Statistical Inference and Probabilistic Modeling in Compositional Vision

by

Wei Zhang

B.A., Fudan University, P.R.China, 2003

M.S., Brown University, RI, 2006,

M.A., Brown University, RI, 2007

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 2009

Abstract of "Statistical Inference and Probabilistic Modeling in Compositional Vision" by Wei Zhang, Ph.D., Brown University, May 2009

This thesis is a mathematical and computational study of compositional vision. Three topics are covered: (1) ROC performance in a compositional world; (2) the construction of a probabilistic model for compositional structure; and (3) the construction of probabilistic model of image gray levels for a given vocabulary of elementary parts.

Chapter 1 introduces compositional vision and a probabilistic framework for modeling hierarchy, reusability, and conditional data models.

Chapter 2 focuses on theoretical questions about the ROC performance of various approaches to recognition in hypothetical compositional worlds. The results suggest that even sub-optimal decisions within a hierarchical framework will substantially outperform a decision process that does not explicitly allow for part-based decomposition.

Chapter 3 focuses on the first component of the Bayesian approach to compositional vision: a prior probability model on hierarchical image interpretations. Non-Markovian (context-sensitive) distributions are investigated, and two theoretical questions are addressed. The existence of a class of non-Markovian distributions is established, and the convergence of an iterative perturbation scheme for achieving these distributions is proven.

Chapter 4 focuses on the second component of the Bayesian approach to compositional vision: a probability model on pixel intensities conditioned on a given hierarchical structure. In particular, a generative approach to modeling object parts is

developed through a probabilistic extension of the idea of fragment-based templates.

Chapter 5 makes some conclusions and suggests future directions.

This dissertation by Wei Zhang 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_____ _____

Elli 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 Wei Zhang

Wei Zhang was born in Baiquan, Heilongjiang province, China on November 15, 1981.

She graduated from First High School of Baiquan. In 1999, she started her undergraduate study in Fudan University and received her Bachelor of Science degree in Mathematics four years later.

She came to the United States in 2003 and has been attending the Ph.D. program in the Division of Applied Mathematics at Brown University. During the Ph.D. program, she received a Master of Science in Applied Mathematics in 2006 and a Master of Arts in Economics in 2007. She received the Stella Dafermos award in 2009 from the Division Applied Mathematics at Brown University. This dissertation was defended on April 13th, 2009.

Acknowledgments

I would like to thank all 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 shape my thought and guide my career. This thesis would not be possible without his persistent help.

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

I want to acknowledge Eran Borenstein and Ya Jin, who have been very helpful during my research. Also I want to thank Sergey Kushnarev, Jingmei Qiu, Lo-Bin Chang, Christian Pfrang, Yanchun Wu, Yanqiu Li, and all the people in the Division of Applied Mathematics for helping me in every aspect.

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

Contents

List of Figures

Chapter 1

Introduction to Compositional Vision

1.1 Introduction

Most approaches to object recognition or detection in computer vision can be broadly characterized as either generative or discriminative according to whether or not a probability distribution of the image features is modeled.

Discriminative approaches are popular in classification related tasks. The idea is to compute a direct mapping from an observed variable Y to a hidden variable X for classification; no direct attempt is made to model the underlying distributions. Various forms of penalty functions, regularization, and kernel functions are used to prevent overfitting. Many discriminative models have been proposed and widely used in computer vision, involving, for example, support vector machines [1], neural networks [2] and boosting [3].

Generative approaches specify a joint probability distribution $P(X, Y)$, by specifying a prior distribution $P(X)$ and a likelihood function $P(Y|X)$. Once the generative model is built, classifiers can be derived in a straightforward way by exploiting the posterior distribution $P(X|Y)$, calculated through Bayes' formula. In contrast

with generative models, discriminative models focus only on the posterior distribution $P(X|Y)$. Hence, a generative model features being able to simulate (i.e., to generate, or to sample) Y, but a discriminative model does not. Gaussian mixture models, hidden Markov models, and Markov random fields are commonly used generative models. In the recent years, hierarchical generative models have gained more attention due to their rich representation and search efficiency. And a growing number of them are achieving state-of-art performance in a growing number of applications. Some are biologically motivated (e.g. [34]), others are computationally motivated (e.g. [4]). Some involve learned hierarchies (e.g. [9]), others are hand-designed (e.g. [16, 15]). Some are deep hierarchies (e.g. [21, 22] and [23]), some are medium ([29]), others are shallow (e.g. POP model [19, 39], Constellation model [20, 36]).

This thesis work falls into the category of hierarchical generative model. It is motivated by both biological evidence in human vision and theoretical evidence for improved ROC performance in compositional (hierarchical) models.

1.2 Motivation

- Primates, especially humans, are remarkably good at learning, recognizing, and generalizing objects from a few examples. The performance of their visual system and its robustness surpasses the best state-of-art computer vision systems. Over the last decade, evidence has accumulated about some key features of the ventral visual pathway. The human vision system depends on a hierarchy of successive layers in the visual cortex. The first layers of the cortex detect an object's simpler features, such as oriented edges, and higher layers integrate that information to form our perception of the object as a whole. It is a simple-to-complex cell hierarchy, [33, 34]. And there is an apparent increase in both invariance and selectivity in moving from the primary visual cortex

to the infero-temporal cortex (see, e.g. [35] and references therein). In addition to the biological evidence from the ventral visual pathway, it is evident that humans have the tendency to represent entities as hierarchies of reusable parts. In a visual world, objects and scenes naturally decompose into hierarchies of meaningful and generic parts. Natural languages are also hierarchical: hierarchies of words, phrases, and sentences whose composition is governed by grammatical rules.

- In a miniature compositional world, better ROC performance of compositional model has been justified theoretically, compared to the models that do not accommodate compositionality. These issues will be introduced and studied in detail in Chapter 2.

  The basic idea is as follows. Consider a very simple and perfectly compositional world. There are two parts—vertical and horizontal bars. Each part, when it appears, appears in a fixed pose. When both parts appear, we declare the presence of an 'L'. The world, then, has four states: H_0 which generates no parts; H_1 which generates only a vertical bar; H_2 which generates only a horizontal bar; and H_3 which generates an L. Figure 1.1 shows the four hypotheses in this compositional world. For each state, a binary scene (image) is generated by adding independent Bernoulli noise to the binary clean image ('1' for foreground, '0' for background), where the addition is binary summation. The problem is to build a classifier that recognizes L's.

  According to the Neyman-Pearson lemma, the optimal classifier would base the decision (L or not L) on the likelihood ratio: If Y is the observed array of pixel intensities (the image) then declare L if

$$\frac{P(Y|H_3)}{P(Y|\bar{H}_3)} \geq c \tag{1.1}$$

  and not L otherwise. Here, $\bar{H}_3$ is short for the compound (mixture) event

$H_0 \cup H_1 \cup H_2$, so

$$P(Y|\bar{H}_3) = P_0 \cdot P(Y|H_0) + P_1 \cdot P(Y|H_1) + P_2 \cdot P(Y|H_2)$$

The threshold c, over the range $[0, \infty)$, sweeps out the optimal ROC curve. Specifically, for each value of c there is a detection probability (i.e. true positive rate) p_d and a false alarm probability (i.e. false positive rate) p_f, and among all classifiers with false alarm probability p_f, p_d is the highest possible detection probability. The problem with the Neyman-Pearson prescription is that it is impractical. Not actually impractical for the thought experiment, but impractical in anything resembling a real experiment, with multiple parts, multiple objects, variable poses, and so on.

An expedient alternative is to devise a "universal-null" model, a serviceable probability on Y under the "not object" condition. The "white noise" model is analyzed here as a universal null (i.e. $P(Y|H_0)$), and there would be a direct generalization to alternative models for $P(Y|H_0)$. In particular, concerning the thought experiment, we will consider the performance of the likelihood ratio test (equation (1.1)) when $P(Y|\bar{H}_3)$ is replaced by the background distribution $P(Y|H_0)$: declare L if

$$\frac{P(Y|H_3)}{P(Y|H_0)} \geq c \tag{1.2}$$

and not L otherwise.

Another alternative is to accommodate the fact that a 'L' is composed of two parts, and to declare L only when there was sufficient evidence for both the vertical and horizontal bars. In and of themselves, these simple building blocks are easy to devise tests for, and in fact in the artificial world of the thought experiment the optimal tests involve no mixtures, as the alternative in each case (the denominator) really is the background, white-noise, model. Consider,

then, the "parts" strategy that declares L if

$$\frac{P(Y|H_1)}{P(Y|H_0)} \geq c \quad and \quad \frac{P(Y|H_2)}{P(Y|H_0)} \geq c \tag{1.3}$$

(or, equivalently, if $\min(\frac{P(Y|H_1)}{P(Y|H_0)}, \frac{P(Y|H_2)}{P(Y|H_0)}) \geq c$), and not L otherwise.

Within this problem framework, two theorems – one under a non-hierarchical setting and the other under a hierarchical settings – have been established in Chapter 2 to compare the ROC performance of these three strategies, as the image resolution (i.e. the number of pixels representing a given area of the image) goes to infinity. The ROC curve produced by the parts strategy is exponentially better than the one produced by the universal-null strategy, and is comparable and eventually merges together with the one produced by the optimal strategy given by the Neyman-Pearson lemma.

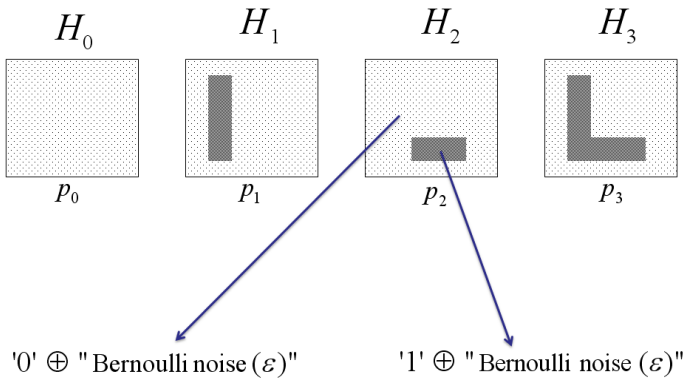

Figure 1.1: Thought experiment. A simple compositional world. There are only four states to the world. Each state generates a binary random image of independent binary pixel intensities. '⊕' in the figure stands for binary summation. Each "part" (a vertical or a horizontal bar) has only one pose, and the presence of both parts constitutes an 'L'.

If the world is indeed compositional, we would do better to take advantage of this fact in devising machines intended to mimic or even just approach human per-

formance. The thought experiment of detecting 'L' above gives us a hint on how to accommodate the structure feature of our world to achieve better performance in computer vision tasks. Certainly, our vision world is way more complicated than the miniature compositional world in the thought experiment. But the philosophy is similar, combining semantic parts into semantic objects. For example, it is obviously crucial to consider the relative coordinates of the parts before combining parts into higher level semantic objects. For example, 'L' and 'T' are both composed of a horizontal bar and a vertical bar, but the relative location of these two parts determines whether to combine them into 'L', or 'T', or neither.

This thesis work is built on the platform of the"compositional machine" – a generative probabilistic model on hierarchies of reusable parts under the Bayesian framework, and has been pioneered by Geman, Bienenstock, and their colleagues [11, 14]. It has been further developed in the Ph.D. work of Huang [12] (preliminary computational experiments), Harrison [13] (some learning related work), and Jin [16, 15] (implementation in license-plate reading).

The compositional machine has two components, the prior distribution on the (image) interpretations, and the conditional data distribution (i.e. the likelihood function) on the image given the interpretations. Section 1.3 through section 1.6 will introduce the framework of compositional machine as follows: Section 1.3 will introduce the Markov distribution on the interpretations. We call this the "Markov Backbone", which serves as a reference distribution. Markov systems like branching processes and probabilistic context-free grammars qualify, but are generally too weak to capture context and content-dependent likelihoods. The Markov property can be saved, but only at the cost of a very large state space. A better approach might be to look for workable non-Markovian distributions. Section 1.4 will extend the "Markov Backbone" to a "compositional system" by introducing a non-Markovian term through a "perturbation" argument. Section 1.5 will describe the conditional data distribution. Section 1.6 will discuss scene parsing via the posterior distribution

and Bayesian inference.

1.3 Markov Backbone

The application that we have in mind is to 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 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 [15, 16], the semantic bricks represent different meanings as shown in Figure 1.2.

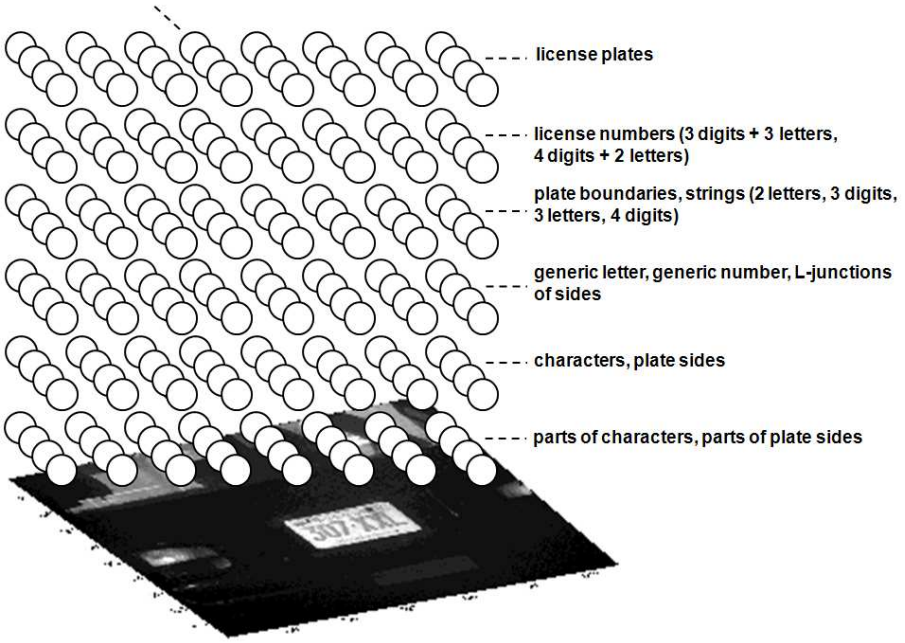

Figure 1.2: Semantic hierarchy for plate-reading application

1.3.1 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, $\alpha \in \mathcal{B}$
$\vec{x} = \{x^\beta : \beta \in \mathcal{B}\}$	state of all bricks
$\{\epsilon_i^\alpha\}_{i=0}^{n^\alpha},\ 0 \leq \epsilon_i^\alpha \leq 1,\ \sum_{i=0}^{n^\alpha} \epsilon_i^\alpha = 1$	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 $i \neq j$)

1.3.2 Interpretations

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.

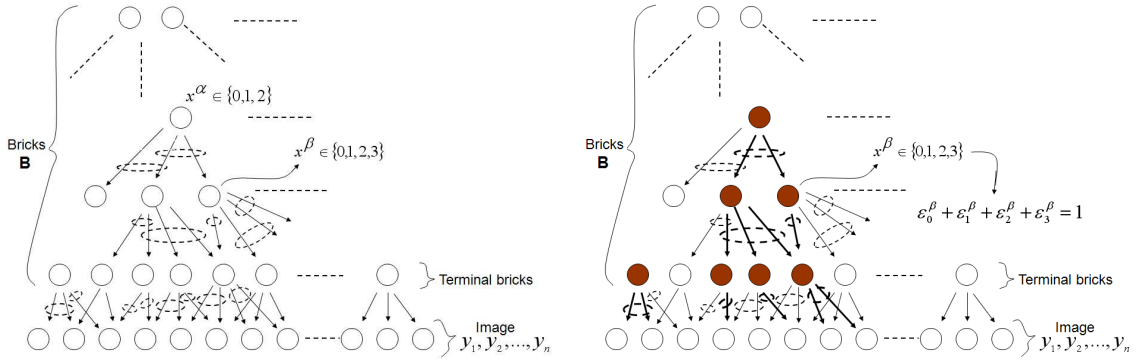

Figure 1.3: Architecture. Left. A hierarchy of "bricks," each representing a disjunction of conjunctions. Bottom row is the image (pixel) data and the row above it is the set of terminal bricks. The state of a brick signals a chosen set of children. Right. An "interpretation," which is an assignment 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.

1.3.3 Markov Probabilities

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

$$P(\vec{x}) = \frac{\prod_{\beta \in \mathcal{B}} (\epsilon^\beta_{x^\beta})}{\prod_{\beta \in B(\vec{x})} (1 - \epsilon^\beta_0)} \tag{1.4}$$

Remarks:

1. $\sum_{\vec{x} \in \mathcal{I}} P(\vec{x}) = 1$, *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. *$P(\vec{x})$ 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 $\alpha \rightarrow \{\beta : \beta \in C_i^\alpha\}$ as a production, chosen with probability $\frac{\epsilon_i^\alpha}{1-\epsilon_0^\alpha}$. 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 [15, 16], Jin and Geman sampled from the Markov backbone, given the semantic assignment of bricks (in Figure 1.2) and the manually hardwired children sets. The left panel of Figure 1.4 shows a 4-digit sample under the Markov backbone. As seen from the figure, although the parts of each digit are present and in roughly the correct locations, neither the parts nor the digits are properly situated.

Figure 1.4: Samples from Markov backbone (left panel, '4850') and compositional distribution (right panel, '8502').

1.4 Content Sensitivity and Non-Markovian Perturbations

Most of the proposed generative models in the literature share the Markov property ([20], [27], [17], [7], [47, 48]), 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 [14]. 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 of β. Depending on the depth of the instantiation of the children, $a^\beta(\vec{x})$ may depend on the states of bricks that are several generations removed from β itself (grandchildren, great grandchildren, etc.).

Start with the Markov probability P, as defined in (1.4), and fix a particular brick $\gamma \in \mathcal{B}$. Under (1.4), a^γ has *some* distribution, $P_0^\gamma(a^\gamma|x^\gamma)$, for every state, $x^\gamma \in \{0, 1, \ldots, n^\gamma\}$. If, say, $a^\gamma(\vec{x})$ is the vector of poses of the chosen children of γ, then there is no reasonable hope that P_0^γ corresponds to the *empirical* (or "real-world") distribution on the positions of the parts of γ. After all, (1.4) 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 $P_0^\gamma(a^\gamma|x^\gamma)$) be the correct conditional distribution on the attribute a^γ. One way to "perturb" P in (1.4), so as to correct the conditional

a^γ distributions, is to choose the new distribution, call it P^*, that is closest to P, subject to the constraint that $P^*(a^\gamma|x^\gamma) = P_c^\gamma(a^\gamma|x^\gamma)$ for all a^γ and all x^γ. If, by "closer," we mean that $D(P||P^*)$ (Kullback-Leibler divergence) is minimized, then it is an easy calculation to show that

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

Remarks:

1. *The particular distribution, $P_c^\gamma(a^\gamma|x^\gamma = 0)$, is largely irrelevant to the problem of modeling an object γ, would be very hard to measure, and in any case can be taken as $P_0^\gamma(a^\gamma|x^\gamma = 0)$ so that there is no perturbation at all unless the γ 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 $x^\gamma > 0$ (γ "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, $a^\gamma(\vec{x})$ would be a function only of x^γ and its possible progeny, meaning every brick that 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," $\gamma_1, \gamma_2, \ldots, \gamma_{|\mathcal{B}|}$, with $\{\gamma_1, \gamma_2, \ldots, \gamma_{|\mathcal{B}|}\} = \mathcal{B}$, we end up with a distribution

$$\begin{aligned} P^*(\vec{x}) &= P(\vec{x})\prod_{v=1}^{|\mathcal{B}|}\frac{P_c^{\gamma_v}(a^{\gamma_v}(\vec{x})|x^{\gamma_v})}{\tilde{P}^{\gamma_v}(a^{\gamma_v}(\vec{x})|x^{\gamma_v})} \\ &= \frac{\prod_{\beta\in\mathcal{B}}(\epsilon^\beta_{x^\beta})}{\prod_{\beta\in B(\vec{x})}(1-\epsilon_0^\beta)}\prod_{v=1}^{|\mathcal{B}|}\frac{P_c^{\gamma_v}(a^{\gamma_v}(\vec{x})|x^{\gamma_v})}{\tilde{P}^{\gamma_v}(a^{\gamma_v}(\vec{x})|x^{\gamma_v})}, \end{aligned} \tag{1.5}$$

where $\tilde{P}^{\gamma_v}(a^{\gamma_v}(\vec{x})|x^{\gamma_v})$ is the distribution on a^{γ_v} given x^{γ_v} *at the time of the visit to the* γ_v *brick*, and $\tilde{P}^{\gamma_v}(a^{\gamma_v}(\vec{x})|x^{\gamma_v}) = P_0^{\gamma_v}(a^{\gamma_v}(\vec{x})|x^{\gamma_v})$ 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 $P_c^{\gamma}(a^{\gamma}|x^{\gamma})$, 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 $\{a^{\gamma}(\vec{x})\}_{\gamma\in\mathcal{B}}$ together with the attribute (conditional) distributions $\{P_c^{\gamma}(a^{\gamma}|x^{\gamma})\}_{\gamma\in\mathcal{B}}$ will in general under-determine the distribution on $\mathcal{I}$: there are typically many distributions with the desired constraints. But we do not yet have a satisfactory theory guaranteeing the existence of such a distribution. See Chapter 3 for a guarantee under some restrictions on the conditional constraints. There is a related question of convergence: is it true that an iterative procedure, that visits every site infinitely often, converges to a distribution with the desired attribute probabilities? This turns out to be true under quite general conditions (Chapter 3).

The license-plate application explored in [15, 16] used a simple approximation. Each pre-perturbation probability, $\tilde{P}^{\gamma}(a^{\gamma}|x^{\gamma}))$ in (1.5), was assumed to be close to, and was replaced by, the corresponding conditional probability under the Markov distribution (1.4), which is denoted by $p_o^{\gamma}(a^{\gamma}|x^{\gamma})$. The right panel of Figure 1.4 shows a compositional 4-digit sample generated by Jin and Geman [15, 16] from this non-Markovian model. As we can see, dramatic improvement is achieved, compared to the sampling result from the Markov backbone (in the left panel of Figure 1.4). 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 (even exact values)

can be derived for the conditional attribute probabilities. Since the target distributions, $\{P_c^\gamma(a^\gamma|x^\gamma)\}_{\gamma\in\mathcal{B}}$, are fixed (by hand or inference) and the "null" probabilities $\{p_o^\gamma(a^\gamma|x^\gamma)\}_{\gamma\in\mathcal{B}}$, all derive from the Markov distribution (1.4), there is no dependence on order. These considerations lead to the useful (and order-independent) approximation:

$$P^*(\vec{x}) \propto \frac{\prod_{\beta\in\mathcal{B}}(\epsilon^\beta_{x^\beta})}{\prod_{\beta\in B(\vec{x})}(1-\epsilon_0^\beta)} \prod_{\beta\in\mathcal{B}} \frac{P^\beta(a^\beta(\vec{x})|x^\beta)}{p_o^\beta(a^\beta(\vec{x})|x^\beta)}$$

A price is paid in that the normalization is no longer exact. On the other hand, the parameters in the Markov "backbone" (1.4), 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.

1.5 Images and Image Probabilities

The Bayesian (generative) framework is completed by specifying a "data model": a probability distribution on images given an interpretation, $\vec{x} \in \mathcal{I}$.

1.5.1 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.

1.5.2 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.*

1.5.3 A Note on Modeling Terminal Bricks

The support of an interpretation $\vec{x}$ is covered by the supports of the active ("on") terminal bricks. These define disjoint 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 $\mathcal{R}_i^\tau$, $i \in \{1, 2, \ldots, n^\tau\}$.

What is a reasonable model for $y_{\mathcal{R}_i^\tau}$, given that $x^\tau = i$? Here is an outline of a model that is learnable and has performed well in experiments.

Fix a terminal brick τ and a state i ($x^\tau = i$). For ease of notation, let $G \stackrel{\triangle}{=} \mathcal{R}_i^\tau$. The task is to model Y_G, under the condition that $x^\tau = i$ and that there is no other (active) support intersecting G. (The extension to overlapping supports follows similar reasoning.)

The idea is that we want to model a distribution $P(Y_G)$ in terms of something resembling a "sufficient statistic" $S(Y_G)$:

$$P(Y_G) = P(S(Y_G))/c(S(Y_G))$$

where

$$c(s) = \#\{\tilde{Y}_G : S(\tilde{Y}_G) = s\}$$

The probability on the data depends only on the statistic ("feature") S, and hence all configurations of Y_G that give the same value of S are equally likely. Notice that, whereas $P(S)$ is the *marginal* distribution on S (conditioned on $x^\tau = i$), it would be a mistake to model the the probability of the actual data, Y_G, as being proportional to $P(S(Y_G))$. The combinatorial factor, $c(s)$, can substantially tilt the data likelihood in favor of configurations that have low "redundancy" (small combinatorial factors, or what are sometimes miss-labeled as low entropies).

A simple, and effective, concrete example is

$$S(Y_G) = Corr(T, Y_G)$$

where $T = \{t_k\}_{k \in G}$ is a template, and $Corr(T, Y_G)$ is the normalized correlation between T and Y_G (and hence between -1 and $+1$). A good fit to real data (say patches of left eyes, or portions of characters or strokes) can be made by taking $P(S = s)$ to be monotone increasing on $s \in [-1, 1]$, as in the backwards exponential $P(S = s) \propto e^{-\lambda(1-s)} \approx \lambda e^{-\lambda(1-s)}$. What is needed is an approximation to the combinatorial factor, $c(s)$, since the direct calculation is intractable.

Chapter 4 studies image probabilities and terminal bricks in detail. It gives a derivation for an approximation for $c(s)$, and an explicit likelihood model. The parameters of the model (the template T and the scalar λ) can be learned from data through the maximum likelihood equations. In fact, templates can be learned despite pose (e.g. scale, rotation, and translation) variation in the training data, and can be learned at super or sub resolution. Chapter 4 also includes recognition experiments with maximum-likelihood templates.

