 × 50, half with glasses and half without glasses.

Another example in this section is to train on a set of 58 faces with size 52 × 50, half with glasses and half without glasses(see Figure 6.15). Note that these two types of images are quite close to each other. An image of people wearing glasses has only a few pixels that are significantly different from the people not wearing glasses.

In Figure 6.16, they seem to try very hard to get convergence, and some templates have a sort of ghost of images of glasses while others do not. While this is an interesting phenomenon, I want to point out that this is definitely not a right way to classify faces wearing glasses. If you believe in compositionality, hierarchy, and reusability, you would not want to model the glasses in this way. You would model the pair of glasses alone and then, in the prior, allow the glasses to set on the face, and build a data model conditioning on the faces with glasses. Thus, this experiment is only meant to get a feeling for what happens in training. Moreover,

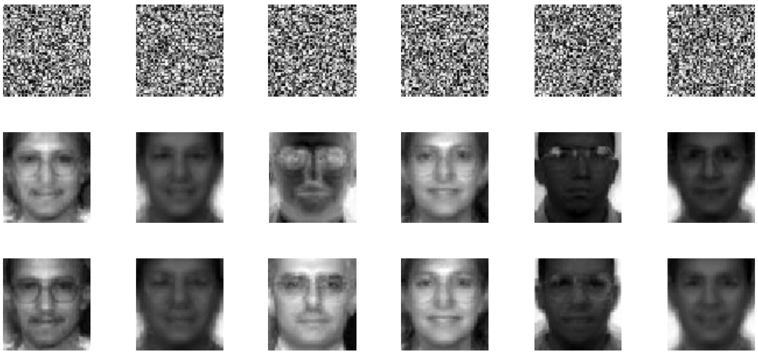


Figure 6.16: 6 templates learned from 58 faces with size 52 × 50, half with glasses and half without glasses.

if you randomly select 8 images from the training set and then order the templates from top to bottom by posterior likelihood like in Figure 6.17, you can see that we are not revealing particular prototype faces, and that they do not belong to anybody. They only belong to someone as a representative of a class under posterior likelihood.

In the recent decade, researchers started doing "coarse to fine" computation not only in the field of computer vision but also in many other fields. The following experiment that we want to present is the "coarse representation" of images. The feature we are using here is the normalized correlation of the down-converted image and the low-resolution template,

$$c(y) = cor(T, D(y))$$

where D is the down-converting operator. We trained on 499 face images and learned 8 coarse templates with size 10 × 10. In Figure 6.18, we can see that some of the down-converted images are very unclear, and they could not be detected even by human eyes. Yet, the learned templates seem to be much smoother and more closely resemble faces.

It is obvious that those down-converted images are not going to be accurate in

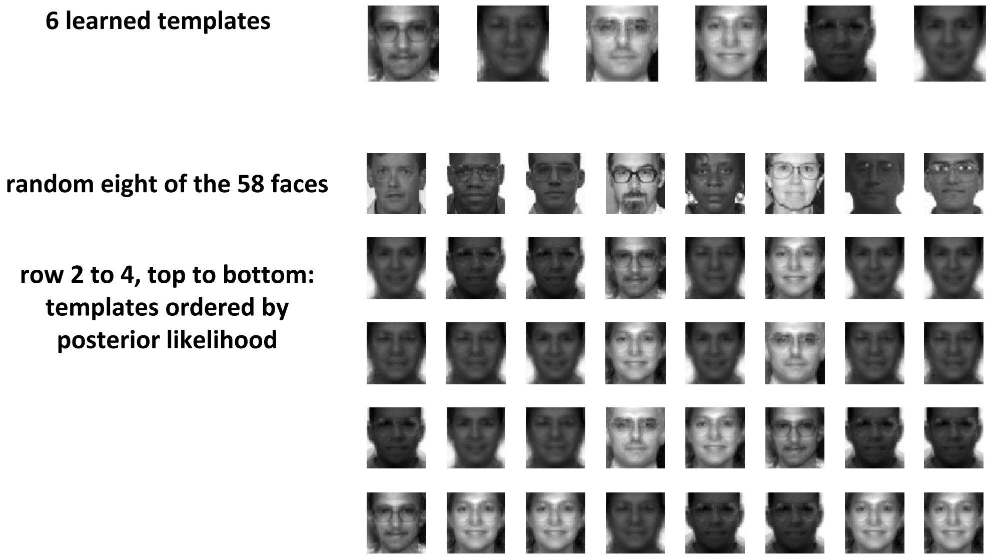


Figure 6.17: Templates ordered by posterior likelihood.

the coarse representation. However, the computer program for this low-resolution representation is very efficient in running the likelihood and getting posterior distribution on classes. Then within these classes, we can run finer templates. From a computational point of view, certainly in a compositional system, the reusable part computation is usually time-consuming, which becomes a big issue. The solution is always "coarse to fine." Moreover, in the computational system of our brains, we also compute in the way of "coarse to fine". We as humans can quickly tell that there is a human in our sight but how old the person is, whether he/she is a female or male, etc are computed on a finer level by somehow focusing our sight on this person.

6.4 Sampling and the choice of null distribution

One way to figure out whether or not we are on the right track is to actually look at samples. Since we have generative models, in principle, we can sample from the model. In practice, it is a little difficult but a few tricks here will get us closer to

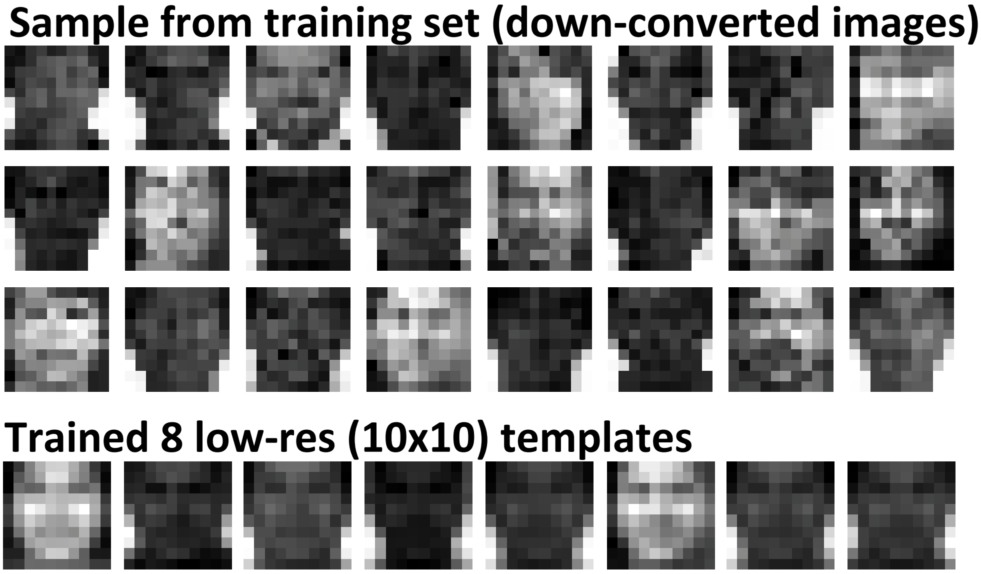

Sample from training set (down-converted images)
Trained 8 low-res (10x10) templates

Figure 6.18: The top panel shows 24 down-converted images with size 10×10. The bottom panel shows 8 trained templates.

exact samples. Let us first review our generative model.

$$P_Y(y) = \sum_{m=1}^{M} \epsilon_m \alpha_{\lambda_m} e^{\lambda_m(1-c_m(y))} P_Y^0(y|C_m = c_m(y)).$$

Then we propose a procedure to approximately sample from the model using the following steps.

Step 1: Standardize the templates and the patches randomly chosen from the background images.

$$T_m \mapsto \frac{T_m - \bar{T_m}}{|T_m - \bar{T_m}|},\ y \mapsto \frac{y - \bar{y}}{|y - \bar{y}|}$$

Note that whether or not we standardize the templates is irrelevant because we will get the same correlation. We can regard any two patches that have the same standard form as in an equivalent class. Thus, they are different only by shift and scale. Now, we are going to put a distribution on the $n-1$ dimensional unit sphere and view P_Y and P_Y^0 as distributions on the unit sphere in R^{n-1}, where $n = |S|$ is the number of template pixels.

Step 2: Choose a mixing component m according to $\epsilon_1, ...\epsilon_M$.

Step 3: Given the mixing component m, choose a correlation c according to $\alpha_{\lambda_m} e^{\lambda_m(1-c_m(y))}$.

Step 4: Choose a sample y according to P_Y^0 and implement the projection method below to get the sample $\tilde{y}$ that we want.

Projection Method : Here we have a unit sphere in R^{n-1} dimension as you can see in Figure 6.19. We take a image y on the sphere and project it on the plane, where all images have a correlation c, and then move it into the sphere.

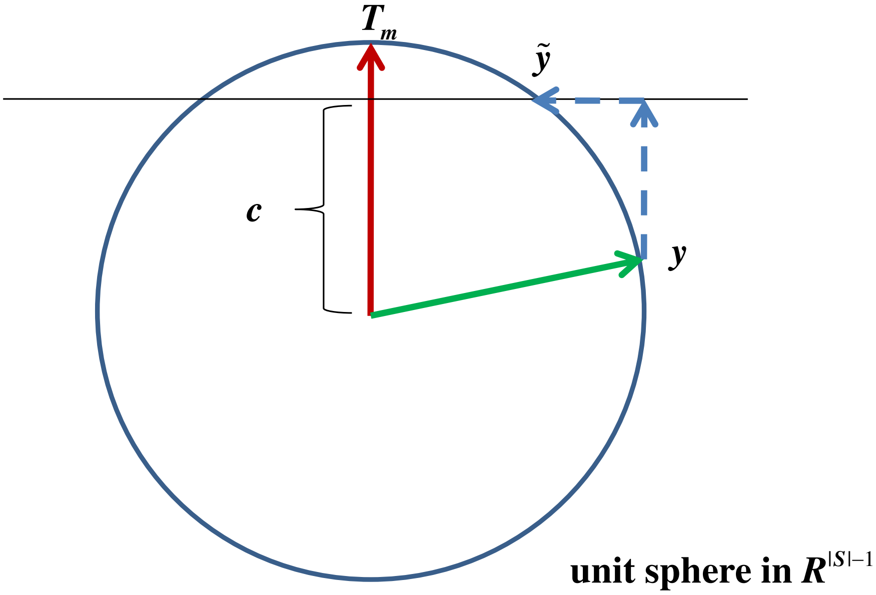


Figure 6.19: Projection method

Remarks:

1. We can prove that if the background model is i.i.d Gaussian then we will get an exact sample by implementing the sampling method above.

2. Certainly, the background model never looks like i.i.d. and it is usually unknown, but we still can get samples by taking random patches of background

images as our samples from the P_Y^0 and by using the projection method above. Although it is not an exact sample but an approximate sample, it is certainly accurate enough to give us a sense of what the sample looks like.

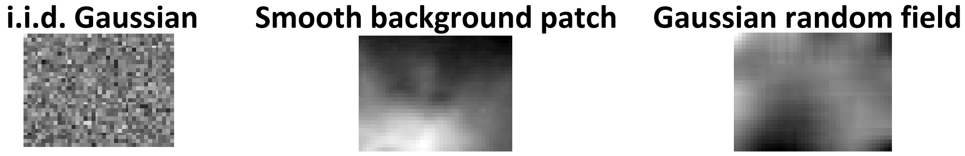


Figure 6.20: Three different background patches, i.i.d Gaussian, smooth real image patch, and Gaussian Random Field

3. The whole sampling idea is to ,instead of sampling from the background model which is difficult to model, randomly pick up background image patches from the background population. Therefore, how to choose the right null model population will be critical. Usually, we would like to have some smooth properties on the background, which are more similar to present real world effects, such as lighting, shadowing, reflecting, etc. Thus, we may want to choose some smooth background patches like the second image in Figure 6.20. The closest artificial background model that we could think of is the Gaussian Random Field, the third image in the Figure 6.20.

We will show some sampling associated with different choices of background populations. Figure 6.21 contains 30 samples from the mixture of noses under the i.i.d Gaussian background model. This is exact sampling if the real null model is i.i.d Gaussian, but the samples do not seem to have regularities, like shadows, smooth properties (neighbors tend to look similar), and so on.

In Figure 6.22, we take the Caltech 101 data which focuses on objects, and we use them to train and to sample from. Then, as you can see, these noses seem to be polluted by various structures that you could guess from the figure. It is

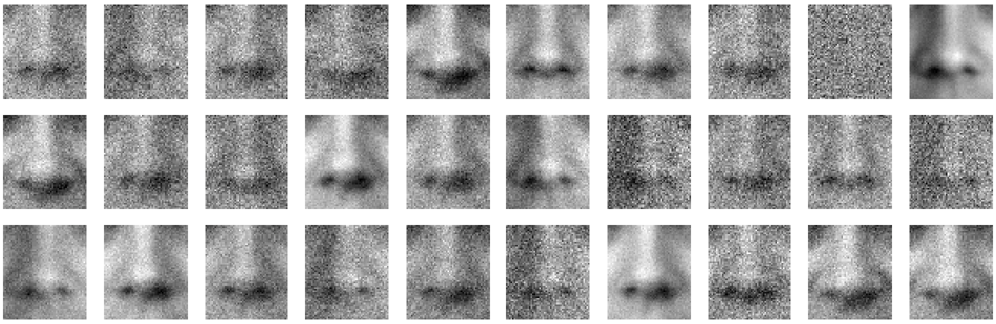


Figure 6.21: Exact sampling: we can get exact sampling if our background model is i.i.d Gaussian

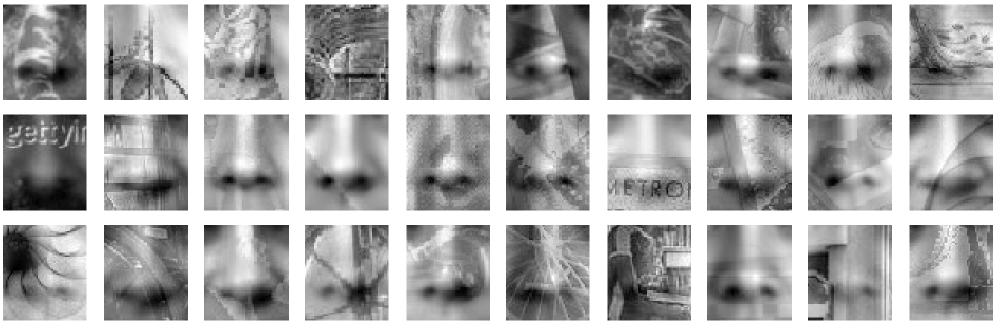


Figure 6.22: Approximate sampling: we pick out background image patches from Caltech 101 data

not reasonable to have samples with some random objects superimposed. Therefore, what we want to use as our background is something that can not be explained using our model. The trick is to build a system that can learn in that sense, so we can explain various pieces or parts of images which previously could not be explained. This always indicates "continuous" as a better notion of background.

Therefore, let us sample from our mixture model with smooth background image patches taken from natural scenes(see Figure 6.23). We take our null model to

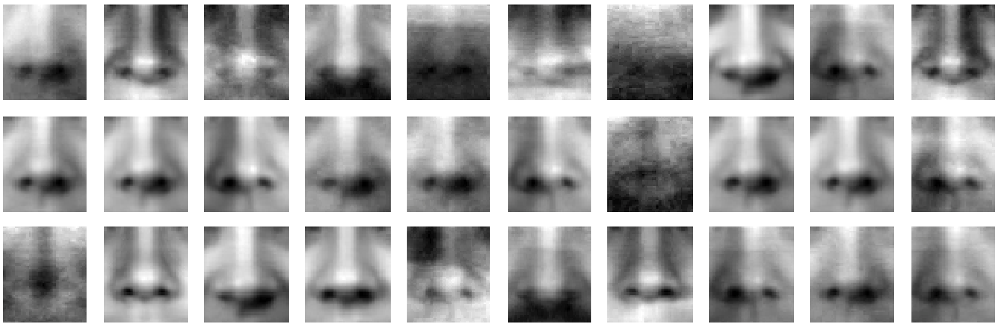


Figure 6.23: Approximate sampling: we pick out background image patches from real SMOOTH image patches.

constrain the gradient of the standardized patches:

$$max_{s\in S}|\nabla(\frac{y_s-\bar{y}}{|y_s-\bar{y}|})|<\eta$$

where η is a threshold. Now these samples look closer to what we expect with both shadows and real world effects. Moreover we do have an artificial smooth background model, namely the Gaussian Random Field, which is close to smooth natural images. Since it is a real model and not just a set of image patches, in principal, we should be able to sample exactly. However, exact sampling is still very difficult under this model due to its high dimensionality. Yet we still can obtain approximate samples by our trick, and Figure 6.24 shows a clean result similar to Figure 6.23.

6.5 Conditional modeling for overlapping templates

Throughout this computer vision part in this thesis, we are analyzing the scene, and each component of the analysis talks about what one part of this sense should look like. We can think of the scene as partitioned, so that there are many opinions and pieces of information from a particular image analysis or composition priors. We try to get a data model given image analysis or given samples from the prior. We do not want to cut out our patches or to train in such a way, partitioning the image,

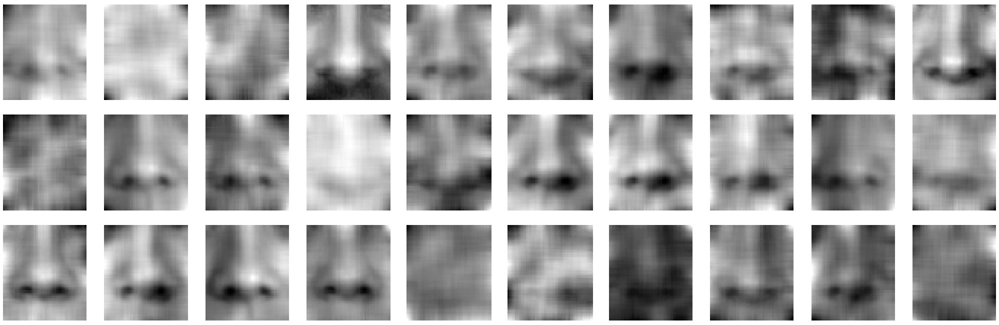


Figure 6.24: Approximate sampling: we pick out background image patches from Gaussian random field image patches.

but rather we want to model overlapping parts. Assume we have a nose template and an eye template that we trained individually. The problem is that the eye patch overlaps with the nose patch as in Figure 6.25.

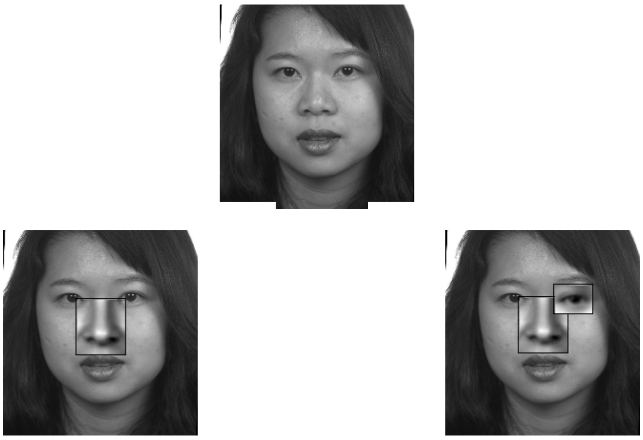


Figure 6.25: Two templates, an eye template and a nose template which have a small overlap.

We need to be able to write down the likelihood function of the overlapping region conditional on there being a nose and an eye. The same trick that we used for one part, say a nose, should also work for two parts here. Remember that using our trick, we ended up with a ratio of distributions of features in the product as follows:

$$L(\lambda, T) \propto \prod_{k=1}^{N} \frac{P_C(c(y_k))}{P_C^0(c(y_k))}.$$

Note that in order to make the topic clear, we simply address it on the basic model, not the mixture model, but the ideas are all extendable. Now we go through the entire procedure that we have done for one part. Then we will end up with the ratio

$$L(\lambda_e, \lambda_n, T_e, T_n) \propto \prod_{k=1}^{N} \frac{P_C(c_e(y_k^{(e)}), c_n(y_k^{(n)}))}{P_C^0(c_e(y_k^{(e)}), c_n(y_k^{(n)}))}.$$

once again, where the subindex $_e$ and $_n$ represent the eye part and the nose part respectively, $y_k^{(e)}$ and $y_k^{(n)}$ are the eye patch and nose patch in the k-th image. Therefore, both the denominator and numerator become *two* dimensional feature distributions. For the small overlapping, we can simply assume that the feature distribution of (c_e, c_n) is the product of their individual probabilities,

$$P_C(c_e(y_k^{(e)}), c_n(y_k^{(n)})) = P_C(c_e(y_k^{(e)}))P_C(c_n(y_k^{(n)})).$$

For the denominator, we can learn the two-dimensional feature distribution of the background by using the background image data. In particular, for the i.i.d background model, we can get an approximation by using the generalized Central limit Theorem as follows:

$$P_C(c_e, c_n) \sim \frac{1}{2\pi\sqrt{|\Sigma|}} exp\left(-\frac{1}{2}(c_e, c_n)\Sigma^{-1}\binom{c_e}{c_n}\right)$$

where the covariance matrix

$$\Sigma = \begin{pmatrix} \sigma_e^2 & \rho\sigma_e\sigma_n \\ \rho\sigma_e\sigma_n & \sigma_n^2 \end{pmatrix}$$

where $\sigma_e = \frac{1}{\sqrt{|S_e|}}$, $\sigma_n = \frac{1}{\sqrt{|S_n|}}$ and $\rho = \tilde{T_e} \cdot \tilde{T_n}$, the inner product of two overlapping parts from the two *standardized* templates T_e and T_n correspondingly.

6.6 Appendix

We usually implement an artificial model as our initial background model in order to get an initial approximation of the templates. Based on these templates, we can update the background distribution of the features. Thus, we need to derive the probability distribution of features under the artificial model. In particular, the i.i.d background model is often used in many applications. The following Lemma gives us an approximation that can be applied to our work.

Lemma: Let $Y^{(n)} = (y_1^{(n)}, y_2^{(n)}, ..., y_n^{(n)})$, where $y_1^{(n)}, ..., y_n^{(n)}$ are i.i.d random variables whose third moment $E((y_i^{(n)})^3)$ exists and is finite. Let our template $T^{(n)} = (t_1^{(n)}, t_2^{(n)}, ..., t_n^{(n)})$, where $\sum_{i=1}^n t_i^{(n)} = 0$ and $\sum_{i=1}^n (t_i^{(n)})^2 = n$. Assume

$$|t_i^{(n)}| \leq M,$$

where M is a constant, independent of n. Then,

$$\sqrt{n} cor(Y^{(n)}, T^{(n)}) = \frac{\sum_{i=1}^n t_i^{(n)} (y_i^{(n)} - \bar{y}^{(n)})}{\sqrt{\sum_{i=1}^n (y_i^{(n)} - \bar{y}^{(n)})^2}}$$

converges to $N(0,1)$ in distribution.

Proof: The key of the proof is to use Lyapunov's Central Limit Theorem. Let us check the Lyapunov Condition. Let $x_i = t_i^{(n)}(y_i^{(n)} - \bar{y}^{(n)})$. Then $E(x_i) = 0$ and

$$s_n^2 = \sum_{i=1}^n E(t_i^{(n)}(y_i^{(n)} - \bar{y}^{(n)}))^2 = nE(y_1^{(n)} - \bar{y}^{(n)})^2.$$

Now third moment,

$$r_n^3 = \sum_{i=1}^n E|t_i^{(n)}(y_i^{(n)} - \bar{y}^{(n)})|^3 \leq MnE|y_1^{(n)} - \bar{y}^{(n)}|^3.$$

Therefore,

$$\lim_{n \to \infty} \frac{r_n}{s_n} \longrightarrow 0.$$

Thus, it satisfies the Lyapunov Condition, and by Lyapunov's Central Limit Theorem,

$$\sqrt{n}cor(Y^{(n)}, T^{(n)}) = \frac{\sum_{i=1}^{n} t_i^{(n)}(y_i^{(n)} - \bar{y}^{(n)})}{\sqrt{\sum_{i=1}^{n}(y_i^{(n)} - \bar{y}^{(n)})^2}} \Longrightarrow N(0,1),$$

and the proof is complete.

Remark: The only condition for the i.i.d background model of the Lemma above is that the third moment exists. Therefore, the two commonly used models, i.i.d Uniform and i.i.d Gaussian, satisfy the condition, and we can approximate the $P_C^0(y)$ by

$$\frac{\sqrt{n}}{\sqrt{2\pi}} e^{-\frac{nc(y)^2}{2}}$$

where $n = |S|$ is the number of pixels.

Chapter 7

Context, Computation, and Optimal ROC Performance in Hierarchical Models

7.1 Introduction

It is widely recognized that human vision relies on contextual information, typically arising from each of many levels of analysis. Local gradient information, otherwise ambiguous, is seen as part of a smooth contour or sharp angle in the context of an object's boundary or corner. A stroke or degraded letter, unreadable by itself, contributes to the perception of a familiar word in the context of the surrounding strokes and letters. The iconic dalmatian dog stays invisible until a multitude of clues about body parts and posture, and figure and ground, are coherently integrated. Context is always based on knowledge about the composition of parts that make up a whole, as in the arrangement of strokes that make up a letter, the arrangement of body parts that make up an animal, or the poses and postures of individuals that make up a mob. From this point of view, the hierarchy of contextual information available to an observer derives from the compositional nature of the world being observed. Here we will formulate this combinatorial viewpoint in terms of probability distributions and examine the computational implications. Whereas optimal recognition performance in this formulation is provably NP-hard, we will give mathematical evidence that a properly orchestrated computational algorithm can achieve nearly optimal recognition within a feasible number of operations.

A frame from a 1920's shot of the expressionless face of the Russian actor Ivan Mozzhukhin is shown, repeatedly, on the right hand side of Figure 7.1. The shot was captured by the director Lev Kuleshov as part of an experiment in context and a study of its role in the cinematic experience. In three separate clips, Kuleshov juxtaposes the shot with a dead child lying in an open coffin, a seductive actress, or a bowl of soup. Asked to interpret Mozzhukhin's expression, audiences reported sadness, lust, or hunger depending on whether the expression followed the images of the dead child, the seductive actress, or the bowl of soup. Many praised the actor's skill. The idea that the movie-going experience is based on composition as much as content became the basis for the so-called montage school of Russian cinema and it remains an essential tool of modern filmmaking.

The effects of context on human perception have been well studied for hundreds of years, and are well illustrated with many familiar illusions involving size and boundary perception, grouping, and shading. But most contextual effects are not illusory. Sighted people are all experts at vision, which makes it difficult, if not impossible, to appreciate the multiple levels of context that critically influence virtually every visual perception. On the other hand, engineers trying to build artificial vision systems invariably discover the ambiguities in the raw pixel data, or in any more-or-less local grouping of pixel data. It is often impossible to decipher cursive, even one's own cursive, without a measure of context, which might come from any one of many levels, including topic, sentence, or just neighboring words or letters. The same effect is striking when applied to auditory signals, where, for example, words spliced from continuous speech are often unintelligible.

Many cognitive scientist would argue that the layers of context that influence the perception of a part or object (e.g. a phoneme or a word) are a manifestation of the compositional nature of mental representation (e.g. [26]). The vision scientist might be tempted to turn this around and say that these representations are themselves manifestations of the compositional nature of the visual or auditory world, but either way, or both, the evidence is that biological-level performance in perceptual tasks relies on knowledge about the relational groupings of parts into wholes, simultaneously at multiple levels of a hierarchy. This combinatorial, or compositional, viewpoint is a common starting point for discriminative or generative models of vision, often within grammar or grammar-like organizations ([5], [52], [70], [41], [69]). The idea in generative models is to use probability distributions to capture likely and unlikely arrangements, starting from arrangements of local features (e.g. local edges or texture elements), and in principle continuing recursively to high-level expert knowledge (e.g. a curator's knowledge about a style of antique furniture, a grandmaster's knowledge about the strengths and weaknesses of an arrangement of pieces on the chess board, a world-class soccer player's knowledge about the posture and likely actions of an opponent). We will adopt the generative compositional viewpoint here, and use it to examine the practical problems of clutter, false alarms, and computational burden in artificial vision systems.

Clutter and the Limits of Artificial Vision Systems. It is one thing to build a classification device that performs on images with single objects placed in simple backgrounds and quite another to find and classify these same objects in unconstrained scenes. Everyone who builds vision systems knows this. Real background has structures, and too often these structures masquerade as bits and pieces of the objects of interest. Run a correlator for an eye, with say a 10x10 patch, on backgrounds in an ensemble of images with bricks and trees and cars (e.g. mid-day Manhattan street scenes as captured by Google's Street View) and you'll probably get many good matches per frame, if "good match" is defined to be at least as good as 5% of the matches to real eyes in the same scenes. This kind of thing is to be expected, if you buy the compositional point of view. In particular, the parts of an object of interest, such as a face, are reusable and can be found among the pieces making up many other structures. It's not that there are actual eyes in and among the bricks, bark, and leaves, but that poorly-resolved oval shapes, with darker centers and lighter surrounds, are not uncommon and certainly not unique to faces. Indeed, if it were otherwise, then excellent performance on face detection tasks could be achieved by looking for nothing but eyes. But the fact is that state-of-the-art face-detection algorithms, still not as good as human observers, require more than just eyes. Google Street View, in order to achieve a high certainty of detecting and obscuring real faces, blurs many false-detections on car wheels, trees, or just about anyplace that includes structured or textured background. When operating at the same detection level, humans get almost no false positives.

In general, artificial vision systems operating at the high-detection end of the ROC curve suffer many more false detections in unconstrained scenes than do human observers. If we think of a "part" as being *defined by* its local appearance, rather than its participation in any particular object, then we can think of these false detections as typically arising from an unlucky arrangement of subparts of the objects of interest. A human interprets these same arrangements for what they are: parts of other objects, or objects in their own right. One could reasonably argue, then, that solving one vision problem, say the detection of a single object, requires solving many vision problems, at least the detection of any other object that shares aspects of its appearance, i.e. shares parts, with the object of interest. How much knowledge is needed to achieve biological-level performance on a single vision task? Is it necessary to know about all objects to accurately detect a single object? In short, is vision "AI-complete"?

We will argue in the opposite direction. We will give evidence that, to the extent the world is compositional, a vision system can achieve nearly optimal performance on a particular vision task, involving a single selected object or a particular library of objects, by modeling only the object or objects of interest. The idea is that most false detections occur at background locations that share bits and pieces of the objects of interest, suggesting that the objects themselves, viewed as compositional, define adequate background models through their own subparts and arrangements of subparts; in a compositional world, objects define their own background models ([41]).

Matching Templates versus Matching Parts. We often think of cascades and other coarse-to-fine strategies as computational imperatives. Even if we had a full-blown model for the appearance of an object, it would not be infeasible to search for it at every pose (already six degrees of freedom for a rigid object). Except in very special circumstances, practical vision systems have to use some form of coarse-to-fine search, usually involving a very simple first pass that highlights candidate poses, followed by a sequence of more refined and constrained searches in the neighborhoods of the candidate poses. Computation might be organized as a tree, for example to search simultaneously for multiple objects, or a cascade, which might be more suitable for single objects. The computational advantages are well documented, both from a practical and a theoretical standpoint ([5], [25], [14], [63]).

But computation might not be the whole story. There might be other reasons for prefering a divide-and-conquer strategy. Consider an imaginary object $\mathcal{O}$ that can appear at only one pose, and an imaginary situation in which we have a fully specified render model for the distribution on images given that $\mathcal{O}$ is present. How would we test for $\mathcal{O}$? How do we compare the hypothesis "$\mathcal{O}$ is present" to the alternative "$\mathcal{O}$ is absent"? It is not enough to have an appearance model for $\mathcal{O}$; we also need an appearance model for scenes without $\mathcal{O}$. The trouble is that "$\mathcal{O}$ absent" is an unimaginably large mixture. What is more, as we have already observed, this mixture will typically include components that represent objects with similarities to $\mathcal{O}$, portions of which might be essentially indistinguishable from portions of $\mathcal{O}$.

An expedient approach would be to adopt a simple"background model," meaning some kind of manageable alternative distribution such as iid pixel intensities, or more generally a random field that might capture local correlations. To the extent that the background model is accurate, the likelihood ratio, the probability of the observed image given that $\mathcal{O}$ is present to the probability under the background model, is an optimal statistic for this two-class problem (i.e. thresholding on the ratio will minimize false alarms at any given detection rate). Another approach, also expedient, would be to sequentially test for the presence of parts of $\mathcal{O}$. If all of the parts are found, then declare that $\mathcal{O}$ is present. The same simple background model could be used, locally, to test for the individual parts.

Both approaches have advantages. The first, which is essentially a template match, is relatively robust to a noisy presentation of the object. The parts may be difficult to confirm, individually, but the collective evidence could be strong. The second, although vulnerable to a poorly rendered part, has an easier time distinguishing false alarms when the actual scene contains parts and objects that resemble pieces of $\mathcal{O}$, but not $\mathcal{O}$ itself. Our purpose is to argue, through mathematical and empirical evidence, that the second approach, parts-based testing, is superior, especially when operating at a high-detection threshold. In fact, it might not be far from optimal. We will propose a particular version of parts-based testing that is suitable for compositional models, and is recursive for hierarchical models.

We will work through a simple thought experiment, not unlike the discussion here of the fictional object $\mathcal{O}$. We will formulate the detection problem in such a way that we can compare three approaches, the optimal approach (based on the Neyman-Pearson Lemma), the template approach, and the parts-based approach. The comparison will be mathematical, via comparisons of the area under the ROC curve for each of the three alternatives, and via experiments with real data chosen to be simple enough that good approximations to each of the three approaches can be computed.

7.2 A Simple World of Parts and Objects

We start with a minimal world of parts and objects, depicted in Figure 7.2. There are two parts, vertical and horizontal bars, and one object, the letter L. The model is generative and includes latent variables, one for each part, that define an "interpretation," and a conditional rendering distribution for the image given an interpretation. The latent variables, denoted X_1 and X_2 for the vertical and horizontal bars, respectively, are each binary ($X_1, X_2 \in \{0,1\}$), representing the absence (0) or presence (1) of a part. The joint probability on parts is $P(x_1, x_2),\ x_1, x_2 \in \{0,1\}$.[1] Referring to Figure 7.3, the set of all pixels is denoted S and the subsets of pixels affected by the presence or absence of the parts are denoted S^1 and S^2, for the horizontal and vertical bars respectively. We will refer to S_1 and S_2 as the "supports" of their respective parts. The intensity of pixel $s \in S$ is treated as a random variable and is denoted Z_s. Generically, given any set of pixels $A \subseteq S$, we use lexicographic (raster) ordering to define a vector of intensities Z_A from the set $\{Z_s : s \in A\}$.

The generative model generates an image (Z_S) by first generating an interpretation according to the joint distribution on X_1 and X_2; then assigning intensities iid in S_1 and, independently, iid in S_2 according to $N(x_1, 1)$ and $N(x_2, 1)$, respectively; and finally, independently of everything else, assigning intensities iid in $S \setminus (S_1 \cup S_2)$ according to $N(0,1)$. In short, $P(z_S, x_1, x_2) = P(z_S | X_1 = x_1, X_2 = x_2)P(x_1, x_2)$, where

$$\begin{aligned} P(z_S|X_1 = x_1, X_2 = x_2) &= P(z_{S^1}|X_1 = x_1)P(z_{S^2}|X_2 = x_2)P(z_{S\setminus(S^1\cup S^2)}) \qquad (7.1)\\ &= \prod_{s\in S^1} G(z_s; x_1, 1) \prod_{s\in S^2} G(z_;x_2, 1) \prod_{s\in S\setminus(S^1\cup S^2)} G(z_s; 0, 1) \end{aligned}$$

and $G(z; \mu, \sigma)$ stands for the normal probability density (mean μ and standard de-

[1] We will reserve the more definite notation $P_{X_1,X_2}(x_1, x_2)$ (instead of just $P(x_1, x_2)$) for cases in which there is a possibility of confusion.

viation σ) evaluated at z.

Imagine now that we are presented with a sample image generated by the model. Our problem is to decide whether or not the image contains the letter L. We will devise and analyze several decision rules, and later relate the conclusions to more general and relevant models, and to the discussion of clutter, context, and computation.

Optimal Decision Rule. In this example, the presence of an L is equivalent to the presence of horizontal and vertical bars, i.e. the event $\{X_1 = x_1 \bigcap X_2 = x_2\}$. This suggests thresholding on the posterior probability, $\mathcal{S}_G(z_S) \doteq P(X_1 = x_1, X_2 = x_2 | Z_S = z_S)$:

$$\textit{Declare “L” if } \mathcal{S}_G(z_S) > t \textit{ and “not L” if } \mathcal{S}_G(z_S) \leq t.$$

The threshold governs the tradeoff between false alarms and missed detections, and the set of all thresholds defines the ROC curve. The decision rule is optimal in that it minimizes the probability of missed detections at any given probability of false alarms. (This follows from the Neyman-Pearson Lemma and the observation that $\mathcal{S}_G(z_S)$ is a monotone increasing function of the likelihood ratio $\frac{P(z_S | L\ present)}{P(z_S | L\ not\ present)}$.)

Observations:

1.

$$\begin{aligned} \mathcal{S}_G(z_S) &= \frac{P(z_S | X_1 = x_1, X_2 = x_2) P(1,1)}{\sum_{x_1=0}^{1} \sum_{x_2=0}^{1} P(z_S | X_1 = x_1, X_2 = x_2) P(x_1, x_2)} \\ &= \frac{P(z_{S^1} | X_1 = 1) P(z_{S^2} | X_2 = 1) P(1,1)}{\sum_{x_1=0}^{1} \sum_{x_2=0}^{1} P(z_{S^1} | X_1 = x_1) P(z_{S^2} | X_2 = x_2) P(x_1, x_2)} \end{aligned} \tag{7.2}$$

which follows from Bayes' formula and the decomposition in equation (7.1).

2. Also by equation (7.1):

$$\begin{aligned}\mathcal{S}_G(z_S) &= P(X_1 = 1, X_2 = 1 | Z_S = z_S) \\ &= P(X_1 = 1 | Z_S = z_S) P(X_2 = 1 | X_1 = 1, Z_S = z_S) \\ &= P(X_1 = 1 | Z_{S^1} = z_{S^1}, Z_{S^2} = z_{S^2}) P(X_2 = 1 | X_1 = 1, Z_{S^2} = z_{S^2}) \quad (7.3)\end{aligned}$$

As this is the product of two conditional probabilities, it suggests a sequential version of the test $\mathcal{S}_G(z_S) > t$. In particular, if $P(X_1 = 1|Z_{S^1} = z_{S^1}, Z_{S^2} = z_{S^2}) > t$ fails then there is no point in computing $P(X_2 = 1|X_1 = 1, Z_{S^2} = z_{S^2})$, since $\mathcal{S}_G(z_S) \leq P(X_1 = 1|Z_{S^1} = z_{S^1}, Z_{S^2} = z_{S^2})$. If it does not fail, then we compute $P(X_2 = 1|X_1 = 1, Z_{S^2} = z_{S^2})$ and compare the product $P(X_1 = 1|Z_{S^1} = z_{S^1}, Z_{S^2} = z_{S^2})P(X_2 = 1|X_1 = 1, Z_{S^2} = z_{S^2})$ to t. We will return to this shortly.

Template Matching. The problem with $\mathcal{S}_G$ is that it can not possibly be computed in anything but a trivial model, as is evident from examining equation (7.2). The denominator is the full likelihood, meaning a mixture over *every possible explanation* of the data. The mixture has one term for "$\{\text{L}\} \bigcap \{Z_S = z_S\}$" and all the rest for "$\{\text{not an L}\} \bigcap \{Z_S = z_S\}$." It is one thing to compute (or estimate) a reasonable likelihood for "nothing there," but quite another to compute a likelihood for "not an L."

A sensible, and in one way or another much-used, alternative is to approximate "$\{\text{not an L}\} \bigcap Z_S$" by "$\{\text{nothing there}\} \bigcap \{Z_S = z_S\}$," i.e. to use the statistic

$$\mathcal{S}_T(z_S) \doteq \frac{P(z_S|X_1 = 1, X_2 = 1)P(1,1)}{P(z_S|X_1 = 0, X_2 = 0)P(0,0) + P(z_S|X_1 = 1, X_2 = 1)P(1,1)} \quad (7.4)$$

Observations:

1. By the same reasoning used for $\mathcal{S}_G$:

   $$\mathcal{S}_T(z_S) = \frac{P(z_{S^1}|X_1 = 1)P(z_{S^2}|X_2 = 1)P(1,1)}{P(z_{S^1}|X_1 = 0)P(z_{S^2}|X_2 = 0)P(0,0) + P(z_{S^1}|X_1 = 1)P(z_{S^2}|X_2 = 1)P(1,1)}$$

2. $\mathcal{S}_T$ is optimal under a different probability, $\tilde{P}$, on the latent variables:

   $$\mathcal{S}_T(z_S) = \tilde{P}(X_1 = x_1, X_2 = x_2|Z_S = z_S)$$

   where

   $$\tilde{P}(z_S, x_1, x_2) = P(z_S, x_1, x_2|(X_1, X_2) = (1,1) \text{ or } (X_1, X_2) = (0,0))$$

   Roughly speaking, we pretend that the world has only two states, "object" or "nothing."