1.6 Scene Parsing

In the previous sections, we have specified the prior distribution $P(\vec{x})$ and the data model $P(Y|\vec{x})$. At this point, the generative model has been fully built up. Equipped with this generative model, the computer vision task of scene parsing (i.e. the image interpretation) can be approached through the posterior distribution on an image interpretation $\vec{x}$ given the input image data Y, $P(\vec{x}|Y)$:

$$P(\vec{x}|Y) = \frac{P(Y|\vec{x})P(\vec{x})}{P(Y)} \propto P(Y|\vec{x})P(\vec{x}). \tag{1.6}$$

The image interpretation $\vec{x}$ provides a rich semantic and syntactic parsing of the scene through the brick variables of the compositional machine. With $\vec{x}$, each image pixel can be identified as either plain background (with no structure), or cluttered background (with structure), or objects at different complexity levels. For example, the license demonstration system [15, 16] could read out from interpretation $\vec{x}$ the detailed locations and identifications of license plate, string, boundary, partial string, L-junction, character, line, and parts, etc., which may occur anywhere across the image.

Ideally, the MAP estimator could be computed exactly given the prior model $P(\vec{x})$ and data model $P(Y|\vec{x})$. But the sample space is very high dimensional considering that a compositional machine can easily contain thousands of bricks. Hence, exact inference is intractable and some computationally feasible approximations are needed. The computation engine explored in [15, 16] took advantage of the coarse-to-fine search strategy, motivated by [4]. Jin and Geman [15, 16] focused on a bottom-up pass and touched on a general depth-first search strategy, aiming at fast object detection, for example, early detection of a license boundary.

Chapter 2

ROC Performance in a Compositional World

2.1 Introduction

Human vision is both highly invariant (invariance refers to the extent to which objects are detected independent of their pose and rendering) and highly selective (selectivity refers to avoiding the misclassification of other structures). Computer vision systems are rarely both. In particular, systems which are highly invariant often suffer from an unacceptable number of false detections. Such an "invariance and selectivity" dilemma is due to a combination of extreme variation in object presentations and the high structure of background clutter.

When facing the problem of testing for the presence of an object O in an input image I, we can not do better than to base our decision on the likelihood ratio, i.e., to threshold the ratio $P(I|O)/P(I|\overline{O})$, where O stands for "object present" and $\overline{O}$ stands for "object absent." This is the Neyman-Pearson Lemma. We refer to the decision model suggested by this lemma as the "optimal model" in this chapter.

The problem with Neyman-Pearson Lemma prescription is that it is not generally practical to enumerate $\overline{O}$. It is one thing to have a model (a likelihood function) of

the data given the presence of a particular object of interest (as in $P(I|O)$), but it is quite another thing to have a model of the data given that the object is *not* present (as in $P(I|\overline{O})$). "Not present" is a mixture—a very big mixture—and it would be bad news indeed if good classification or object detection required a good data model under this mixture.

One way around this is to approximate $P(I|\overline{O})$ by $P(I)$, since the set $\overline{O}$ is very nearly the entire image set. Then $P(I)$ is usually referred as the background or natural-scene model. But of course $P(I)$ is complicated to calculate as well.

An expedient alternative is to devise a "universal-null model," a serviceable probability on I under the "object absent" condition. We will analyze here the "white noise" model as a universal null. Given that the world is highly structured, this naive assumption does not seem proper, although it is widely used (often implicitly) in recognition algorithms. For example, it is common to see in the literature that $P(I|O)$ is thresholded for object O detection. But thresholding $P(I|O)$ is the same as thresholding $P(I|O)/P(I|\overline{O})$ while assuming the denominator to be a constant, for example $(1/256)^{|I|}$, as in the i.i.d. white noise background model.

We argue that we can do better than the universal-null model, at least if we accept that the principles of hierarchy and reusability are operating in the real world. Since the clutter of background shares the same reusable parts as the target (object), the idea of compositionality suggests that we devise tests to address the inaccuracy caused by an unnecessary binary choice between target and universal null. The idea is to perform multiple hypothesis tests on object parts, i.e., to detect the parts of the object O, and only claim the existence of O if all the part tests succeed. We call this model the "parts model."

In this chapter, we demonstrate the efficient discrimination available through compositional and hierarchical representation via two theorems establishing a better receiver operating characteristic (ROC) curve. We study the asymptotic (as the resolution of the data goes to infinity) ROC performance of three models: the optimal

model (given by the Neyman-Person Lemma), the parts model, and the universal-null model. This chapter is organized as follows: Section 2.2 will introduce some background materials about ROC curves. Section 2.3 will employ the Large Deviation Theory to prove that the parts model is comparable to the optimal model, and is strictly better than the universal-null model (in terms of asymptotic ROC performance). Section 2.4 will compare the three models within a hierarchical problem setting. Finally, Section 2.5 will conclude with a discussion and a summary.

2.2 ROC Curves

In signal detection theory, a ROC curve is a graphical plot of the true positive rate (TPR) versus the false positive rate(FPR) for a binary classification system as its discrimination threshold varies. ROC analysis is an efficient tool for researchers to select possibly optimal models and to discard suboptimal ones. It is widely used in multiple-disciplines – for example, decision making in medical diagnosis, model selecting in biosciences, machine learning and data mining, etc..

We now introduce some basic concepts in ROC analysis that will be used in this Chapter. Let Y be the observed data. The goal is to make a binary decision between two hypotheses $Y \in H_1$ or $Y \in H_0 = H_1^c$. The discrimination rule is $Y \in H_1$ if $F(Y) \geq c$ and $Y \in H_0$ otherwise, where F is a determinant function designed by the user, mapping Y to a real number, and c is a threshold, $c \in R$.

Theoretically, if H_1 is associated with a probability distribution P_1 on domain Ω, and H_0 is associated with a distribution P_2 on domain Ω, we can calculate the theoretical true positive rate and false positive rate as

$$\text{TPR} = P_1(\{Y \in \Omega : F(Y) \geq c\}), \quad \text{FPR} = P_2(\{Y \in \Omega : F(Y) \geq c\}).$$

In practice, we can only observe samples from H_1 and H_0. The observed dataset

S can be divided into positive dataset O and negative dataset B, where $O \cap B = \emptyset$. O contains data Y sampled from H_1, while B contains data sampled from H_0. Let $D = \{Y \in S : F(Y) \geq c\}$; then TPR and FPR are calculated as follows:

$$\text{TPR} = \frac{|D \cap O|}{|O|}, \quad \text{FPR} = \frac{|D \cap B|}{|B|},$$

where $|\cdot|$ is the measure of a set.

For both the theoretical case and the practical case, if we plot true positive rate versus false positive rate in the X-Y plane, $\forall c \in (-\infty, +\infty)$, we get the ROC curve. Each single threshold c corresponds to a point on the ROC curve. The best possible prediction result corresponds to the point (0,1) in the X-Y plane, which stands for 100 percent detection rate and 0 percent false positive rate. The (0,1) point is also called a perfect classification. A random classification, say by flipping a biased coin, would give a point along the diagonal line from the left bottom to the top right corners (the so-called line of no-discrimination). In general, the more closely the ROC curve approaches the (0,1) point, the better the classification result. If one ROC curve is always strictly higher on the Y axis, then we refer to this case as "the first ROC curve is strictly better than the other." See the example in Figure 2.1.

In statistics, the Neyman-Pearson Lemma states that when performing a hypothesis test between two point hypotheses $H_0 : \theta = \theta_0$ and $H_1 : \theta = \theta_1$, among the likelihood-ratio test which rejects H_0 in favor of H_1, the following testing rule is optimal:

$$F(Y) = \frac{P(Y|\theta_1)}{P(Y|\theta_0)} \geq c, \ \forall \ c \in R.$$

i.e., for any other test $\hat{F}$ with equal or smaller false positive rate than F, its true positive rate must be no greater than the one given by F. In other words, the ROC curve given by F is at least as good as the one given by $\hat{F}$.

Another concept that we will need later is the area under the ROC curve, denoted by AUROCC. The discussion and notation are simplified by assuming that Y is a

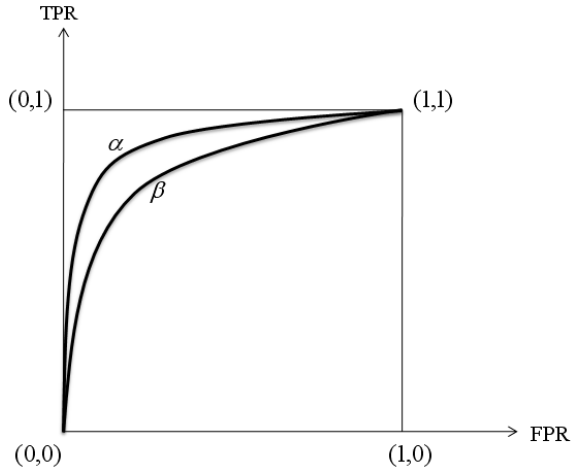

Figure 2.1: An example of two ROC curves. The ROC curve α is always strictly higher on the Y axis than the ROC curve β. Hence the decision rule associated with α is strictly better than the one associated with β.

continuous random variable under both hypotheses H_0 and H_1. AUROCC is a good measurement to assess the classifier, especially when evaluating two ROC curves which cross each other. AUROCC reaches its maximum 1, when the classifier makes error-free decisions. Green and Swets gave a concise formula to evaluate AUROCC theoretically, [24]:

$$\text{AUROCC} = P(\{(Y_1, Y_2) \in H_1 \times H_0 : F(Y_1) \geq F(Y_2)\}).$$

Accordingly, the area above the ROC curve, denoted as AAROCC, is equal to one minus AUROCC, i.e.,

$$\text{AAROCC} = P(\{(Y_1, Y_2) \in H_1 \times H_0 : F(Y_1) \leq F(Y_2)\}). \tag{2.1}$$

2.3 Comparison of Asymptotic ROC Curves of the Optimal Model, the Parts Model and the Universal-Null Model

Consider a thought experiment. Imagine a target (object) composed of two parts. The generative model is $Y_n = f_n(X) + Z_n$, where X stands for the presence of one target with two parts. X takes values in the set $\{(0,0),(1,0),(0,1),(1,1)\}$ (standing for four cases: nothing, only part 1 present, only part 2 present, the whole target present) with probability p_0, p_1, p_2, p_3 respectively ($p_0 + p_1 + p_2 + p_3 = 1, p_i > 0, \forall i$). See Figure 2.2A for a graphical representation. $f_n(X)$ is obtained by repeating the left part of X n times, followed by repeating the right part of X n times. For example, if $n = 3$ and $X = (1,0)$, we will have $f_n(X) = (1,1,1,0,0,0)$. Z_n represents random noise. Z_n is a vector with the same length as $f_n(X)$, containing $2n$ i.i.d Bernoulli(ϵ) random variables, $1 < \epsilon < 0.5$. So $Z_n = (z_1, \ldots, z_{2n})$, and $P(z_i = 1) = \epsilon, P(z_i = 0) = 1 - \epsilon$. The addition between $f_n(X)$ and Z_n follows the rule of binary summation. For example, if $n = 3, f_n(X) = (1,1,1,0,0,0), Z_n = (1,0,0,1,1,0)$, Y_n will be $(0,1,1,1,1,0)$. Our task is this: Observing Y_n, make a decision whether Y_n is derived from X=(1,1) or not.

Based on the general philosophy of the optimal model, the parts model and the universal-null model that we stated earlier, we define these three concepts in detail as follows, according to the current problem setting.

optimal model: We conclude $X = (1,1)$ if Y_n passes the following test:

$$\frac{P(Y_n|X=(1,1))}{P(Y_n|X=(0,0),(1,0),\text{ or }(0,1))} \geq c. \tag{2.2}$$

parts model: Let Y_{n1} be the first n elements of Y_n, and Y_{n2} be the second n elements of Y_n. Let X_1 stand for the first element of X, and X_2 be the second element of X.

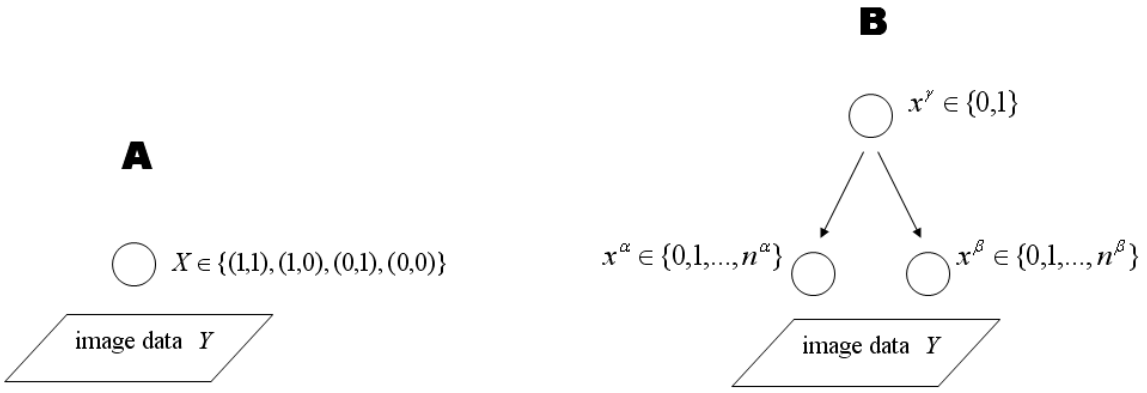

Figure 2.2: Graphical model. Panel "A" is a graphical representation for the non-hierarchical setting in Section 2.3. Panel "B" is a graphical representation for the hierarchical settings in Section 2.4.

We conclude $X = (1, 1)$, if Y_n passes both of the following tests:

$$\begin{cases} \frac{P(Y_{n1}|X_1=1)}{P(Y_{n1}|X_1=0)} & \geq & c \\ \\ \frac{P(Y_{n2}|X_2=1)}{P(Y_{n2}|X_2=0)} & \geq & c \end{cases} \tag{2.3}$$

universal-null model: We conclude $X = (1, 1)$ if Y_n passes the following test:

$$\frac{P(Y_n|X = (1, 1))}{P(Y_n|X = (0, 0))} = \frac{P(Y_n|X_1 = 1, X_2 = 1)}{P(Y_n|X_1 = 0, X_2 = 0)} \geq c. \tag{2.4}$$

2.3.1 Theoretical Comparison of Asymptotic ROC Curves

We are interested in the performance of the ROC curves associated with the three models as n goes to infinity. We will study this topic through analyzing the area above the three ROC curves. To ease the notations, we denote the area above the ROC curve associated with the optimal model by AG, the one associated with the

parts model by AP, and the one associated with the universal-null model by AW. It is obvious that in the limit, all three ROC curves go to the best point (0,1). Hence, there is no difference between asymptotic ROC curves given by these three models, and $AG, AP, AW \to 0$, as $n \to \infty$. However, what we are interested in is the convergence rate of the three ROC curves. We argue that the ROC curve of the parts model converges to (0,1) exponentially faster than the ROC curve of the universal-null model, and its convergence rate is comparable with that of the optimal model.

Theorem 1.

(1) $\lim_{n\to\infty} \frac{AP}{AW} = 0$*, and the ratio converges to zero in an exponential order.*

(2) $\frac{AP}{AG} \leq C$ *when* n *is large, where* C *is a positive constant.*

Proof. We will first prove the first part of the theorem, (1).

Let $I(Y_{n1})$ and $I(Y_{n2})$ denote the numbers of 1's in Y_{n1} and Y_{n2} separately. Then,

$$(2.3) \quad \Longleftrightarrow \quad \begin{cases} \frac{(1-\epsilon)^{I(Y_{n1})}\epsilon^{n-I(Y_{n1})}}{(1-\epsilon)^{n-I(Y_{n1})}\epsilon^{I(Y_{n1})}} & \geq \quad c \\ \frac{(1-\epsilon)^{I(Y_{n2})}\epsilon^{n-I(Y_{n2})}}{(1-\epsilon)^{n-I(Y_{n2})}\epsilon^{I(Y_{n2})}} & \geq \quad c \end{cases}$$

$$(2.4) \quad \Longleftrightarrow \quad \frac{(1-\epsilon)^{I(Y_{n1})+I(Y_{n2})} \cdot \epsilon^{2n-I(Y_{n1})-I(Y_{n2})}}{(1-\epsilon)^{2n-I(Y_{n1})-I(Y_{n2})} \cdot \epsilon^{I(Y_{n1})+I(Y_{n2})}} \geq c$$

Since $\epsilon < 0.5$,

$$(2.3) \quad \Longleftrightarrow \quad \min\{I(Y_{n1}), I(Y_{n2})\} \geq \hat{c};$$

$$(2.4) \quad \Longleftrightarrow \quad (I(Y_{n1}) + I(Y_{n2})) \geq \hat{c}.$$

Let

$$\begin{aligned} S &= \{Y_n : Y_n \text{ is derived from } X = (1,1)\}, \\ \overline{S} &= \{Y_n : Y_n \text{ was derived from } X = (1,0) \text{ or } (0,1) \text{ or } (0,0)\}, \\ S_{10} &= \{Y_n : Y_n \text{ is derived from } X = (1,0)\}, \\ S_{01} &= \{Y_n : Y_n \text{ is derived from } X = (0,1)\}, \\ S_{00} &= \{Y_n : Y_n \text{ is derived from } X = (0,0)\}. \end{aligned}$$

We have,

$$\overline{S} = S_{10} \cup S_{01} \cup S_{00}.$$

By (2.1),

$$\begin{aligned} AP &= P(\ \min\{I(\tilde{Y}_{n1}), I(\tilde{Y}_{n2})\} \geq \min\{I(Y_{n1}), I(Y_{n2})\} \mid Y_n \in S, \tilde{Y}_n \in \overline{S}). \\ AW &= P(\ (I(\tilde{Y}_{n1}) + I(\tilde{Y}_{n2})) \geq (I(Y_{n1}) + I(Y_{n2})) \mid Y_n \in S, \tilde{Y}_n \in \overline{S}). \end{aligned}$$

We make the following notations for later use:

$$\begin{aligned} AP_{10} &= P(\ \min\{I(\tilde{Y}_{n1}), I(\tilde{Y}_{n2})\} \geq \min\{I(Y_{n1}), I(Y_{n2})\} \mid Y_n \in S, \tilde{Y}_n \in S_{10}). \\ AP_{01} &= P(\ \min\{I(\tilde{Y}_{n1}), I(\tilde{Y}_{n2})\} \geq \min\{I(Y_{n1}), I(Y_{n2})\} \mid Y_n \in S, \tilde{Y}_n \in S_{01}). \\ AP_{00} &= P(\ \min\{I(\tilde{Y}_{n1}), I(\tilde{Y}_{n2})\} \geq \min\{I(Y_{n1}), I(Y_{n2})\} \mid Y_n \in S, \tilde{Y}_n \in S_{00}). \\ AW_{10} &= P(\ (I(\tilde{Y}_{n1}) + I(\tilde{Y}_{n2})) \geq (I(Y_{n1}) + I(Y_{n2})) \mid Y_n \in S, \tilde{Y}_n \in S_{10}). \\ AW_{01} &= P(\ (I(\tilde{Y}_{n1}) + I(\tilde{Y}_{n2})) \geq (I(Y_{n1}) + I(Y_{n2})) \mid Y_n \in S, \tilde{Y}_n \in S_{01}). \\ AW_{00} &= P(\ (I(\tilde{Y}_{n1}) + I(\tilde{Y}_{n2})) \geq (I(Y_{n1}) + I(Y_{n2})) \mid Y_n \in S, \tilde{Y}_n \in S_{00}). \end{aligned}$$

Note, when n is large, there exist positive constants C_1, C_2, C_3, C_4, and C such that

$$\begin{aligned}
AP_{10} &= AP_{01} \leq C_1 \cdot P(I(\tilde{Y}_{n2}) \geq I(Y_{n2}) \mid Y_n \in S, \tilde{Y}_n \in S_{10}), \\
AP_{00} &\leq C_2 \cdot P(I(\tilde{Y}_{n2}) \geq I(Y_{n2}) \mid Y_n \in S, \tilde{Y}_n \in S_{00}) \\
&\leq C_3 \cdot P(I(\tilde{Y}_{n2}) \geq I(Y_{n2}) \mid Y_n \in S, \tilde{Y}_n \in S_{10}), \\
AP &\leq C_4 \cdot \sum_{(i,j)\in\{(1,0),(0,1),(0,0)\}} AP_{ij} \\
&\leq C \cdot P(I(\tilde{Y}_{n2}) \geq I(Y_{n2}) \mid Y_n \in S, \tilde{Y}_n \in S_{10}), \qquad (2.5)
\end{aligned}$$

where the last inequality above is due to the fact that $\tilde{Y}_n \in S_{10}$ and $\tilde{Y}_n \in S_{00}$ influence $I(\tilde{Y}_{n2})$ the same way. Since $AW \leq AW_{10}$, from (2.5) we have

$$\begin{aligned}
\frac{AP}{AW} &= \frac{P(\ \min\{I(\tilde{Y}_{n1}), I(\tilde{Y}_{n2})\} \geq \min\{I(Y_{n1}), I(Y_{n2})\} \mid Y_n \in S, \tilde{Y}_n \in \overline{S})}{P(\ (I(\tilde{Y}_{n1}) + I(\tilde{Y}_{n2})) \geq (I(Y_{n1}) + I(Y_{n2})) \mid Y_n \in S, \tilde{Y}_n \in \overline{S})} \\
&\leq C \cdot \frac{P(I(\tilde{Y}_{n2}) \geq I(Y_{n2}) \mid Y_n \in S, \tilde{Y}_n \in S_{10})}{AW_{10}}.
\end{aligned}$$

Let $\{U_i\}_{i=1}^n, \{V_i\}_{i=1}^n, \{G_i\}_{i=1}^n$ and $\{Q_i\}_{i=1}^n$ be i.i.d Bernoulli(ϵ) r.v.'s, then $\forall\ Y_n \in S, \tilde{Y}_n \in S_{10}$ we have

$$I(\tilde{Y}_{n2}) = \sum_i V_i,\ I(Y_{n2}) = n - \sum_i Q_i,\ I(\tilde{Y}_{n1}) = n - \sum_i G_i,\ I(Y_{n1}) = n - \sum_i U_i.$$

Hence,

$$\begin{aligned}
&P(I(\tilde{Y}_{n2}) \geq I(Y_{n2}) \mid Y_n \in S, \tilde{Y}_n \in S_{10}\ ) = P(\ \sum_{i=1}^n (V_i + Q_i) \geq n\ ), \\
&AW_{10} = P(\ \sum_{i=1}^n [(V_i + Q_i) + (U_i - G_i)] \geq n\ ).
\end{aligned}$$

Therefore,

$$\begin{aligned}\frac{AP}{AW} &\leq C \cdot \frac{P(\ \sum_{i=1}^{n}(V_i+Q_i) \geq n\ \}}{P\{\ \sum_{i=1}^{n}[(V_i+Q_i)+(U_i-G_i)] \geq n\ )} \\ &= C \cdot \frac{P(\ \frac{1}{n}\sum_{i=1}^{n}(V_i+Q_i) \geq 1\ \}}{P\{\ \frac{1}{n}\sum_{i=1}^{n}(V_i+Q_i+U_i-G_i) \geq 1\ )}\end{aligned}$$

Let

$$\begin{aligned}J &= V_1+Q_1; \\ K &= V_1+Q_1+U_1-G_1; \\ H_J(\alpha) &= \log Ee^{\alpha J}; \\ H_K(\alpha) &= \log Ee^{\alpha K}; \\ L_J(\beta) &= \sup_{\alpha}(\alpha\beta - H_J(\alpha)); \\ L_K(\beta) &= \sup_{\alpha}(\alpha\beta - H_K(\alpha)).\end{aligned}$$

Note $L_J(\beta)$ and $L_K(\beta)$ are convex functions. Since $E[J] = E[K] = 2\epsilon$, both $L_J(\beta)$ and $L_K(\beta)$ achieve their minimum value 0 at $\beta = 2\epsilon$, and are positive for any $\beta \neq 2\epsilon$. Since $\epsilon < 0.5$, $2\epsilon < 1$, we have

$$\begin{aligned}\inf_{\beta\geq 1} L_J(\beta) &= L_J(1) > 0; \\ \inf_{\beta\geq 1} L_K(\beta) &= L_K(1) > 0.\end{aligned}$$

For $\alpha \neq 0$, by Jensen Inequality,

$$\begin{aligned} E[e^{\alpha K}] &= E[E[e^{\alpha K}|J]] \\ &= E[E[e^{\alpha(J+(U_1-G_1))}|J]] \\ &> E[e^{\alpha E[J+(U_1-G_1)|J]}] \\ &= E[e^{\alpha(J+E(U_1-G_1))}] \\ &= E[e^{\alpha J}] \end{aligned}$$

Thus,

$$H_K(\alpha) > H_J(\alpha),\ \forall \alpha \neq 0.$$

Assume,

$$\begin{aligned} L_K(1) &= \sup_{\alpha}(\alpha - H_K(\alpha)) \\ &= \alpha_0 - H_K(\alpha_0). \end{aligned}$$

Note that $L_K(1) > 0$ and $0 - H_K(0) = 0$, thus $\alpha_0 \neq 0$. Hence,

$$\begin{aligned} L_J(1) &= \sup_{\alpha}(\alpha - H_J(\alpha)) \\ &\geq \alpha_0 - H_J(\alpha_0) \\ &> \alpha_0 - H_K(\alpha_0) \\ &= L_K(1). \end{aligned} \tag{2.6}$$

By Cramer's Theorem,

$$\lim_{n\to\infty}\frac{1}{n}\log P(\ \frac{1}{n}\sum_{i=1}^{n}(V_i+Q_i)\geq 1\ ) \ = \ -\inf_{\beta\geq 1} L_J(\beta) = -L_J(1),$$

$$\lim_{n\to\infty}\frac{1}{n}\log P(\ \frac{1}{n}\sum_{i=1}^{n}(V_i+Q_i+U_i-G_i)\geq 1\ ) \ = \ -\inf_{\beta\geq 1} L_K(\beta) = -L_K(1).$$