Sequential Testing for Parts. This is a modification of the sequential version of the optimal decision rule (7.3). The second, postponed, computation is of $P(X_2 = 1|X_1 = 1, Z_{S^2} = z_{S^2})$. This is local to S^2 and scales efficiently to larger systems. (We will have more to say about scaling later.) On the other hand, the first computation, of $P(X_1 = 1|Z_{S^1} = z_{S^1}, Z_{S^2} = z_{S^2})$, is global in the sense that it involves the evaluation of likelihoods for every state of every other part in the object. This is exponential in the number of parts. These observations suggest a third statistic, derived by approximating $P(X_1 = 1|Z_{S^1} = z_{S^1}, Z_{S^2} = z_{S^2})$ with the corresponding local probability $P(X_1 = 1|Z_{S^1} = z_{S^1})$:

$$\mathcal{S}_P(z_S) \doteq P(X_1 = 1|Z_{S^1} = z_{S^1})P(X_2 = 1|X_1 = 1, Z_{S^2} = z_{S^2})$$

The test $\mathcal{S}_P(z_S) > t$ can be performed sequentially. The first test is for the first part ($P(X_1 = 1|Z_{S^1} = z_{S^1}) > t$), *ignoring* information in the pixel data about the second part. If $P(X_1 = 1|Z_{S^1} = z_{S^1}) > t$ then the second part is tested (via $P(X_1 = 1|Z_{S^1} = z_{S^1})P(X_2 = 1|X_1 = 1, Z_{S^2} = z_{S^2}) > t$), using the pixels in the support of the second part and a probability that is computed in the context of the presumed presence of the first part.

Foveal Limit. We want to compare these three strategies. The optimal serves as a benchmark against which the performance of template matching and the sequential parts tests can be measured. The set up is simple enough that both mathematical and exhaustive computational analyses are possible. Concerning mathematical analysis, we will examine relative performances by comparing the ROC curves in the limit as the density of pixels goes to infinity (the "foveal limit"). In other words, spacing between pixels of the uniform grid S in Figure 7.3 is decreased to zero.

All three approaches are perfect in the foveal limit. Hence the areas under the three ROC curves converge to one. We will compare the rates at which the areas *above* the ROC curves converge to zero. Obviously, neither template matching nor sequential testing can do better than optimal. But which of the two suboptimal approaches should we expect to better approximate optimal performance? One way to think about this is to anticipate the primary sources of confusions for each of the suboptimal tests. Consider two sets of circumstances. In the first, both parts are present ($X_1 = 1$ and $X_2 = 1$) but one or the other of the parts is substantially degraded. A template takes into account all of the data, and from this point of view the situation is not really different from a uniform, but less severe, degradation of the entire L. As for the sequential test, it is vulnerable to missing the degraded part, especially when the degraded part is tested first.[2] On the other hand either part can appear alone, and in such cases template matching, in that it is essentially making a forced decision between $X_1 = X_2 = 1$ and $X_1 = X_2 = 0$, is vulnerable to false alarms. It turns out that the consequences of the second circumstance dominate, overwhelmingly.

[2] The reader might be tempted to conclude that the optimal test should suffer the same vulnerability to a degraded part, in light of the sequential representation of equation (7.3). But, to the contrary, the first test takes into account the appearances of both parts, and the second test, when performed, is based on both the presence of the first part and the intensities in the support of the second part.

To make this precise, given a statistic $\mathcal{S} = \mathcal{S}(z_S)$, let $\mathcal{A}_\mathcal{S}$ be the area above the ROC curve generated by the test $\mathcal{S}(z_S) > t$. Necessarily, $\mathcal{A}_{\mathcal{S}_T} \geq \mathcal{A}_{\mathcal{S}_G}$ and $\mathcal{A}_{\mathcal{S}_P} \geq \mathcal{A}_{\mathcal{S}_G}$ (Neyman Pearson). Concerning the simple two-part world constructed above:

Theorem. If $P_{X_1,X_2}(x_1,x_2) > 0$ for every pair $(x_1,x_2) \in \{0,1\}^2$, then in the foveal limit

1. $\mathcal{A}_{\mathcal{S}_P}/\mathcal{A}_{\mathcal{S}_G}$ remains bounded;

2. $\mathcal{A}_{\mathcal{S}_T}/\mathcal{A}_{\mathcal{S}_P} \to \infty$ exponentially fast.

Remarks.

1. The first result, that $\mathcal{A}_{\mathcal{S}_P}/\mathcal{A}_{\mathcal{S}_G}$ remains bounded, is a little surprising given that in the foveal limit both $\mathcal{S}_G(z_S) > t$ and $\mathcal{S}_P(z_S) > t$ are based on the ratios of likelihoods of increasing numbers of independent random variables. Likelihood ratios generally converge to zero or to infinity exponentially fast, as in the second result.

2. The conclusions are the same if $\mathcal{S}_P(z_S) = P(X_1 = 1|Z_{S^1} = z_{S^1})P(X_2 = 1|X_1 = 1, Z_{S^2} = z_{S^2})$ is replaced by $\mathcal{S}_Q(z_S) \doteq P(X_1 = 1|Z_{S^1} = z_{S^1})P(X_2 = 1|Z_{S^2} = z_{S^2})$ (i.e. if the discovery of the first part is ignored in testing for the second part), but this does not mean that the test $\mathcal{S}_Q(z_S) > t$ is as good as the test $\mathcal{S}_P(z_S) > t$. To the contrary, the experimental evidence strongly favors $\mathcal{S}_P(z_S)$ for moderate pixel densities, but a mathematical proof might call for a more delicate analysis.

3. The proof requires a large-deviation principle. From this point of view, there is nothing special about the Gaussian assumption. More generally, any "ordinary" distribution (there is a moment condition) could be used instead.

Proof. Generically, for any statistic $\mathcal{S}(z_S)$

$$\mathcal{A}_\mathcal{S} = Prob\{\mathcal{S}(Z_S) < \mathcal{S}(\tilde{Z}_S)\}$$

where Z_S and $\tilde{Z}_S$ are independent samples from $P(z_S|\{X_1 = 1\} \cap \{X_2 = 1\})$ and $P(z_S|\{\{X_1 = 1\} \cap \{X_2 = 1\}\}^C)$, respectively. This, along with the various independence assumptions, makes the comparisons relatively straightforward. The detailed proof is in the Appendix.

The three ROC curves can be computed numerically. Figure 7.4 explores performance of all three classifiers as a function of resolution, for small and moderate pixel densities as well as the larger densities corresponding to the "foveal limit" covered by the theorem. At the lowest density there are only two pixels in the support of the horizontal bar and four in the support of the vertical. Template matching is not far from optimal, and better than parts-based testing. The order is already reversed when there are just four and eight pixels representing the horizontal and vertical bars, respectively. With eight and sixteen pixels, parts-based testing is nearly indistinguishable from optimal, and substantially outperforms the template model. A glance at higher resolutions confirms that the template model converges to perfect classification much more slowly than the other two.

Saliency. Even without occlusion (which we would argue should be addressed by a layered model), different parts of an object will likely show up with more or less clarity. The local evidence for some parts will be stronger than for others. In as much as sequential testing is most vulnerable to a degraded view of the first part tested, it makes sense to look first at those parts for which the evidence is strongest (loosely speaking, the "most salient parts"). When there are many parts, instead of just two, then the idea can be applied recursively: first test for the most salient part, then test for the conditionally most salient part, given the part already found, and so on. The result is order dependent because tests for all but the first part are conditioned on the presence of the earlier parts. Here we take a closer look at these ideas, by extending the theorem from two parts to an arbitrary number of parts, and from a fixed-order sequential test to a data-dependent ordering. We illustrate with some additional experiments.

Suppose that our object, call it $\mathcal{O}$, is made of N parts rather than just two. Extending the notation in the obvious way, we let $X_k \in \{0,1\}$ indicate the absence or presence of the $k'th$ part, $1 \leq k \leq N$, let $S^k \subseteq S$ be the pixels in the support of the $k'th$ part, and let Z_{S^k} be the corresponding pixel intensities. We assume that $S_j \cap S_k = \emptyset$, for all $j \neq k$, though there will be some discussion of this in the next section. The joint distribution of $Z_S, X_1, X_2, \ldots, X_N$ is modeled by extension of the L model: $P(z_S, x_1, x_2, \ldots, x_N) = P(z_S|X_1 = x_1, X_2 = x_2, \ldots, X_N = x_N)P(x_1, x_2, \ldots, x_N)$, where

$$\begin{aligned} P(z_S|X_1 = x_1, X_2 = x_2, \ldots, X_N = x_N) &= P(z_{S \setminus \bigcup_{k=1}^n S^k}) \prod_{k=1}^N P(z_{S^k}|X_k = x_k) \qquad (7.5) \\ &= G(z_{S \setminus \bigcup_{k=1}^n S^k}; 0, 1) \prod_{k=1}^N G(z_{S^k}; x_k, 1) \end{aligned}$$

and $G(z_A; \mu, \sigma)$ stands for $\prod_{s \in A} G(z_s; 0, 1)$ (iid normal) for any $A \subseteq S$. Finally, we say that the object $\mathcal{O}$ is present if and only if all of its parts are present.

The extensions of the optimal decision rule $(\mathcal{S}_G(z_S) > t)$ and template matching $(\mathcal{S}_T(z_S) > t)$ involve straightforward changes in the statistics:

$$\mathcal{S}_G(z_S) \doteq P(X_1 = 1, X_2 = 1, \ldots, X_N = 1|Z_S = z_S)$$

and

$$\mathcal{S}_T(z_S) \doteq \frac{P(z_S|X_1 = 1, X_2 = 1, \ldots, X_N = 1)P(1, 1, \ldots, 1)}{P(z_S|X_1 = 0, \ldots, X_N = 0)P(0, \ldots, 0) + P(z_S|X_1 = 1, \ldots, X_N = 1)P(1, \ldots, 1)}$$

All of the various observations about these two statistics, made earlier for the case $N = 2$, still hold when $N \geq 2$, with obvious changes in the formulas.

As for the sequential test $(\mathcal{S}_P(z_S) > t)$, we want to make a more fundamental change by extending it to allow for a data-dependent sequence of tests. The first test is directed at the "most salient part," by which we mean the most probable part when only local evidence is taken into account (i.e. based on Z_{S^k} and not Z_S):

$$k_1 \doteq \arg\max_k P(X_k = 1 | Z_{S^k} = z_{S^k}) \tag{7.6}$$

The first test is $P(X_{k_1} = 1 | Z_{S^{k_1}} = z_{S^{k_1}}) > t$. If it succeeds, then we compute the most salient of the remaining parts, but now in the context of $X_{k_1} = 1$:

$$k_2 \doteq \arg\max_{k \neq k_1} P(X_k = 1 | X_{k_1} = 1, Z_{S^k} = z_{S^k})$$

Iterating through N parts generates a *random sequence* $k_i = k_i(Z_S)$, $i = 1, 2, \ldots, N$, and defines a random (data-dependent) factorization:

$$\mathcal{S}_P(z_S) \doteq \prod_{i=1}^{N} P(X_{k_i} = 1 | X_{k_1} = 1, \ldots, X_{k_{i-1}} = 1, Z_{S^{k_i}} = z_{S^{k_i}}) \tag{7.7}$$

Corollary. The theorem holds under the extended definitions of $\mathcal{S}_G$, $\mathcal{S}_T$, and $\mathcal{S}_P$.

Proof. There is very little different from the proof as already given in the Appendix, and we forgo the details.

The difference between a fixed-order testing of parts and a saliency-based testing can be illustrated by returning to the simple L world and performing the same experiment as reported in Figure 7.4, but with

$$\mathcal{S}_P(z_S) = P(X_{k_1} = 1 | Z_{S^{k_1}} = z_{S^{k_1}}) P(X_{k_2} = 1 | X_{k_1} = 1, Z_{S^{k_2}} = z_{S^{k_2}})$$

instead of

$$\mathcal{S}_P(z_S) = P(X_1 = 1 | Z_{S^1} = z_{S^1}) P(X_2 = 1 | X_1 = 1, Z_{S^2} = z_{S^2})$$

Figure 7.5 is identical to Figure 7.4, except that the random-order parts-based test was used. Comparing to Figure 7.4, parts-based testing is now nearly equivalent to optimal testing at all resolutions. It is intuitive that visiting parts in the order of saliency is better than using a fixed order, especially in the low-resolution domain, and no doubt something can be proven along these lines. But the approach will need to be different, since the analysis behind the theorem and corollary is asymptotic, in the foveal (high-resolution) limit.

For a final illustration, we chose a problem that is still easy enough that versions of the optimal classifier and template matching classifier can be computed, but is no longer entirely artificial. Starting with an ASCII (e-book) version of Ernest Hemingway's novel "For Whom the Bell Tools," we built an image of every page by choosing a resolution (pixel dimensions per page) and creating a JPEG image. The first page, at an intermediate resolution, can be seen on the left-hand side of Figure 7.5. There is no noise, per se, but the moderate resolution and random positioning of characters relative to pixels creates significant degradation. The task was to search the manuscript for specific words, "at" and "the" in the experiments reported in the figure.

For each character in a word we built a random model by assuming (wrongly) that the pixels in the support are iid, with different distributions for the two conditions "character present" and "character absent". Every page was partitioned into blocks, within which there could be a character, a symbol, or a blank. For each letter in the word and each of the two conditions, "present" or absent", the manuscript was used to build two empirical distributions for the pixels in the character's support. These empirical distributions were used for the data model. Notice that typically other characters would be present when a given character was absent – the iid assumption is crude. Referring to Figure 7.5, then, the "optimal decision rule" isn't really optimal since the data model is merely an approximation.

These approximations do not seem to have affected the relative performances, as compared to the L example in which the model was exact. ROC performance of parts testing with saliency-based ordering is indistinguishable from the (approximate) optimal, and substantially better than template matching, for detecting "at" and "the" (right-hand side of the figure). Obviously, there are many levels of relevant context, including word strings that are more or less usual, sentence structure, the topic of a paragraph or a chapter, and even an author's style and preferred vocabulary. In the next two sections we will discuss generalizations and propose a hierarchical version of sequential parts-based testing.

We end this chapter with several observations about $\mathcal{S}_P(z_S)$, related to computation and interpretation:

1. **Computation.** How much does it cost to use the factorization in equation (7.7)? The first step already requires examining all of the data in the support of object $\mathcal{O}$ (i.e. $z_{S^1}, \ldots, z_{S^N}$), which was avoided in the fixed-order sequential scheme. On the other hand, once these N conditional probabilities have been computed the remaining tests come down to computing the conditional probability of one part of $\mathcal{O}$ given the presence of a set of other parts of $\mathcal{O}$, as we shall see. This is the contextual term, which involves a computation on the prior distribution (the distribution on latent variables) but does not involve the data or the data model. In a Markov system it is often of little cost (e.g. in a Bayes net or in the "Markov Backbone" that we use in [41]), but in any case some version of this computation, either in closed form or by approximation, is unavoidable in any formulation of contextual reasoning.

   To make the connection between the calculation of

   $$P(X_{k_i} = 1 | X_{k_1} = 1, \ldots, X_{k_{i-1}} = 1, Z_{S^{k_i}} = z_{S^{k_i}})$$

   and the calculation of

   $$P(X_{k_i} = 1 | Z_{S^{k_i}} = z_{S^{k_i}})$$

   define

   $$M(q,l) \doteq \frac{lq}{lq + (1-q)} \text{ and } H(q,m) = \frac{1-q}{q}\frac{m}{1-m}$$

   If

   $$l = \frac{P(z_S^{k_i} | X^{k_i} = 1)}{P(z_S^{k_i} | X^{k_i} = 1)}$$

   and if $q_1 = P(X_{k_i} = 1)$ and $q_i = P(X_{k_i} = 1 | X_{k_1} = 1, \ldots, X_{k_{i-1}} = 1)$, then

   $$P(X_{k_i} = 1 | Z_{S^{k_i}} = z_{S^{k_i}}) = M(q_1, l)$$

and

$$P(X_{k_i}=1|X_{k_1}=1,\ldots,X_{k_{i-1}}=1,Z_{S^{k_i}})=z_{S^{k_i}}=M(q_i,l)$$

(The last expression is a consequence of (7.5).) Since $l=H(q_1,M(q_1,l))$, the computation of $P(X_{k_i}=1|X_{k_1}=1,\ldots,X_{k_{i-1}}=1,Z_{S^{k_i}}=z_{S^{k_i}})$ comes down to computing $P(X_{k_i}=1|X_{k_1}=1,\ldots,X_{k_{i-1}}=1)$.

The key observation then is that $P(X_{k_i}=1|X_{k_1}=1,\ldots,X_{k_{i-1}}=1,Z_{S^{k_i}}=z_{S^{k_i}})$ is really the same computation as $P(X_{k_i}=1|Z_{S^{k_i}}=z_{S^{k_i}})$ (already computed), but with the probability $P(X_{k_i}=1)$ replaced by $P(X_{k_i}=1|X_{k_1}=1,\ldots,X_{k_{i-1}}=1)$. Typically, $P(X_{k_i}=1|X_{k_1}=1,\ldots,X_{k_{i-1}}=1)>P(X_{k_i}=1)$, by virtue of the accumulating evidence for the object $\mathcal{O}$, and hence the threshold for object k_i is effectively reduced if it comes late in the testing.

2. This condition for a contextual effect, $P(X_{k_i}=1|X_{k_1}=1,\ldots,X_{k_{i-1}}=1)>P(X_{k_i}=1)$, is reminiscent of other discussions of learning in hierarchies. Iterating the expression, and dropping the cumbersome ordering (which is irrelevant to the interpretation of the inequality) we arrive at the condition

$$\frac{P(X_1=1,X_2=1,\ldots,X_N=1)}{\prod_{k=1}^{N}P(X_k=1)}>1 \tag{7.8}$$

which implies an analogous expression for any subset of the parts of $\mathcal{O}$. The ratio on the left-hand side of (7.8) is a measure of departure from independence, in the direction of a strong positive contextual effect. In fact, as a first cut to developing a learning rule for hierarchical systems, it would not be unreasonable to take the empirical estimate of this ratio as evidence in favor of explicitly representing the composition of these parts, and thereby leading to the "discovery" of the object $\mathcal{O}$.

3. It is instructive to compare the optimal statistic to the parts-based statistic. Unlike the part-based strategy, where each ordering of the visitation to parts defines a different statistic, the statistic defining the optimal strategy (i.e. $\mathcal{S}_G$)

is independent of ordering, whether or not the ordering is random. In particular

$$\mathcal{S}_G(z_S) = \prod_{i=1}^{N} P(X_{k_i} = 1 | X_{k_1} = 1, \ldots, X_{k_{i-1}} = 1, Z_{S^{k_i}} = z_{S^{k_i}}, \ldots, Z_{S^{k_N}} = z_{S^{k_N}})$$

which follows by straightforward extension of the reasoning used to derive (7.3). Compared to $\mathcal{S}_P$, as expressed in equation (7.7), the sequential parts-based test is local at every stage of the computation. Contextual influence from the pixel data associated with parts not yet visited is ignored, relying only on the contextual influence from the parts already visited and presumed present. It is not hard to see, then, that the re-ordering of the part-visitation schedule according to saliency can have a substantial impact on performance, consistent with the experiments reported in this chapter.

Appendix

Proof of theorem: Generically, for any statistic $\mathcal{S}(z_S)$

$$\mathcal{A}_{\mathcal{S}} = Prob\{\mathcal{S}(Z_S) < \mathcal{S}(\tilde{Z}_S) \mid Z_S \sim P_1, \tilde{Z}_S \sim P_0\}$$

where $P_1(z_S) = P(z_S \mid \{X_1 = 1\} \cap \{X_2 = 1\})$ and $P_0(z_S) = P(z_S \mid \{\{X_1 = 1\} \cap \{X_2 = 1\}\}^c)$. Let

$$\tilde{\epsilon}_0 = P(X_1 = 0, X_2 = 0 \mid \{\{X_1 = 1\} \cap \{X_2 = 1\}\}^c),$$

$$\tilde{\epsilon}_1 = P(X_1 = 1, X_2 = 0 \mid \{\{X_1 = 1\} \cap \{X_2 = 1\}\}^c)$$

and

$$\tilde{\epsilon}_2 = P(X_1 = 0, X_2 = 1 \mid \{\{X_1 = 1\} \cap \{X_2 = 1\}\}^c).$$

We have

$$P_0(z_S) = \tilde{\epsilon}_0 P(z_S \mid X_1 = 0, X_2 = 0) + \tilde{\epsilon}_1 P(z_S \mid X_1 = 1, X_2 = 0) + \tilde{\epsilon}_2 P(z_S \mid X_1 = 0, X_2 = 1)$$

W.L.O.G, let $Z_{S^1} = [x_1, x_2, \ldots, x_n] = x_1^n$, $Z_{S^2} = [y_1, y_2, \ldots, y_n] = y_1^n$, $\tilde{Z}_{S^1} = [\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_n] = \tilde{x}_1^n$ and $\tilde{Z}_{S^2} = [\tilde{y}_1, \tilde{y}_2, \ldots, \tilde{y}_n] = \tilde{y}_1^n$. Let $\epsilon_0 = P(X_1 = 0, X_2 = 0)$, $\epsilon_1 = P(X_1 = 1, X_2 = 0)$, $\epsilon_2 = P(X_1 = 0, X_2 = 1)$. Then, we will prove the theorem by the following Lemma.

Lemma. Let random variables

$$v_i = log\Big(\frac{p_0(y_i)p_1(\tilde{y}_i)}{p_1(y_i)p_0(\tilde{y}_i)}\Big),\ w_i = log\Big(\frac{p_0(x_i)p_1(\tilde{x}_i)}{p_1(x_i)p_0(\tilde{x}_i)}\Big)$$

for $i = 1, ..., n$, where $y_1^n \sim p_1$, $\tilde{y}_1^n \sim p_0$, $x_1^n \sim p_1$ and $\tilde{x}_1^n \sim p_1$. Then

$$A_{S_G} \geq 0.5(\tilde{\epsilon}_1 + \tilde{\epsilon}_2) Pro\Big(\frac{1}{n}\sum_{i=1}^{n}(v_i - \bar{v}) \geq -\bar{v}\Big), \tag{7.9}$$

$$A_{S_P} \leq (2\tilde{\epsilon}_0 + 3\tilde{\epsilon}_1 + 3\tilde{\epsilon}_2) Pro\Big(\frac{1}{n}\sum_{i=1}^{n}(v_i - \bar{v}) \geq -\bar{v} + \frac{1}{n}log(c)\Big), \tag{7.10}$$

and

$$A_{S_T} \geq \tilde{\epsilon}_1 Pro\Big(\frac{1}{n}\sum_{i=1}^{n}(v_i - \bar{v} + w_i) \geq -\bar{v}\Big), \tag{7.11}$$

where $\bar{v} = E(v_i) < 0$ by Jensen's inequality and $c \leq 1$ is a constant.

Now, we will use the deviation theorem in Bahadur [9] to prove it. Let us first review some notations in the page 1 of the paper. For estimating

$$Pro\Big(\frac{1}{n}\sum_{i=1}^{n}(v_i - \bar{v}) \geq -\bar{v}\Big),$$

let $\varphi(t) = Ee^{t(v_1-\bar{v})}$ and $\psi(t) = e^{-(-\bar{v})t}\varphi(t)$. Since $\varphi(t) < \infty$ for all t and $Pro(v_1 - \bar{v} > -\bar{v}) > 0$, there exists a positive $\tau < \infty$ such that

$$\psi(\tau) = \inf_{t\in R}\psi(t) \equiv \rho.$$

By theorem 1 of Bahadur's paper,

$$Pro\Big(\frac{1}{n}\sum_{i=1}^{n}(v_i - \bar{v}) \geq -\bar{v}\Big) = \frac{\rho^n}{\sqrt{n}}O(1). \tag{7.12}$$

Similarly, for estimating

$$Pro\Big(\frac{1}{n}\sum_{i=1}^{n}(v_i - \bar{v}) \geq -\bar{v} + \frac{1}{n}log(c)\Big),$$

let $\psi_n(t) = e^{-(-\bar{v}+\frac{1}{n}log(c))t}\varphi(t) = e^{-\frac{1}{n}log(c)t}\psi(t)$. Then, we have

$$\begin{aligned}&Pro\Big(\frac{1}{n}\sum_{i=1}^{n}(v_i - \bar{v}) \geq -\bar{v} + \frac{1}{n}log(c)\Big)\\ =&\frac{(\inf_{t\in R}\psi_n(t))^n}{\sqrt{n}}O(1) \leq \frac{\psi_n(\tau)^n}{\sqrt{n}}O(1) = \frac{e^{-log(c)\tau}\psi(\tau)^n}{\sqrt{n}}O(1) = \frac{\rho^n}{\sqrt{n}}O(1).\end{aligned} \tag{7.13}$$

Therefore, by equation (7.12) and (7.13) and the lemma, we obtain that $\frac{A_{S_P}}{A_{S_G}}$ is bounded.

Next, let us consider

$$Pro\Big(\frac{1}{n}\sum_{i=1}^{n}(v_i - \bar{v} + w_i) \geq -\bar{v}\Big).$$

Let

$$\tilde{\varphi}(t) = Ee^{t(v_1-\bar{v}+w_1)} = \varphi(t)Ee^{tw_1},$$

and let $\tilde{\psi}(t) = e^{-(-\bar{v})t}\tilde{\varphi}(t) = \psi(t)Ee^{tw_1}$. Since $\tilde{\varphi}(t) < \infty$ for all t and $Pro(v_1 - \bar{v} +$

$w_1 > -\bar{v}) > 0$, there exists a positive $\tilde{\tau} < \infty$ such that

$$\psi(\tilde{\tau}) = \inf_{t \in R} \tilde{\psi}(t).$$

Thus, we have

$$\begin{aligned} & Pro\Big(\frac{1}{n}\sum_{i=1}^{n}(v_i - \bar{v} + w_i) \geq -\bar{v}\Big) \\ = & \frac{\tilde{\psi}(\tilde{\tau})^n}{\sqrt{n}} O(1) = \frac{\psi(\tilde{\tau})^n (Ee^{\tilde{\tau} w_1})^n}{\sqrt{n}} O(1) \geq \frac{\rho^n (Ee^{\tilde{\tau} w_1})^n}{\sqrt{n}} O(1). \end{aligned} \tag{7.14}$$

Now, since $\tilde{\tau} > 0$ and $Ew_1 = 0$, $Ee^{\tilde{\tau} w_1} > 1$ by Jensen's inequality. Therefore, by comparing (7.14) and (7.13), and by the lemma, we obtain that $\frac{A_{S_T}}{A_{S_P}} \to \infty$ exponentially fast. The proof is completed.

Proof of Lemma:

$$\begin{aligned} A_{S_G} = & Pro((S_G(Z_S))^{-1} \geq (S_G(\tilde{Z}_S))^{-1} \mid Z_S \sim P_1, \tilde{Z}_S \sim P_0) \\ = & Pro\Big(\epsilon_0 \prod_{i=1}^{n} \frac{p_0(x_i)p_0(y_i)}{p_1(x_i)p_1(y_i)} + \epsilon_1 \prod_{i=1}^{n} \frac{p_0(y_i)}{p_1(y_i)} + \epsilon_2 \prod_{i=1}^{n} \frac{p_0(x_i)}{p_1(x_i)} \geq \epsilon_0 \prod_{i=1}^{n} \frac{p_0(\tilde{x}_i)p_0(\tilde{y}_i)}{p_1(\tilde{x}_i)p_1(\tilde{y}_i)} \\ & + \epsilon_1 \prod_{i=1}^{n} \frac{p_0(\tilde{y}_i)}{p_1(\tilde{y}_i)} + \epsilon_2 \prod_{i=1}^{n} \frac{p_0(\tilde{x}_i)}{p_1(\tilde{x}_i)} \;\Big|\; (x_1^n, y_1^n) \sim P_1, (\tilde{x}_1^n, \tilde{y}_1^n) \sim P_0 \Big) \\ \geq & (\tilde{\epsilon}_1 + \tilde{\epsilon}_2) Pro\Big(\prod_{i=1}^{n} \frac{p_0(y_i)}{p_1(y_i)} \geq \prod_{i=1}^{n} \frac{p_0(\tilde{y}_i)}{p_1(\tilde{y}_i)} \;\Big|\; y_1^n \sim p_1, \tilde{y}_1^n \sim p_0 \Big) \cdot \\ & Pro\Big(\prod_{i=1}^{n} \frac{p_0(x_i)}{p_1(x_i)} \geq \prod_{i=1}^{n} \frac{p_0(\tilde{x}_i)}{p_1(\tilde{x}_i)} \;\Big|\; x_1^n \sim p_1, \tilde{x}_1^n \sim p_1 \Big) \end{aligned}$$

The last inequality is because the set

$$\Big\{ \prod_{i=1}^{n} \frac{p_0(y_i)}{p_1(y_i)} \geq \prod_{i=1}^{n} \frac{p_0(\tilde{y}_i)}{p_1(\tilde{y}_i)}, \prod_{i=1}^{n} \frac{p_0(x_i)}{p_1(x_i)} \geq \prod_{i=1}^{n} \frac{p_0(\tilde{x}_i)}{p_1(\tilde{x}_i)} \Big\}$$

is contained in the set

$$\begin{aligned} & \Big\{ \epsilon_0 \prod_{i=1}^{n} \frac{p_0(x_i)p_0(y_i)}{p_1(x_i)p_1(y_i)} + \epsilon_1 \prod_{i=1}^{n} \frac{p_0(y_i)}{p_1(y_i)} + \epsilon_2 \prod_{i=1}^{n} \frac{p_0(x_i)}{p_1(x_i)} \geq \epsilon_0 \prod_{i=1}^{n} \frac{p_0(\tilde{x}_i)p_0(\tilde{y}_i)}{p_1(\tilde{x}_i)p_1(\tilde{y}_i)} \\ & + \epsilon_1 \prod_{i=1}^{n} \frac{p_0(\tilde{y}_i)}{p_1(\tilde{y}_i)} + \epsilon_2 \prod_{i=1}^{n} \frac{p_0(\tilde{x}_i)}{p_1(\tilde{x}_i)} \Big\}. \end{aligned}$$

Now, since

$$Pro\Big(\prod_{i=1}^{n}\frac{p_0(x_i)}{p_1(x_i)} \geq \prod_{i=1}^{n}\frac{p_0(\tilde{x}_i)}{p_1(\tilde{x}_i)} \;\Big|\; x_1^n \sim p_1, \tilde{x}_1^n \sim p_1\Big) = 0.5,$$

we have

$$\begin{aligned} A_{S_G} \geq & 0.5(\tilde{\epsilon}_1 + \tilde{\epsilon}_2)Pro\Big(\prod_{i=1}^{n}\frac{p_0(y_i)}{p_1(y_i)} \geq \prod_{i=1}^{n}\frac{p_0(\tilde{y}_i)}{p_1(\tilde{y}_i)} \;\Big|\; y_1^n \sim p_1, \tilde{y}_1^n \sim p_0\Big). \\ = & 0.5(\tilde{\epsilon}_1 + \tilde{\epsilon}_2)Pro\Big(\frac{1}{n}\sum_{i=1}^{n}(v_i - \bar{v}) \geq -\bar{v}\Big). \end{aligned}$$

Thus, we get equation (7.9).

Next, Let

$$G_1(x_1^n) = \frac{\epsilon_1}{\epsilon_1 + (1-\epsilon_1)\prod_{i=1}^{n}\frac{p_0(x_i)}{p_1(x_i)}}$$

and

$$G_2(y_1^n) = \frac{\epsilon_{2|1}}{\epsilon_{2|1} + (1-\epsilon_{2|1})\prod_{i=1}^{n}\frac{p_0(y_i)}{p_1(y_i)}}.$$

where $\epsilon_{2|1} = P(X_2 = 1 \mid X_1 = 1)$. Then,

$$\begin{aligned} A_{S_P} = & Pro(S_P(Z_S) \leq S_P(\tilde{Z}_S) \mid Z_S \sim P_1, \tilde{Z}_S \sim P_0) \\ = & Pro(G_1(x_1^n)G_2(y_1^n) \leq G_1(\tilde{x}_1^n)G_2(\tilde{y}_1^n) \mid (x_1^n, y_1^n) \sim P_1, (\tilde{x}_1^n, \tilde{y}_1^n) \sim P_0) \\ = & \tilde{\epsilon}_0 I_0 + \tilde{\epsilon}_1 I_1 + \tilde{\epsilon}_2 I_2 \end{aligned}$$

where

$$I_0 = Pro(G_1(x_1^n)G_2(y_1^n) \leq G_1(\tilde{x}_1^n)G_2(\tilde{y}_1^n) \mid x_1^n \sim p_1, y_1^n \sim p_1, \tilde{x}_1^n \sim p_0, \tilde{y}_1^n \sim p_0),$$

$$I_1 = Pro(G_1(x_1^n)G_2(y_1^n) \leq G_1(\tilde{x}_1^n)G_2(\tilde{y}_1^n) \mid x_1^n \sim p_1, y_1^n \sim p_1, \tilde{x}_1^n \sim p_1, \tilde{y}_1^n \sim p_0)$$

and

$$I_2 = Pro(G_1(x_1^n)G_2(y_1^n) \leq G_1(\tilde{x}_1^n)G_2(\tilde{y}_1^n) \mid x_1^n \sim p_1, y_1^n \sim p_1, \tilde{x}_1^n \sim p_0, \tilde{y}_1^n \sim p_1).$$

Now

$$\begin{aligned}I_0 \leq & Pro(G_1(x_1^n) \leq G_1(\tilde{x}_1^n) \mid x_1^n \sim p_1, \tilde{x}_1^n \sim p_0) \\ & \qquad\qquad + Pro(G_2(y_1^n) \leq G_2(\tilde{y}_1^n) \mid y_1^n \sim p_1, \tilde{y}_1^n \sim p_0) \\ = & 2Pro\Big(\prod_{i=1}^{n} \frac{p_0(y_i)p_1(\tilde{y}_i)}{p_1(y_i)p_0(\tilde{y}_i)} \geq 1 \Bigm| y_1^n \sim p_1, \tilde{y}_1^n \sim p_0\Big)\end{aligned}$$

Next,

$$\begin{aligned}I_1 \leq & Pro(G_1(x_1^n)G_2(y_1^n) \leq G_2(\tilde{y}_1^n) \mid x_1^n \sim p_1, y_1^n \sim p_1, \tilde{y}_1^n \sim p_0) \\ = & Pro\Big(\epsilon_1(1-\epsilon_{2|1}) \prod_{i=1}^{n} \frac{p_0(\tilde{y}_i)}{p_1(\tilde{y}_i)} \leq \epsilon_{2|1}(1-\epsilon_1) \prod_{i=1}^{n} \frac{p_0(x_i)}{p_1(x_i)} + \epsilon_1(1-\epsilon_{2|1}) \prod_{i=1}^{n} \frac{p_0(y_i)}{p_1(y_i)} \\ & + (1-\epsilon_1)(1-\epsilon_{2|1}) \prod_{i=1}^{n} \frac{p_0(x_i)}{p_1(x_i)} \prod_{i=1}^{n} \frac{p_0(y_i)}{p_1(y_i)} \Bigm| x_1^n \sim p_1, y_1^n \sim p_1, \tilde{y}_1^n \sim p_0\Big) \\ \leq & J_1 + J_2 + J_3\end{aligned}$$

where

$$J_1 = Pro\Big(\epsilon_1(1-\epsilon_{2|1}) \prod_{i=1}^{n} \frac{p_0(\tilde{y}_i)}{p_1(\tilde{y}_i)} \leq 3\epsilon_{2|1}(1-\epsilon_1) \prod_{i=1}^{n} \frac{p_0(x_i)}{p_1(x_i)} \Bigm| x_1^n \sim p_1, \tilde{y}_1^n \sim p_0\Big),$$

$$J_2 = Pro\Big(\epsilon_1(1-\epsilon_{2|1}) \prod_{i=1}^{n} \frac{p_0(\tilde{y}_i)}{p_1(\tilde{y}_i)} \leq 3\epsilon_1(1-\epsilon_{2|1}) \prod_{i=1}^{n} \frac{p_0(y_i)}{p_1(y_i)} \Bigm| y_1^n \sim p_1, \tilde{y}_1^n \sim p_0\Big)$$

and

$$\begin{aligned}J_3 = & Pro\Big(\epsilon_1(1-\epsilon_{2|1}) \prod_{i=1}^{n} \frac{p_0(\tilde{y}_i)}{p_1(\tilde{y}_i)} \\ & \leq 3(1-\epsilon_1)(1-\epsilon_{2|1}) \prod_{i=1}^{n} \frac{p_0(x_i)}{p_1(x_i)} \prod_{i=1}^{n} \frac{p_0(y_i)}{p_1(y_i)}) \Bigm| x_1^n \sim p_1, y_1^n \sim p_1, \tilde{y}_1^n \sim p_0\Big).\end{aligned}$$

As n large enough, we have

$$max(J_1, J_2, J_3) \leq Pro\Big(\prod_{i=1}^{n} \frac{p_0(y_i)p_1(\tilde{y}_i)}{p_1(y_i)p_0(\tilde{y}_i)} \geq c_1 \Bigm| y_1^n \sim p_1, \tilde{y}_1^n \sim p_0\Big),$$

where

$$c_1 = min\Big(\frac{1}{3}, \frac{\epsilon_1(1-\epsilon_{2|1})}{3\epsilon_{2|1}(1-\epsilon_1)}\Big)$$

is a constant. Therefore,

$$I_1 \leq 3Pro\Big(\prod_{i=1}^{n} \frac{p_0(y_i)p_1(\tilde{y}_i)}{p_1(y_i)p_0(\tilde{y}_i)} \geq c_1 \;\Big|\; y_1^n \sim p_1, \tilde{y}_1^n \sim p_0\Big).$$

Similarly,

$$I_2 \leq 3Pro\Big(\prod_{i=1}^{n} \frac{p_0(y_i)p_1(\tilde{y}_i)}{p_1(y_i)p_0(\tilde{y}_i)} \geq c_2 \;\Big|\; y_1^n \sim p_1, \tilde{y}_1^n \sim p_0\Big),$$

where c_2 is a constant. Then we have

$$\begin{aligned} A_{S_P} &\leq (2\tilde{\epsilon}_0 + 3\tilde{\epsilon}_1 + 3\tilde{\epsilon}_2)Pro\Big(\prod_{i=1}^{n} \frac{p_0(y_i)p_1(\tilde{y}_i)}{p_1(y_i)p_0(\tilde{y}_i)} \geq c \;\Big|\; y_1^n \sim p_1, \tilde{y}_1^n \sim p_0\Big) \\ &= (2\tilde{\epsilon}_0 + 3\tilde{\epsilon}_1 + 3\tilde{\epsilon}_2)Pro\Big(\frac{1}{n}\sum_{i=1}^{n}(v_i - \bar{v}) \geq -\bar{v} + \frac{1}{n}log(c)\Big), \end{aligned}$$

where $c = min(c_1, c_2, 1) \leq 1$. Thus, we get equation (7.10).

Finally,

$$\begin{aligned} A_{S_T} =& Pro\Big(\prod_{i=1}^{n} \frac{p_0(x_i)}{p_1(x_i)}\frac{p_0(y_i)}{p_1(y_i)} \geq \prod_{i=1}^{n} \frac{p_0(\tilde{x}_i)}{p_1(\tilde{x}_i)}\frac{p_0(\tilde{y}_i)}{p_1(\tilde{y}_i)} \;\Big|\; (x_1^n, y_1^n) \sim P_1, (\tilde{x}_1^n, \tilde{y}_1^n) \sim P_0\Big) \\ \geq& \tilde{\epsilon}_1 Pro\Big(\prod_{i=1}^{n} \frac{p_0(y_i)}{p_1(y_i)}\frac{p_1(\tilde{y}_i)}{p_0(\tilde{y}_i)}\frac{p_0(x_i)}{p_1(x_i)}\frac{p_1(\tilde{x}_i)}{p_0(\tilde{x}_i)} \geq 1 \;\Big|\; x_1^n \sim p_1, y_1^n \sim P_1, \tilde{x}_1^n \sim p_1, \tilde{y}_1^n \sim p_0\Big). \\ =& \tilde{\epsilon}_1 Pro\Big(\frac{1}{n}\sum_{i=1}^{n}(v_i - \bar{v} + w_i) \geq -\bar{v}\Big) \end{aligned}$$

Therefore, we get equation (7.11). The proof is completed.