Therefore,

$$\begin{aligned}\frac{AP}{AW} &\leq C\cdot\frac{P(\ \frac{1}{n}\sum_{i=1}^{n}(V_i+Q_i)\geq 1\ )}{P(\ \frac{1}{n}\sum_{i=1}^{n}(V_i+Q_i+U_i-G_i)\geq 1\ )}\\ &= C\cdot\frac{e^{-(L_J(1)+o(1))n}}{e^{-(L_K(1)+o(1))n}}\end{aligned}$$

We have concluded above that $L_J(1) > L_K(1)$ in (2.6), hence

$$\frac{AP}{AW} \text{ goes to zero exponentially, as } n \text{ goes to } \infty.$$

The first part of the theorem has been proved. Now we will prove the second part. Remember $I(Y_{n1})$ and $I(Y_{n2})$ stand for the numbers of 1's in Y_{n1} and Y_{n2} respectively. The decision rule of the optimal model in (2.2) is equivalent to

$$\frac{(1-\epsilon)^{I(Y_{n1})+I(Y_{n2})}\cdot\epsilon^{2n-I(Y_{n1})-I(Y_{n2})}\cdot p_3}{\left[\begin{array}{c}(1-\epsilon)^{I(Y_{n1})+n-I(Y_{n2})}\cdot\epsilon^{n-I(Y_{n1})+I(Y_{n2})}\cdot p_0\\ +\ (1-\epsilon)^{n-I(Y_{n1})+I(Y_{n2})}\cdot\epsilon^{I(Y_{n1})+n-I(Y_{n2})}\cdot p_1\\ +\ (1-\epsilon)^{2n-I(Y_{n1})-I(Y_{n2})}\cdot\epsilon^{I(Y_{n1})+I(Y_{n2})}\cdot p_2\end{array}\right]} \geq \ c$$

Considering the denominator above is a summation over three items, we flip the numerator and the denominator of LHS above, and get an equivalent decision rule

as

$$\left(\begin{array}{ll} & \frac{p_0}{p_3}\cdot\left(\frac{1-\epsilon}{\epsilon}\right)^{2n-2I(Y_{n1})-2I(Y_{n2})} \\ + & \frac{p_1}{p_3}\cdot\left(\frac{1-\epsilon}{\epsilon}\right)^{n-2I(Y_{n2})} \\ + & \frac{p_2}{p_3}\cdot\left(\frac{1-\epsilon}{\epsilon}\right)^{n-2I(Y_{n1})} \end{array}\right) \leq \ c \tag{2.7}$$

Let

$$\begin{aligned} & Z(Y_{n1}) = n - 2I(Y_{n1}), \quad Z(Y_{n2}) = n - 2I(Y_{n2}), \\ & a = \frac{1-\epsilon}{\epsilon} > 1, \quad b_0 = \frac{p_0}{p_3} \geq 0, \\ & b_1 = \frac{p_1}{p_3} \geq 0, \qquad b_2 = \frac{p_2}{p_3} \geq 0, \end{aligned}$$

and (2.7) becomes

$$b_0 \cdot a^{Z(Y_{n1})+Z(Y_{n2})} + \ b_1 \cdot a^{Z(Y_{n1})} + \ b_2 \cdot a^{Z(Y_{n2})} \leq c. \tag{2.8}$$

By (2.1), the Area above ROC curve of the optimal model is

$$AG = P\left(\begin{array}{l|l} b_0 \cdot a^{Z(Y_{n1})+Z(Y_{n2})} + b_1 \cdot a^{Z(Y_{n1})} + b_2 \cdot a^{Z(Y_{n2})} & \\ \geq \ b_0 \cdot a^{Z_1(\tilde{Y}_n)+Z_2(\tilde{Y}_n)} + b_1 \cdot a^{Z_1(\tilde{Y}_n)} + b_2 \cdot a^{Z_2(\tilde{Y}_n)} & Y_n \in S, \tilde{Y}_n \in \overline{S} \end{array}\right).$$

Since $\overline{S} = S_{10} \cup S_{01} \cup S_{00}$, and Y_{n1} and Y_{n2} are independent,

$$\begin{aligned} AG &\geq P\left(\begin{array}{l} b_0 \cdot a^{Z(Y_{n1})+Z(Y_{n2})} + b_1 \cdot a^{Z(Y_{n1})} + b_2 \cdot a^{Z(Y_{n2})} \\ \geq\ b_0 \cdot a^{Z_1(\tilde{Y}_n)+Z_2(\tilde{Y}_n)} + b_1 \cdot a^{Z_1(\tilde{Y}_n)} + b_2 \cdot a^{Z_2(\tilde{Y}_n)} \end{array} \middle| \ Y_n \in S, \tilde{Y}_n \in S_{10} \right) \\ &\geq P(Z(Y_{n1}) \geq Z_1(\tilde{Y}_n),\ Z(Y_{n2}) \geq Z_2(\tilde{Y}_n)|\ Y_n \in S, \tilde{Y}_n \in S_{10}) \\ &= P(Z(Y_{n1}) \geq Z_1(\tilde{Y}_n)|\ Y_n \in S, \tilde{Y}_n \in S_{10}) \cdot P(Z(Y_{n2}) \geq Z_2(\tilde{Y}_n)|\ Y_n \in S, \tilde{Y}_n \in S_{10}) \\ &= P(I(\tilde{Y}_{n1}) \geq I(Y_{n1})\ |Y_n \in S, \tilde{Y}_n \in S_{10}) \cdot P(I(\tilde{Y}_{n2}) \geq I(Y_{n2})\ |Y_n \in S, \tilde{Y}_n \in S_{10}) \\ &\geq \frac{1}{2} \cdot P(I(\tilde{Y}_{n2}) \geq I(Y_{n2})\ |Y_n \in S, \tilde{Y}_n \in S_{10}). \end{aligned}$$

From (2.5),

$$AP \leq C \cdot P(I(\tilde{Y}_{n2}) \geq I(Y_{n2})\ |Y_n \in S, \tilde{Y}_n \in S_{10}).$$

Therefore,

$$\begin{aligned} \frac{AP}{AG} &\leq \frac{C \cdot P(I(\tilde{Y}_{n2}) \geq I(Y_{n2})\ |Y_n \in S, \tilde{Y}_n \in S_{10})}{\frac{1}{2} \cdot P(I(\tilde{Y}_{n2}) \geq I(Y_{n2})\ |Y_n \in S, \tilde{Y}_n \in S_{10})} \\ &= 2 \cdot C. \end{aligned}$$

□

2.3.2 Demonstration of the Three ROC Curves for Finite n

Within the problem setting in Section 2.3.1, we did a simple experiment to compare the empirical ROC curves for the three models: the optimal model, the parts model, and the universal-null model. It also shows how ROC curves vary with respect to the different image resolutions, i.e., n.

We gave a prior distribution of X: $P(X = (1,1)) = 0.09, P(X = (1,0)) = P(X = (0,1)) = 0.21, P(X = (0,0)) = 0.49$. We took $\epsilon = 0.3$ for the superposed noise.

We generated $N = 100,000$ random samples, and calculated empirical ROC curves, for $n = 3, 6, 10, 18$. In Figure 2.3, we plotted four figures comparing the three ROC curves, where each figure corresponds to a particular n. Solid line represents the ROC curve for the parts model, dashed line represents the one for the universal-null model, while dotted line represents the one for the optimal model. As shown in the experiment, when n is small (n=3 or 6), the universal-null ROC curve can be better than or cross the parts ROC curve. But when n is large ($n \geq 10$) and as n gets larger, the parts model gives strictly better ROC curve than the universal-null model, and it merges to the optimal ROC curve.

2.3.3 Generalization

In Section 2.3.1, Theorem 1 was established based on the case that X is composed of two parts. Actually, Theorem 1 remains true when X is composed of more than two parts. Without loss of generality, assume that X contains m parts. The problem setting is similar: $X = (X_1, X_2, \cdots, X_m)$, where $\{X_i\}_{i=1}^{m}$ takes binary value 0 or 1. $Y_n = f_n(X) + Z_n = (f_n(X_1), f_n(X_2), \cdots, f_n(X_m)) + Z_n$, where $f_n(X)$ is obtained by repeating X_1 n times, followed by repeating X_2 n times, ..., followed by repeating X_m n times. For example, if $m = 3, n = 2, X = (1, 0, 1)$, we will have $f_n(X) = (1, 1, 0, 0, 1, 1)$.

Z_n represents the superposed random noise. Z_n is a vector with the same length as $f_n(X)$, containing $m \cdot n$ i.i.d Bernoulli(ϵ) random variables, $1 < \epsilon < 0.5$. So $Z_n = (z_1, \ldots, z_{mn})$, and $P(z_i = 1) = \epsilon, P(z_i = 0) = 1 - \epsilon$. The addition between $f_n(X)$ and Z_n follows the rule of binary summation. For example, if $m = 3, n = 2, f_n(X) = (1, 1, 0, 0, 1, 0), Z_n = (1, 0, 0, 1, 1, 0)$, Y_n will be $(0, 1, 0, 1, 0, 0)$. Our task is this: Given a fixed m, observing Y_n, make a decision whether Y_n is derived from X=(1,1,...,1) or not.

To solve the problem, we can formalize the parts model and the universal-null model in almost the same manner as before. And again by Large Deviation Theory,

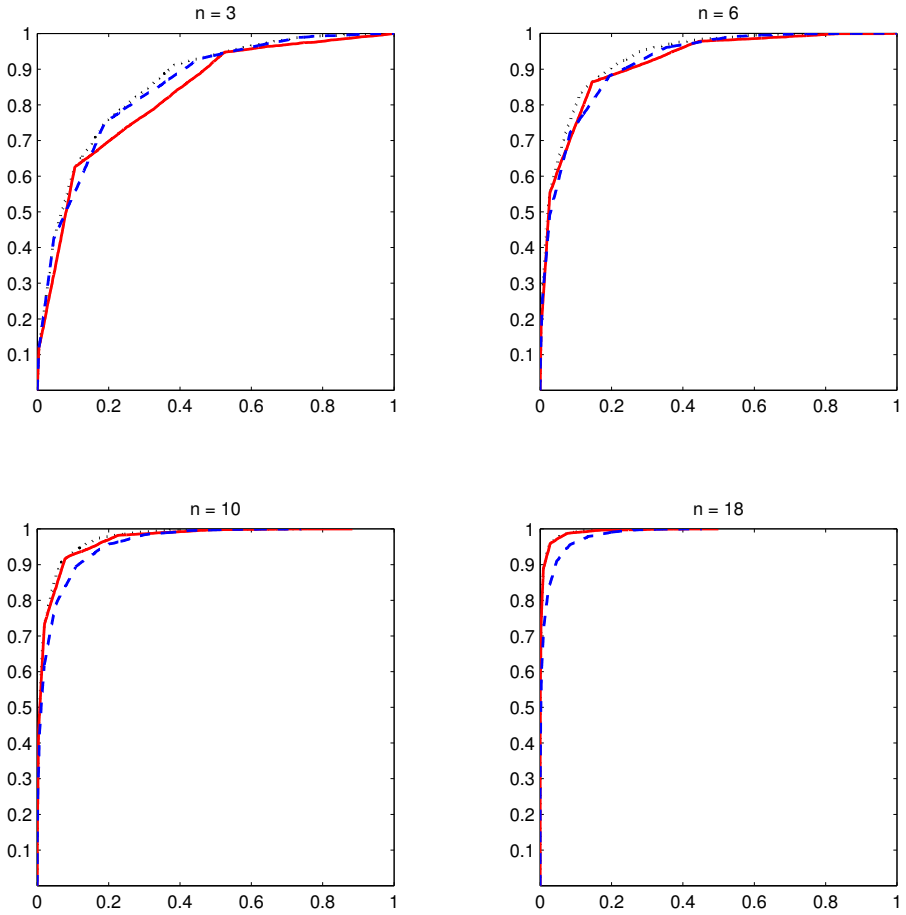

Figure 2.3: Comparison of empirical ROC curves of the three models. The upper left panel, the upper right panel, the lower left panel, and the lower right panel show the comparison results associated with n=3, 6, 9, and 12 respectively. The dotted lines represent the ROC curves produced by the optimal ROC curve given by the Neyman-Pearson Lemma; The solid lines represent the ROC curves produced by the parts model; The dashed lines represent the ROC curves produced by the universal-null model.

we can achieve the same asymptotic result that the ratio between two AAROCCs of the parts model and the universal-null model goes to zero exponentially, as n goes to infinity.

2.3.4 Note

Theorem 1 does not hold if the limit condition is $\epsilon \to 0$ instead of $n \to \infty$. Since as $\epsilon \to 0$,

$$\begin{aligned} AP_{10} = AP_{01} &= C_1 \cdot P(I(\tilde{Y}_{n2}) \geq I(Y_{n2}) \mid Y_n \in S, \tilde{Y}_n \in S_{10}) \\ &= C_1 \cdot P(\sum_{i=1}^{n}(V_i + Q_i) \geq n\ ) \\ &= C_1 \cdot O(\epsilon^n(1-\epsilon)^n) \\ AW_{10} = AW_{01} &= P(\sum_{i=1}^{n}[(V_i + Q_i) + (U_i - G_i)] \geq n\ ) \\ &= O(\epsilon^n(1-\epsilon)^{3n}) \end{aligned}$$

(See the Appendix for detailed derivation of the big O approximations above.) Since $AP_{00} \leq AP_{10}, AW_{00} \leq AW_{10}$, there exists $b > a > 0$ such that

$$\frac{AP}{AW} \in [a \cdot \frac{AP_{10}}{AW_{10}}, b \cdot \frac{AP_{10}}{AW_{10}}].$$

Note that

$$\frac{AP_{10}}{AW_{10}} \to C, \quad \text{as } \epsilon \to 0.$$

Hence $\frac{AP}{AW}$ is bounded by positive numbers, as $\epsilon \to 0$.

2.4 Comparison of ROC Curves of the Optimal Model, the Parts Model and the Universal-Null Model within a Hierarchical Setting

Consider a simple hierarchical tree structure with only three nodes, γ, α, β. The node γ is the parent node, pointing to two child nodes α and β. The states of brick γ, α, and β are donated as x^γ, x^α, x^β, where $x^\gamma \in \{0,1\}$, $x^\alpha \in \{0,1,\ldots,n^\alpha\}$, and $x^\beta \in \{0,1,\ldots,n^\beta\}$. From here through out this chapter, we will use $X^\gamma, X^\alpha, X^\beta$ to represent the random variables, while using $x^\gamma, x^\alpha, x^\beta$ to represent the realization of these three random variables. We generate image data Y in the following three steps. First, we randomly select X^γ according to a Bernoulli(p) distribution, where $p \in (0,1)$. Second, we generate X^α and X^β given X^γ, as follows:

- If $X^\gamma = 1$, we generate X^α and X^β according to a joint conditional distribution $P(X^\alpha, X^\beta | X^\gamma = 1)$. We require that $P(X^\alpha = 0 \text{ or } X^\beta = 0 | X^\gamma = 1) = 0$, i.e. both X^α and X^β have to be positive when $X^\gamma = 1$.

- If $X^\gamma = 0$, we generate X^α and X^β independently, i.e. $P(X^\alpha, X^\beta | X^\gamma = 0) = P(X^\alpha | X^\gamma = 0) \cdot P(X^\beta | X^\gamma = 0)$. We require that $P(X^\alpha = x^\alpha | X^\gamma = 0) > 0$ and $P(X^\beta = x^\beta | X^\gamma = 0) > 0$, $\forall x^\alpha, x^\beta$.

Finally, we generate image Y given x^α and x^β, according to $P(Y | x^\alpha, x^\beta)$. See Figure 2.2B for a graphical representation.

The task is to decide whether $X^\gamma = 1$ or not, given image data Y.

To ease the notations, we define the testing rules associated with three models (the optimal model, the parts model, and the universal-null model) as follows: Given a threshold c,

$$X^\gamma = 1 \quad \text{if} \quad F_o(Y) \geq c, \quad \forall o \in \{g, p, w\},$$

where "g," "p," and "w" stands for the optimal model, the parts model, and the universal-null model respectively. With this hierarchical problem setting, we formalize F_g, F_p, and F_w in a slightly different way from Section 2.3:

optimal model:

$$F_g(Y) = \frac{P(Y|X^\gamma = 1)}{P(Y|X^\gamma = 0)} = \frac{\sum_{x^\alpha>0, x^\beta>0} P(Y|x^\alpha, x^\beta)P(x^\alpha, x^\beta|X^\gamma = 1)}{\sum_{x^\alpha, x^\beta} P(Y|x^\alpha, x^\beta)P(x^\alpha, x^\beta|X^\gamma = 0)}$$

parts model:

$$F_p(Y) = \min \begin{cases} \frac{\sum_{x^\alpha>0} P(Y|x^\alpha)P(x^\alpha|X^\alpha>0)}{P(Y|X^\alpha=0)}, & \text{denoted by } F_p^\alpha(Y) \\ \frac{\sum_{x^\beta>0} P(Y|x^\beta)P(x^\beta|X^\beta>0)}{P(Y|X^\beta=0)}, & \text{denoted by } F_p^\beta(Y) \\ \frac{\sum_{x^\alpha>0, X^\beta>0} P(Y|x^\alpha,x^\beta)P(x^\alpha,x^\beta|X^\gamma=1)}{\sum_{x^\alpha>0,x^\beta>0} P(Y|x^\alpha,x^\beta)P(x^\alpha,x^\beta|X^\gamma=0)}, & \text{denoted by } \hat{F}_p(Y) \end{cases}$$

Or, alternatively,

$$F_p(Y) = \min \begin{cases} \frac{\max_{x^\alpha>0} P(Y|x^\alpha)P(x^\alpha|X^\alpha>0)}{P(Y|X^\alpha=0)}, & \text{denoted by } F_p^\alpha(Y) \\ \frac{\max_{x^\beta>0} P(Y|x^\beta)P(x^\beta|X^\beta>0)}{P(Y|X^\beta=0)}, & \text{denoted by } F_p^\beta(Y) \\ \frac{\max_{x^\alpha>0, X^\beta>0} P(Y|x^\alpha,x^\beta)P(x^\alpha,x^\beta|X^\gamma=1)}{\max_{x^\alpha>0,x^\beta>0} P(Y|x^\alpha,x^\beta)P(x^\alpha,x^\beta|X^\gamma=0)}, & \text{denoted by } \hat{F}_p(Y) \end{cases}$$

universal-null model:

$$F_w(Y) \;=\; \frac{P(Y|X^\gamma = 1)}{P(Y|X^\alpha = 0, X^\beta = 0)} \;=\; \frac{\sum_{x^\alpha>0, x^\beta>0} P(Y|x^\alpha, x^\beta)P(x^\alpha, x^\beta|X^\gamma = 1)}{P(Y|X^\alpha = 0, X^\beta = 0)}$$

Equivalently (since the $F_w(Y)$ below is an increasing function of the $F_w(Y)$ above),

$$F_w(Y) = \frac{\sum_{x^\alpha>0,x^\beta>0} P(Y|x^\alpha, x^\beta)P(x^\alpha, x^\beta|X^\gamma = 1)}{\sum_{x^\alpha>0,x^\beta>0} P(Y|x^\alpha, x^\beta)P(x^\alpha, x^\beta|X^\gamma = 1) + P(Y|X^\alpha = 0, X^\beta = 0)}$$

2.4.1 Theoretical Comparison of Asymptotic ROC Curves

We want to study mathematically the ROC performance of the three models within this hierarchical setting. A few notations need to be defined for later use. Let $\{(i_k, j_k)\}$ be a re-ordered sequence of $\{1, 2, \ldots, n^\alpha\} \times \{1, 2, \ldots, n^\beta\}$, such that $\forall k \in \{1, 2, \ldots, (n^\alpha \cdot n^\beta - 1)\}$,

$$\theta_k \geq \theta_{k+1},$$

where θ_k is defined as

$$\theta_k = \frac{P(X^\alpha = i_k, X^\beta = j_k|X^\gamma = 1)}{P(X^\alpha = i_k, X^\beta = j_k|X^\gamma = 0)}, \quad \forall k \in \{1, 2, \ldots, n^\alpha \cdot n^\beta\}. \tag{2.9}$$

The binary decision problem within this hierarchical setting is called "trivial" if

$$\theta_1 = \theta_2 = \ldots = \theta_k = \ldots = \theta_{n^\alpha \cdot n^\beta - 1} = \theta_{n^\alpha \cdot n^\beta}.$$

By this definition, $n^\alpha \cdot n^\beta = 1$ automatically falls into the category of "trivial." This decision problem is called "trivial" if there exist $k_1, k_2 \in \{1, 2, \ldots, n^\alpha \cdot n^\beta\}$ such that $\theta_{k_1} \neq \theta_{k_2}$. The following theorem addresses the non-trivial case. The trivial case essentially reduces to Theorem 1, as discussed in Section 2.5.

Theorem 2. *As resolution of data Y goes to ∞, the ROC curves produced by the optimal model and the parts model merge together, and strictly better than the one produced by the universal-null model for non-trivial cases.*

Proof. For all $c \in [0, \infty]$, let

$$\begin{aligned} O &= \{Y : \text{ generated by } X^\gamma = 1\}, \\ B &= \{Y : \text{ generated by } X^\gamma = 0\}, \\ D &= \{Y : F(Y) \geq c\}. \end{aligned}$$

With the definitions above, TPR and FPR associated with threshold c become

$$\begin{aligned} \text{TPR}(c) &= \frac{|D \cap O|}{|O|}, \\ \text{FPR}(c) &= \frac{|D \cap B|}{|B|}. \end{aligned}$$

For the time being, we assume that the decision problem is non-trivial. Let

$$\begin{aligned} S^{00} &= \{Y : \text{ generated by } X^\alpha = X^\beta = 0\}, \\ S^{01} &= \{Y : \text{ generated by } X^\alpha = 0, X^\beta > 0\}, \\ S^{10} &= \{Y : \text{ generated by } X^\alpha > 0, X^\beta = 0\}, \\ S_k &= \{Y : \text{ generated by } X^\alpha = i_k > 0, X^\beta = j_k > 0\}, \quad \text{for } k \in \{1, 2, \ldots, (n^\alpha \cdot n^\beta)\}. \end{aligned}$$

Claims without proof: As resolution n goes to infinity,

(1) $F_p(Y) \longrightarrow 0, \ \forall\ Y \in S^{00} \cup S^{01} \cup S^{10}$.

(2) $F_p(Y) \longrightarrow \theta_k, \ \forall\ Y \in S_k, \ \forall k \in \{1, 2, \ldots, (n^\alpha \cdot n^\beta)\}$.

This indicates that, as n goes to infinity, all the $F_p(Y)$s will concentrate on finitely many values: $\{0\} \cup \{\theta_k : k \in \{1, 2, \ldots, (n^\alpha \cdot n^\beta)\}$. Note that $\theta_k \geq \theta_{k+1}$, hence as the threshold c decreases from ∞ to 0, we stay at the origin first, then all the Ys in S_1 pass the test when $c = \theta_1$, followed by all the Ys in S_2 passing the test when $c = \theta_2$, ..., followed by all the Ys in $S_(n^\alpha \cdot n^\beta)$ pass test when $c = \theta_(n^\alpha \cdot n^\beta)$, and finally

all the Ys pass the test when $c = 0$. Therefore,

$$
\begin{aligned}
\text{when } c > \theta_1, \quad & \mathrm{TPR}_0 \triangleq \mathrm{TPR}(c) = 0, \\
& \mathrm{FPR}_0 \triangleq \mathrm{FPR}(c) = 0, \\
\text{when } \theta_2 < c \leq \theta_1, \quad & \mathrm{TPR}_1 \triangleq \mathrm{TPR}(c) = \frac{|S_1 \cap O|}{|O|} = P(X^\alpha = i_1, X^\beta = j_1 | X^\gamma = 1), \\
& \mathrm{FPR}_1 \triangleq \mathrm{FPR}(c) = \frac{|S_1 \cap B|}{|B|} = P(X^\alpha = i_k, X^\beta = j_k | X^\gamma = 0), \\
& \cdots\cdots\cdots\cdots \\
\text{when } \theta_{k+1} < c \leq \theta_k, \quad & \mathrm{TPR}_k \triangleq \mathrm{TPR}(c) = \frac{|(\cup_{l=1}^{k} S_l) \cap O|}{|O|} \\
& \qquad = \sum_{l=1}^{k} P(X^\alpha = i_l, X^\beta = j_l | X^\gamma = 1), \\
& \mathrm{FPR}_k \triangleq \mathrm{FPR}(c) = \frac{|(\cup_{l=1}^{k} S_l) \cap B|}{|B|} \\
& \qquad = \sum_{l=1}^{k} P(X^\alpha = i_l, X^\beta = j_l | X^\gamma = 0), \\
& \cdots\cdots\cdots\cdots \\
\text{when } 0 < c \leq \theta_{n^\alpha \cdot n^\beta}, \quad & \mathrm{TPR}_{n^\alpha \cdot n^\beta} \triangleq \mathrm{TPR}(c) = \frac{|(\cup_{l=1}^{n^\alpha \cdot n^\beta} S_l) \cap O|}{|O|} = 1, \\
& \mathrm{FPR}_{n^\alpha \cdot n^\beta} \triangleq \mathrm{FPR}(c) = \frac{|(\cup_{l=1}^{n^\alpha \cdot n^\beta} S_l) \cap B|}{|B|} \\
& \qquad = P(X^\alpha > 0, X^\beta > 0 | X^\gamma = 0), \\
\text{when } c = 0, \quad & \mathrm{TPR}_{n^\alpha \cdot n^\beta + 1} \triangleq \mathrm{TPR}(c) = 1, \\
& \mathrm{FPR}_{n^\alpha \cdot n^\beta + 1} \triangleq \mathrm{FPR}(c) = 1.
\end{aligned}
$$

Hence, as c decreases from ∞ to 0, we get $(n^\alpha \cdot n^\beta + 2)$ points $\{(\mathrm{FPR}_k, \mathrm{TPR}_k)\}_{k=0}^{n^\alpha \cdot n^\beta + 1}$ in the X-Y plane, if we connect these points one by one, we get the asymptotic ROC curve for the parts model. Note that this piecewise linear asymptotic ROC is convex, due to the fact that $\theta_k \geq \theta_{k+1}$. The reason is that $\forall k \in \{1, 2, \ldots, n^\alpha \cdot n^\beta\}$, θ_k is just the slope of the k^{th} linear piece of the asymptotic ROC curve, where "k^{th}" is counted from the point (0,0) to the point (1,1) along the asymptotic ROC curve.