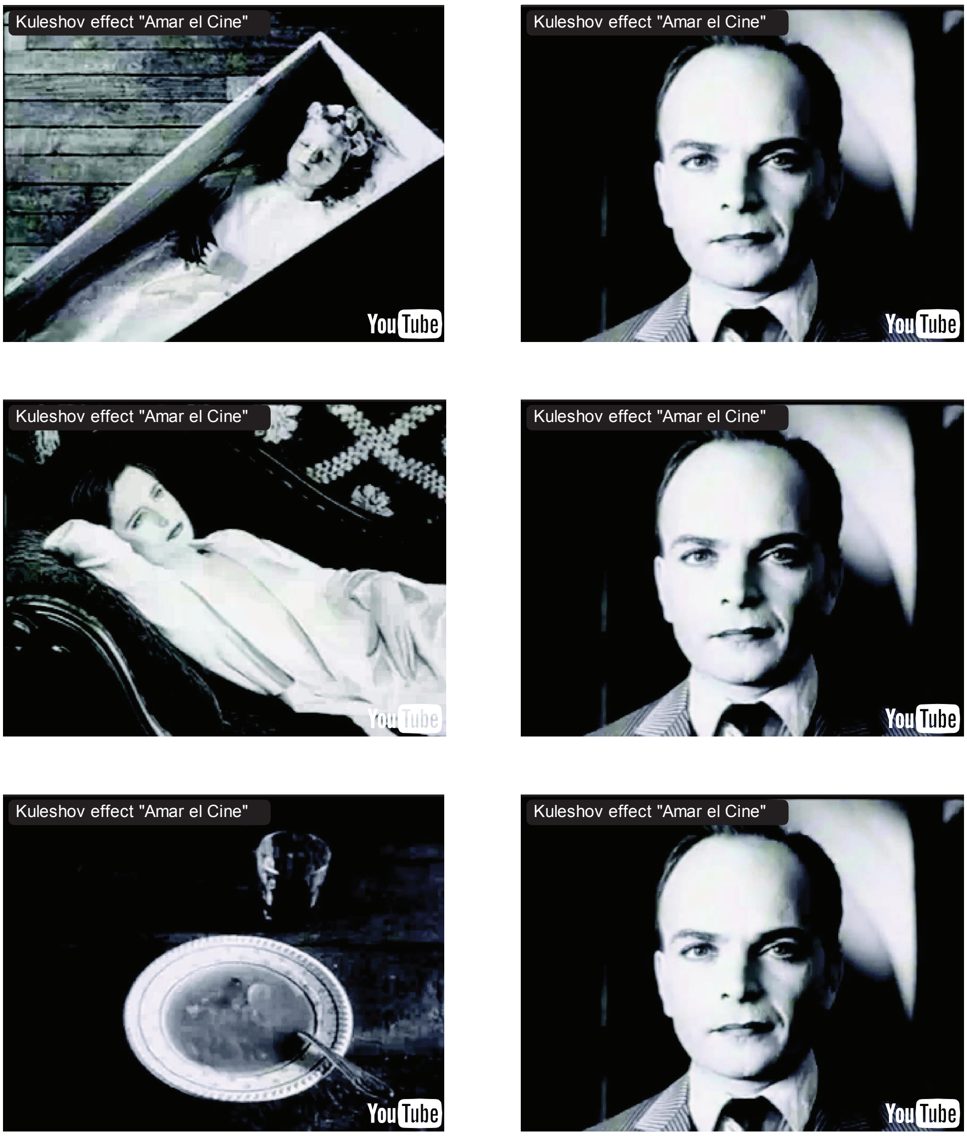

Kuleshov effect "Amar el Cine"
Kuleshov effect "Amar el Cine"
Kuleshov effect "Amar el Cine"
Kuleshov effect "Amar el Cine"
Kuleshov effect "Amar el Cine"
Kuleshov effect "Amar el Cine"

Figure 7.1: **The Kuleshov Effect. Top Row.** Frames from a sequence with a dead child followed by a shot of Ivan Mozzhukhin's face. **Middle Row.** Frames from a sequence with an actress in a seductive pose, followed again the same shot of Mozzhukhin's face. **Bottom Row.** Frames from a sequence with a bowl of soup, followed by the same shot of Mozzhukhin's face. Audiences viewing the clips ascribe different emotions to same expression, sadness, hunger, or lust, depending on the context.

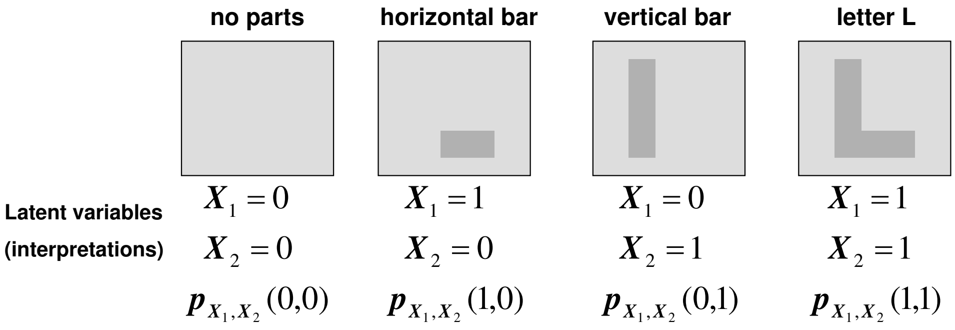


Figure 7.2: **Minimal Compositional World.** There are only two parts, horizontal bar and vertical bar, and one object, the letter L. The presence of both parts is always interpreted as the letter L.

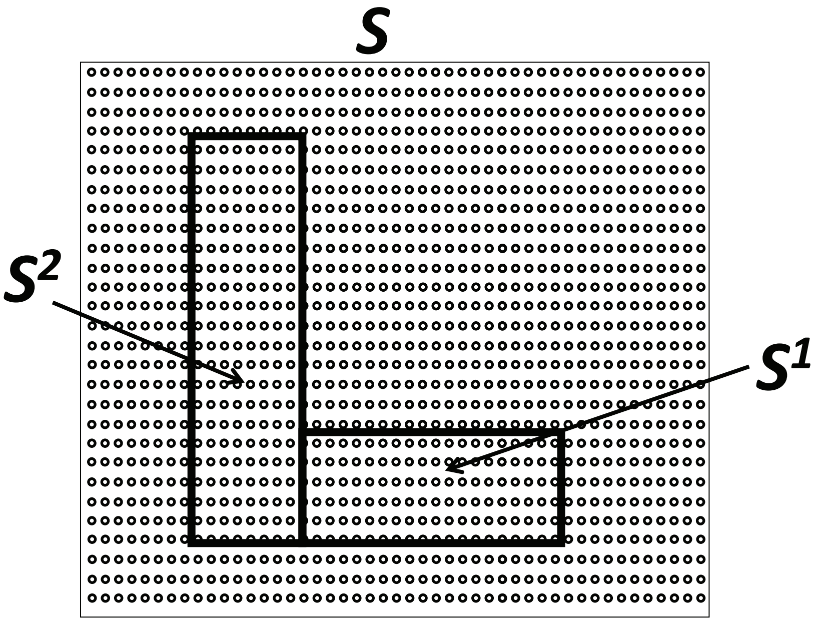


Figure 7.3: **Pixel Lattice.** S is the set of pixels. $S^1 \subseteq S$ (the "support of part 1") and $S^2 \subseteq S$ (the "support of part 2") are the subsets of pixels at which horizontal and vertical bars appear, respectively.

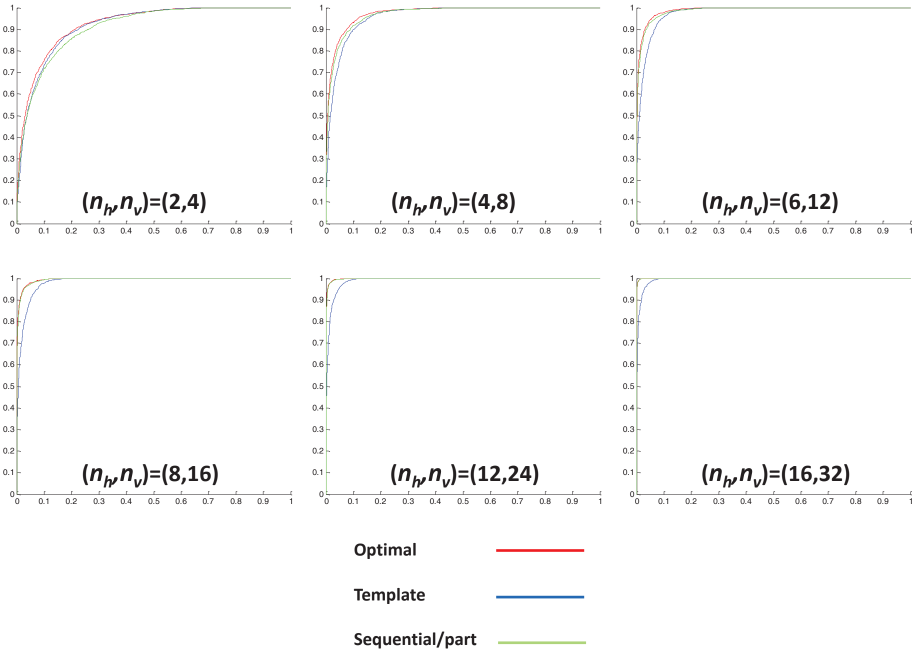


Figure 7.4: **Illustration of Comparison Theorem.** Each panel contains three ROC curves, for the optimal ($\mathcal{S}_G(z_S) > t$, in red), for template matching ($\mathcal{S}_T(z_S) > t$, in blue) and for sequential testing of parts ($\mathcal{S}_P(z_S) > t$, in green). Resolution is progressively increased, left-to-right and top-to-bottom ("foveal limit"). In each panel the numbers of pixels on the horizontal and vertical bars (the "supports") are indicated by (n_h, n_v) (so $n_h = |S^1|$ and $n_v = |S^2|$). At low resolution, $(n_h, n_v) = (2, 4)$, template matching outperforms parts-based testing. At higher resolutions parts-based testing is better, and nearly optimal. Template matching is slow to converge to perfect performance.

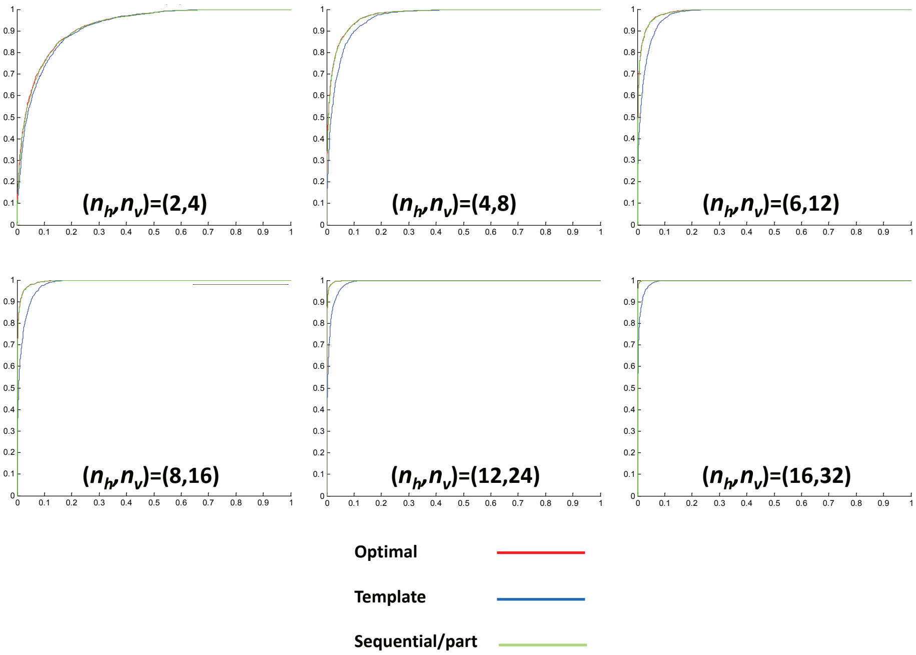
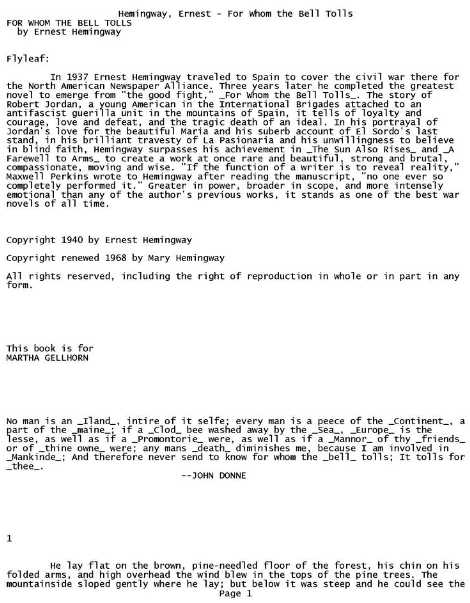


Figure 7.5: **Saliency.** Same as Figure 7.4, except that parts are tested in the order of saliency (i.e. their conditional probabilities given only *local* pixel data) for the sequential algorithm (green curve). Compared to Figure 7.4, parts-based testing is now essentially optimal at all resolutions.



Hemingway, Ernest - For Whom the Bell Tolls

FOR WHOM THE BELL TOLLS
by Ernest Hemingway

Flyleaf:

In 1937 Ernest Hemingway traveled to Spain to cover the civil war there for the North American Newspaper Alliance. Three years later he completed the greatest novel to emerge from "the good fight," _For Whom the Bell Tolls_. The story of Robert Jordan, a young American in the International Brigades attached to an antifascist guerilla unit in the mountains of Spain, it tells of loyalty and courage, love and defeat, and the tragic death of an ideal. In his portrayal of Jordan's love for the beautiful Maria and his suberb account of El Sordo's last stand, in his brilliant travesty of La Pasionaria and his unwillingness to believe in blind faith, Hemingway surpasses his achievement in _The Sun Also Rises_ and _A Farewell to Arms_ to create a work at once rare and beautiful, strong and brutal, compassionate, moving and wise. "If the function of a writer is to reveal reality," Maxwell Perkins wrote to Hemingway after reading the manuscript, "no one ever so completely performed it." Greater in power, broader in scope, and more intensely emotional than any of the author's previous works, it stands as one of the best war novels of all time.

Copyright 1940 by Ernest Hemingway

Copyright renewed 1968 by Mary Hemingway

All rights reserved, including the right of reproduction in whole or in part in any form.

This book is for
MARTHA GELLHORN

No man is an _Iland_, intire of it selfe; every man is a peece of the _Continent_, a part of the _maine_; if a _Clod_ bee washed away by the _Sea_, _Europe_ is the lesse, as well as if a _Promontorie_ were, as well as if a _Mannor_ of thy _friends_ or of _thine owne_ were; any mans _death_ diminishes me, because I am involved in _Mankinde_; And therefore never send to know for whom the _bell_ tolls; It tolls for _thee_.

--JOHN DONNE

1

He lay flat on the brown, pine-needled floor of the forest, his chin on his folded arms, and high overhead the wind blew in the tops of the pine trees. The mountainside sloped gently where he lay; but below it was steep and he could see the

Page 1

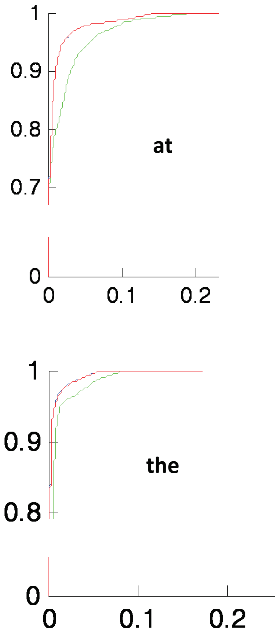


Figure 7.6: **Word Search.** On the left is an image of the first page of Ernest Hemingway's novel, "For Whom the Bell Tolls." The ASCII e-book was converted to a relatively low-resolution JPEG image. The image was used to search for all instances of the words "at" and "the" in the novel. A simple model was estimated and the ROC performance of each of the three decision algorithms (optimal, template, and salient-based sequential testing) was computed. Sequential testing of parts was indistinguishable from the optimal test, and substantially better than template matching for both word searches. (Color scheme is the same one used in Figures 7.4 and 7.5.)

Chapter 8

Experiments with a composition system

Digital image techniques have increased tremendously since 1970. The rapid growth of digital medical images, such as standard radiographs (X-ray), computed tomography(CT) images and magnetic resonance(MR) images, has generated a critical need for automatic classification engines and retrieval systems to help the radiologists in prioritization and in the diagnosis of findings.

Many researchers have worked on medical image classification and retrieval, and they have provided numerous approaches for this purpose. In Lu [48], they used texture analysis in the X-ray image classification. In Mueen [50], they used multi-level image features and the implement Support Vector Machine(SVM) to classify X-ray images. In Shim [59], they proposed a color structure descriptor for color features and they proposed an edge histogram descriptor for texture features. They then applied them to multiclass-SVM and K-nearest images for X-ray image classification and retrieval. In Unay [51], they explored the effect of principal component analysis based feature selection on the X-ray image classification performance. In Avni [8], their methodology is based on local patch representation of the image content and on a bag-of-features approach for defining X-ray image categories, with a kernel based SVM classifier.

The classification rates in most publications are about 90%, but the rates are not uniform for every category of X-ray images. For some categories, they can get close to a 100% detection rate, but for others, they can only reach about 80%. In this section, we will focus on one of the most difficult categories, the X-ray images of ankles. There are many different views of an ankle, for example the Anteroposterior(AP) view, the Mortise view, the Lateral(side) view, the Oblique view and so on(see Figure 8.1). We will focus on the lateral view and build a composition model of the lateral ankle images, following the procedures of Chapter 5 and Chapter 6. Then we can use this probability model to classify the lateral ankle images and other images(either ankle images of different views or non-ankle images) by looking at their probabilities. In this work, we utilize the image data from the ImageCLEFmed 2008 for training and testing. Figure 8.2 shows 36 images of ankles and non-ankles.

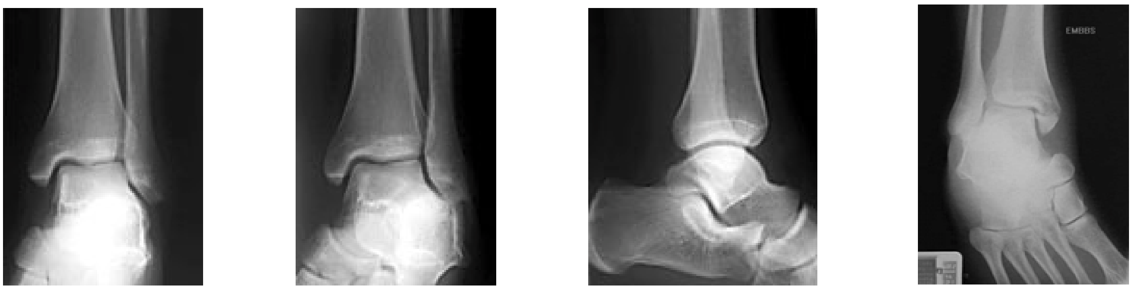


Figure 8.1: Different views of the X-ray images: Anteroposterior(AP) view, Mortise view, Lateral(side) view and Oblique view.

8.1 Image interpretations of non-terminal bricks

To build a composition model of lateral ankle images, we can decompose it into two parts: the prior distribution on image interpretations and the conditional data distribution on the image given its interpretation. To build a prior distribution, we can think of the object as being composed of several parts. These parts have some relation to each other, and each part has its own probability model. For a normal lateral ankle image, we can imagine the ankle is composed of four parts as in the

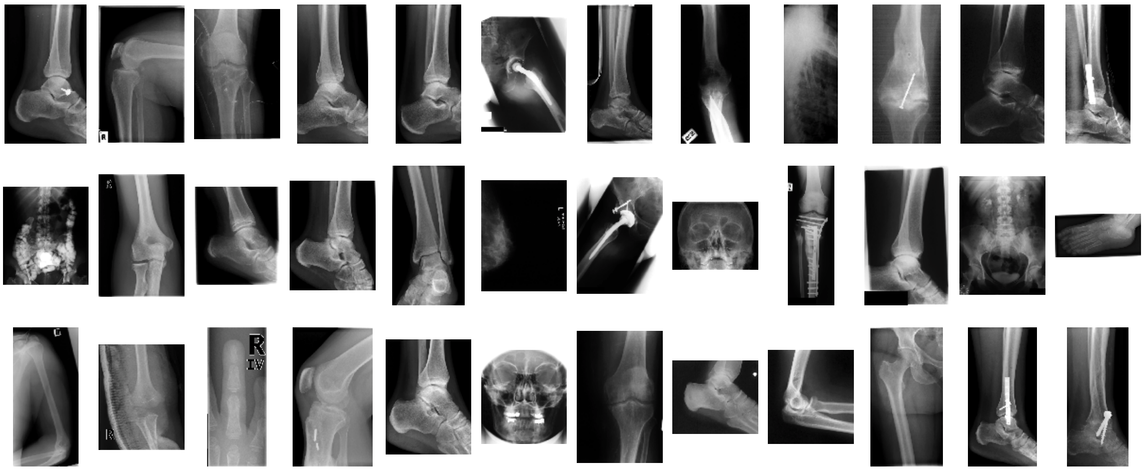


Figure 8.2: Classification of the X-ray images: Pick out the ankle images.

Figure 8.3. Let the middle blue one be part α_1, the top red one be part α_2, the left

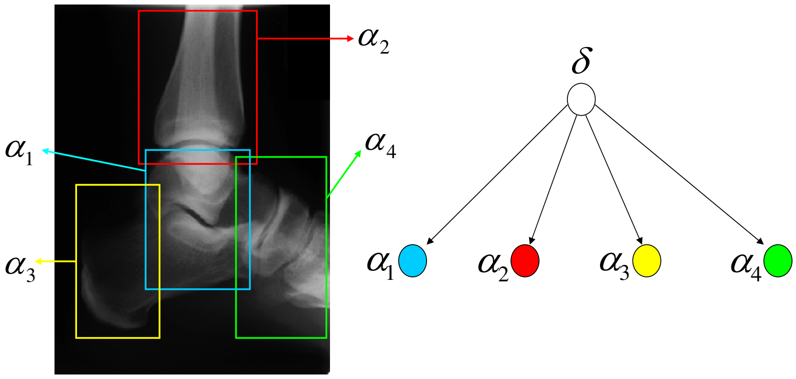


Figure 8.3: an ankle can be composed of four parts.

yellow one be part α_3 and the right green one be part α_4. Next, in order to reduce the time expense of computation, we usually consider the coarse representation for each part as shown in the left panel of Figure 8.4. Of course, we lose some detail of the ankle in the coarse representation, like the joints of the ankle. Thus, we need to work on finer representations for some parts(see the middle panel of Figure 8.4), so that these parts are composed of the low resolution part and of the high resolution

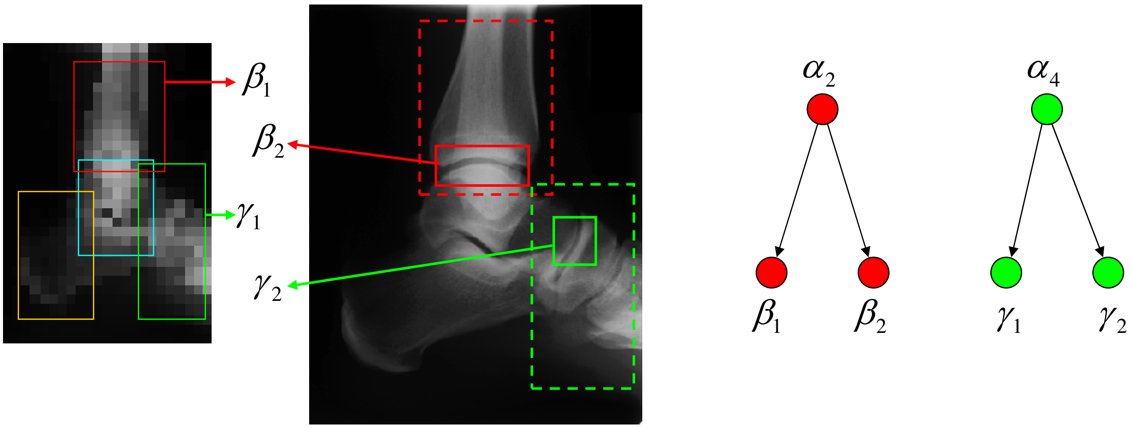


Figure 8.4: the left panel shows the coarse representation for the four parts; the middle panel shows the finer representation for part α_2 and part α_4; the right panel shows that α_2 is composed of the low resolution part β_1 and high resolution part β_2, and α_4 is composed of the low resolution part γ_1 and high resolution part γ_2.

part as in the right panel of Figure 8.4. Then, we can build the foundation of a composition structure as in Figure 8.5. Now we are going to parse the composition structure and set the prior probability on the parsing of the structure.

We first assume that the toes of the images are equally likely to point towards the right and the left, and we assume that the model is invariant for flipping images. Observing the training data, Figure 8.6, we find two kinds of severe occlusion problems: window occlusion(e.g. the fifth image in the first row and the tenth image in the second row) and object occlusion(e.g. the fourth image in the second row). The result is that the ankle classification rate is always lower for most classification methods. To solve the occlusion problem, we let our model allow some occlusions in the procedure of parsing. We start from the interpretations of brick δ and let $n^\delta = 5$ so that $x^\delta \in \{0, 1, 2, 3, 4, 5\}$. The following table is the list of each "on" state and its corresponding children bricks:

1. $x^\delta = 1$: $C_1^\delta = \{\alpha_1, \alpha_2, \alpha_3, \alpha_4\}$,

2. $x^\delta = 2$: $C_2^\delta = \{\alpha_1, \alpha_2, \alpha_3\}$,

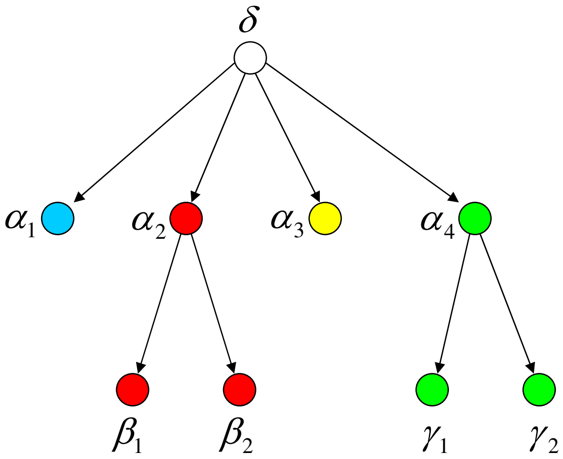


Figure 8.5: the directed graph of the ankle composition structure.

3. $x^\delta = 3$: $C_3^\delta = \{\alpha_1, \alpha_2, \alpha_4\}$,

4. $x^\delta = 4$: $C_4^\delta = \{\alpha_1, \alpha_3, \alpha_4\}$,

5. $x^\delta = 5$: $C_5^\delta = \{\alpha_1, \alpha_2\}$.

For $x^\delta = 2$, we only look at three bricks, $\alpha_1, \alpha_2, \alpha_3$, and the fourth part, α_4 brick is occluded by the window or some objects. Similarly for $x^\delta = 3, 4, 5$ the model we build is flexible enough to allow for some occlusions and other considerations

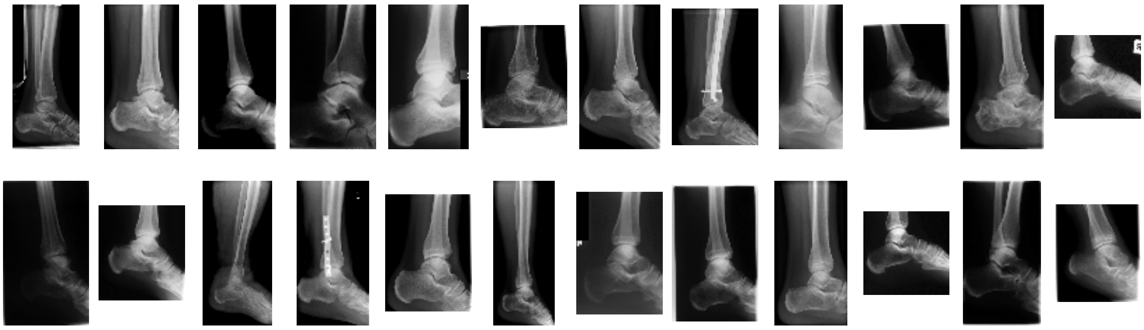


Figure 8.6: The set of the ankle images

about the ankle images. Next, we look at those children bricks. Let α_1 brick and α_3 brick be terminal bricks corresponding to the coarse representations(see the next subsection). Now, let us consider the two non-terminal bricks, α_2 brick and α_4 brick. Let $n^{\alpha_2} = 3$ and $n^{\alpha_4} = 3$. Then, let their children bricks be as follows:

1. $x^{\alpha_2} = 1$: $C_1^{\alpha_2} = \{\beta_1, \beta_2\}$,
2. $x^{\alpha_2} = 2$: $C_2^{\alpha_2} = \{\beta_1\}$,
3. $x^{\alpha_2} = 3$: $C_3^{\alpha_2} = \{\beta_2\}$,
4. $x^{\alpha_4} = 1$: $C_1^{\alpha_4} = \{\gamma_1, \gamma_2\}$,
5. $x^{\alpha_4} = 2$: $C_2^{\alpha_4} = \{\gamma_1\}$,
6. $x^{\alpha_4} = 3$: $C_3^{\alpha_4} = \{\gamma_2\}$.

For $x^{\alpha_2} = 1$, we have both coarse and finer representations for α_2 brick. However, for some ankle images, we can not see the detail(the joint) of the second part, α_2, just like the third ankle image of the second row in Figure 8.6 because of object occlusions or incompleteness of the broken ankle. Thus $x^{\alpha_2} = 2$ only considers the coarse representation, which could roughly interpret α_2, so that we can get some benefits for weak evidence image. On the other hand, the window occlusion may occlude only the top of part α_2 like the second or tenth ankle images of the second row in Figure 8.6, but we still can see the joint of α_2. Thus, $x^{\alpha_2} = 3$ takes care of this case. We can have a similar expansion in the setting of α_4. Finally, the setup of the states can be shown in the Figure 8.7. The discussion of the terminal bricks $\alpha_1, \alpha_3, \beta_1, \beta_2, \gamma_1, \gamma_2$ will be in the next subsection.

8.2 Template learning and image interpretations of terminal bricks

To finish the interpretations of the entire composition model, we need to set up the states of the terminal bricks that connect to the data images. Since for the local

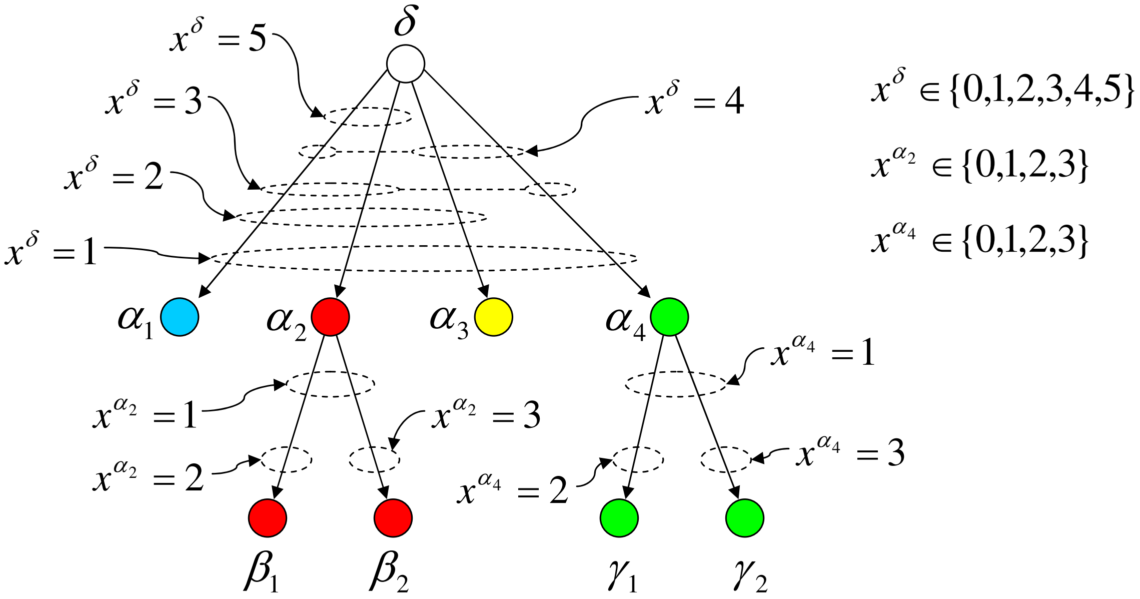


Figure 8.7: the directed graph with prior structure

image of each part there are many configurations including locations, orientations, scales and different prototypical images(templates), each terminal brick will have a huge number of states. In this subsection, we will first learn templates by the methods of Chapter 6, and then we will have a data model given the interpretations.

For template learning, we first resize them to create both low resolution and high resolution images, and we crop out the associated regions from training images. Then we implement our methods to learn the templates. Notice that the methods we are using need a background model or random background image patches, but instead of using smooth background image patches, we randomly choose image patches from a negative data set(non-ankle image set). Figure 8.8 and Figure 8.9 show both low resolution and high resolution templates as well as their corresponding regions in the training images.

Now, we are going to write down the distribution of data given that the non-

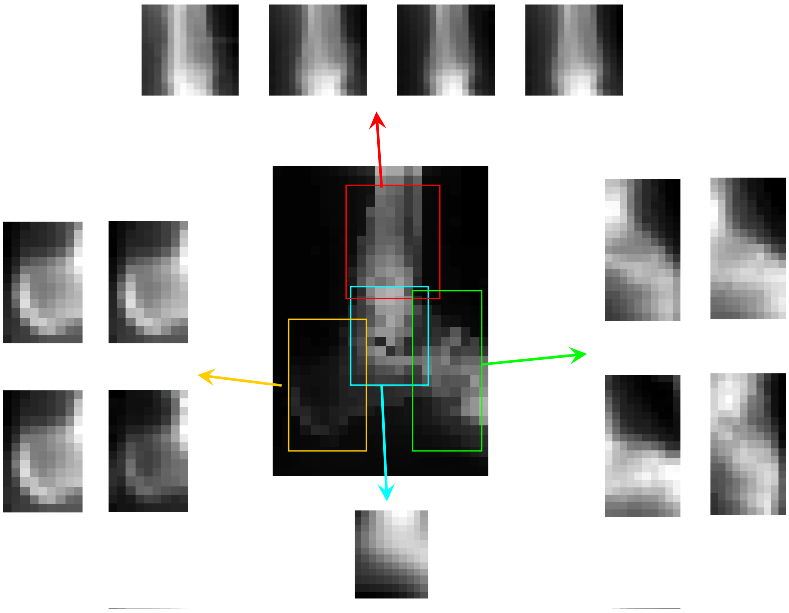


Figure 8.8: Low resolution templates: the top panel shows 4 learned templates with size 14×15; the left panel shows 4 learned templates with size 14×9; the bottom panel shows 1 learned templates with size 12×10; the right panel shows 4 learned templates with size 19×10.

terminal brick is "on." For example,

$$\begin{aligned} P(y|x^{\delta}=2,x^{\alpha_2}=2) = \textstyle\sum_{i=1}^{n^{\alpha_1}}\sum_{j=1}^{n^{\beta_1}}\sum_{k=1}^{n^{\alpha_3}} \quad & P(y|x^{\delta}=2,x^{\alpha_2}=2,x^{\alpha_1}=i,x^{\beta_1}=j,x^{\alpha_3}=k)\cdot \\ & P(x^{\alpha_1}=i,x^{\beta_1}=j,x^{\alpha_3}=k|x^{\delta}=2,x^{\alpha_2}=2), \end{aligned} \tag{8.1}$$

which is the sum of the product of two terms. Let us first consider the first term of the product.

$$\begin{aligned} & P(y|x^{\delta}=2,x^{\alpha_2}=2,x^{\alpha_1}=i,x^{\beta_1}=j,x^{\alpha_3}=k) \\ = \quad & P(y_{R_D}|x^{\delta}=2,x^{\alpha_2}=2,x^{\alpha_1}=i,x^{\beta_1}=j,x^{\alpha_3}=k)\cdot P^0(y_{R_D^c}), \end{aligned}$$

and R_D is the union of the sets of image locations in the support of the associated terminal bricks,

$$R_D = R_i^{\alpha_1}\cup R_j^{\beta_1}\cup R_k^{\alpha_3}.$$

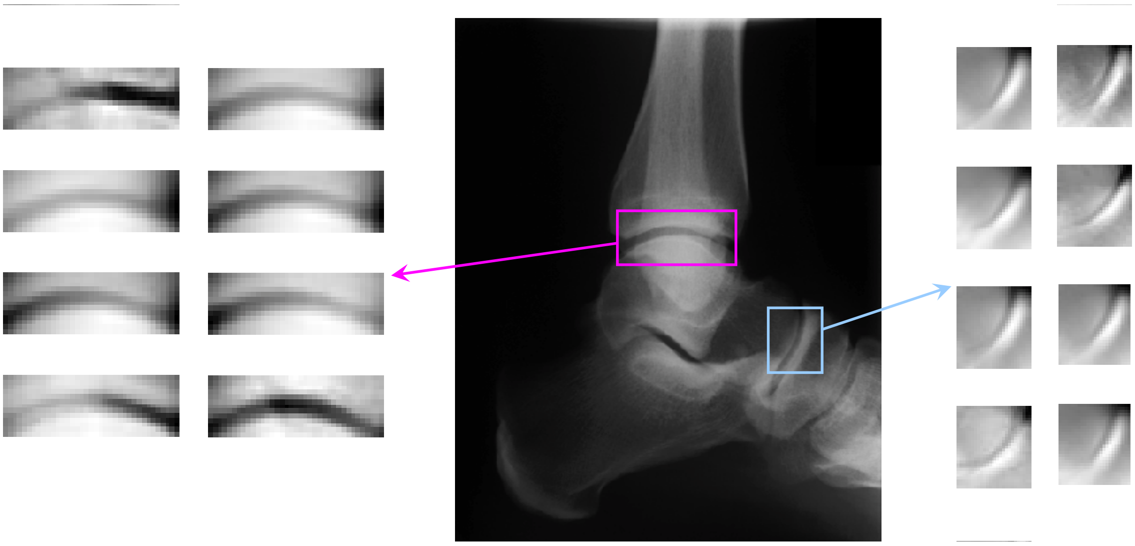


Figure 8.9: High resolution templates: the left panel shows 8 learned templates with size 13×37; the right panel shows 8 learned templates with size 29×26.