All the arguments above work the same for the optimal model, hence in the limit, the parts model and the optimal model share the same ROC curve.

For the universal-null model: Denote

$$\begin{aligned} S^{00} &= \{Y : \text{ generated by } X^\alpha = X^\beta = 0\}, \\ S^{01} &= \{Y : \text{ generated by } X^\alpha = 0, X^\beta > 0\}, \\ S^{10} &= \{Y : \text{ generated by } X^\alpha > 0, X^\beta = 0\}, \\ S &= \{Y : \text{ generated by } X^\alpha > 0, X^\beta > 0\}. \end{aligned}$$

Claims without proof: As resolution n goes to infinity,

(1) $F_w(Y) \longrightarrow 0, \ \forall\ Y \in S^{00}$.

(2) $F_w(Y) \longrightarrow \infty, \ \forall\ Y \in S$.

(3) $F_w(Y_1) < F_w(Y_2) < F_w(Y_3), \ \forall\ Y_1 \in S^{00}, Y_2 \in S^{01} \cup S^{10}, Y_3 \in S$.

For $Y \in S^{01} \cup S^{10}$, we do not have a clear statement about the convergence of $F_w(Y)$, but the three claims above are enough to portrait the asymptotic ROC curve for the universal-null model. As threshold c decreases from ∞ to 0, we stay at the origin first, then when $c < \infty$ and c keeps decreasing, all the Ys in S will pass the test, earlier than any Y from the set $S^{01} \cup S^{10}$ passes the test. Let $M = \max_{Y \in S^{01} \cup S^{10}} F_w(Y)$.

Asymptotically we have,

$$
\begin{aligned}
\text{when } c = \infty, \quad & \mathrm{TPR}(c) = 0, \\
& \mathrm{FPR}(c) = 0, \\
\text{when } M < c < \infty, \quad & \mathrm{TPR}(c) = \frac{|S \cap O|}{|O|} = P(X^\alpha > 0, X^\beta > 0 | X^\gamma = 1) = 1, \\
& \mathrm{FPR}(c) = \frac{|S \cap B|}{|B|} = P(X^\alpha > 0, X^\beta > 0 | X^\gamma = 0), \\
\text{when } 0 < c \leq M, \quad & \mathrm{TPR}(c) \geq \frac{|S \cap O|}{|O|} = P(X^\alpha > 0, X^\beta > 0 | X^\gamma = 1) = 1, \\
& (\text{hence, here, } \mathrm{TPR}(c) = 1) \\
& \mathrm{FPR}(c) \geq \frac{|S \cap B|}{|B|} = P(X^\alpha > 0, X^\beta > 0 | X^\gamma = 0), \\
\text{when } c = 0, \quad & \mathrm{TPR}(c) = 1, \\
& \mathrm{FPR}(c) = 1.
\end{aligned}
$$

Thus, as c decreases from ∞ to 0, only three points are distinguishable and matter for the ROC curve: (0,0), ($P(X^\alpha > 0, X^\beta > 0 | X^\gamma = 0)$,1), and (1,1). When we connect them one by one, we get the asymptotic ROC curve for the universal-null model, as the resolution n goes to infinity.

From the computation above, we can conclude that

(1) As the resolution n goes to infinity, the ROC curves of the parts model and the optimal model merge together. Their asymptotic ROC curves are convex, piecewise linear, and uniquely determined by the $(n^\alpha \cdot n^\beta + 2)$ points $\{(\mathrm{FPR}_k, \mathrm{TPR}_k)\}_{k=0}^{n^\alpha \cdot n^\beta + 1}$.

(2) As the resolution n goes to infinity, the ROC curve of the universal-null model converges to a convex and piecewise linear curve, uniquely determined by the three points (0,0), ($P(X^\alpha > 0, X^\beta > 0 | X^\gamma = 0)$,1), and (1,1).

(3) The asymptotic ROC curve of the parts model is no worse than the asymptotic

ROC curve of the universal-null model. For the non-trivial case that $n^\alpha \cdot n^\beta > 1$ and $\exists\, k_1, k_2 \in \{1, 2, \ldots, n^\alpha \cdot n^\beta\}$ such that $\theta_{k_1} \neq \theta_{k_2}$, the asymptotic ROC curve of the parts model is strictly better than that of the universal-null model. For the trivial case that $n^\alpha = n^\beta = 1$ or $\theta_1 = \theta_2 = \ldots = \theta_{n^\alpha \cdot n^\beta}$, the asymptotic ROC curves of the optimal model, the parts model, and the universal-null model converge to the same convex and piecewise linear curve, that is uniquely determined by the three points (0,0), $(P(X^\alpha > 0, X^\beta > 0 | X^\gamma = 0)$,1), and (1,1).

□

2.4.2 Demonstration of the Three ROC Curves for Finite n Within the Hierarchical Setting

Consider a simple thought experiment where $n^\alpha = 2, n^\beta = 2$. The probability parameters were set as follows.

$$P(X^\gamma = 1) = 0.09,\ P(X^\gamma = 0) = 0.91.$$
$$P(X^\alpha = 1, X^\beta = 1 | X^\gamma = 1) = 0.4, \quad P(X^\alpha = 1, X^\beta = 2 | X^\gamma = 1) = 0.1,$$
$$P(X^\alpha = 2, X^\beta = 1 | X^\gamma = 1) = 0.1, \quad P(X^\alpha = 2, X^\beta = 2 | X^\gamma = 1) = 0.4.$$
$$P(X^\alpha = 0 | X^\gamma = 0) = 0.2, \quad P(X^\alpha = 1 | X^\gamma = 0) = 0.3, \quad P(X^\alpha = 2 | X^\gamma = 0) = 0.5,$$
$$P(X^\beta = 0 | X^\gamma = 0) = 0.3, \quad P(X^\beta = 1 | X^\gamma = 0) = 0.2, \quad P(X^\beta = 2 | X^\gamma = 0) = 0.5.$$

Given X^α and X^β, we gave the data model a similar form as in Section 2.3.2, $Y_n = (f_n(X^\alpha) + Z_n^\alpha, f_n(X^\beta) + Z_n^\beta)$, where $f_n(X^\alpha)$ is obtained by repeating X^α n times, while $f_n(X^\beta)$ is obtained by repeating X^β n times. Z_n^α and Z_n^β represent the added random noise. Z_n^α is a vector with length n, containing n i.i.d random variable z, $P(z = 0) = 1 - \epsilon, P(z = k) = \epsilon / n^\alpha, \forall k \in \{1, \ldots, n^\alpha\}$. Z_n^β is a vector with length n, containing n i.i.d random variable z', $P(z' = 0) = 1 - \epsilon, P(z' = k) = \epsilon / n^\beta, \forall k \in$

$\{1, \ldots, n^\beta\}$. The addition of the i^{th} element of $f_n(X^\alpha)$ (call it "a") and the i^{th} element of Z_n^α (call it "b") is defined as the remainder of (a+b) divided by $(n^\alpha+1)$. Note that the value of this remainder is in the set $\{0, 1, \ldots, n^\alpha\}$. The addition of the i^{th} element of $f_n(X^\beta)$ and the i^{th} element of Z_n^β is defined similarly, and takes values in the set $\{0, 1, \ldots, n^\beta\}$. For example, if $n = 2, X^\alpha = 1, X^\beta = 2, Z_n^\alpha = (0, 2), Z_n^\beta = (1, 0)$, we will have $Y_n = ((1, 1) + (0, 2), (2, 2) + (1, 0)) = (1, 0, 0, 2)$. Our task is this: Given Y_n, make a decision whether $X^\gamma = 1$ or not. In this experiment, we picked $\epsilon = 0.3$.

Let $(i_1, i_2, i_3, i_4) = (1, 2, 2, 1), (j_1, j_2, j_3, j_4) = (1, 2, 1, 2)$, and define

$$\theta_1 = \frac{P(X^\alpha = i_1, X^\beta = j_1 | X^\gamma = 1)}{P(X^\alpha = i_1, X^\beta = j_1 | X^\gamma = 0)} = 6.6667,$$
$$\theta_2 = \frac{P(X^\alpha = i_2, X^\beta = j_2 | X^\gamma = 1)}{P(X^\alpha = i_2, X^\beta = j_2 | X^\gamma = 0)} = 1.6,$$
$$\theta_3 = \frac{P(X^\alpha = i_3, X^\beta = j_3 | X^\gamma = 1)}{P(X^\alpha = i_3, X^\beta = j_3 | X^\gamma = 0)} = 1,$$
$$\theta_4 = \frac{P(X^\alpha = i_4, X^\beta = j_4 | X^\gamma = 1)}{P(X^\alpha = i_4, X^\beta = j_4 | X^\gamma = 0)} = 0.6667.$$

We can see that $\theta_k \geq \theta_{k+1}, \ \forall k \in \{1, 2, 3\}$. The asymptotic theoretical ROC curve of the optimal model and the parts model is determined by the following six points

$\{(\mathrm{FPR}_k, \mathrm{TPR}_k)\}_{k=0}^{5}$:

$$
\begin{aligned}
\text{when } c > \theta_1, \quad & \mathrm{TPR}_0 = 0, \\
& \mathrm{FPR}_0 = 0, \\
\text{when } \theta_2 < c \le \theta_1, \quad & \mathrm{TPR}_1 = P(X^\alpha = 1, X^\beta = 1 | X^\gamma = 1) = 0.4, \\
& \mathrm{FPR}_1 = P(X^\alpha = 1, X^\beta = 1 | X^\gamma = 0) = 0.06, \\
\text{when } \theta_3 < c \le \theta_2, \quad & \mathrm{TPR}_2 = \sum_{l=1}^{2} P(X^\alpha = i_l, X^\beta = j_l | X^\gamma = 1) = 0.8, \\
& \mathrm{FPR}_2 = \sum_{l=1}^{2} P(X^\alpha = i_l, X^\beta = j_l | X^\gamma = 0) = 0.31, \\
\text{when } \theta_4 < c \le \theta_3, \quad & \mathrm{TPR}_3 = \sum_{l=1}^{3} P(X^\alpha = i_l, X^\beta = j_l | X^\gamma = 1) = 0.9, \\
& \mathrm{FPR}_3 = \sum_{l=1}^{3} P(X^\alpha = i_l, X^\beta = j_l | X^\gamma = 0) = 0.41, \\
\text{when } 0 < c \le \theta_4, \quad & \mathrm{TPR}_4 = \sum_{l=1}^{4} P(X^\alpha = i_l, X^\beta = j_l | X^\gamma = 1) = 1, \\
& \mathrm{FPR}_4 = \sum_{l=1}^{4} P(X^\alpha = i_l, X^\beta = j_l | X^\gamma = 0) = 0.56, \\
\text{when } c = 0, \quad & \mathrm{TPR}_5 = 1, \\
& \mathrm{FPR}_5 = 1.
\end{aligned}
$$

We want to compare the empirical ROC curves of three models for different ns, and to examine whether the empirical ROC curves produced by the optimal model and the parts model will converge to the asymptotic theoretical ROC curve determined by the six points above.

To get the empirical ROC curves, we pick four different ns: $n = 2, 5, 10, 50$. For each n, we randomly generated 100000 random Y_n according the generating procedure described earlier, and calculated the empirical ROC curves. For each n, we plotted the three ROC curves in one single figure. Within the same figure we also

plotted the six points $\{(\mathrm{FPR}_k, \mathrm{TPR}_k)\}_{k=0}^{5}$ for reference. Figure 2.4 shows the four cases associated with four different ns. As we can see, as n is large enough, the ROC curves of the optimal model and the parts model merge together, and are strictly better than the ROC curve produced by the universal-null model. Also, when n is large (=50), the empirical ROC curves of the optimal model and the parts model merge to the asymptotic theoretical ROC curve determined by $\{(\mathrm{FPR}_k, \mathrm{TPR}_k)\}_{k=0}^{5}$. In addition, when n is large, the ROC curve of the universal-null model also converges in the same way as we concluded in Theorem 2.

2.5 Discussion

The Neyman-Pearson Lemma gives us the optimal model for hypothesis testing in object detection or recognition. However, this optimal model is impractical. Not actually impractical for the thought experiment, but impractical in anything resembling a real experiment, with multiple parts, multiple objects, variable poses, and so on. Hence we need to seek alternative models.

We studied and compared three models in this chapter – the optimal model, the parts model, and the universal-null model. We have shown for both the non-hierarchical setting and the hierarchical setting that the parts model has better ROC performance than the universal-null model and has comparable performance with the optimal model, at least when n is large. From the testing rule associated with the parts model, it seems that, at least in the thought experiments, the parts model does not save us a dramatic amount of computation, compared to the optimal model. However, it does in practice, if we follow a sequence of testings for the parts model. For example, within the setting of Theorem 2, we can examine brick α first, and only move forward to examine brick β if $F_p^{\alpha}(Y) \geq c$, and only move forward further to examine brick γ if $F_p^{\beta}(Y) \geq c$. By this way, it decreases the computation to a great extent, compared to the optimal model. This amounts to a simple example

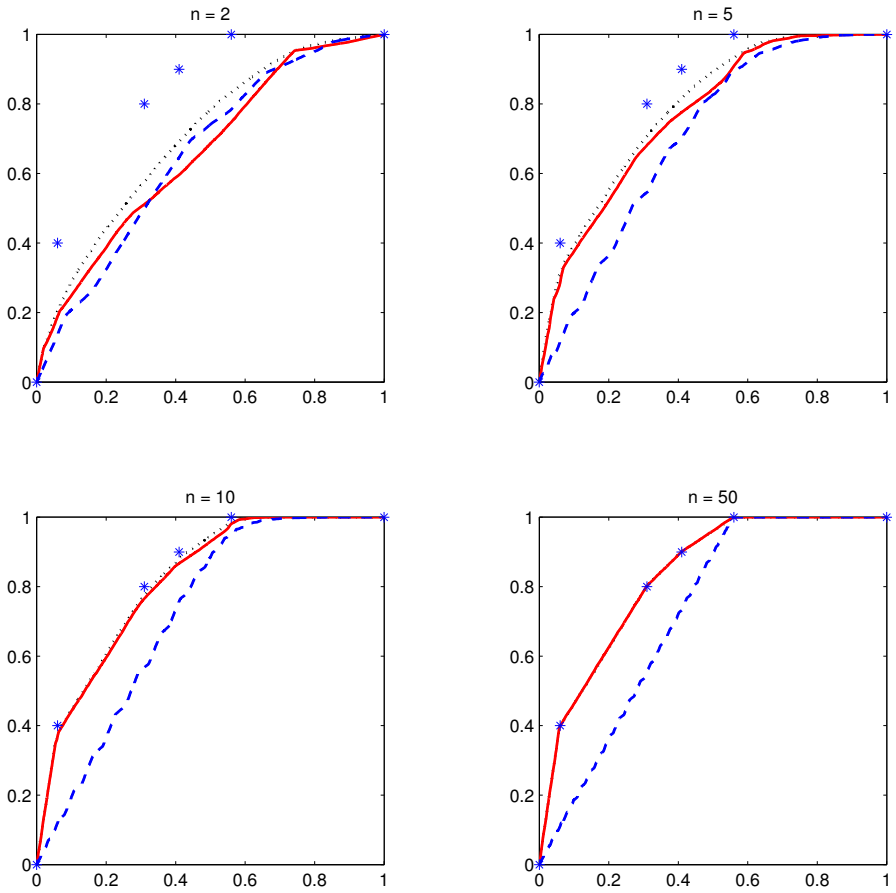

Figure 2.4: Comparison of empirical ROC curves of the three models within the hierarchical setting, for a non-trivial case. The upper left panel, the upper right panel, the lower left panel, and the lower right panel show the comparison results associated with n=2, 5, 10, and 50 respectively. The dotted lines represent the ROC curves produced by the optimal ROC curve given by the Neyman-Pearson Lemma; The solid lines represent the ROC curves produced by the parts model; The dashed lines represent the ROC curves produced by the universal-null model. The six points $\{(\mathrm{FPR}_k, \mathrm{TPR}_k)\}_{k=0}^{5}$ were plotted as six star points (*) on the X-Y plane in each panel.

of the coarse-to-fine algorithm developed in [6, 4, 8].

We introduced two theorems on the asymptotic ROC performance of these three models. Now, the question is how the two theorems, Theorem 1 and Theorem 2, are related with each other.

As we concluded earlier in Theorem 2, the asymptotic ROC curve of the parts model is no worse than the asymptotic ROC curve of the universal-null model. Especially, for the non-trivial cases that $\exists\ k_1, k_2 \in \{1, 2, \ldots, n^\alpha \cdot n^\beta\}$ such that $\theta_{k_1} \neq \theta_{k_2}$, the asymptotic ROC curve of the parts model is strictly better than the asymptotic ROC curve of the universal-null model. This has been demonstrated in the thought experiment earlier. In Theorem 2, for the trivial cases that $\theta_1 = \theta_2 = \ldots = \theta_{n^\alpha \cdot n^\beta}$, the asymptotic ROC curves of the optimal model, the parts model, and the universal-null model converge to the same convex and piecewise linear curve (call it η), that is determined by the three points (0,0), ($P(X^\alpha > 0, X^\beta > 0 | X^\gamma = 0)$,1), and (1,1). In this case, how to compare the ROC performance of the three models? This is where Theorem 1 comes in. Theorem 1 can be adapted to prove that even though the three ROC curves merge together to η, the ROC curve given by the parts model converges to η exponentially faster than the universal-null model. The only difference from the proof of Theorem 1 is that η here is determined by the three points (0,0), ($P(X^\alpha > 0, X^\beta > 0 | X^\gamma = 0)$,1), and (1,1), while the η in Theorem 1 is determined by the three points (0,0), (0,1), and (1,1). The difference between two ηs from two theorems is due to their different problem settings. In Theorem 2, the confusion resulted by the case when $X^\alpha > 0, X^\beta > 0$ can not be eliminated, due to the fact that $P(X^\alpha > 0, X^\beta > 0 | X^\gamma = 0) > 0$. Luckily, this difference does not prevent us to adapt the proof of Theorem 1 to the trivial case of Theorem 2.

To demonstrate the connection between the two theorems, we did a simple experiment for one of the trivial cases, $n^\alpha = n^\beta = 1$, in the hierarchical setting of

Theorem 2:

$$
\begin{aligned}
&P(X^\gamma = 1) = 0.09, P(X^\gamma = 0) = 0.91.\\
&P(X^\alpha = 1, X^\gamma = 1 | X^\gamma = 1) = 1,\\
&P(X^\alpha = 0 | X^\gamma = 0) = 0.5,\ P(X^\alpha = 1 | X^\gamma = 0) = 0.5,\\
&P(X^\beta = 0 | X^\gamma = 0) = 0.5,\ P(X^\beta = 1 | X^\gamma = 0) = 0.5.
\end{aligned}
$$

We generated data Y_n in the same way as in Section 2.4.2 with $\epsilon = 0.3$. Our task is this: Observing Y_n, make a decision whether $X^\gamma = 1$ or not. First we plotted ROC curves for n=2, 5, 10, 50 in Figure 2.5. We can see that as n gets larger and larger, the three empirical ROC curves converge to the same curve η determined by the three points (0,0), ($P(X^\alpha > 0, X^\beta > 0 | X^\gamma = 0) = 0.25$,1), and (1,1). To illustrate better that the ROC curve of the parts model converges to η exponentially faster than that of the universal-null model, we plotted the ROC curves for more ns: n=12, 15, 20, 25, in Figure 2.6.

2.5.1 Appendix

The large O derivation in section 2.3.4:

$$
\begin{aligned}
AP_{10} = AP_{01} &= C_1 \cdot P(I(\tilde{Y}_{n2}) \geq I(Y_{n2}) \mid Y_n \in S, \tilde{Y}_n \in S_{10})\\
&= C_1 \cdot P(\ \sum_{i=1}^{n} (V_i + Q_i) \geq n\ )\\
&= C_1 \cdot \sum_{k=0}^{n} P(\ \sum_{i=1}^{n} (V_i + Q_i) = n + k\ )\\
&= C_1 \cdot \sum_{k=0}^{n} \epsilon^{n+k} (1-\epsilon)^{n-k}.
\end{aligned}
$$

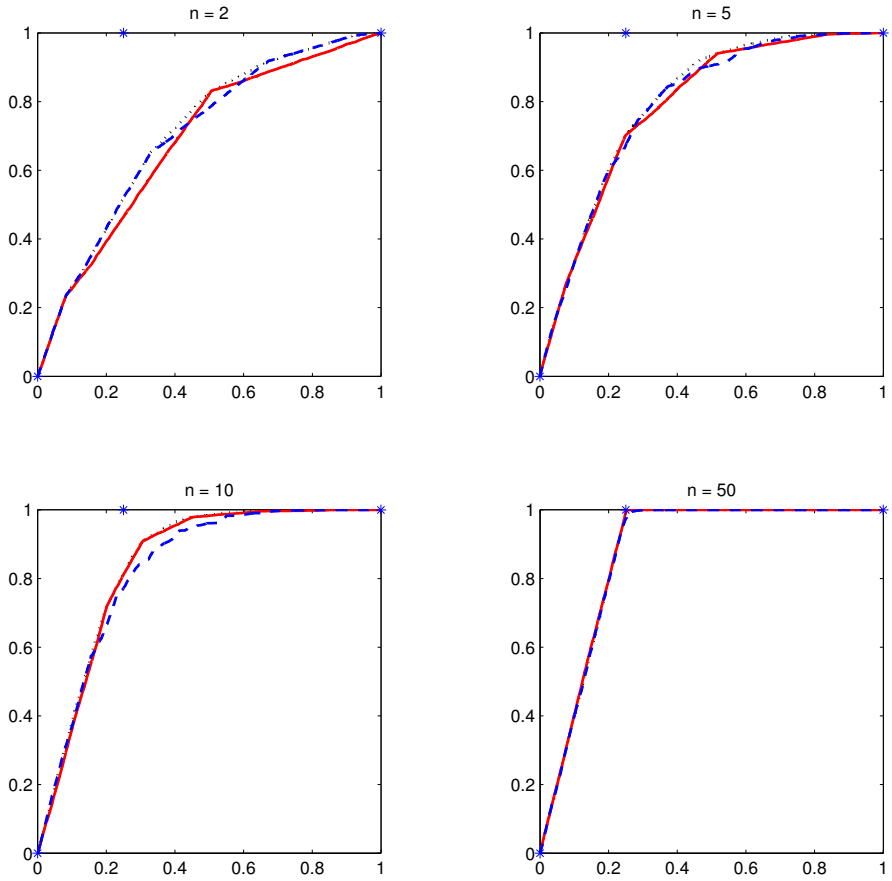

Figure 2.5: Comparison of empirical ROC curves of the three models within the hierarchical setting, for a trivial case. The upper left panel, the upper right panel, the lower left panel, and the lower right panel show the comparison results associated with n=2, 5, 10, and 50 respectively. The dotted lines represent the ROC curves produced by the optimal ROC curve given by the Neyman-Pearson Lemma; The solid lines represent the ROC curves produced by the parts model; The dashed lines represent the ROC curves produced by the universal-null model. The three star points (*) in each panel correspond to points (0,0), ($P(X^{\alpha} > 0, X^{\beta} > 0 | X^{\gamma} = 0) =$ 0.25,1), and (1,1).

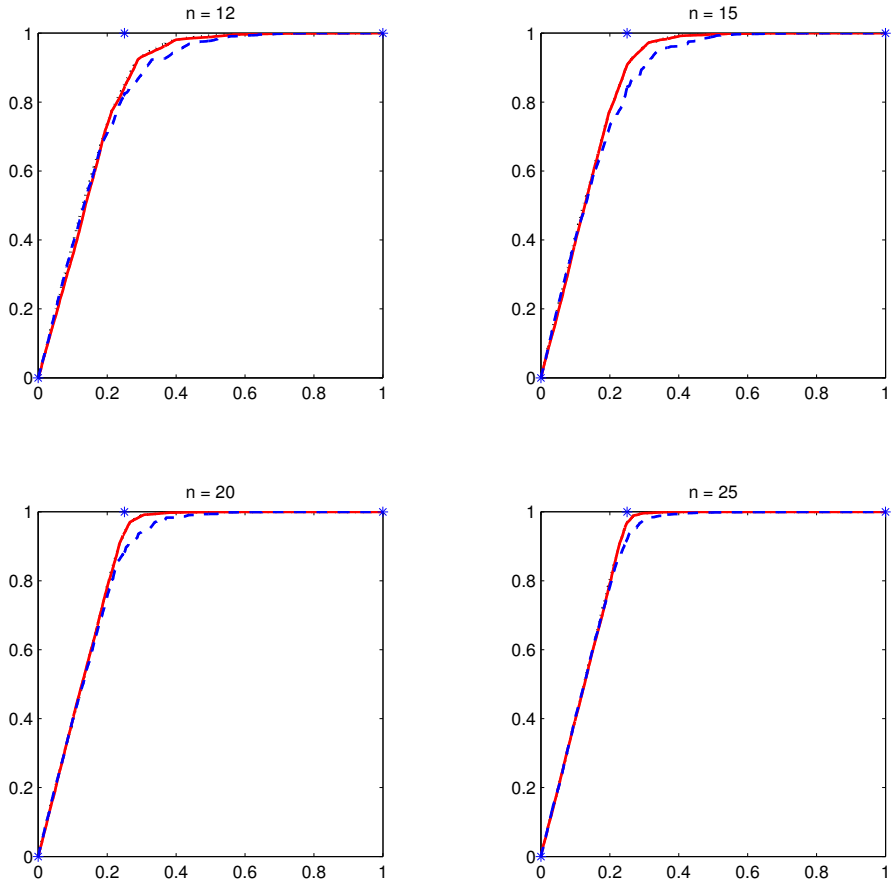

Figure 2.6: Comparison of empirical ROC curves of the three models within the hierarchical setting, for a non-trivial case. The upper left panel, the upper right panel, the lower left panel, and the lower right panel show the comparison results associated with n=12, 15, 20, and 25 respectively. The dotted lines represent the ROC curves produced by the optimal ROC curve given by the Neyman-Pearson Lemma; The solid lines represent the ROC curves produced by the parts model; The dashed lines represent the ROC curves produced by the universal-null model. The three star points (*) in each panel correspond to points (0,0), $(P(X^{\alpha} > 0, X^{\beta} > 0 | X^{\gamma} = 0) = 0.25$,1), and (1,1).

Since $\epsilon < 0.5$, we have $AP_{10} = AP_{01} = C_1 \cdot O(\epsilon^n(1-\epsilon)^n)$.

$$\begin{aligned}
AW_{10} = AW_{01} &= P(\ \sum_{i=1}^{n}[(V_i + Q_i) + (U_i - G_i)] \geq n\ ) \\
&= \sum_{k=0}^{3n} P(\ \sum_{i=1}^{n}[(V_i + Q_i) + (U_i - G_i)] = n + k\ ) \\
&= \sum_{k=0}^{3n}\sum_{j=0}^{n} P(\ \sum_{i=1}^{n}[(V_i + Q_i) + (U_i - G_i)] = n + k,\ G_i = j\ ) \\
&= \sum_{k=0}^{3n}\sum_{j=0}^{n} P(\ \sum_{i=1}^{n} V_i + Q_i + U_i = n + k + j,\ G_i = j\ ) \\
&= \sum_{k=0}^{3n}\sum_{j=0}^{n} \epsilon^{n+2k+j}(1-\epsilon)^{3n-2k-j}
\end{aligned}$$

Since $\epsilon < 0.5$, we have $AW_{10} = AW_{01} = O(\epsilon^n(1-\epsilon)^{3n})$.

Chapter 3

On the Correctness of Compositional Probabilities

Chapter 1 introduced a probabilistic framework for hierarchical generative models, which is composed of a prior distribution on image interpretations and a conditional data distribution on an image given an interpretation. This chapter will focus on the study of the prior distribution, especially to answer the question proposed in Chapter 1, how to perturb a Markovian distribution of interpretations to obtain a non-Markovian distribution that satisfies a set of conditional (attribute) constraints. This question is actually two-fold: First, given the conditional distributions of a set of attribute functions, does there exist any distribution on interpretations that satisfies all of the conditional (attribute) constraints? Second, if there exists such a distribution, how would we achieve it starting with a Markov distribution on interpretations? This Chapter will answer these two questions through two theorems. The first theorem will prove that there exits a distribution on interpretations satisfying a "special" set of conditional constraints. The second theorem will give a complete solution to the second question above. Its proof is general and is independent of the proof for the first theorem.

We will first review the problem setting and some important concepts defined in

Chapter 1. Consider a directed acyclic graph $\mathcal{G}$ defined by

- A vertex for every brick $\alpha \in \mathcal{B}$
- 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 $\{x^\alpha\}_{\alpha \in \mathcal{B}}$ such that $\alpha \in \mathcal{B} \setminus \mathcal{T}$ and $x^\alpha > 0 \Rightarrow x^\beta > 0 \; \forall \beta \in C_{x^\alpha}^\alpha$. ($\mathcal{T}$ is the set of terminal bricks defined in Section 1.3.1.) 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 $x^\alpha > 0$, then an interpretation is a state vector $\vec{x}$ in which the chosen children of every non-terminal on brick are themselves on. (See Figure 3.1.)

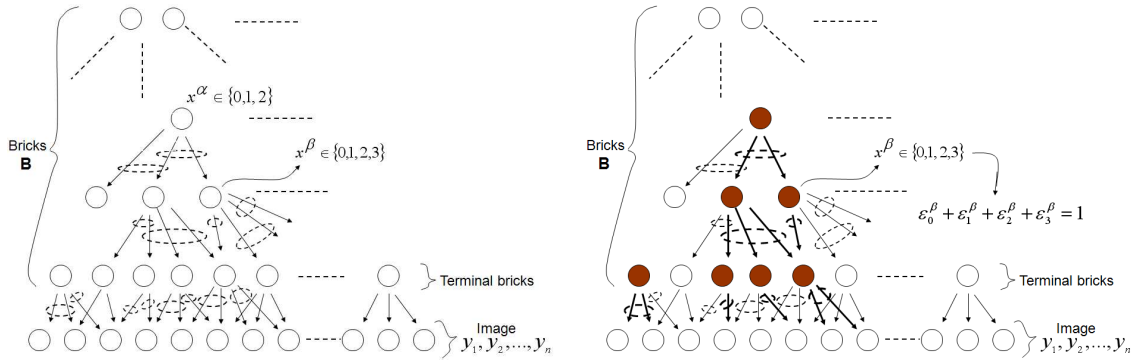

Figure 3.1: Architecture. Left. A hierarchy of "bricks," each representing a disjunction of conjunctions. Bottom row is the image (pixel) data and the row above it is the set of terminal bricks. The state of a brick signals a chosen set of children. Right. An "interpretation," which is an assignment 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.

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

$$P_0(\vec{x}) = \frac{\prod_{\beta \in \mathcal{B}} (\epsilon_{x^\beta}^\beta)}{\prod_{\beta \in B(\vec{x})} (1 - \epsilon_0^\beta)}, \tag{3.1}$$

where $B = B(\vec{x})$ is the *below set* defined in Section 1.3.3.

$a^{x^\alpha}(\vec{x})$ is an attribute function (scaler valued or vector valued) associated with non-terminal brick $\alpha \in \mathcal{B} \setminus \mathcal{T}$, when α takes value x^α and $x^\alpha > 0$. To ease the notation, from here through the end of this Chapter, we will use $a^\alpha(\cdot)$ to stand for $a^{x^\alpha}(\cdot)$. We require that $a^\alpha(\vec{x})$ depends only on the state of the descendants (i.e., children, grandchildren, great grandchildren, etc.) of α, by the graph structure $\mathcal{G}$. Let D_α be the set of the descendant bricks of α, i.e. for any brick $\beta \in D_\alpha$, there either exists a directed edge pointing from α to β ('$\alpha \to \beta$'), or there exist bricks $\eta_1,\ldots,\eta_m$ s.t. '$\alpha \to \eta_1 \to \cdots \to \eta_m \to \beta$.' Let x^{D_α} be the state of D_α. Then under this notation, $a^\alpha(\vec{x}) = a^\alpha(x^{D_\alpha})$.

A prototypical example is the set of pose coordinates (or *relational* pose coordinates) of the chosen children of α. Depending on the depth of the instantiation of the children, $a^{x^\alpha}(\vec{x})$ may depend on the states of bricks that are several generations removed from α itself (grandchildren, great grandchildren, etc.). $P_c^{x^\alpha}(a^\alpha|x^\alpha)$ (shorted as $P_c(a^\alpha|x^\alpha)$) is the desired conditional distribution on the attribute a^α, which will not, in general, agree with P_0 defined in (3.1). The questions we address here are:

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

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

Theorem 3 (Section 3.1) and Theorem 4 (Section 3.2) will answer these two questions, respectively.

3.1 Existence of Probability Distribution Satisfying the Conditional Constraints

In this section, we will prove the existence of a distribution on interpretations that satisfies all the conditional constraints under a "positive constraint" assumption:

$$P_c(a^\alpha(S)|x^\alpha) = 1,\ \alpha \in \mathcal{B} \setminus \mathcal{T}, \forall x^\alpha > 0, \tag{3.2}$$

where S is a subset of $\mathcal{I}$, defined as $S = \{\vec{x} \in \mathcal{I} : x^\beta > 0, \forall \beta \in D_\alpha\}$. We call $\{P_c(a^\alpha|x^\alpha) : x^\alpha > 0\}_{\alpha \in \mathcal{B}\setminus\mathcal{T}}$ a set of "positive conditional constraints," if it satisfies (3.2).

Theorem 3. *For the directed acyclic graph $\mathcal{G}$, given any set of positive conditional constraints $\{P_c^\alpha : x^\alpha > 0\}_{\alpha \in \mathcal{B}\setminus\mathcal{T}}$, there exists at least one distribution P on $\mathcal{I}$ such that $P(a^\alpha|x^\alpha) = P_c(a^\alpha|x^\alpha)$, $\forall \alpha \in \mathcal{B} \setminus \mathcal{T}$, $\forall x^\alpha > 0$.*

Proof. We will start from a special case and make this case more and more general – We will first prove the existence of P for a special hierarchy structure and a special set of conditional constraints in Step 1. Then we will move one step forward, to prove the existence of P in Step 2 for the same special hierarchy structure (as in Step 1) and the original conditional constraints $\{P_c(a^\alpha|x^\alpha) : x^\alpha > 0\}_{\alpha \in \mathcal{B}\setminus\mathcal{T}}$. Finally, in Step 3, we will prove the existence of P for the original hierarchy structure as show in Figure 3.1 and the original conditional constraints $\{P_c(a^\alpha|x^\alpha) : x^\alpha > 0\}_{\alpha \in \mathcal{B}\setminus\mathcal{T}}$. To ease the notation, let N be the number of bricks in $\mathcal{B}$. Also denote $\mathcal{B}$ as $\mathcal{B} = \{\alpha_1, \alpha_2, \ldots, \alpha_N\}$.

Step 1. We define a special hierarchy structure $\hat{\mathcal{G}}$ (one example is shown in Figure 3.2) as follows,

- A directed edge from brick α_i to brick α_j if and only if $j < i, \forall i, j \in \{1, \ldots, N\}$,

and a special set of given conditional constraints as follows,

- $\{P_c(\{x^{\alpha_j}\}_{j<i}|x^{\alpha_i}) : x^{\alpha_i} > 0\}_{i=2}^{N}$ satisfying $\forall x^{\alpha_i} > 0$,

$$P_c(\{x^{\alpha_j}\}_{j<i}|x^{\alpha_i}) > 0, \forall x^{\alpha_1}, \ldots, x^{\alpha_{i-1}} > 0, \tag{3.3}$$

$$P_c(\{x^{\alpha_j}\}_{j<i}|x^{\alpha_i}) = 0, \text{ if there exists } j < i \text{ s.t. } x^{\alpha_j} = 0. \tag{3.4}$$

This special set of conditional constraints is a conditional distribution directly on $\vec{x}$, instead of on an attribute function of $\vec{x}$. This set of condition constraints implies that, $\forall A \subset \{1, 2, \ldots, i-1\}$, the marginal distribution $P_c(\{x^{\alpha_j}\}_{j\in A}|x^{\alpha_i})$ is given as well, since it can be computed through integration. We will construct a distribution P within this special hierarchy structure s.t. $P(\{x^{\alpha_j}\}_{j<i}|x^{\alpha_i}) = P_c(\{x^{\alpha_j}\}_{j<i}|x^{\alpha_i})$, $\forall i > 1$ and $x^{\alpha_i} > 0$. And this will be done through four successive minor steps: Step 1.1, Step 1.2, Step 1.3, and Step 1.4.

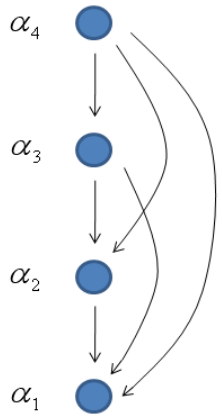

Figure 3.2: The special hierarchy structure in Step 1 and Step 2, when N=4.

Step 1.1. In this step, we only deal with the ancestor brick α_N, the brick on top of the hierarchy as shown in Figure 3.2. Let $P(\{x^{\alpha_j}\}_{j<N}|x^{\alpha_N}) = P_c(\{x^{\alpha_j}\}_{j<N}|x^{\alpha_N})$, $\forall x^{\alpha_N} > 0$.

Step 1.2. In this step, we will focus on the second to the lowest brick, α_2, in addition to brick α_N when $x^{\alpha_N} = 0$ (See Figure 3.2). $\forall s \in \{2, 3, \ldots, n^{\alpha_1}\}$, $\forall t \in \{1, 2, \ldots, n^{\alpha_2}\}$,

the following equation always holds,

$$\frac{P(x^{\alpha_1}=1|x^{\alpha_2}=t)}{P(x^{\alpha_1}=s|x^{\alpha_2}=t)} = \frac{\left(\begin{array}{l}\sum_{k=1}^{n^{\alpha_N}} P(x^{\alpha_1}=1,x^{\alpha_2}=t|x^{\alpha_N}=k)\cdot P(x^{\alpha_N}=k)\\ +\ P(x^{\alpha_1}=1,x^{\alpha_2}=t|x^{\alpha_N}=0)\cdot P(x^{\alpha_N}=0)\end{array}\right)}{\left(\begin{array}{l}\sum_{k=1}^{n^{\alpha_N}} P(x^{\alpha_1}=s,x^{\alpha_2}=t|x^{\alpha_N}=k)\cdot P(x^{\alpha_N}=k)\\ +\ P(x^{\alpha_1}=s,x^{\alpha_2}=t|x^{\alpha_N}=0)\cdot P(x^{\alpha_N}=0)\end{array}\right)}. \quad (3.5)$$

We require $P(x)$ to satisfy the following constraints: $\forall s \in \{2,3,\ldots,n^{\alpha_1}\}$, $\forall t \in \{1,2,\ldots,n^{\alpha_2}\}$, $\forall k \in \{1,2,\ldots,n^{\alpha_N}\}$,

$$\begin{cases} P(x^{\alpha_1}=1|x^{\alpha_2}=t) &= P_c(x^{\alpha_1}=1|x^{\alpha_2}=t),\\ P(x^{\alpha_1}=s|x^{\alpha_2}=t) &= P_c(x^{\alpha_1}=s|x^{\alpha_2}=t),\\ P(x^{\alpha_1}=1,x^{\alpha_2}=t|x^{\alpha_N}=0) &= \frac{1}{2\cdot n^{\alpha_2}}\cdot P_c(x^{\alpha_1}=1|x^{\alpha_2}=t),\\ P(x^{\alpha_N}=k) &= \delta, \quad \text{where } \delta \text{ is a tiny positive number.}\end{cases} \quad (3.6)$$

From (3.3), $P(x^{\alpha_1}=1,x^{\alpha_2}=t|x^{\alpha_N}=0)$ constructed above is positive. Since from Step 1.1, we have already had

$$P(x^{\alpha_1}=1,x^{\alpha_2}=t|x^{\alpha_N}=k) = P_c(x^{\alpha_1}=1,x^{\alpha_2}=t|x^{\alpha_N}=k), \quad (3.7)$$

$$P(x^{\alpha_1}=s,x^{\alpha_2}=t|x^{\alpha_N}=k) = P_c(x^{\alpha_1}=s,x^{\alpha_2}=t|x^{\alpha_N}=k). \quad (3.8)$$

After plugging (3.6), (3.7), and (3.8) into (3.5), we solve out

$$P(x^{\alpha_1}=s,x^{\alpha_2}=t|x^{\alpha_N}=0), \quad \forall s \in \{2,3,\ldots,n^{\alpha_1}\}, \forall t \in \{1,2,\ldots,n^{\alpha_2}\}.$$

Before moving forward to other bricks, we first need to show that $P(x^{\alpha_1}=s,x^{\alpha_2}=t|x^{\alpha_N}=0)$ defined above is proper, i.e. the summation of $P(x^{\alpha_1}=s,x^{\alpha_2}=t|x^{\alpha_N}=0)$ over positive s and positive t is in [0,1]. Since $P(x^{\alpha_N}=k)=\delta$ and δ is very small,

from equation (3.5) we have the following approximation,

$$\frac{P(x^{\alpha_1}=1, x^{\alpha_2}=t|x^{\alpha_N}=0)}{P(x^{\alpha_1}=s, x^{\alpha_2}=t|x^{\alpha_N}=0)} \approx \frac{P(x^{\alpha_1}=1|x^{\alpha_2}=t)}{P(x^{\alpha_1}=s|x^{\alpha_2}=t)}.$$

It implies that $\forall s \in \{1,\ldots,n^{\alpha_1}\}, \forall t \in \{1,\ldots,n^{\alpha_2}\}$, $P(x^{\alpha_1}=s, x^{\alpha_2}=t|x^{\alpha_N}=0) > 0$ and

$$\begin{aligned}
& P(x^{\alpha_1}=s, x^{\alpha_2}=t|x^{\alpha_N}=0) \\
\leq\ & 2 \cdot P(x^{\alpha_1}=1, x^{\alpha_2}=t|x^{\alpha_N}=0) \cdot \frac{P(x^{\alpha_1}=s|x^{\alpha_2}=t)}{P(x^{\alpha_1}=1|x^{\alpha_2}=t)} \\
=\ & \frac{1}{n^{\alpha_2}} P(x^{\alpha_1}=1|x^{\alpha_2}=t) \cdot \frac{P(x^{\alpha_1}=s|x^{\alpha_2}=t)}{P(x^{\alpha_1}=1|x^{\alpha_2}=t)} \\
=\ & \frac{1}{n^{\alpha_2}} P_c(x^{\alpha_1}=s|x^{\alpha_2}=t).
\end{aligned}$$

Hence,

$$\sum_{t=1}^{n^{\alpha_2}} \sum_{s=1}^{n^{\alpha_1}} P(x^{\alpha_1}=s, x^{\alpha_2}=t|x^{\alpha_N}=0) \leq \frac{1}{n^{\alpha_2}} \sum_{t=1}^{n^{\alpha_2}} \sum_{s=1}^{n^{\alpha_1}} P_c(x^{\alpha_1}=s|x^{\alpha_2}=t) \leq 1.$$

Step 1.3. In this step, we will focus on the third to the lowest brick, α_3, in addition to brick α_N when $x^{\alpha_N}=0$. $\forall s \in \{1,2,\ldots,n^{\alpha_1}\}, \forall t \in \{1,2,\ldots,n^{\alpha_2}\}, \forall r \in$

$\{1, 2, \ldots, n^{\alpha_3}\}$ and $(s,t) \neq (1,1)$, the following equality holds

$$\begin{aligned}
&\frac{P(x^{\alpha_1}=1, x^{\alpha_2}=1 | x^{\alpha_3}=r)}{P(x^{\alpha_1}=s, x^{\alpha_2}=t | x^{\alpha_3}=r)} \\
&= \frac{\sum_{k=1}^{n^{\alpha_N}} \left(\begin{array}{c} P(x^{\alpha_1}=1, x^{\alpha_2}=1, x^{\alpha_3}=r | x^{\alpha_N}=k) \cdot P(x^{\alpha_N}=k) \\ + \\ P(x^{\alpha_1}=1, x^{\alpha_2}=1, x^{\alpha_3}=r | x^{\alpha_N}=0) \cdot P(x^{\alpha_N}=0) \end{array} \right)}{\sum_{k=1}^{n^{\alpha_N}} \left(\begin{array}{c} P(x^{\alpha_1}=s, x^{\alpha_2}=t, x^{\alpha_3}=r | x^{\alpha_N}=k) \cdot P(x^{\alpha_N}=k) \\ + \\ P(x^{\alpha_1}=s, x^{\alpha_2}=t, x^{\alpha_3}=r | x^{\alpha_N}=0) \cdot P(x^{\alpha_N}=0) \end{array} \right)}. \qquad (3.9)
\end{aligned}$$

From Step 1.2, we have already defined $P(x^{\alpha_N}=k)=\delta$, $\forall k \in \{1,2,\ldots,n^{\alpha_N}\}$. Now we require $P(x)$ to satisfy more constraints as follows: $\forall s \in \{2,\ldots,n^{\alpha_1}\}, \forall t \in \{2,\ldots,n^{\alpha_2}\}, \forall r \in \{1,\ldots,n^{\alpha_3}\}, \forall k \in \{1,2,\ldots,n^{\alpha_N}\}$,

$$\left\{ \begin{array}{rcl} P(x^{\alpha_1}=1, x^{\alpha_2}=1 | x^{\alpha_3}=r) & = & P_c(x^{\alpha_1}=1, x^{\alpha_2}=1 | x^{\alpha_3}=r), \\ P(x^{\alpha_1}=s, x^{\alpha_2}=t | x^{\alpha_3}=r) & = & P_c(x^{\alpha_1}=s, x^{\alpha_2}=t | x^{\alpha_3}=r), \\ P(x^{\alpha_1}=1, x^{\alpha_2}=1, x^{\alpha_3}=r | x^{\alpha_N}=0) & = & \frac{1}{2n^{\alpha_3}} \cdot P_c(x^{\alpha_1}=1, x^{\alpha_2}=1 | x^{\alpha_3}=r) \\ & & \cdot \min_{s,t>0}\{P(x^{\alpha_1}=s, x^{\alpha_2}=t | x^{\alpha_N}=0)\}. \end{array} \right. \qquad (3.10)$$

From Step 1.2, $\forall s \in \{1,\ldots,n^{\alpha_1}\}, \forall t \in \{1,\ldots,n^{\alpha_2}\}$, $P(x^{\alpha_1}=s, x^{\alpha_2}=t | x^{\alpha_N}=0) > 0$, hence $P(x^{\alpha_1}=1, x^{\alpha_2}=1, x^{\alpha_3}=r | x^{\alpha_N}=0)$ defined above is positive. Since from Step 1.1, we have already had

$$\begin{aligned}
& P(x^{\alpha_1}=1, x^{\alpha_2}=1, x^{\alpha_3}=r | x^{\alpha_N}=k) \\
= \; & P_c(x^{\alpha_1}=1, x^{\alpha_2}=1, x^{\alpha_3}=r | x^{\alpha_N}=k) \qquad (3.11) \\
& P(x^{\alpha_1}=s, x^{\alpha_2}=t, x^{\alpha_3}=r | x^{\alpha_N}=k) \\
= \; & P_c(x^{\alpha_1}=s, x^{\alpha_2}=t, x^{\alpha_3}=r | x^{\alpha_N}=k). \qquad (3.12)
\end{aligned}$$

After plugging (3.10), (3.11), and (3.12) into (3.9), we solve out

$$P(x^{\alpha_1} = s, x^{\alpha_2} = t, x^{\alpha_3} = r | x^{\alpha_N} = 0).$$

$\forall s \in \{1, \ldots, n^{\alpha_1}\}, \forall t \in \{1, \ldots, n^{\alpha_2}\}, \forall r \in \{1, \ldots, n^{\alpha_3}\}$. Similar as in Step 1.2, we have to justify that $P(x^{\alpha_1} = s, x^{\alpha_2} = t, x^{\alpha_3} = r | x^{\alpha_N} = 0)$ defined above is proper. It is easy to see that we only need to justify $\sum_{r=1}^{n^{\alpha_3}} P(x^{\alpha_1} = s, x^{\alpha_2} = t, x^{\alpha_3} = r | x^{\alpha_N} = 0) \leq P(x^{\alpha_1} = s, x^{\alpha_2} = t | x^{\alpha_N} = 0)$. Due to the fact that $P(x^{\alpha_N} = k) = \delta$ and δ is a small positive number, we have the following approximation from equation (3.9),

$$\frac{P(x^{\alpha_1} = 1, x^{\alpha_2} = 1, x^{\alpha_3} = r | x^{\alpha_N} = 0)}{P(x^{\alpha_1} = s, x^{\alpha_2} = t, x^{\alpha_3} = r | x^{\alpha_N} = 0)} \approx \frac{P(x^{\alpha_1} = 1, x^{\alpha_2} = 1 | x^{\alpha_3} = r)}{P(x^{\alpha_1} = s, x^{\alpha_2} = t | x^{\alpha_3} = r)}.$$

It implies

$$\begin{aligned}
& \sum_{r=1}^{n^{\alpha_3}} P(x^{\alpha_1} = s, x^{\alpha_2} = t, x^{\alpha_3} = r | x^{\alpha_N} = 0) \\
\leq & \sum_{r=1}^{n^{\alpha_3}} 2 \cdot P(x^{\alpha_1} = 1, x^{\alpha_2} = 1, x^{\alpha_3} = r | x^{\alpha_N} = 0) \cdot \frac{P(x^{\alpha_1} = s, x^{\alpha_2} = t | x^{\alpha_3} = r)}{P(x^{\alpha_1} = 1, x^{\alpha_2} = 1 | x^{\alpha_3} = r)} \\
= & \sum_{r=1}^{n^{\alpha_3}} \frac{1}{n^{\alpha_2}} P(x^{\alpha_1} = 1, x^{\alpha_2} = 1 | x^{\alpha_3} = r) \cdot \min_{s,t>0} \{P(x^{\alpha_1} = s, x^{\alpha_2} = t | x^{\alpha_N} = 0)\} \\
& \cdot \frac{P(x^{\alpha_1} = s, x^{\alpha_2} = t | x^{\alpha_3} = r)}{P(x^{\alpha_1} = 1, x^{\alpha_2} = 1 | x^{\alpha_3} = r)} \\
= & \min_{s,t>0} \{P(x^{\alpha_1} = s, x^{\alpha_2} = t | x^{\alpha_N} = 0)\} \cdot \frac{1}{n^{\alpha_3}} \sum_{r=1}^{n^{\alpha_3}} P_c(x^{\alpha_1} = s, x^{\alpha_2} = t | x^{\alpha_3} = r) \\
\leq & P(x^{\alpha_1} = s, x^{\alpha_2} = t | x^{\alpha_N} = 0).
\end{aligned}$$

Step 1.4. For brick $\alpha_4, \alpha_5, \ldots, \alpha_{N-1}$, by going through the similar procedure as in Step 1.3, we construct $P(\{x^{\alpha_l}\}_{l=1}^{j} | x^{\alpha_N} = 0)$, $\forall j \in \{4, \ldots, N-1\}$. And similarly as in Step 1.3, it is not hard to be justified that these constructed conditional probabilities

$P(\{x^{\alpha_l}\}_{l=1}^{j}|x^{\alpha_N}=0)$ are proper and do not contradict with each other. So far, we have constructed

$$P(\{x^{\alpha_l}\}_{l=1}^{N-1}|x^{\alpha_N}=k)=P_c(\{x^{\alpha_l}\}_{l=1}^{N-1}|x^{\alpha_N}=k),\ \ \forall k>0, \tag{3.13}$$

and we have defined

$$P(x^{\alpha_N}=k)\quad\&\quad P(\{x^{\alpha_l}\}_{l=1}^{N-1}|x^{\alpha_N}=0),\ \ \forall k>0,\ \forall\{x^{\alpha_l}>0\}_{l=1}^{N-1}. \tag{3.14}$$

In order to specify the full probability $P(\{x^{\alpha_l}\}_{l=1}^{N})$, we only need to fill in the conditional probability $P(\{x^{\alpha_l}\}_{l=1}^{N-1}\mid x^{\alpha_N}=0)$ for the case when there exists at least one x^{α_l} equal to zero, and this filling procedure is trivial. According to the construction process of P from Step 1.2 through Step 1.4, automatically, P satisfies

$$P(\{x^{\alpha_j}\}_{j<i}|x^{\alpha_i})=P_c(\{x^{\alpha_j}\}_{j<i}|x^{\alpha_i}),\ \ \forall i\in\{2,\ldots,N\},\ \forall x^{\alpha_i}>0.$$

At this point, we have proved the existence of P for the special hierarchy structure (as shown in Figure 3.2) and the special set of conditional constraints. Now let us move forward, to prove the existence of P for the same hierarchy structure as in Step 1 and the original conditional constraints $\{P_c(a^\alpha|x^\alpha):x^\alpha>0\}_{\alpha\in\mathcal{B}\setminus\mathcal{T}}$.