Using the method in Section 6.2, we only need to consider the ratio

$$\begin{aligned} & \frac{P(y|x^{\delta}=2, x^{\alpha_2}=2, x^{\alpha_1}=i, x^{\beta_1}=j, x^{\alpha_3}=k)}{P^0(y)} \\ = & \frac{P(y_{R_D}|x^{\delta}=2, x^{\alpha_2}=2, x^{\alpha_1}=i, x^{\beta_1}=j, x^{\alpha_3}=k)}{P^0(y_{R_D})} \\ = & \frac{P_C(cor(T_i^{\alpha_1}, y_{R_i^{\alpha_1}}), cor(T_j^{\beta_1}, y_{R_j^{\beta_1}}), cor(T_k^{\alpha_3}, y_{R_k^{\alpha_3}}))}{P_C^0(cor(T_i^{\alpha_1}, y_{R_i^{\alpha_1}}), cor(T_j^{\beta_1}, y_{R_j^{\beta_1}}), cor(T_k^{\alpha_3}, y_{R_k^{\alpha_3}}))}, \end{aligned}$$

where $T_i^{\alpha_1}$, $T_j^{\beta_1}$ and $T_k^{\alpha_3}$ are the templates associated to the states $x^{\alpha_1}=i$, $x^{\beta_1}=j$ and $x^{\alpha_3}=k$ respectively. Similar to Section 6.5, we assume that the numerator is the product of individual probabilities:

$$\begin{aligned} & P_C(cor(T_i^{\alpha_1}, y_{R_i^{\alpha_1}}), cor(T_j^{\beta_1}, y_{R_j^{\beta_1}}), cor(T_k^{\alpha_3}, y_{R_k^{\alpha_3}})) \\ = \; & P_C(cor(T_i^{\alpha_1}, y_{R_i^{\alpha_1}})) \cdot P_C(cor(T_j^{\beta_1}, y_{R_j^{\beta_1}})) \cdot P_C(cor(T_k^{\alpha_3}, y_{R_k^{\alpha_3}})), \end{aligned}$$

and the denominator can be approximated by a multivariate normal distribution:

$$\begin{aligned} & P_C^0(cor(T_i^{\alpha_1}, y_{R_i^{\alpha_1}}), cor(T_j^{\beta_1}, y_{R_j^{\beta_1}}), cor(T_k^{\alpha_3}, y_{R_k^{\alpha_3}})) \\ \sim \;& N(cor(T_i^{\alpha_1}, y_{R_i^{\alpha_1}}), cor(T_j^{\beta_1}, y_{R_j^{\beta_1}}), cor(T_k^{\alpha_3}, y_{R_k^{\alpha_3}}); (0,0,0), \Sigma_{i,j,k}), \end{aligned}$$

where the covariance matrix is

$$\Sigma_{i,j,k} = \begin{pmatrix} (\sigma_i^{\alpha_1})^2 & \rho_{i,j}^{(\alpha_1,\beta_1)} \sigma_i^{\alpha_1} \sigma_j^{\beta_1} & \rho_{i,k}^{(\alpha_1,\alpha_3)} \sigma_i^{\alpha_1} \sigma_k^{\alpha_3} \\ \rho_{i,j}^{(\alpha_1,\beta_1)} \sigma_i^{\alpha_1} \sigma_j^{\beta_1} & (\sigma_j^{\beta_1})^2 & \rho_{j,k}^{(\beta_1,\alpha_3)} \sigma_j^{\beta_1} \sigma_k^{\alpha_3} \\ \rho_{i,k}^{(\alpha_1,\alpha_3)} \sigma_i^{\alpha_1} \sigma_k^{\alpha_3} & \rho_{j,k}^{(\beta_1,\alpha_3)} \sigma_j^{\beta_1} \sigma_k^{\alpha_3} & (\sigma_k^{\alpha_3})^2, \end{pmatrix}$$

and $\rho_{i,j}^{(\alpha_1,\beta_1)} = \tilde{T}_i^{\alpha_1} \cdot \tilde{T}_j^{\beta_1}$, $\rho_{i,k}^{(\alpha_1,\alpha_3)} = \tilde{T}_i^{\alpha_1} \cdot \tilde{T}_k^{\beta_3}$, $\rho_{j,k}^{(\beta_1,\alpha_3)} = \tilde{T}_j^{\beta_1} \cdot \tilde{T}_k^{\alpha_3}$(the inner product of two overlapping parts from the two *standardized* templates). However, $(\sigma_i^{\alpha_1})^2$, $(\sigma_j^{\beta_1})^2$ and $(\sigma_k^{\alpha_3})^2$ are learned from the correlations between the corresponding templates and the random background patches, instead of being equal to $\frac{1}{|T_i^{\alpha_1}|}$, $\frac{1}{|T_j^{\beta_1}|}$ and $\frac{1}{|T_k^{\alpha_3}|}$, since we are not assuming that the background is i.i.d.

Next, we will consider the second term of the product in equation 8.1. As we have mentioned, there are many configurations for each terminal brick, including locations, orientations, scales and specific templates, so that each state of terminal bricks contains information about the absolute coordinate of its corresponding template. In fact, when we learned the templates, we also obtained the corresponding mixing probabilities. Thus, we only need to deal with the coordinate distribution. First let the mixing probabilities of the templates corresponding to $x^{\alpha_1} = i$, $x^{\beta_1} = j$ and $x^{\alpha_3} = k$ be q_1^i, q_2^j and q_3^k(actually, $q_1^i = 1$,since we only learned one template for first part). Let the absolute coordinates associated with $x^{\alpha_1} = i$, $x^{\beta_1} = j$ and $x^{\alpha_3} = k$ be $X_1^{(i)}$, $X_2^{(j)}$ and $X_3^{(k)}$. Then, we have

$$\begin{aligned} & P(x^{\alpha_1} = i, x^{\beta_1} = j, x^{\alpha_3} = k | x^{\delta} = 2, x^{\alpha_2} = 2) \\ = \;& q_1^i \cdot q_2^j \cdot q_3^k \cdot P(X_1^{(i)}, X_2^{(j)}, X_3^{(k)} | x^{\delta} = 2, x^{\alpha_2} = 2, T_i^{\alpha_1}, T_j^{\beta_1}, T_k^{\alpha_3}). \end{aligned}$$

Starting from the Markov Backbone model, x^{α_1}, x^{β_1} and x^{α_3} are conditionally independent given x^{δ}. In other words,

$$P(X_1^{(i)}, X_2^{(j)}, X_3^{(k)} | x^{\delta} = 2, x^{\alpha_2} = 2, T_i^{\alpha_1}, T_j^{\beta_1}, T_k^{\alpha_3}) = P(X_1^{(i)} | x^{\delta} = 2, x^{\alpha_2} = 2, T_i^{\alpha_1}) \cdot \\ P(X_2^{(j)} | x^{\delta} = 2, x^{\alpha_2} = 2, T_j^{\beta_1}) \cdot P(X_3^{(k)} | x^{\delta} = 2, x^{\alpha_2} = 2, T_k^{\alpha_3}).$$

For simplification, we write the above equation as

$$P^{MB}(X_1^{(i)}, X_2^{(j)}, X_3^{(k)}) = P(X_1^{(i)})P(X_2^{(j)})P(X_3^{(k)}).$$

Using the r-cube law(see Subsection 5.5.1) and assuming that the X_i's are uniform on locations and orientations, we have $P(X_i) \sim \frac{1}{r^3}$, and P^{MB} is well specified. Next, by perturbing P^{MB}, we can obtain the following composed distribution:

$$P^c(X_1^{(i)}, X_2^{(j)}, X_3^{(k)}) = P^{MB}(X_1^{(i)}, X_2^{(j)}, X_3^{(k)}) \cdot \frac{P(R(X_2^{(j)}; X_1^{(i)}), R(X_3^{(k)}; X_1^{(i)}))}{P^{MB}(R(X_2^{(j)}; X_1^{(i)}), R(X_3^{(k)}; X_1^{(i)}))},$$

where $P(R(X_2^{(j)}; X_1^{(i)}), R(X_3^{(k)}; X_1^{(i)}))$ is our expected distribution of $R(X_2^{(j)}; X_1^{(i)})$ and $R(X_3^{(k)}; X_1^{(i)})$. In order to estimate $P(R(X_2^{(j)}; X_1^{(i)}), R(X_3^{(k)}; X_1^{(i)}))$, we simply assume that $R(X_2^{(j)}; X_1^{(i)})$ and $R(X_3^{(k)}; X_1^{(i)})$ are independent(in other words, $X_2^{(j)}$ and $X_3^{(k)}$ are conditionally independent given $X_1^{(i)}$), so that we can estimate $P(R(X_2^{(j)}; X_1^{(i)}))$ and $P(R(X_3^{(k)}; X_1^{(i)}))$ separately. Since we learned the low resolution templates under the same scale of training images as well as the high resolution templates, we assume that $X_1^{(i)}$, $X_2^{(j)}$ and $X_3^{(k)}$ have the same scale. Thus, the last component of $R(X_2^{(j)}; X_1^{(i)})$ and $R(X_3^{(k)}; X_1^{(i)})$ are fixed to be 0.[1] For the other components of $R(X_2^{(j)}; X_1^{(i)})$ and $R(X_3^{(k)}; X_1^{(i)})$, we assume that they are Gaussian distributed. Now, we have both the training images and our learned templates so that we can learn those Gaussian parameters by the EM algorithm.

Eventually, we build the second term of the product in equation 8.1 and, in

[1] Notice that if the template associated with X is high resolution template, then the last component of $R(X; X_1^{(i)})$ would be 7.1 since the scale for high resolution templates is 7.1 times of the scale for low resolution templates

addition to the first term, the ratio can be written as below

$$\frac{P(y|x^{\delta}=2,x^{\alpha_2}=2)}{P^0(y)}=\sum_{i=1}^{n^{\alpha_1}}\sum_{j=1}^{n^{\beta_1}}\sum_{k=1}^{n^{\alpha_3}}\frac{P_C(c_i^{\alpha_1})P_C(c_j^{\beta_1})P_C(c_k^{\alpha_3})}{N(c_i^{\alpha_1},c_j^{\beta_1},c_k^{\alpha_3};(0,0,0),\Sigma_{i,j,k})}$$

$$\cdot q_1^i q_2^j q_3^k\cdot P(X_1^{(i)})P(X_2^{(j)}|X_1^{(i)})P(X_3^{(k)}|X_1^{(i)}) \qquad (8.2)$$

, where $c_i^{\alpha_1}$, $c_j^{\beta_1}$ and $c_k^{\alpha_3}$ are $cor(T_i^{\alpha_1},y_{R_i^{\alpha_1}})$, $cor(T_j^{\beta_1},y_{R_j^{\beta_1}})$ and $cor(T_k^{\alpha_3},y_{R_k^{\alpha_3}})$ respectively. However, it is too expensive to compute equation 8.2, since n^{α_1}, n^{β_1} and n^{α_3} are very large numbers involving locations, scales, orientations and templates. Since most terms in the sum are negligible, we are going to provide a "pruning method" to reduce the computational expense. We first compute the ratio

$$\frac{q_1^iP_C(c_i^{\alpha_1})}{N(c_i^{\alpha_1};0,(\sigma_i^{\alpha_1})^2)},\ \frac{q_2^jP_C(c_j^{\beta_1})}{N(c_j^{\beta_1};0,(\sigma_j^{\beta_1})^2)},\ \frac{q_3^kP_C(c_k^{\alpha_3})}{N(c_k^{\alpha_3};0,(\sigma_k^{\alpha_3})^2)}$$

for $i=1,2,..,n^{\alpha_1}$, $j=1,2,..,n^{\beta_1}$ and $k=1,2,..,n^{\alpha_3}$. Then, for each group, we keep the biggest 400(say the index $i\in\{i_1,i_2,...,i_{400}\}\equiv I_i$, the index $j\in\{j_1,j_2,...,j_{400}\}\equiv I_j$ and the index $k\in\{k_1,k_2,...,k_{400}\}\equiv I_k$). Next we compute

$$\frac{q_1^iq_2^jP_C(c_i^{\alpha_1})P_C(c_j^{\beta_1})}{N(c_i^{\alpha_1},c_j^{\beta_1};(0,0),\Sigma_{i,j})}P(X_1^{(i)})P(X_2^{(j)}|X_1^{(i)})$$

for $i\in I_i$ and $j\in I_j$, where the covariance matrix is

$$\Sigma_{i,j}=\left(\begin{array}{cc}(\sigma_i^{\alpha_1})^2 & \rho_{i,j}^{(\alpha_1,\beta_1)}\sigma_i^{\alpha_1}\sigma_j^{\beta_1}\\ \rho_{i,j}^{(\alpha_1,\beta_1)}\sigma_i^{\alpha_1}\sigma_j^{\beta_1} & (\sigma_j^{\beta_1})^2\end{array}\right).$$

Among the 1600 terms, we choose the biggest 400(say the pair index

$$(i,j)\in\{(\bar{i}_1,\bar{j}_1),(\bar{i}_2,\bar{j}_2),...,(\bar{i}_{400},\bar{j}_{400})\}\equiv I_{i,j}\ ).$$

Finally, we compute

$$\frac{P(y|x^{\delta}=2,x^{\alpha_2}=2)}{P^0(y)}=\sum_{(i,j)\in I_{i,j}}\sum_{k\in I_k}\frac{P_C(c_i^{\alpha_1})P_C(c_j^{\beta_1})P_C(c_k^{\alpha_3})}{N(c_i^{\alpha_1},c_j^{\beta_1},c_k^{\alpha_3};(0,0,0),\Sigma_{i,j,k})}$$

$$\cdot q_1^iq_2^jq_3^k\cdot P(X_1^{(i)})P(X_2^{(j)}|X_1^{(i)})P(X_3^{(k)}|X_1^{(i)}).$$

Notice that there are only 1600 terms in this sum so that it is manageable.

Similarly, we can obtain the other ratios of the probabilities of y given the non-zero states of non-terminal bricks and $P^0(y)$: $\frac{P(y|x^\delta=1,x^{\alpha_2}=i,x^{\alpha_4}=j)}{P^0(y)}$, $\frac{P(y|x^\delta=2,x^{\alpha_2}=i)}{P^0(y)}$, $\frac{P(y|x^\delta=3,x^{\alpha_2}=i,x^{\alpha_4}=j)}{P^0(y)}$, $\frac{P(y|x^\delta=4,x^{\alpha_4}=j)}{P^0(y)}$ and $\frac{P(y|x^\delta=5,x^{\alpha_2}=i)}{P^0(y)}$, for $i = 1, 2, 3$ and $j = 1, 2, 3$. Next, the probability distribution of y, given that it is an ankle image,(or given that $x^\delta > 0$) can be written as

$$\begin{aligned} P(y|x^\delta > 0) = \ & \textstyle\sum_{i,j} P(y|x^\delta = 1, x^{\alpha_2} = i, x^{\alpha_4} = j)P(x^\delta = 1, x^{\alpha_2} = i, x^{\alpha_4} = j|x^\delta > 0) \\ + \ & \textstyle\sum_i P(y|x^\delta = 2, x^{\alpha_2} = i)P(x^\delta = 2, x^{\alpha_2} = i|x^\delta > 0) \\ + \ & \textstyle\sum_{i,j} P(y|x^\delta = 3, x^{\alpha_2} = i, x^{\alpha_4} = j)P(x^\delta = 3, x^{\alpha_2} = i, x^{\alpha_4} = j|x^\delta > 0) \\ + \ & \textstyle\sum_j P(y|x^\delta = 4, x^{\alpha_4} = j)P(x^\delta = 4, x^{\alpha_4} = j|x^\delta > 0) \\ + \ & \textstyle\sum_i P(y|x^\delta = 5, x^{\alpha_2} = i)P(x^\delta = 5, x^{\alpha_2} = i|x^\delta > 0). \end{aligned}$$

Therefore, in order to obtain the likelihood ratio

$$\frac{P(y|x^\delta > 0)}{P^0(y)}$$

we must learn those conditional probabilities, $P(x^\delta = 1, x^{\alpha_2} = i, x^{\alpha_4} = j|x^\delta > 0)$, $P(x^\delta = 2, x^{\alpha_2} = i|x^\delta > 0)$, $P(x^\delta = 3, x^{\alpha_2} = i, x^{\alpha_4} = j|x^\delta > 0)$, $P(x^\delta = 4, x^{\alpha_4} = j|x^\delta > 0)$ and $P(x^\delta = 5, x^{\alpha_2} = i|x^\delta > 0)$ by using the maximum likelihood estimate. Finally, the likelihood ratio is specified and will be applied to image classification as in the next subsection.

8.3 X-ray image classification

In this subsection, we will apply our compositional model to the classification problem. From the previous subsection, we have built a composition model for lateral ankle images. Now we can use the model to do classification. Given an image y, we can compute the ratio

$$ratio(y) \equiv \frac{P(y|x^\delta > 0)}{P^0(y)}$$

and then, depending on how big the ratio is, we know how likely it is to be a lateral ankle image. To be clear, we define the positive set to be a set of lateral ankle images that have at least part α_1 in the Figure 8.3. The negative set is a set of the remaining images. Therefore, the four images in Figure 8.10 are negative images.

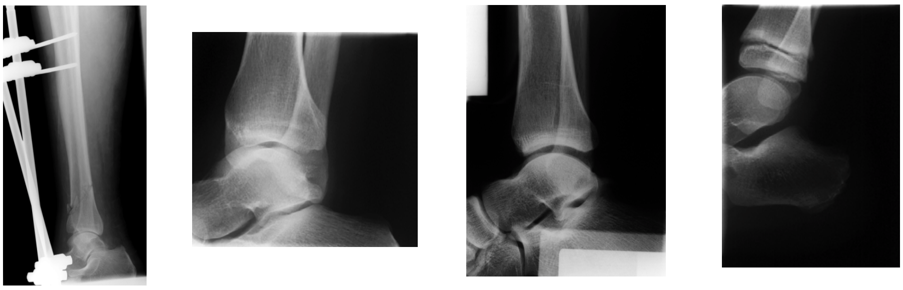


Figure 8.10: The particular examples of negative images

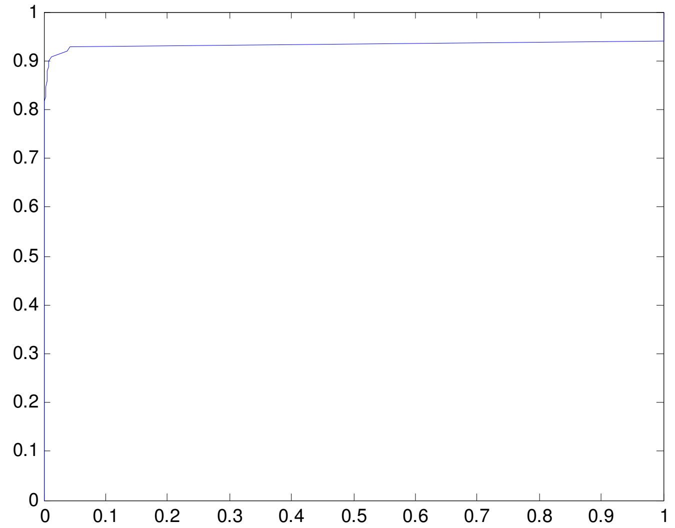


Figure 8.11: The ROC curve of the classification experiment

More specifically, we will categorize the first image as a lower leg image, the second and third images as ankle joint images, and the fourth image as a calcaneus image.

Now, we calculate the ratio on 2072 negative images and on 99 positive images, and we plot the ROC curve as shown in Figure 8.11. From the curve, we know that we can get a 90.91% detection rate with only a 1.35% false positive. Figure 8.12

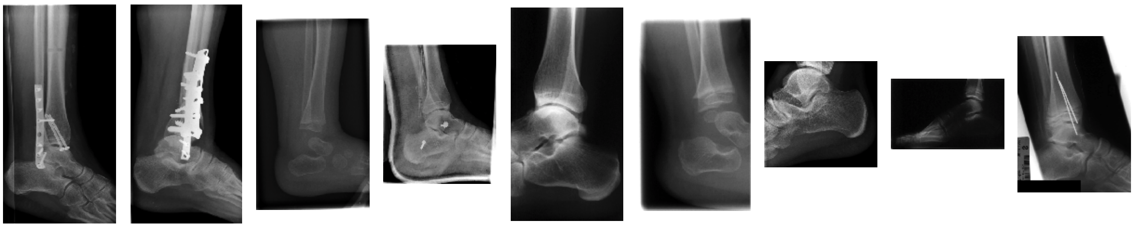


Figure 8.12: False negative images

shows the false negative images. The first, second, fourth and ninth false negative images are caused by occlusions, like metals, wires, screws, etc. If we want to reduce the false negative rate, we need to model these occlusions and let our priors allow the occlusions to be set on the images. Moreover, the third and the sixth false negative images are baby or child ankle images, but our training image set does not contain such images. Therefore, if we want to detect them, we will need many child ankle training images to create a model for children ankle images and to build a branch for it in our composition system.

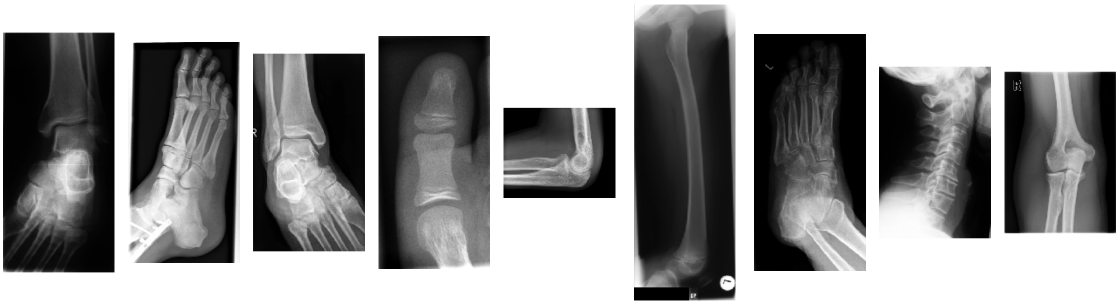


Figure 8.13: False positive images

In addition, if we look at the false positive images, we find that some parts of the image look like some parts of an ankle. However, the entire image is not an ankle. For example the fifth image in Figure 8.13 is an elbow image, but the right hand side of the image looks like an ankle. Of course, a human has no problem recognizing that it is not an ankle, because he/she knows what an elbow looks like. Our brains have different models for different categories that tell us that it is more likely to be an elbow image. Therefore, in order to get better performance, we should also model the elbow images in the same way. Then we can see that the probability for the elbow model will be greater than the probability for the ankle model, which indicates that it should be an elbow image.