Step 2. Consider the same compositional machine structure $\hat{\mathcal{G}}$ as in Step 1, but the original given conditional constraints $\{P_c(a^\alpha|x^\alpha):x^\alpha>0\}_{\alpha\in\mathcal{B}\setminus\mathcal{T}}$. In order to exploit the result from Step 1, we need to make a connection between the special set of conditional constraints and the original conditional constraints. The connection is that there necessarily exists at least one conditional distribution $\tilde{P}_c(x^{\alpha_1},\ldots,x^{\alpha_{i-1}}|x^{\alpha_i})$, satisfying (3.3) and (3.4), and $\forall i\in\{2,\ldots,N\}$, $\forall x^{\alpha_i}>0$, $\forall a$ in the domain of $a^{\alpha_i}(x^{\alpha_1},x^{\alpha_2},\ldots,x^{\alpha_{i-1}})$,

$$\int_{\{x^{\alpha_j}\}_{j=1}^{i-1},\ \text{s.t.}\ a^{\alpha_i}(x^{\alpha_1},x^{\alpha_2},\ldots,x^{\alpha_{i-1}})=a}\tilde{P}_c(x^{\alpha_1},\ldots,x^{\alpha_{i-1}}|x^{\alpha_i})\ =\ P_c(a^{\alpha_i}=a|x^{\alpha_i}).$$

This existence of $\tilde{P}_c(x^{\alpha_1}, \ldots, x^{\alpha_{i-1}}|x^{\alpha_i})$ is guaranteed by the assumption that $\{P_c^\alpha : x^\alpha > 0\}_{\alpha \in \mathcal{B}\setminus\mathcal{T}}$ is a set of "positive conditional constraints." Hence, as long as we require the distribution $P(\vec{x})$ to satisfy the conditional constraint posed by $\tilde{P}_c$ as in Step 1, P will satisfy the conditional constraint posed by $\{P_c(a^\alpha|x^\alpha) : x^\alpha > 0\}_{\alpha \in \mathcal{B}\setminus\mathcal{T}}$ automatically.

At this point, we have proved the existence of P for the special hierarchy structure $\hat{\mathcal{G}}$ (as shown in Figure 3.2) and the original conditional constraints $\{P_c(a^\alpha|x^\alpha) : x^\alpha > 0\}_{\alpha \in \mathcal{B}\setminus\mathcal{T}}$. Now let us move to the final stage to prove the existence of P for the original hierarchy structure $\mathcal{G}$ (as shown in Figure 3.1) and the original conditional constraints.

Step 3. Consider the original hierarchy structure $\mathcal{G}$ (shown in Figure 3.1) and the original conditional constraints $\{P_c(a^\alpha|x^\alpha) : x^\alpha > 0\}_{\alpha \in \mathcal{B}\setminus\mathcal{T}}$. We construct P through the following procedure:

(1). Transform the original hierarchy structure $\mathcal{G}$ (shown in Figure 3.1) to the special hierarchy structure $\hat{\mathcal{G}}$ (as shown in Figure 3.2), by adding more edges into the graph.

(2). Do Step 2, and get $P(\vec{x})$.

It is easy to see that the resulting distribution $P(\vec{x})$ satisfies $\forall \alpha \in \mathcal{B} \setminus \mathcal{T}$, $\forall x^\alpha > 0$, $P(a^\alpha|x^\alpha) = P_c(a^\alpha|x^\alpha)$. But is it a proper distribution on set of interpretations $\mathcal{I}$ defined by the hierarchy structure $\mathcal{G}$? The answer is yes. We reason as follows: Let $\hat{\mathcal{I}}$ represent the set of interpretations defined by the special hierarchy structure $\hat{\mathcal{G}}$ in Step 3.(1) above. By the definition of "an interpretation" (refer to the beginning of this Chapter), we have $\hat{\mathcal{I}} \subseteq \mathcal{I}$, considering that $\hat{\mathcal{G}}$ was obtained after more directed edged were added to $\mathcal{G}$. From Step 3.(1), this $P(\vec{x})$ is supported on $\hat{\mathcal{I}}$, hence $P(\vec{x})$ is supported on $\mathcal{I}$ as well.

□

3.2 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 the conditional constraints $\{P_c(a^\alpha|x^\alpha) : x^\alpha > 0\}_{\alpha\in\mathcal{B}\setminus\mathcal{T}}$, if this set of conditional constraints is a set of "positive" conditional constraints. In point of fact, the attribute functions $\{a^\alpha(\vec{x})\}_{\alpha\in\mathcal{B}\setminus\mathcal{T}}$ together with the attribute (conditional) distributions $\{P_c(a^\alpha|x^\alpha)\}_{\alpha\in\mathcal{B}\setminus\mathcal{T}}$ will in general under-determine the distribution on $\mathcal{I}$. Hence, given the existence of one distribution P satisfying the desired constraints, there are many such distributions satisfying the desired constraints. In this section, we will present a scheme to iteratively perturb the Markovian distribution P_0 defined in (3.1) such that, under a hypothesis of "non triviality," it will converge to an asymptotic distribution P^* on $\mathcal{I}$, and P^* will satisfy the conditional constraints as well. "Non trivial" means that $\forall\alpha \in \mathcal{B} \setminus \mathcal{T}$, $\forall x^\alpha > 0$,

$$P_c(a^\alpha(\vec{x})|x^\alpha) \text{ has the same support as } P_0(a^\alpha(\vec{x})|x^\alpha), \tag{3.15}$$

where $P_0(a^\alpha(\vec{x})|x^\alpha)$ is the marginal conditional distribution of $a^\alpha(\vec{x})$ under P_0 defined in (3.1).

Let N be the number of bricks in $\mathcal{B}$, and denote $\mathcal{B}$ as $\mathcal{B} = \{\alpha_1, \alpha_2, \ldots, \alpha_N\}$. Let $M_i \stackrel{\triangle}{=} n^{\alpha_i}$, 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, $\forall m \in \{0,1,2,\ldots\}$, $\forall t \in \{1,2,\ldots,M_s\}$, $\forall l \in \{1,2,\ldots,N\}$, 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:

$$P_{mM_s+t}(\vec{x}) = P_{mM_s+t-1}(\vec{x}) \cdot \left(\frac{P_c(a^{\alpha_l}(\vec{x})|x^{\alpha_l}=t-\sum_{i=1}^{l-1} M_i)}{P_{mM_s+t-1}(a^{\alpha_l}(\vec{x})|x^{\alpha_l}=t-\sum_{i=1}^{l-1} M_i)} \right)^{1_{\{x^{\alpha_l}=t-\sum_{i=1}^{l-1} M_i\}}} \tag{3.16}$$

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 $k = mM_s + t$. We continue otherwise. (The "non-triviality" condition defined in (3.15) 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 (3.16) for all the brick $\alpha_l \in \mathcal{B}$, 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 4. *Under the non-triviality condition, if 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. $P_k(\vec{x})) \stackrel{k\to\infty}{\longrightarrow} P^*(\vec{x})$, $\forall \vec{x} \in \mathcal{I}$. 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. We will prove the convergence of $\{P_k\}$ and the property with P^* in three steps, Step 1, Step 2, and Step 3.

Step 1. $\forall k \in \{0, 1, \ldots\}$, w.o.l.g., we perturb brick α at $x^\alpha = j, j > 0$, at $(k+1)^{th}$ step:

$$P_{k+1}(\vec{x}) = P_k(\vec{x}) \cdot \left(\frac{P(a^\alpha(\vec{x})|x^\alpha = j)}{P_k(a^\alpha(\vec{x})|x^\alpha = j)} \right)^{1_{\{x^\alpha = j\}}} \tag{3.17}$$

We want to show $D(P\|P_{k+1}) \leq D(P\|P_k)$ in Step 1. Let $G = \{\vec{x} : x^\alpha = j\}$. From (3.17) we have four observations:

$$P_{k+1}(\vec{x}) = P_k(\vec{x}), \ \forall \vec{x} \in G^c, \tag{3.18}$$

$$P_k(G) = P_{k+1}(G), \text{ since } P_k(G^c) = P_{k+1}(G^c), \tag{3.19}$$

$$P_{k+1}(a^\alpha(\vec{x})|G) = P(a^\alpha(\vec{x})|G), \tag{3.20}$$

$$P_k(\vec{x}|a^\alpha(\vec{x}), G) = P_{k+1}(\vec{x}|a^\alpha(\vec{x}), G), \tag{3.21}$$

where (3.18) through (3.20) are straightforward, and (3.21) can be derived as follows:

$$\begin{aligned} P_k(\vec{x}|a^\alpha(\vec{x}), G) &= \frac{P_k(\vec{x}|G)}{P_k(a^\alpha(\vec{x})|G)}. \\ P_{k+1}(\vec{x}|a^\alpha(\vec{x}), G) &= \frac{P_{k+1}(\vec{x}|G)}{P_{k+1}(a^\alpha(\vec{x})|G)} \\ &= \frac{P_k(\vec{x}|G) \cdot \frac{P(a^\alpha(\vec{x})|x^\alpha=j)}{P_k(a^\alpha(\vec{x})|x^\alpha=j)}}{P(a^\alpha(\vec{x})|G)} \\ &= \frac{P_k(\vec{x}|G)}{P_k(a^\alpha(\vec{x})|G)}. \end{aligned}$$

Now we split $D(P\|P_k)$ into two parts,

$$\begin{aligned} D(P\|P_k) &= \int_{\vec{x}} P(\vec{x}) \log \frac{P(\vec{x})}{P_k(\vec{x})} \\ &= \int_G P(\vec{x}) \log \frac{P(\vec{x})}{P_k(\vec{x})} + \int_{G^c} P(\vec{x}) \log \frac{P(\vec{x})}{P_k(\vec{x})} \\ &= \int_G P(\vec{x}) \log \frac{P(G) \cdot P(a^\alpha(\vec{x})|G) \cdot P(\vec{x}|a^\alpha(\vec{x}), G)}{P_k(G) \cdot P_k(a^\alpha(\vec{x})|G) \cdot P(\vec{x}|a^\alpha(\vec{x}), G)} + \int_{G^c} P(\vec{x}) \log \frac{P(\vec{x})}{P_k(\vec{x})}. \end{aligned}$$

With the observations (3.18) to (3.21), we get

$$\begin{aligned} & D(P\|P_k) - D(P\|P_{k+1}) \\ = & \left[\int_G P(\vec{x}) \log \frac{P(G) \cdot P(\vec{x}|a^\alpha(\vec{x}), G)}{P_k(G) \cdot P_k(\vec{x}|a^\alpha(\vec{x}), G)} - \int_G P(\vec{x}) \log \frac{P(G) \cdot P(\vec{x}|a^\alpha(\vec{x}), G)}{P_{k+1}(G) \cdot P_{k+1}(\vec{x}|a^\alpha(\vec{x}), G)}\right] \\ & + \left[\int_G P(\vec{x}) \log \frac{P(a^\alpha(\vec{x})|G)}{P_k(a^\alpha(\vec{x})|G)} - \int_G P(\vec{x}) \log \frac{P(a^\alpha(\vec{x})|G)}{P_{k+1}(a^\alpha(\vec{x})|G)}\right] \\ & + \left[\int_{G^c} P(\vec{x}) \log \frac{P(\vec{x})}{P_k(\vec{x})} - \int_{G^c} P(\vec{x}) \log \frac{P(\vec{x})}{P_{k+1}(\vec{x})}\right] \\ = & \int_G P(\vec{x}) \log \frac{P(a^\alpha(\vec{x})|G)}{P_k(a^\alpha(\vec{x})|G)} \\ = & P(G) \int_G P(\vec{x}|G) \log \frac{P(a^\alpha(\vec{x})|G)}{P_k(a^\alpha(\vec{x})|G)} \\ = & P(G) \int_{a^\alpha} \int_{\{\vec{x}\in G:\ a^\alpha(\vec{x})=a^\alpha\}} P(\vec{x}|G) \log \frac{P(a^\alpha(\vec{x})|G)}{P_k(a^\alpha(\vec{x})|G)} d\vec{x}\ da^\alpha \\ = & P(G) \int_{a^\alpha} \left[\int_{\{\vec{x}\in G:\ a^\alpha(\vec{x})=a^\alpha\}} P(\vec{x}|G) d\vec{x}\right] \log \frac{P(a^\alpha|G)}{P_k(a^\alpha|G)} da^\alpha \\ = & P(G) \int_{a^\alpha} P(a^\alpha|G) \log \frac{P(a^\alpha|G)}{P_k(a^\alpha|G)} da^\alpha \\ = & P(G) \cdot D(P(a^\alpha|G)\|P_k(a^\alpha|G)) \\ \geq & 0. \end{aligned}$$

At this moment, we have proved $\forall k \in \{0, 1, \ldots\}$, $D(P\|P_k) \geq D(P\|P_{k+1})$, i.e.,

$$\{D(P\|P_k)\}_{k=0}^{\infty} \text{ is monotonically decreasing.} \tag{3.22}$$

Since $D(P\|P_k) \geq 0, \forall k$, we conclude that $\{D(P\|P_k)\}_{k=0}^{\infty}$ is converging. Hence $\{D(P\|P_k) - D(P\|P_{k+1})\}_{k=0}^{\infty}$ is non-negative and converges to 0.

Step 2. $\forall l \in \{1, 2, \ldots, N\}, \forall j \in \{1, 2, \ldots, M_j\}$, (i.e., considering brick α_l, when $x^{\alpha_l} = j$,) $\forall m \in \{1, 2, \ldots\}$, let $k_l^j(m) = mM_s + \sum_{i=1}^{l-1} M_i + j$, $G_l^j = \{\vec{x} : x^{\alpha_l} = j\}$.

$$D(P\|P_{k_l^j(m)-1}) - D(P\|P_{k_l^j(m)}) = P(G_l^j) \cdot D(P(a^{\alpha_l}|G_l^j)\|P_{k_l^j(m)-1}(a^{\alpha_l}|G_l^j)).$$

Since LHS above is converging to 0, as $m \to \infty$, and $P(G_l^j) \neq 0$, we conclude

$$D(P(a^{\alpha_l}|G_l^j)\|P_{k_l^j(m)-1}(a^{\alpha_l}|G_l^j)) \text{ is converging to 0, as } m \to \infty. \tag{3.23}$$

From here until the end of Step 2, our goal is to find a universal bound for

$$\frac{P_{k_l^j(m)}(a^{\alpha_r}|G_r^s)}{P_{k_l^j(m)-1}(a^{\alpha_r}|G_r^s)},$$

$\forall l, r \in \{1, 2, \ldots, N\}$, $\forall j, s \in \{1, 2, \ldots, M_r\}$, $\forall a^{\alpha_r} \in \{a^{\alpha_r}(\vec{x}) : \vec{x} \in \mathcal{I}\}$, where $G_r^s = \{\vec{x} : x^{\alpha_r} = s\}$. First, we need to derive a few inequalities. $\forall l \in \{1, 2, \ldots, N\}$, $\forall j \in \{1, 2, \ldots, M_l\}$, $\forall a^{\alpha_l} \in \{a^{\alpha_l}(\vec{x})\}$, from the non-trivial assumption of P_c, we have

$$P(a^{\alpha_l}|x^{\alpha_l} = j) \;=\; P_c(a^{\alpha_l}|x^{\alpha_l} = j) \;>\; 0.$$

In addition, $\vec{x}$ has a finite domain, hence the domain of $a^{\alpha}(\vec{x})$ is also finite. Therefore, the number MIN defined below is positive,

$$MIN \;\stackrel{\triangle}{=}\; \min_{\{l \,:\, \alpha_l \in\, \mathcal{B}\setminus\mathcal{T}\}} \;\min_{j \in \{1,\ldots,M_l\}} \;\min_{a^{\alpha_l} \in \{a^{\alpha_l}(\vec{x})\}} P(a^{\alpha_l}|x^{\alpha_l} = j) \;>\; 0.$$

By (3.23), $\forall \epsilon > 0$ s.t.

$$\epsilon < \frac{(MIN)^2}{128 \ln 2} \cdot 2^{-2M_s}, \tag{3.24}$$

$\exists\ m^*$ s.t. $\forall m > m^*$, $\forall l$, $\forall j$,

$$D(P(a^{\alpha_l}|G_l^j) \| P_{k_l^j(m)-1}(a^{\alpha_l}|G_l^j)) < \epsilon.$$

Lemma 12.6.1 from Elements of Information Theory by Cover and Thomas, [25], gives a lower bound for the K-L divergence between any two density functions,

$$D(P_1 \| P_2) \geq \frac{1}{2 \ln 2} \| P_1 - P_2 \|_1^2.$$

Here it implies

$$\begin{aligned} D(P(a^{\alpha_l}|G_l^j) \| P_{k_l^j(m)-1}(a^{\alpha_l}|G_l^j)) &\geq \frac{1}{2 \ln 2} \| P(a^{\alpha_l}|G_l^j) - P_{k_l^j(m)-1}(a^{\alpha_l}|G_l^j) \|_1^2 \\ &= \frac{1}{2 \ln 2} \left[\sum_{a^{\alpha_l}} |P(a^{\alpha_l}|G_l^j) - P_{k_l^j(m)-1}(a^{\alpha_l}|G_l^j)| \right]^2 \\ &\geq \frac{1}{2 \ln 2} \left(P(a^{\alpha_l}|G_l^j) - P_{k_l^j(m)-1}(a^{\alpha_l}|G_l^j) \right)^2, \quad \forall a^{\alpha_l}. \end{aligned}$$

Since $\vec{x}$ has a finite domain, the inequality above implies $\forall a^{\alpha_l}$,

$$|P(a^{\alpha_l}|G_l^j) - P_{k_l^j(m)-1}(a^{\alpha_l}|G_l^j)| \ \leq \ \sqrt{2 \ln 2} \cdot \sqrt{\epsilon}.$$

Hence $P_{k_l^j(m)-1}(a^{\alpha_l}|G_l^j) \in (P(a^{\alpha_l}|G_l^j) - \sqrt{2\ln 2}\cdot\sqrt{\epsilon},\ P(a^{\alpha_l}|G_l^j) + \sqrt{2\ln 2}\cdot\sqrt{\epsilon})$, and

$$\frac{P(a^{\alpha_l}|G_l^j)}{P_{k_l^j(m)-1}(a^{\alpha_l}|G_l^j)} - 1 \in (\frac{-\sqrt{2\ln 2}\cdot\sqrt{\epsilon}}{P(a^{\alpha_l}|G_l^j) - \sqrt{2\ln 2}\cdot\sqrt{\epsilon}},\ \frac{\sqrt{2\ln 2}\cdot\sqrt{\epsilon}}{P(a^{\alpha_l}|G_l^j) - \sqrt{2\ln 2}\cdot\sqrt{\epsilon}}).$$

$$\begin{aligned}
\left|\frac{P(a^{\alpha_l}|G_l^j)}{P_{k_l^j(m)-1}(a^{\alpha_l}|G_l^j)} - 1\right| &\leq \frac{\sqrt{2\ln 2}\cdot\sqrt{\epsilon}}{P(a^{\alpha_l}|G_l^j) - \sqrt{2\ln 2}\cdot\sqrt{\epsilon}} \\
&\leq \frac{\sqrt{2\ln 2}\cdot\sqrt{\epsilon}}{P(a^{\alpha_l}|G_l^j) - \frac{1}{2}P(a^{\alpha_l}|G_l^j)} \qquad \text{(due to (3.24))} \\
&= \frac{2\sqrt{2\ln 2}}{P(a^{\alpha_l}|G_l^j)}\cdot\sqrt{\epsilon} \\
&\leq \frac{2\sqrt{2\ln 2}}{MIN}\cdot\sqrt{\epsilon} \\
&\triangleq C\cdot\sqrt{\epsilon} \qquad (< \frac{1}{4}\cdot 2^{-M_s}), \qquad (3.25)
\end{aligned}$$

where the last inequality in the parentheses is due to the upper bound of ϵ in (3.24). Now we consider all the bricks at $k_l^j(m)$ step, after perturbing the brick l. $\forall r \in$

$\{1,2,\ldots,N\}$, $\forall s \in \{1,2,\ldots,M_r\}$, $\forall a^{\alpha_r}$, from the definition of $P_{k_l^j(m)}(\vec{x})$,

$$\begin{aligned}
& P_{k_l^j(m)}(G_r^s) - P_{k_l^j(m)-1}(G_r^s) \\
= & \int_{\vec{x}\in G_r^s} P_{k_l^j(m)}(\vec{x}) - \int_{\vec{x}\in G_r^s} P_{k_l^j(m)-1}(\vec{x}) \\
= & \left[\int_{\vec{x}\in G_r^s\cap G_l^j} P_{k_l^j(m)}(\vec{x}) - \int_{\vec{x}\in G_r^s\cap G_l^j} P_{k_l^j(m)-1}(\vec{x})\right] \\
& + \left[\int_{\vec{x}\in G_r^s\cap (G_l^j)^c} P_{k_l^j(m)}(\vec{x}) - \int_{\vec{x}\in G_r^s\cap (G_l^j)^c} P_{k_l^j(m)-1}(\vec{x})\right] \\
= & \left[\int_{\vec{x}\in G_r^s\cap G_l^j} P_{k_l^j(m)-1}(\vec{x})\cdot\frac{P(a^{\alpha_l}(\vec{x})|G_l^j)}{P_{k_l^j(m)-1}(a^{\alpha_l}(\vec{x})|G_l^j)} - \int_{\vec{x}\in G_r^s\cap G_l^j} P_{k_l^j(m)-1}(\vec{x})\right] \\
& + \left[\int_{\vec{x}\in G_r^s\cap (G_l^j)^c} P_{k_l^j(m)-1}(\vec{x}) - \int_{\vec{x}\in G_r^s\cap (G_l^j)^c} P_{k_l^j(m)-1}(\vec{x})\right] \\
= & \int_{\vec{x}\in G_r^s\cap G_l^j} P_{k_l^j(m)-1}(\vec{x})\cdot\left(\frac{P(a^{\alpha_l}(\vec{x})|G_l^j)}{P_{k_l^j(m)-1}(a^{\alpha_l}(\vec{x})|G_l^j)} - 1\right).
\end{aligned}$$

Hence,

$$\begin{aligned}
|P_{k_l^j(m)}(G_r^s) - P_{k_l^j(m)-1}(G_r^s)| \leq & \int_{\vec{x}\in G_r^s\cap G_l^j} P_{k_l^j(m)-1}(\vec{x})\cdot\left|\frac{P(a^{\alpha_l}(\vec{x})|G_l^j)}{P_{k_l^j(m)-1}(a^{\alpha_l}(\vec{x})|G_l^j)} - 1\right| \\
\leq & C\cdot\sqrt{\epsilon}\cdot P_{k_l^j(m)-1}(G_r^s\cap G_l^j) \\
\leq & C\cdot\sqrt{\epsilon}\cdot P_{k_l^j(m)-1}(G_r^s),
\end{aligned}$$

where $C\cdot\epsilon$ is from (3.25). We can rewrite the inequality above as

$$P_{k_l^j(m)-1}(G_r^s)\cdot(1-C\cdot\sqrt{\epsilon}) \leq P_{k_l^j(m)}(G_r^s) \leq P_{k_l^j(m)-1}(G_r^s)\cdot(1+C\cdot\sqrt{\epsilon}). \quad (3.26)$$

With the inequalities derived by far, now we are ready to look at the ratio between

$P_{k_l^j(m)}(a^{\alpha_r}|G_r^s)$ and $P_{k_l^j(m)-1}(a^{\alpha_r}|G_r^s)$.

$$
\begin{aligned}
& P_{k_l^j(m)}(a^{\alpha_r}|G_r^s) \\
= & \int_{\{\vec{x}\in G_r^s:\ a^{\alpha_r}(\vec{x})=a^{\alpha_r}\}} P_{k_l^j(m)}(\vec{x}|G_r^s) \\
= & \frac{1}{P_{k_l^j(m)}(G_r^s)} \int_{\{\vec{x}\in G_r^s:\ :\ a^{\alpha_r}(\vec{x})=a^{\alpha_r}\}} P_{k_l^j(m)}(\vec{x}) \\
= & \frac{1}{P_{k_l^j(m)}(G_r^s)} \int_{\{\vec{x}\in G_r^s\cap G_l^j:\ a^{\alpha_r}(\vec{x})=a^{\alpha_r}\}} P_{k_l^j(m)}(\vec{x}) \\
& + \frac{1}{P_{k_l^j(m)}(G_r^s)} \int_{\{\vec{x}\in G_r^s\cap (G_l^j)^c:\ a^{\alpha_r}(\vec{x})=a^{\alpha_r}\}} P_{k_l^j(m)}(\vec{x}) \\
= & \frac{1}{P_{k_l^j(m)}(G_r^s)} \int_{\{\vec{x}\in G_r^s\cap G_l^j:\ a^{\alpha_r}(\vec{x})=a^{\alpha_r}\}} P_{k_l^j(m)-1}(\vec{x}) \cdot \frac{P(a^{\alpha_l}|G_l^j)}{P_{k_l^j(m)-1}(a^{\alpha_l}(\vec{x})|G_l^j)} \\
& + \frac{1}{P_{k_l^j(m)}(G_r^s)} \int_{\{\vec{x}\in G_r^s\cap (G_l^j)^c:\ a^{\alpha_r}(\vec{x})=a^{\alpha_r}\}} P_{k_l^j(m)-1}(\vec{x}).
\end{aligned}
$$

By (3.26),

$$
\begin{aligned}
& P_{k_l^j(m)}(a^{\alpha_r}|G_r^s) \\
\leq & \frac{1}{P_{k_l^j(m)-1}(G_r^s)\cdot(1-C\cdot\sqrt{\epsilon})} \cdot \int_{\{\vec{x}\in G_r^s\cap G_l^j:\ a^{\alpha_r}(\vec{x})=a^{\alpha_r}\}} P_{k_l^j(m)-1}(\vec{x})\cdot(1+C\cdot\sqrt{\epsilon}) \\
& + \frac{1}{P_{k_l^j(m)-1}(G_r^s)\cdot(1-C\cdot\sqrt{\epsilon})} \cdot \int_{\{\vec{x}\in G_r^s\cap (G_l^j)^c:\ a^{\alpha_r}(\vec{x})=a^{\alpha_r}\}} P_{k_l^j(m)-1}(\vec{x})\cdot(1+C\cdot\sqrt{\epsilon}) \\
= & P_{k_l^j(m)-1}(a^{\alpha_r}|G_r^s)\cdot\frac{1+C\cdot\sqrt{\epsilon}}{1-C\cdot\sqrt{\epsilon}}.
\end{aligned}
$$

Similarly, by (3.26),

$$
P_{k_l^j(m)}(a^{\alpha_r}|G_r^s) \geq P_{k_l^j(m)-1}(a^{\alpha_r}|G_r^s)\cdot\frac{1-C\cdot\sqrt{\epsilon}}{1+C\cdot\sqrt{\epsilon}}.
$$

Hence, $\forall l, r \in \{1, 2, \ldots, N\}$, $\forall j, s \in \{1, 2, \ldots, M_r\}$, $\forall a^{\alpha_r}$,

$$\begin{aligned} \frac{P_{k_l^j(m)}(a^{\alpha_r}|G_r^s)}{P_{k_l^j(m)-1}(a^{\alpha_r}|G_r^s)} &\in [\frac{1-C\cdot\sqrt{\epsilon}}{1+C\cdot\sqrt{\epsilon}},\ \frac{1+C\cdot\sqrt{\epsilon}}{1-C\cdot\sqrt{\epsilon}}] \\ &\subset [1-4C\cdot\sqrt{\epsilon},\ 1+4C\cdot\sqrt{\epsilon}]. \end{aligned} \tag{3.27}$$

Step 3. From the general definition of the perturbation (3.16), we can see that $\forall r \in \{1, 2, \ldots, N\}$, $\forall s \in \{1, 2, \ldots, M_r\}$, there exists $k_r^s \in \{1, 2, \ldots, M_s\}$ s.t.

$$P_{k_l^j(m)-k_r^s}(a^{\alpha_r}|G_r^s) = P(a^{\alpha_r}|G_r^s), \quad \forall \alpha_r.$$

Hence, by (3.27),

$$\begin{aligned} \frac{P_{k_l^j(m)}(a^{\alpha_r}|G_r^s)}{P_{k_l^j(m)-k_r^s}(a^{\alpha_r}|G_r^s)} &= \prod_{i=0}^{k_r^s-1} \frac{P_{k_l^j(m)-i}(a^{\alpha_r}|G_r^s)}{P_{k_l^j(m)-(i+1)}(a^{\alpha_r}|G_r^s)} \\ &\in [(1-4C\cdot\sqrt{\epsilon})^{k_r^s}, (1+4C\cdot\sqrt{\epsilon})^{k_r^s}] \\ &\subseteq [(1-4C\cdot\sqrt{\epsilon})^{M_s}, (1+4C\cdot\sqrt{\epsilon})^{M_s}] \\ &\subseteq [1-2^{M_s}\cdot 4C\cdot\sqrt{\epsilon}, 1+2^{M_s}\cdot 4C\cdot\sqrt{\epsilon}]. \end{aligned} \tag{3.28}$$

(See the Appendix for the derivation of (3.28).) Since $P_{k_l^j(m)-k_r^s}(a^{\alpha_r}|G_r^s) = P(a^{\alpha_r}|G_r^s)$, we have

$$\frac{P_{k_l^j(m)}(a^{\alpha_r}|G_r^s)}{P(a^{\alpha_r}|G_r^s)} \in [1-2^{M_s}\cdot 4C\cdot\sqrt{\epsilon}, 1+2^{M_s}\cdot 4C\cdot\sqrt{\epsilon}].$$

By far, we have proved: $\forall \epsilon > 0$ s.t. $\epsilon < \frac{(MIN)^2}{128\ln 2}\cdot 2^{-2M_s}$, $\exists m^*$ s.t. $\forall m > m^*$, $\forall l, r \in \{1, 2, \ldots, N\}$, $\forall j, s \in \{1, 2, \ldots, M_r\}$, $\forall a^{\alpha_r}$,

$$\frac{P_{k_l^j(m)}(a^{\alpha_r}|G_r^s)}{P(a^{\alpha_r}|G_r^s)} \in [1-2^{M_s}\cdot 4C\cdot\sqrt{\epsilon}, 1+2^{M_s}\cdot 4C\cdot\sqrt{\epsilon}],$$

where

$$\begin{aligned} C &= \frac{2\sqrt{2\ln 2}}{MIN} > 0 \\ MIN &= \min_{\{l\,:\,\alpha_l \in \mathcal{B}\setminus\mathcal{T}\}} \min_{j\in\{1,\ldots,M_l\}} \min_{a^{\alpha_l}\in\{a^{\alpha_l}(\vec{x})\}} P(a^{\alpha_l}|x^{\alpha_l}=j) > 0. \end{aligned}$$

The upperbound of ϵ guarantees $2^{M_s}\cdot 4C\cdot\sqrt{\epsilon} < 1$. For any $0 < \delta << 1$, take $\epsilon = \frac{(MIN)^2}{128\ln 2}\cdot 2^{-2M_s}\cdot\delta^2$, and there exists m^* s.t. $\forall m > m^*$, $\forall l, r \in \{1,2,\ldots,N\}$, $\forall j, s \in \{1,2,\ldots,M_r\}$, $\forall a^{\alpha_r}$,

$$\begin{aligned} \frac{P_{k_l^j(m)}(a^{\alpha_r}|G_r^s)}{P(a^{\alpha_r}|G_r^s)} &\in [1-2^{M_s}\cdot 4C\cdot\sqrt{\epsilon}, 1+2^{M_s}\cdot 4C\cdot\sqrt{\epsilon}] \\ &= [1-\delta, 1+\delta]. \end{aligned}$$

Since this is true $\forall l, \forall j$, it is straightforward to rephrase it as: $\forall\delta > 0$ small enough, $\exists k^*$, $\forall k > k^*$, $\forall r \in \{1,2,\ldots,N\}$, $\forall s \in \{1,2,\ldots,N_r\}$, $\forall a^{\alpha_r}$,

$$\frac{P_k(a^{\alpha_r}|G_r^s)}{P(a^{\alpha_r}|G_r^s)} \in [1-\delta, 1+\delta].$$

This implies

$$\begin{aligned} D(P_k(a^{\alpha_r}|G_r^s)\|P(a^{\alpha_r}|G_r^s)) &= \int P_k(a^{\alpha_r}|G_r^s)\cdot\log\frac{P_k(a^{\alpha_r}|G_r^s)}{P(a^{\alpha_r}|G_r^s)}\,da^{\alpha_r} \\ &\leq \int P_k(a^{\alpha_r}|G_r^s)\log(1+\delta)\,da^{\alpha_r} \\ &\leq \delta. \end{aligned}$$

Note that $D(P_k(a^{\alpha_r}|G_r^s)\|P(a^{\alpha_r}|G_r^s))$ is non-negative, therefore

$$D(P_k(a^{\alpha_r}|G_r^s)\|P(a^{\alpha_r}|G_r^s)) \to 0, \text{ uniformly for } r, s, a^{\alpha_r}. \tag{3.29}$$

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. $\forall \vec{x}$,

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

This also indicates

$$D(P_{k_l}(a^{\alpha_r}|G_r^s) \| P^*(a^{\alpha_r}|G_r^s)) \to 0, \quad \text{as } l \to \infty. \tag{3.30}$$

Combining (3.30) with (3.29), we have

$$D(P^*(a^{\alpha_r}|G_r^s) \| P(a^{\alpha_r}|G_r^s)) = 0. \quad \forall r, s, a^{\alpha_r}.$$

With this equality, we can replace P by P^* in the arguments for (3.22) and get

$$D(P_k \| P^*) \text{ monotonically decreases w.r.t } k.$$

But on the other hand we also have $D(P_{k_l} \| P^*) \to 0$, as $l \to \infty$, therefore,

$$D(P_k \| P^*) \to 0, \text{ as } k \to \infty.$$

□

Chapter 4

Maximum-Likelihood Templates

4.1 Introduction

In Bayesian image analysis, a "prior" probability distribution expresses common or learned knowledge about likely and unlikely interpretations, and a (conditional) data model places a probability on image-based observations given any particular interpretation. In conjunction, a prior distribution with a conditional data distribution constitute the "forward" or "generative" model. To the extent that the generative model generates features, as opposed to pixel intensities, the "inverse" or "posterior distribution" on interpretations given images is based on incomplete information; feature vectors are generally insufficient to recover the original intensities. Therefore, it is of great interest to develop a class of data models that generate pixel intensities rather than image features.

The models covered in this chapter are based on image-fragment templates, similar to that introduced by Ullman and his collaborators [26, 27, 28]. Ullman et al. selected as templates the informative image patches that have the highest mutual information with the object class they represent, and use these templates for object classification and segmentation. Other researchers applied interest point detectors on training images, and then extracted image patches around the interest

points obtained, [20], [36], [37] and [38]. They used these cropped image patches as templates or template candidates. Heisele et al. designed a SVM algorithm to select good image patches around manually selected seed points – that minimized the foreground-versus-background classification error – as their templates for facial parts, [30, 31]. In all these methods the templates were selected from image patches cropped out of natural images. However, if there actually exists a model that governs the natural image generation from the templates, it would be improper to use the generated image patches as the hidden templates. Allassonniere et al. accordingly developed a Bayesian framework to learn deformable templates and proposed a detailed parameter estimation by the Expectation-Maximization (EM) algorithm, [39]. Its application on recognition of handwritten digits showed promising results. However, due to the additive noise model, this generative probabilistic model still suffers from not allowing linear change of the gray-scale map. In addition, it is not possible to use this data model to formulate a credible background model for observations outside the object. The deformable templates may handle small variation on scales, but perform badly on spatial shifts, rotations and big scale variations. Sabuncu et al. adapted the method ([39]) on clustering and registering MRI images, [40]. From a training data set of whole brain MR volumns of subjects, they used the EM algorithm to learn two (or three) templates, which corresponded to two (or three) age groups, and used them to partition new datasets. Their model still suffers from the same drawbacks as [39]'s.

Similar to [39], we learn templates through a generative probabilistic model. Good performance in most image analysis applications requires some degree of invariance. In the context of a probability model, the notion of invariance is closely connected to the statistical notion of sufficiency. The normalized correlation between an image patch and an image template is a convenient statistic that is invariant to linear transformations. Only gray-scale images are considered in this chapter. We model pixel grey levels by assuming that their distribution depends only on the

normalized correlation. Rather than using a correlation between a template and a given image patch as an extracted feature, we define a conditional-data model on pixel intensities under which the correlation is assumed to be a sufficient statistic. This produces a tractable forward model, and furthermore a fully specified likelihood function that can be used to learn templates from image data. A training set of eyes, for example, yields an ensemble of templates of left and right eyes, of familiar and natural character, but not actually coming from any particular individuals in the training set. By including mixtures over spatial shifts, scale and rotation variations, our model is able to accommodate invariance with respect to the spatial transformations of objects, in addition to linear transformations of pixel intensities.

The same idea can be adapted to model background image patches. When a small window slides across a natural image, most of patches seen through the window are (or are close to) uniform-colored. This indicates that a uniform-color template needs to be included into the mixture model for background image patches. However, the normalized correlation between an image patch and a uniform-color template is not well defined. Certain adaptation and modification needs to be made, and that will discussed later in this Chapter.

Chapter 1 has presented the compositional machinery, that is composed of two parts: a prior model on image interpretations, and a data model given the terminal bricks of the hierarchy structure. This chapter will focus on the second part of the compositional machinery, i.e. the data model, and is organized as follows. Section 4.2 will develop a probabilistic framework for modeling image patches based on the concepts of templates and sufficient statistics. A basic probabilistic model with a single template will be introduced, and followed by a generalization to multiple templates. A full generalization will be presented after that, to embrace the invariance to scales, rotations and spatial shifts. All the unknown parameters, including the templates themselves, will be learned through an Expectation-Maximization (EM)

algorithm. Section 4.3 will implement this method to model facial parts, and demonstrates its direct application on ethnicity classification of face images. Section 4.4 will focus on the adapted model for background image patches. Finally, Section 4.5 will conclude with a discussion about this model and suggestions about future work.

4.2 Generative Probabilistic Model

Key notations: T represents an image template. Y represents an observed gray-scale image patch with n pixels. Each pixel of Y takes its value from the finite set $\{0, 1, \ldots, 255\}$, unless otherwise specified. $S = S(Y, T)$ is the normalized correlation between T and Y. (Unless otherwise specified, we will use $S(Y, T)$, $S(Y)$, and S interchangeably.)

4.2.1 Basic Probabilistic Model with a Single Template

We want to model the probability distribution of Y conditioned on it representing a certain object category. For the purpose of discussion, let this object category be an "eye." (Unless otherwise specified, $P(Y|\text{eye})$ and $P(Y)$ are going to be used interchangeably through this chapter, to represent the conditional probability of Y.)

Our plan is to model $P(Y)$ based on S being a sufficient statistics of Y, by assuming that all the Ys are equally likely given the same S. We will first describe the sampling procedure of Y from $P(Y)$, and this will uniquely determine the expression of $P(Y)$. As specified earlier, Y is a discrete variable from the domain $\{0, 1, \ldots, 255\}^n$, but at this moment, let us suppose for a moment that Y is continuous – assuming each pixel of Y takes its value from interval $[0, 256)$ – and this temporary modification will ease the understanding of the sampling procedure of Y. Assume T is an eye template, the same size as Y. S is defined as the normalized correlation between Y and T, hence, it takes its value from the interval $[-1, 1]$. The sampling procedure of Y is composed of two steps: step 1, $S = s$ is sampled from

$[-1,+1]$ according to a certain probability distribution of S, denoted as P_S; step 2, given $S=s$, Y is sampled uniformly from the set $\{Y\in[0,256)^n : S(Y,T)=s\}$, i.e., the set of all possible image patches satisfying $S(Y,T)=s$. (Now it becomes clear that why Y's continuity is needed for describing its sampling procedure here, since otherwise the set $\{Y\in[0,256)^n : S(Y,T)=s\}$ might be empty.) The generating procedure of Y described above uniquely determines $P(Y)$,

$$P(Y)=P(Y,S(Y))=P(S(Y))\cdot P(Y|S(Y))=\frac{P(S(Y))}{c(S(Y))},\quad \forall Y\in[0,256)^n,$$

where $c(S(Y))$ is the combinatorial factor, the measure of the set $\{\tilde{Y}\in[0,256)^n : S(\tilde{Y})=S(Y)\}$.,

Notice. An essential assumption is made here, and will be in effect through out this chapter: Given $S(Y,T)=s$, all the Ys are equally likely. S can be seen as a sufficient statistics of Y.

Now let us come back to the discrete case, i.e. $Y\in\{0,1,\ldots,255\}^n$. From this point until the end of this Chapter, Y will be a discrete variable taking its value from $\{0,1,\ldots,255\}^n : S(Y,T)=s\}$. We will model the distribution of a discrete Y similarly as the continuous case, i.e.

$$P(Y)=\frac{P(S(Y))}{c(S(Y))},\quad \forall Y\in\{0,1,\ldots,255\}^n, \tag{4.1}$$

where $c(S(Y))$ is the combinatorial factor, counting the number of $\tilde{Y}$s that satisfy $S(\tilde{Y})=S(Y)$, i.e., $c(S(Y))=\#\{\tilde{Y}\in\{0,1,\ldots,255\}^n : S(\tilde{Y})=S(Y)\}$. But we need to be careful, since the set $\{Y\in\{0,1,\ldots,255\}^n : S(Y,T)=s\}$ might be empty for a random $s\in[-1,+1]$. (See the Appendix for the importance sampling method we proposed for sampling discrete Y uniformly from the set $\{Y\in\{0,1,\ldots,255\}^n : S(Y,T)=s\}$.) Some rigorous mathematical derivation is needed here, in order to evaluate $P(Y)$. Two things need to be specified in (4.1): the marginal distribution of S, $P(S(Y))$, and the combinatorial factor, $c(S(Y))$.

Let us first look at $c(S(Y))$. It is not hard to realize that a direct calculation of $c(S(Y) = s)$ is intractable, considering the high dimensionality of Y. What is needed is an approximation to $c(s)$. Here, we approximate it through the Central Limit Theorem (CLT). Consider a thought experiment: Let $\tilde{Y}$ be another image patch with n pixels. Each pixel of $\tilde{Y}$ is distributed according to i.i.d. uniform distribution (denoted as $\tilde{P}$) on $\{0, 1, \ldots, 255\}$. Since $\tilde{P}(\tilde{Y}) = (1/256)^n \ \forall \ \tilde{Y}$, the "sufficiency assumption" is exactly true under $\tilde{P}$ - all configurations of $\tilde{Y}$ that give the same value of $S(\tilde{Y}) = s$ are equally likely. The marginal distribution of S under $\tilde{P}$ is

$$\tilde{P}(S = s) = \sum_{\tilde{Y}: S(\tilde{Y})=s} \tilde{P}(\tilde{Y}) = (\frac{1}{256})^n \cdot c(s).$$

It implies,

$$c(s) = (256)^n \cdot \tilde{P}(S = s).$$

Let $\tilde{F}_s(S)$ be the cumulative distribution function of S under $\tilde{P}$. By the Central Limit Theorem (CLT), as n goes to infinity, $\tilde{F}_s(S)$ converges to the cumulative function of a normal distribution with mean μ and variance σ^2, where $\mu = 0, \sigma = \frac{1}{\sqrt{n}}$. (See the Appendix for the detailed derivation.) Hence, when n is large,

$$\tilde{F}_s(S = s) \approx \int_{-\infty}^{s} \frac{\sqrt{n}}{\sqrt{2\pi}} \, e^{-\frac{nx^2}{2}} \, dx. \tag{4.2}$$

Since the asymptotic distribution $N(\mu, \sigma^2)$ is continuous while the distribution of S is discrete for any finite n, we have to be more careful in order to take advantage of the density function of $N(\mu, \sigma^2)$. Let δ be a small positive number. $\delta_k = k \cdot \delta$, $k = 0, \pm 1, \pm 2, \ldots$. Let $s_\delta(Y)$ be $S(Y)$ truncated to the largest δ_k s.t. $\delta_k \leq S(Y)$, i.e.,

$$s_\delta(Y) \stackrel{\triangle}{=} \delta_{k^*}, \quad \text{where } k^* = \text{argmax}_{\{k \in \{0, \pm 1, \pm 2, \ldots\}: \ \delta_k \leq S(Y)\}} \ \delta_k.$$

A few notations are defined as follows for later use:

$$\begin{aligned} P_\delta(s_\delta) &\triangleq P(\ S(Y) \in [s_\delta, s_\delta + \delta)\ ), \quad s_\delta \in \{i\delta\}_{i=-\infty}^{\infty} \\ \tilde{P}_\delta(s_\delta) &\triangleq \tilde{P}(\ S(\tilde{Y}) \in [s_\delta, s_\delta + \delta)\ ), \\ c_\delta(s_\delta) &\triangleq \#\{\ Y:\ S(Y) \in [s_\delta, s_\delta + \delta)\ \} \ = \ \#\{\ \tilde{Y}:\ S(\tilde{Y}) \in [s_\delta, s_\delta + \delta)\ \}. \end{aligned}$$

Simply by the definition of the notations above, we have

$$P_\delta(s_\delta) \ = \sum_{Y:\ S(Y)\in[s_\delta, s_\delta+\delta)} P(Y) \ \approx \ P(Y) \cdot c_\delta(s_\delta). \tag{4.3}$$

$$\tilde{P}_\delta(s_\delta) \ = \sum_{\tilde{Y}:\ S(\tilde{Y})\in[s_\delta, s_\delta+\delta)} \tilde{P}(\tilde{Y}) \ = \ (\frac{1}{256})^n \cdot c_\delta(s_\delta). \tag{4.4}$$

Eqn. (4.4) gives

$$c_\delta(s_\delta) = (256)^n \cdot \tilde{P}_\delta(s_\delta(Y)).$$

After plugging it into (4.3), we get

$$P(Y) \approx \frac{P_\delta(s_\delta)}{c_\delta(s_\delta)} = \frac{P_\delta(s_\delta)}{(256)^n \cdot \tilde{P}_\delta(s_\delta)}. \tag{4.5}$$

In order to get the final expression for $P(Y)$, we need to specify $P_\delta(s_\delta)$ and $\tilde{P}_\delta(s_\delta)$. From (4.2), when n is large,

$$\begin{aligned} \tilde{P}_\delta(s_\delta) &= \tilde{F}_s(s_\delta + \delta) - \tilde{F}_s(s_\delta) \\ &\approx \int_{s_\delta}^{s_\delta+\delta} \frac{\sqrt{n}}{\sqrt{2\pi}}\ e^{-\frac{nx^2}{2}}\ dx \\ &\approx \delta \cdot \frac{\sqrt{n}}{\sqrt{2\pi}}\ e^{-\frac{n(s_\delta)^2}{2}} \\ &\approx \delta \cdot \frac{\sqrt{n}}{\sqrt{2\pi}}\ e^{-\frac{n(S(Y))^2}{2}}, \end{aligned} \tag{4.6}$$

where the last approximation in (4.6) is due to $s_\delta \approx S(Y)$, by the definition of s_δ.

Concerning $P_\delta(s_\delta)$, theoretically, this is a discrete probability function of $\{S(Y) : Y \in \{0, 1, \ldots, 255\}^n\}$. Given that $\hat{F}_s(S = s)$ converges to a cumulative distribution function of a normal distribution, the discrete domain of $S(Y)$ is dense in [-1,1] asymptotically, as n goes to infinity. Therefore, when n is large, the cumulative distribution function of $S(Y)$, namely $F(S = s)$, can be approximated by a cumulative distribution function $\hat{F}_s(S = s)$ of a "continuous" distribution with a density function $\hat{P}_s(S = s)$. (The hat " ˆ " stands for "continuous.") Hence,

$$\begin{aligned} P_\delta(s_\delta) &= F(s_\delta + \delta) - F(s_\delta) \\ &\approx \hat{F}_s(s_\delta + \delta) - \hat{F}_s(s_\delta) \\ &= \int_{s_\delta}^{s_\delta+\delta} \hat{P}_s(x)\, dx \\ &\approx \delta \cdot \hat{P}_s(s_\delta) \\ &\approx \delta \cdot \hat{P}_s(S(Y)). \end{aligned} \tag{4.7}$$

Combining (4.5), (4.6), and (4.7), we have,

$$P(Y) \approx \frac{P_\delta(s_\delta)}{(256)^n \cdot \tilde{P}_\delta(s_\delta)} \tag{4.8}$$

$$\approx \frac{\delta \cdot \hat{P}_s(S(Y))}{(256)^n \cdot \delta \cdot \frac{\sqrt{n}}{\sqrt{2\pi}}\, e^{-\frac{n(S(Y))^2}{2}}} \tag{4.9}$$

$$= \frac{\hat{P}_s(S(Y))}{(256)^n \cdot \frac{\sqrt{n}}{\sqrt{2\pi}}\, e^{-\frac{n(S(Y))^2}{2}}}. \tag{4.10}$$

Now we need to specify $\hat{P}_s(S)$, a density function of a continuous random variable on $[-1, 1]$, approximating $P(S)$. Given that Y is generated from an eye template T, it is reasonable to model $\hat{P}_s(S = s)$ to be monotone increasing on $s \in [-1, 1]$. We model $\hat{P}_s(S = s)$ as a backwards truncated exponential distribution. (Actually, this distribution is close to the empirical histogram of $S(Y, T)$ as well. For example,

simply pick T to be one of the training eye image patches, and plot the histogram of the normalized correlation S between T and all the rest of eye image patches. With enough eye image patches, say 100, we will observe that the histogram of S is roughly proportional to a backwards truncated exponential distribution.)

$$\hat{P}_s(S = s) \propto \lambda \, e^{-\lambda \, (1-s)}, \qquad \forall s \in [-1, 1],$$

where λ is the distribution parameter, controlling the steepness. The greater λ is, the more S concentrates at $+1$. After normalization, $\hat{P}_s(S = s)$ becomes

$$\hat{P}_s(S = s) = \frac{1}{1 - e^{-2\lambda}} \, \lambda \, e^{-\lambda \, (1-s)}, \qquad \forall s \in [-1, 1]. \tag{4.11}$$

After plugging (4.11) into (4.10), we get the final expression for $P(Y)$,

$$P(Y) = \frac{\frac{1}{1-e^{-2\lambda}} \, \lambda \, e^{-\lambda \, (1-S(Y))}}{Q \cdot (256)^n \cdot \frac{\sqrt{n}}{\sqrt{2\pi}} \, e^{-\frac{n(S(Y))^2}{2}}}, \tag{4.12}$$

where Q is a number that changes the approximations from (4.8, 4.9) to the equality from (4.12). Q is very close to 1 when n is large.