Chapter 9

Conclusion and future direction

9.1 Conclusion

Pattern recognition has been a challenging and interesting research field for several decades, especially in computer vision. Researchers want to build machines to imitate how people recognize objects. We believe that the only way to accomplish this task is through an effective usage of hierarchy, reusability and compositionality. In this part of the thesis, we have introduced a general approach of how to present an object hierarchically and of how to build a generative model of the object. Under the Bayesian framework, we model the object by two steps. In the first step, we decompose the object into many reusable parts and then organize them hierarchically. Any possible way to present or explain the object corresponds to an interpretation, but the challenge is to set a proper probability distribution on those interpretations. The Markov Backbone is easy to deal with through a directed graph, but the problem is that the model usually has to be context-sensitive. To overcome this problem, we iteratively perturbed the Markov Backbone model into a context-sensitive model, which is consistent with the desired conditional distribution. Theoretically, we proposed a perturbation theorem that guarantees the convergence of iteratively perturbing. In practice, we have an approximation form with which to proceed.

In the second step, we proposed a general approach for modeling image data given the upper level image interpretations. This approach can be applied not only to image analysis but also to other extremely high-dimensional data sets. The key is to reduce the complexity of the data by focusing, through conditioning, on a collection of low-dimensional features of the data. We factored the image distribution into a low-dimensional feature distribution and into a high-dimensional background distribution given its feature. Then, we encounter the difficulty of modeling background distribution. To avoid this difficulty, we created a method to skirt around the background distribution and to end up with a likelihood ratio of the low-dimensional feature distributions. The feature is the subject of interest, and it can be any function of the data. In particular, we used the template correlation, which involves the templates as representatives of the object. The templates and the corresponding parameters can be learned by maximum likelihood estimation. Combined with the first step, the generative model is concrete. Since the generative model is based on a concrete probability distribution, it tells us how likely it is that our concerns are involved. It is also convenient for extending it or embedding it into higher hierarchical compositions. Therefore, there are many applications for generative models including classifications, detections, trackings etc. Moreover, the composition system in the generative model is full of representations which provides the capacity to model a more complicated object naturally.

9.2 Future Directions

Sequential Testing.

The theoretical and experimental ROC results reported in Chapter 7 are encouraging, and suggest several directions for generalization and application.

1. Decision cascades are routinely used in image processing and image analysis. Usually, a classifier is built for every stage of the cascade, and thresholds are set through more-or-less ad hoc methods. But we can think of the sequence of classifiers from a different point of view: Each classifier can be thought of as performing a test for a critical component of the object (e.g. face) or

overall classification (e.g. scratch or dirt in an image-processing setting). By viewing the cascade this way, we can formulate the decision problem as one of verifying the presence of a set of parts, much like we verified the presence of a sequence of letters in the word-detection examples in Chapter 7. This opens the door to using conditional probabilities, and using a single threshold. The cascade of calculations becomes a product of conditional probabilities, necessarily non-increasing with each additional stage. The process exits when and if the product drops below threshold. We intend to pursue the approach for some practical vision problems.

2. The motivation behind our sequential-testing result is to define a feasible scheme for orchestrating computations in a fully generative compositional model. We believe that all of the theoretical pieces are in place for such a model (as addressed in some of the other chapters of this thesis), and that it should now be feasible to implement a prototype system for a non-trivial vision task. Possibly the most important element is the computational engine, which we envision utilizing a hierarchy of sequential tests.

   In principle, when detecting an object by a generative model, we compute the probability of the object being present, given the data, and compare it to a given threshold. This probability is an integration over all of the possible instantiations of the object. This integral is obviously computationally expensive and not feasible in all but the simplest examples. In the X-ray image experiment, we calculated the approximated probabilities of an object being present by recursively approximating the probabilities of the subparts being present, each itself an integral over instantiations. The point is that the instantiations at one level can be pruned to remove the negligible contributions. The integral at a given level is thereby supported by a relatively small number of terms. These are "passed up" to the next higher level, and the procedure continues until an approximate probability is found for the presence of the object of interest. But this is still too much computation for a scalable system. The difficulty is that the probabilities of all objects need to be approximated.

One way to save computation is to reject an interpretation before completing the integration process. For example, if we consider the depth-first method, we can set up a threshold for the first decision step. If the probability is greater than the given threshold, we pass the first step and move to the next decision step; otherwise we reject the interpretation. Using this method, most negative images will be rejected in the first several decisions, so there will be no need to compute the entire integral. Still there is the problem of thresholds: for each decision step, we will need an appropriate threshold associated with it. This too is clearly impractical, and brings us back to the sequential testing "cascade method." Our goal is to set only one threshold that can be used in every decision step of the entire procedure, under the conditional framework. Furthermore, we want to build a prototype that takes advantage of highly parallel computing systems, by attacking sub-trees, simultaneously, with separate processors.

3. This next idea is hugely speculative: Many believe, as do we, that the folding of bio-molecules proceeds in prototypical sequences, with each stage producing additional secondary-structure bindings. It is not unreasonable to speculate that these bindings are the ones that produce the largest, or nearly largest, drops in energy. The resulting probability of a structure would then be the product of a sequence of conditional probabilities, each one chosen to have the highest conditional probability given the current structure. Of course there are many variations on this theme (e.g. including a redundancy term that accounts for the very important combinatorial factor that captures the number of pathways to a given additional binding structure), but there is at least a superficial resemblance to the testing sequence used to verify the presence of parts in a composition of parts. It might be possible to test this hypothesis by looking at known structures and their phylogenetic histories. One might

expect that the oldest substructures (bindings) are the ones that occur the earliest in the sequence of bindings in an individual molecule.

Multi-dimensional features for more precise patch models.

The conditioning method provided in Chapter 6.2 is a very general approach and it can be theoretically applied to any feature or to any statistic. We have used "template correlation" as the feature in the model due to its nice properties. However, the template correlation will not in general take into account all of critical features of the appearance of an object, for instance skin tones and texture characteristic. Depending on the application, we may require many critical features (in other words, multi-dimensional features) in order to properly characterize the object. Especially in image recognition, we will likely need more than one feature in order to achieve high-performance discrimination.

The difficulty is that the univariate distribution of a single feature, which is easy to handle statistically, becomes a multivariate. How do we build a model for the joint distribution $P(c_1(y), c_2(y), ..., c_n(y))$ of the features $c_1(y), c_2(y), ..., c_n(y)$? The easiest way, of course, is to assume independence among those features, but it is not likely that this will be a good assumption. We will explore the multi-feature problem, probably through parameterized multivariate distributions, in order to build a more effective model in the future.

Learning.

Parameter learning is a standard step in Bayesian modeling. Given the data, the MLE is a standard method to learn the parameters, and the EM algorithm is the most common tool to handle the latent variables. However, there are some limitations to both the MLE and the EM algorithm, including consistency and computational issues revolving around local maxima. Learning the state probabilities, the ϵ_i^α's, will require both positive training sets and negative training sets. In particular, ϵ_0^α, the probability of brick α being "off," is usually very big (close to one), and depends critically on the training data. It is extremely difficult to learn since it is extremely difficult to obtain a "random sample" of images with and without the object of interest. For most objects, a random set of images would have to be very large indeed to yield an accurate estimate of the a priori probability of having a particular object. On the other hand, the *conditional* probability of finding an object in a particular state, given that it is present (i.e. $\epsilon_k^\alpha/(1-\epsilon_0^\alpha)$ in the Markov backbone) can be reasonably estimated from a training set made up of instances of the object. In our X-ray image experiment, we only learned the conditional probabilities of the states given that the brick is "on." The additional parameter, ϵ_0^α, only changes the threshold for deciding between object-present and object-absent.

The sequential testing approach, on the other hand, requires estimates of the unconditioned probabilities, and this becomes an important problem for future study. Additionally, there is the challenging and important problem of learning the compositions themselves (i.e. the architecture), directly from unlabeled data.

Bibliography

[1] D.J. Aldous, *The random walk construction for spanning trees and uniform labelled trees*, SIAM J. Discrete Math. **3** (1990), 450–465.

[2] S. Allassonnière, Y. Amit, and A. Trouvé, *Towards a coherent statistical framework for dense deformable template estimation*, Journal Of The Royal Statistical Society Series B **69** (2007), 3–29.

[3] S.F. Altschul and B.W. Erickson, *Significance of nucleotide sequence alignments: a method for random sequence permutation that preserves dinucleatide and codon usage*, Mol. Biol. Evol. **2** (1985), 526–538.

[4] Luis Alvarez, Yann Gousseau, and Jean-Michel Morel, *The size of objects in natural images*, Advance in Imaging and Electron Physics **111** (1999), 167–242.

[5] Y. Amit and A. Trouv, *Pop: Patchwork of parts models for object recognition*, International Journal of Computer Vision **75(2)** (2007).

[6] Y. Amit and A. Trouvé, *Generative models for labeling multi-object configurations in images*, Lecture Notes in Computer Science, vol. 4170/2006, Springer, 2006, pp. 362–381.

[7] T. Ané and H. Geman, *Order flow, transaction clock, and normality of asset returns*, Journal of Finance **55** (2000), no. 5, 2259–2284.

[8] Uri Avni, Hayit Greenspan, Michal Sharon, Eli Konen, and Jacob Goldberger, *X-ray image categorization and retrieval using patch-based visual words representation*, ISBI'09: Proceedings of the Sixth IEEE international conference on

Symposium on Biomedical Imaging (Piscataway, NJ, USA), IEEE Press, 2009, pp. 350–353.

[9] R. R. Bahadur and R. Ranga Rao, *On deviations of the sample mean*, Annals of Mathematical Statistics **31** (1960), 1015–1027.

[10] J. Besag and D. Mondal, *Exact goodness-of-fit tests for markov chains*, 2004.

[11] E. Bienenstock, *Notes on the growth of a composition machine*, Proceedings of the Royaumont Interdisciplinary Workshop on Compositionality in Cognition and Neural Networks (D. Andler, E. Bienenstock, and B. Laks, eds.), 1991.

[12] Christopher M. Bishop, *Pattern recognition and machine learning*, Springer, 2006.

[13] T. Bjork and H. Hult, *A note on wick products and the fractional black-scholes model*, Finance and Stochastics **9** (2005), 197–209.

[14] G. Blanchard and D. Geman, *Hierarchical testing designs for pattern recognition*, Annals of Statistics **33** (2005), 1155–1202.

[15] E. Borenstein and S. Ullman, *Feature hierarchies for object classification*, IEEE International Conference on Computer Vision (2005).

[16] A. Broder, *Generating random spanning trees*, Proceedings of the 30th IEEE Symposium on Foundations of Computer Science, 1989, pp. 442–447.

[17] L.-B. Chang, S.-C. Chen, F. Hsieh, C.-R. Hwang, and M. Palmer, *Empirical invariance in stock market and related problems, draft*, Submitted for publication, 2009.

[18] L.-B. Chang, A. Goswami, F. Hsieh, and C.-R. Hwang, *An invariance for the large-sample empirical distribution of waiting time between successive extremes*, Submitted for publication, 2009.

[19] P.K. Clark, *A subordinated stochastic process model with finite variance for speculative prices*, Econometrica **41** (1973), no. 1, 135–155.

[20] J.C. Cox, S.A. Ross, and M. Rubinstein, *Option pricing: A simplified approach*, Financial Economics **7** (1979), 229–263.

[21] P. Diaconis and D. Freedman, *Finite exchangeable sequences*, Annals of Probability **8** (1980), no. 4, 745–764.

[22] R.J. Elliott and J.V. Hoek, *A general fractional white noise theory and applications to finance*, Mathematical Finance **13** (2003), no. 2, 301–330.

[23] R. Fergus, P. Perona, and A. Zisserman, *Object class recognition by unsupervised scale-invariant learning*, IEEE Conference on Computer Vision and Pattern Recognition (2003).

[24] S. Fidler, M. Boben, and A. Leonardis, *Similarity-based cross-layered hierarchical representation for object categorization*, IEEE Conference on Computer Vision and Pattern Recognition (2008).

[25] F. Fleuret and D. Geman, *Coarse-to-fine face detection*, International Journal of Computer Vision **41** (2001), 85–107.

[26] J. Fodor and Z. Pylyshyn, *Connectionism and cognitive architecture: a critical analysis*, Cognition **28** (1988), 3–71.

[27] W. T. Freeman, J. Yedidia, and Y. Weiss, *Understanding belief propagation and its generalizations*, International Joint Conference on Artificial Intelligence (2001).

[28] Brendan J. Frey, *Transformation-invariant clustering using the em algorithm*, IEEE transcations on pattern analysis and machine intelligence **25** (2003), 1–17.

[29] Brendan J. Frey and Nebojsa Jojic, *Transformed component analysis: Joint estimation of spatial transformations and image components*, International Conference on Computer Vision **2** (1999), 1190.

[30] H. Geman, D.B. Madan, and M. Yor, *Time changes for Lévy processes*, Mathematical Finance **11** (2001), no. 1, 79–96.

[31] S. Geman, *Invariance and selectivity in the ventral visual pathway*, J. of Physiology – Paris **100** (2006), 212–224.

[32] S. Geman and M. Johnson, *Article title probability and statistics in computational linguistics, a brief review*, Mathematical foundations of speech and language processing (2004), 1–26.

[33] S. Geman, D. F. Potter, and Z. Chi, *Composition systems*, Quarterly of Applied Mathematics **LX** (2002), 707–736.

[34] M. Harrison and S. Geman, *A pattern-preserving resampling algorithm for neural spike trains*, Neural Computation **21** (2009), 1244–1258.

[35] N. Hatsopoulos, S. Geman, A. Amarasingham, and E. Bienenstock, *At what time scale does the nervous system operate?*, Neurocomputing **52-54** (2003), 25–29.

[36] T. Serre Heisele, B. and T. Poggio, *A component-based framework for face detection and identification*, International Journal of Computer Vision **74(2)** (2007), 167–181.

[37] S.L. Heston, *A closed-form solution for options with stochastic volatility with applications to bond and currency options*, The Review of Financial Studies **6** (1993), no. 2, 327–343.

[38] M. Hollander and D.A. Wolfe, *Nonparametric statistical methods, 2nd edition*, Wiley, Hoboken, NJ, 1999.

[39] Y. Hu and B. Oksendal, *Fractional white noise calculus and applications to finance*, Infin. Dimens. Anal. Quantum Probab. Relat. Top. **6** (2003), 1–32.

[40] J. Hull and A. White, *The pricing of options on assets with stochastic volatilities*, Journal of Finance **42** (1987), 281–300.

[41] Y. Jin and S. Geman, *Context and hierarchy in a probabilistic image model*, CVPR'06, vol. (2), IEEE, 2006, pp. 2145–2152.

[42] Ya Jin, *Non-markovian hierarchy vision system*, Ph.D. thesis, Brown University, Division of Applied Mathematics, 2006.

[43] D. Kandel, Y. Matias, R. Unger, and P. Winkler, *Shuffling biological sequences*, Discrete Applied Mathematics **71** (1996), 171–185.

[44] Anitha Kannan, Nebojsa Jojic, and Brendan Frey, *Fast transformation-invariant factor analysis*, in Advances in Neural Information Processing Systems **15** (2002).

[45] I. Kokkinos and P. Maragos, *Synergy between image segmentation and object recognition using the expectation maximization algorithm*, IEEE Transactions on Pattern Analysis and Machine Intelligence **31** (2009), 1486–1501.

[46] Ann B. Lee, David Mumford, and Jinggang Huang, *Occlusion models for natural images: A statistical study of a scale-invariant dead leaves model*, International Journal of Computer Vision **41** (2001), 35–59.

[47] E.L. Lehmann, *Nonparametrics: Statistical methods based on ranks*, Holden-Day, San Francisco, CA, 1975.

[48] Jun Lu and Qiuqi Ruan, *Aluminum alloy x-ray image classification using texture analysis*, Signal Processing, 2006 8th International Conference, 2006.

[49] F. Min, J.L. Suo, and S.C. Zhu, *An and-or graph model for face representation, sketching and aging*, Chapter in Encyclopedia of Biometric Recognition , Springer (2009).

[50] A. Mueen, M.S. Baba, and R. Zainuddin, *Multilevel feature extraction and x-ray image classification.*, Journal of Applied Science **7** (2007), 1224–1229.

[51] Devrim nay, Octavian Soldea, Ahmet Ekin, Mjdat etin, and Aytl Eril, *Automatic annotation of x-ray images: a study on attribute selection*, Medical Content-based Retrieval for Clinical Decision Support: In conjunction with MICCAI 2009 (12th International Conference on Medical Image Computing and Computer Assisted Intervention), London, UK, 2009.

[52] B. Ommer and J.M. Buhmann, *Learning the compositional nature of visual objects*, CVPR'07, IEEE, 2007.

[53] N. Reid, *The roles of conditioning in inference*, Statistical Science (1995), 138–157.

[54] S. Roth and M. J. Black, *Fields of experts: A framework for learning image priors*, IEEE Conference on Computer Vision and Pattern Recognition **II** (2005), 860–867.

[55] ———, *Fields of experts*, International Journal of Computer Vision **82** (2009), 205–229.

[56] Daniel L. Ruderman, *Origins of scaling in natural images*, Vission Research **37** (1997), 3385–3398.

[57] T. Serre, A. Oliva, and T. Poggio, *A feedforward architecture accounts for rapid categorization*, Proceedings of the National Academy of Science **104** (2007), 6424–6429.

[58] N. Shephard, *Stochastic volatility*, Oxford University Press, Oxford, 2005.

[59] Jeonghee Shim, Kihee Park, Byoungchul Ko, and Jaeyeal Nam, *X-ray image classification and retrieval using ensemble combination of visual descriptors*, PSIVT '09: Proceedings of the 3rd Pacific Rim Symposium on Advances in Image and Video Technology (Berlin, Heidelberg), Springer-Verlag, 2008, pp. 738–747.

[60] Z. Tu, X. Chen, A. Yuille, and S. Zhu, *Image parsing: unifying segmentation, detection and recongnition*, International Journal of Computer Vision **63** (2005), 113–140.

[61] S. Ullman, E. Sali, and M. Vidal-Niquet, *A fragment-based approach to object representation and classification*, International Workshop on Visual Form (2001), 85–100.

[62] D.C. Van Essen, C.H. Anderson, and D.J. Felleman, *Information processing in the primate visual system: an integrated systems perspective*, Science **255** (1992), 419–423.

[63] P. Viola and M. J. Jones, *Robust real-time face detection*, Proc. ICCV01, 2001, p. II: 747.

[64] C. von der Malsburg, *Synaptic plasticity as a basis of brain organization*, The Neural and Molecular Bases of Learning (J.P. Changeux and M. Konishi, eds.), John Wiley and Sons, 1987, pp. 411–432.

[65] M. Weber, M. Welling, and P. Perona, *Unsupervised learning of models for recognition*, Proc. Sixth European Conf. computer Vision (2000), 18–32.

[66] Z.J. Xu, H. Chen, and S.C. Zhu, *A high resolution grammatical model for face representation and sketching*, Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), San Diego, June (2005).

[67] W. Zhang, *Statistical inference and probabilistic modeling in compositional vision*, Ph.D. thesis, Brown University, Division of Applied Mathematics, May, 2009.

[68] L. Zhu, Y. Chen, X. Ye, and A. Yuille, *Structure-perceptron learning of a hierarchical log-linear model*, IEEE Conference on Computer Vision and Pattern Recognition (2008).

[69] L. Zhu, C. Lin, H. Huang, Y. Chen, and A. Yuille, *Unsupervised structure learning: Hierarchical recursive composition, suspicious coincidence and competitive exclusion*, ECCV 2008, 2008.

[70] S.C. Zhu and D. Mumford, *A stochastic grammar of images*, Foundations and Trends in Computer Graphics and Vision **2(4)** (2006), 259–362.

[71] S.C. Zhu and D. Mumford, *Quest for a stochastic grammar of images:*, Foundations and Trends in Computer Graphics and Vision (2007).