4.2.2 Generalized Probabilistic Model with Multiple Templates.

In general, objects from a particular category can not be well represented by only one single template. Object classes are often defined as a combination of structures with distinct characteristics. Taking the "eye" category for example, there is a big variation of its appearance among humans from different ethnic groups. It is therefore natural to generalize the model to embrace multiple templates, and learn the associated weights of each mixture together with all the other unknown parameters.

Suppose there exist N_t eye templates $\{T_1, \ldots, T_t, \ldots, T_n\}$, each with the same

size as Y and each associated with a weight ϵ_t, where $\sum_t \epsilon_t = 1$. We model the distribution of Y as a mixture over N_t templates,

$$P(Y|\text{eye}) = \sum_{t=1}^{N_t} \epsilon_t \ \cdot \ P(Y|T_t), \tag{4.13}$$

where each $P(Y|T_t)$ is modeled the same as $P(Y)$ with a single template from Section 4.2.1. After (4.12) is plugged in, the mixture distribution (4.13) becomes

$$\begin{aligned} P(Y|\text{eye}) &= \sum_{t=1}^{N_t} \epsilon_t \ \cdot \ P(Y|T_t) \\ &= \sum_{t=1}^{N_t} \epsilon_t \ \cdot \ \frac{\frac{1}{1-e^{-2\lambda_t}} \ \lambda_t \ e^{-\lambda_t \ (1-S_t(Y))}}{Q_t \cdot (256)^n \frac{\sqrt{n}}{\sqrt{2\pi}} e^{-\frac{nS_t(Y)^2}{2}}}, \end{aligned} \tag{4.14}$$

where $S_t(Y) = S(Y, T_t)$ is the normalized correlation between Y and the t^{th} template T_t; Q_t is a constant, and very close to 1 when n is large.

Parameter Learning. There are a large amount of unknown parameters in this probabilistic model (4.14), that include all the pixel values of N_t image templates, $\{T_t = (\tau_1^{(t)}, \tau_2^{(t)}, \ldots, \tau_n^{(t)})'\}_{t=1}^{N_t}$, in addition to $2N_t$ scalar parameters, $\{\lambda_t\}$ and $\{\epsilon_t\}$. As mentioned earlier in this chapter, we learn all the unknown parameters (including the templates themselves) by Expectation-Maximization algorithm. Suppose there are N eye image patches $\{Y_i = (y_1^{(i)}, y_2^{(i)}, \ldots, y_n^{(i)})'\}_{i=1}^{N}$ for training. Assuming that they are i.i.d. samples from the model $P(Y|\text{eye})$, the likelihood function is:

$$\begin{aligned} P(\vec{Y}|\text{eye}) &= \prod P(Y_i|\text{eye}) \\ &= \prod_{i=1}^{N} \sum_{t=1}^{N_t} \epsilon_t \ \cdot \ \frac{\frac{1}{1-e^{-2\lambda_t}} \ \lambda_t \ e^{-\lambda_t \ (1-S_t(Y_i))}}{Q_t \ (256)^n \frac{\sqrt{n}}{\sqrt{2\pi}} e^{-\frac{nS_t(Y_i)^2}{2}}}. \end{aligned} \tag{4.15}$$

Note that S is invariant to linear transformation of templates, hence in order to uniquely identify the templates, we require that all the templates are normalized,

with mean 0 and variance 1. To simplify the computation, all the training eye image patches are pre-normalized as well, such that $\sum_k y_k^{(i)} = 0, \sum_k {y_k^{(i)}}^2 = 1$. In this way, $S_t(Y_i)$ becomes the inner product of Y and T_t, i.e., $S_t(Y_i) = \sum_k y_k^{(i)} \cdot \tau_k^{(t)}$. Since in (4.14) Q_t is very close to 1 when n is large, we replace Q_t with 1 to simplify our computation. The vector of all the unknown parameters is denoted as $\vec{\theta}$.

Expectation Step. $\forall i$,

$$\begin{aligned} \hat{P}_t^{(i)} &= P(X_i = t | Y_i, \vec{\theta}^{(c)}) \\ &= \frac{P_t^{(c)}(Y_i) \cdot \epsilon_t^{(c)}}{\sum_{t=1}^{N_t} P_t^{(c)}(Y_i) \cdot \epsilon_t^{(c)}}, \end{aligned}$$

where $X_i = t$ means that the t^{th} template T_t generated Y_i; the form $\theta^{(c)}$ stands for the "current" guess of θ and

$$P_t^{(c)}(Y_i) = \frac{\frac{1}{1-e^{-2(\lambda_t)^{(c)}}} \ (\lambda_t)^{(c)} \ e^{-(\lambda_t)^{(c)} \ (1-S_t(Y_i))}}{Q_t \ (256)^n \frac{\sqrt{n}}{\sqrt{2\pi}} e^{-\frac{nS_t(Y_i)^2}{2}}}.$$

Maximization Step. Maximize

$$\begin{aligned} B &= \sum_t \sum_i \hat{P}_t^{(i)} \cdot \log \left[\epsilon_t \cdot P(Y_i | X_i = t, \vec{\theta}) \right] \\ &= \sum_t \sum_i \hat{P}_t^{(i)} \cdot \log(\epsilon_t) + \sum_t \sum_i \hat{P}_t^{(i)} \cdot \log \left(\frac{\frac{1}{1-e^{-2\lambda_t}} \ \lambda_t \ e^{-\lambda_t \ (1 - \sum_k y_k^{(i)} \cdot \tau_k^{(t)})}}{(256)^n \frac{\sqrt{n}}{\sqrt{2\pi}} e^{-\frac{n(\sum_k y_k^{(i)} \cdot \tau_k^{(t)})^2}{2}}} \right), \end{aligned}$$

over $\{\epsilon_t\}_t, \{\lambda_t\}_t, \{T_t\}_t$ subject to:

$$\sum_{t=1}^{N_t} \epsilon_t = 1; \quad \sum_{k=1}^{n} \tau_k^{(t)} = 0; \quad \sum_{k=1}^{n} (\tau_k^{(t)})^2 = 1.$$

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

$$\begin{aligned}\epsilon_t &= \frac{1}{N}\sum_{i=1}^{N}\hat{P}_t^{(i)} \\ \lambda_t &= \frac{2\lambda_t e^{-2\lambda_t}+e^{-2\lambda_t}-1}{e^{-2\lambda_t}-1}\cdot\frac{\sum_{i=1}^{N}\hat{P}_t^{(i)}}{\sum_{i=1}^{N}\hat{P}_t^{(i)}\cdot(1-\sum y_k^{(i)}\tau_k^{(t)})},\end{aligned}$$

where λ_t can be identified by a simple numerical method, for example, Newton's method or binary search. The non-trivial part is to solve out the N_t unknown templates, $\{T_t=(\tau_1^{(t)},\tau_2^{(t)},\ldots,\tau_n^{(t)})'\}_{t=1}^{N_t}$. For any $t\in\{1,2,\ldots,N_t\}$, maximizing B over T_t is equivalent to

$$\max_{\{\tau_k^{(t)}\}_{k=1}^{n}}\hat{B}, \qquad \text{s.t. } \sum_{k=1}^{n}\tau_k^{(t)}=0,\ \sum_{k=1}^{n}(\tau_k^{(t)})^2=1, \tag{4.16}$$

where

$$\hat{B}=\sum_i\hat{P}_t^{(i)}\left[\lambda_t\cdot\sum_k y_k^{(i)}\tau_k^{(t)}+\frac{n}{2}\cdot(\sum_k y_k^{(i)}\tau_k^{(t)})^2\right]$$

This is a constrained quadratic maximization problem, and can be solved through traditional methods, for example, the gradient ascent method. However, there is a better way to attack this problem directly, through matrix decomposition and changing variables. This direct method will yield a *c*losed form solution in one *s*inge step. Let

$$V=\sum_i\lambda_t\hat{P}_t^{(i)}Y_i \qquad A=\sum_i\frac{n}{2}\hat{P}_t^{(i)}Y_i\ (Y_i)^\top.$$

V is a column vector; A is a $n\times n$ semi-positive definite matrix. Let $\{(a_m,e_m)\}_{m=1}^{n}$ be the pairs of eigenvalues and corresponding eigen vectors of A. Let $\{v_m\}_{m=1}^{n}$ and $\{\hat{\tau}_m\}_{m=1}^{n}$ be the decomposition coefficients of V and T_t with respect to the new base

vectors $\{e_m\}_m$. We have the following decomposition of A, V, and T_t:

$$A = \sum_m a_m e_m e_m^\top, \quad V = \sum_m v_m e_m, \quad T_t = \sum_m \hat{\tau}_m e_m.$$

Denote $e = (1, \ldots, 1)^\top$, a column vector with n 1s. Let $s_m = e^\top \cdot e_m$, then the constrained maximization problem defined in (4.16) becomes

$$\max_{\{\hat{\tau}_m\}_{m=1}^n} \hat{B}, \qquad \text{s.t. } \sum_m \hat{\tau}_m s_m = 0, \ \sum_m \hat{\tau}_m^2 = 1, \tag{4.17}$$

where

$$\hat{B} = V^\top T_T + T_t^\top A T_t = \sum_m v_m \hat{\tau}_m + \sum_m a_m \hat{\tau}_m^2.$$

Now the question is how to solve this transformed constrained optimization problem (4.17). We will take the traditional Lagrange Multipliers method. Define

$$\hat{L} = \sum_m v_m \hat{\tau}_m + \sum_m a_m \hat{\tau}_m^2 - \zeta(\sum_m \hat{\tau}_m s_m) - \eta(\sum_m \hat{\tau}_m^2 - 1).$$

Take derivatives of $\hat{L}$ w.r.t. $\hat{\tau}_m$, ζ, and η, and we have

$$\frac{\partial \hat{L}}{\partial \hat{\tau}_m} = 0 \quad \Longrightarrow \quad \hat{\tau}_m = \frac{v_m - \zeta s_m}{2(\eta - a_m)}. \tag{4.18}$$

$$\sum_m \hat{\tau}_m s_m = 0 \quad \Longrightarrow \quad \zeta = \frac{\sum \frac{v_m s_m}{\eta - a_m}}{\sum \frac{s_m^2}{\eta - a_m}}. \tag{4.19}$$

$$\sum_m \hat{\tau}_m^2 = 1 \quad \Longrightarrow \quad \sum_m \frac{v_m^2}{4(\eta - a_m)^2} = 1. \tag{4.20}$$

Equation (4.19) implies

$$\zeta = 0,$$

due to the fact that

$$v_m \cdot s_m = 0, \ \forall m \in \{1, 2, \ldots, n\}. \tag{4.21}$$

(See the Appendix for the proof of (4.21)). Now consider the equation w.r.t. η in (4.20). Generally, there exist $(n-2)$ roots, $\{\eta_r\}_{r=1}^{n-2}$, that solve this equation. Among these $(n-2)$ roots of η, we pick η^* that maximizes $\hat{B}$, i.e.,

$$\begin{aligned}
\eta^* &= \underset{\{\eta_r\}_{r=1}^{n-2}}{\text{argmax}}\ \hat{B} \\
&= \underset{\{\eta_r\}_{r=1}^{n-2}}{\text{argmax}}\ \sum_m v_m\hat{\tau}_m + \sum_m a_m\hat{\tau}_m^2 \\
&= \underset{\{\eta_r\}_{r=1}^{n-2}}{\text{argmax}}\ \frac{v_m^2}{2(\eta - a_m)} + \frac{a_m v_m^2}{4(\eta - a_m)^2}.
\end{aligned}$$

After plugging $\zeta = 0$ and $\eta = \eta^*$ into (4.18), we get

$$\hat{\tau}_m^* = \frac{v_m}{2(\eta^* - a_m)}.$$

And this uniquely determines T_t,

$$T_t = \sum_m \hat{\tau}_m^* e_m.$$

4.2.3 Further Generalized Probabilistic Model with Multiple Scales, Rotations and Location Shifts.

This section studies the case when an eye in an image patch is neither horizontally located in the center nor with a fixed scale. To capture the invariance of the object to different scales, rotations and spatial shifts, we generalize the previous model (4.14) to include mixtures over discrete scales, rotations and spatial shifts of templates within the image. Besides shifts by pixels, the discrete location shifts can be fraction of a pixel, that enables the templates to find better locations to fit the image data. With this generalized model, to generate an eye image patch Y, we first randomly select a template T, a scale s, a rotation r, and a spatial shift l, then scale and rotate T according to s and r. Second, we use this scaled and rotated template to generate

$\tilde{Y}$ – a sub-region of Y that is determined uniquely by (s, r, l) – within Y through sufficient statistics as before. Finally, we fill the rest of Y, i.e. the complement of $\tilde{Y}$, with i.i.d. random noise from $\{0, 1, \ldots, 255\}$.

Notice. In this further generalized model, T can have a different size from Y due to the fact that T can scale, rotate, and shift within Y. For example, if Y is 10×15, then T can be 3×2 or 30×40, as long as all the templates have the same size. This indicates that T and $\tilde{Y}$ (mentioned in the sampling procedure above) will have different resolutions. Hence, in order to generate $\tilde{Y}$ from T, we will have to "stretch" T (thinking T as a rubber sheet) to match the pixel coordinate of Y first. We call this process a "projection" of T. This projection – at rotation r, scale s, and spatial shift l – will be obtained through a projection matrix $M_{r,s,l}$, whose derivation will be described in the Appendix.

Suppose each template T_t 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 $\forall l \in Q_{s,r}$, let $g_{s,r,l}(T_t)$ be the projection of T_t down to the pixel coordinate of Y, under scale s, rotation r, and spatial shift l. Let ϵ_t be the prior weight associated with each template T_t, δ_s^t be the chance that scale s is selected for T_t, and η_r^t be the chance that rotation r is selected for T_t. Hence, $\sum_{t=1}^{N_t} \epsilon_t = 1$, and $\forall t, \sum_{s=1}^{N_s} \delta_s^t = 1$, $\sum_{r=1}^{N_r} \eta_r^t = 1$. Considering an object is almost equally likely to appear anywhere in an image patch, we model the distribution of l is uniform from the set $Q_{s,r}$, i.e. $P(l) = \frac{1}{\|Q_{s,r}\|}, \forall l \in Q_{s,r}$, where $\|Q_{s,r}\|$ is the counting measure of set $Q_{s,r}$. Let $Y^{s,r,l}$(i.e. $\tilde{Y}$ in the previous paragraph) be the sub-region of Y covered by $g_{s,r,l}(T_t)$. Let n be the total number of pixels in Y, while $n_{s,r,l}$ is the number of pixels in $Y^{s,r,l}$.

$$P(Y|\text{eye}) = \sum_{t=1}^{N_t} \sum_{s=1}^{N_s} \sum_{r=1}^{N_r} \sum_{l \in Q_{s,r}} \epsilon_t \, \delta_s^t \, \eta_r^t \, \frac{1}{\|Q_{s,r}\|} \cdot \, P(Y^{s,r,l} | g_{s,r,l}(T_t)) \cdot \, P_o((Y^{s,r,l})^c), \quad (4.22)$$

where $P_o((Y^{s,r,l})^c)$ stands for the probability density function of the area of Y that

is not covered by $g_{s,r,l}(T_t)$. Since $(Y^{s,r,l})^c$ is filled with i.i.d. uniform noise,

$$P_o((Y^{s,r,l})^c) = (\frac{1}{256})^{n-n_{s,r,l}}. \tag{4.23}$$

Now consider $P(Y^{s,r,l}|g_{s,r,l}(T_t))$. Note that $Y^{s,r,l}$ and $g_{s,r,l}(T_t)$ contain the same amount of pixels, $n_{s,r,l}$. It is no different to model the distribution of $Y^{s,r,l}$ given $g_{s,r,l}(T_t)$, from modeling the distribution of an image patch given a single template in Section 4.2.1. Therefore,

$$P(Y^{s,r,l}|g_{s,r,l}(T_t)) = \frac{\frac{1}{1-e^{-2\lambda_t}}\ \lambda_t\ e^{-\lambda_t\ (1-S_{t,s,r,l}(Y^{s,r,l}))}}{Q_{t,s,r,l}\cdot(256)^{n_{s,r,l}}\cdot\frac{\sqrt{n_{s,r,l}}}{\sqrt{2\pi}}e^{-\frac{n_{s,r,l}\cdot(S_{t,s,r,l}(Y^{s,r,l}))^2}{2}}}, \tag{4.24}$$

where $S_{t,s,r,l}(Y)$ is the normalized correlation between $Y^{s,r,l}$ and $g_{s,r,l}(T_t))$, λ_t is the parameter associated with the backward truncated exponential distribution of $S_{t,s,r,l}(Y)$, and $Q_{t,s,r,l}$ is a constant, that is close to 1 when $n_{s,r,l}$ is large enough. Plugging (4.23) and (4.24) back into (4.22), we get

$$P(Y|\text{eye}) \ = \ \sum_{t,s,r,l}\epsilon_t\ \delta_s^t\ \eta_r^t\ \times\ (\frac{1}{256})^n\ \frac{\frac{1}{1-e^{-2\lambda_t}}\ \lambda_t\ e^{-\lambda_t\ (1-S_{t,s,r,l}(Y^{s,r,l}))}}{Q_{t,s,r,l}\cdot\frac{\sqrt{n_{s,r,l}}}{\sqrt{2\pi}}e^{-\frac{n_{s,r,l}\cdot(S_{t,s,r,l}(Y^{s,r,l}))^2}{2}}}. \tag{4.25}$$

Parameter Learning. All the unknown parameters in (4.25) are learned by the EM algorithm, similarly as in Section 4.2.2. However, the matrix decomposition method in Section 4.2.2 does not work here for updating T_t in M-step. We will describe later an alternative way to update T_t, the direct gradient ascent method. Define

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

where $M_{s,r,l}$ stands for the projection matrix of T_t down to the coordinate of image patch Y, under scale s, rotation r, and location shift 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 example, $M_{s,r,l}(i,j)$ can be simply replaced with ($M_{s,r,l}(i,j)-$ the mean of the j^{th}column of $M_{s,r,l}$). (See the Appendix for the derivation of $M_{s,r,l}$.) Suppose there are N eye images $\{Y_i = (y_1^{(i)}, y_2^{(i)}, \ldots, y_n^{(i)})'\}_{i=1}^N$ (can be of different sizes) for training. Assuming that they are i.i.d. samples from the density function (4.25), the likelihood function of these N training images is:

$$\begin{aligned} P(\vec{Y}|\text{eye}) &= \prod_{i=1}^{N} P(Y_i|\text{eye}) \\ &= \prod_i \sum_{t,s,r,l} \epsilon_t \ \delta_s^t \ \eta_r^t \ \times \ (\frac{1}{256})^n \ \frac{\frac{1}{1-e^{-2\lambda_t}} \ \lambda_t \ e^{-\lambda_t \ (1-S_{t,s,r,l}(Y_i^{s,r,l}))}}{Q_{t,s,r,l} \cdot \frac{\sqrt{n_{s,r,l}}}{\sqrt{2\pi}} e^{-\frac{n_{s,r,l} \cdot S_{t,s,r,l}(Y_i^{s,r,l})^2}{2}}}. \end{aligned}$$

To simply the computation and notation, we pre-normalized all $Y_i^{s,r,l}$ s.t. it has mean 0 and variance 1, and that leads to

$$S_{t,s,r,l}(Y_i^{s,r,l}) = \frac{(Y_i^{s,r,l})^\top M_{s,r,l} T_t}{\sqrt{T_t^\top M_{s,r,l}^\top M_{s,r,l} T_t}}.$$

Again we approximated $Q_{t,s,r,l}$ by 1, if we assume $n_{s,r,l}$ is large enough for all s, r, l. For example, $\min_{s,r,l} n_{s,r,l} \geq 100$.

Expectation Step. $\forall i$,

$$\begin{aligned} \hat{P}_{(t,s,r,l)}^{(i)} &= P(X_i = (t,s,r,l)|Y_i, \vec{\theta}^{(c)}) \qquad (4.26) \\ &= \frac{1_{l \in Q_{s,r}} \cdot \frac{1}{\|Q_{s,r}\|} \epsilon_t^{(c)} (\delta_s^t)^{(c)} (\eta_r^t)^{(c)} \cdot P_{(t,s,r,l)}^{(c)}(Y_i)}{\sum_{t,s,r,l} \frac{1}{\|Q_{s,r}\|} \epsilon_t^{(c)} (\delta_s^t)^{(c)} (\eta_r^t)^{(c)} \cdot P_{(t,s,r,l)}^{(c)}(Y_i)}, \end{aligned}$$

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

$$P_{(t,s,r,l)}^{(c)}(Y_i) = P(Y_i|X_i = (t,s,r,l), \vec{\theta}^{(c}) = (\frac{1}{256})^n \ \frac{\frac{1}{1-e^{-2\lambda_t^{(c)}}} \ \lambda_t^{(c)} \ e^{-\lambda_t^{(c)} \ (1-S_{t,s,r,l}(Y_i^{s,r,l}))}}{\frac{\sqrt{n_{s,r,l}}}{\sqrt{2\pi}} e^{-\frac{n_{s,r,l} \cdot (S_{t,s,r,l}(Y_i^{s,r,l}))^2}{2}}}.$$

Actually, $(\frac{1}{256})^n$ get canceled in both the numerator and the denominator in (4.27).

Maximization Step. Maximize

$$
\begin{aligned}
B &= \sum_{t,s,r,l}\sum_{i}\hat{P}^{(i)}_{(t,s,r,l)}\cdot\log\left[\frac{1}{\|Q_{s,r}\|}\epsilon_t\ \delta_s^t\ \eta_r^t\cdot P(Y_i|X_i=(t,s,r,l),\vec{\theta}\,)\right] \\
&= \sum_{t,s,r,l}\sum_{i}\hat{P}^{(i)}_{(t,s,r,l)}\cdot\log(\frac{1}{\|Q_{s,r}\|}\epsilon_t\ \delta_s^t\ \eta_r^t) \\
&\quad + \sum_{t,s,r,l}\sum_{i}\hat{P}^{(i)}_{(t,s,r,l)}\cdot\log\left((\frac{1}{256})^n\ \frac{\frac{1}{1-e^{-2\lambda_t}}\ \lambda_t\ e^{-\lambda_t\ (1-S_{t,s,r,l}(Y_i^{s,r,l}))}}{\frac{\sqrt{n_{s,r,l}}}{\sqrt{2\pi}}e^{-\frac{n_{s,r,l}\cdot(S_{t,s,r,l}(Y_i^{s,r,l}))^2}{2}}}\right),
\end{aligned}
$$

over $\left\{\{T_t\}_{t=1}^{N_t},\{\epsilon_t\}_{t=1}^{N_t},\{\lambda_t\}_{t=1}^{N_t},\{\delta_s^t\}_{s,t},\{\eta_r^t\}_{r,t}\right\}$ subject to:

$$
\sum_{t=1}^{N_t}\epsilon_t=1;\ \sum_{s=1}^{N_s}\delta_s^t=1,\ \sum_{r=1}^{N_r}\eta_r^t=1,\ \forall t.
$$

It is straightforward to solve out $\{\lambda_t\}$, $\{\epsilon_t\}$, $\{\delta_s^t\}$ and $\{\eta_r^t\}$,

$$
\begin{aligned}
\epsilon_t &= \frac{\sum_{i,s,r,l}\hat{P}^{(i)}_{(t,s,r,l)}}{\sum_{i,t,s,r,l}\hat{P}^{(i)}_{(t,s,r,l)}} \\
&= \frac{1}{N}\sum_{i,s,r,l}\hat{P}^{(i)}_{(t,s,r,l)}, \\
\delta_s^t &= \frac{\sum_{i,r,l}\hat{P}^{(i)}_{(t,s,r,l)}}{\sum_{i,s,r,l}\hat{P}^{(i)}_{(t,s,r,l)}}, \\
\eta_r^t &= \frac{\sum_{i,s,l}\hat{P}^{(i)}_{(t,s,r,l)}}{\sum_{i,s,r,l}\hat{P}^{(i)}_{(t,s,r,l)}}, \\
\lambda_t &= \frac{2\lambda_t e^{-2\lambda_t}+e^{-2\lambda_t}-1}{e^{-2\lambda_t}-1}\cdot\frac{\sum_{i,s,r,l}\hat{P}^{(i)}_{(t,s,r,l)}}{\sum_{i,s,r,l}\hat{P}^{(i)}_{(t,s,r,l)}\cdot(1-S_{t,s,r,l}(Y_i^{s,r,l}))},
\end{aligned}
$$

where λ_t can be identified by a numerical searching method, e.g., Newton's method or binary search. The non-trivial part is to solve out the unknown templates $\{T_t\}_{t=1}^{N_t}$.

$\forall l \in \{1, 2, \ldots, L\}$, maximizing B over T_t is equivalent to

$$\max_{\{T_t\}_{t=1}^{N_t}} \hat{B},$$

where,

$$\begin{aligned}
\hat{B} &= \sum_{i,s,r,l} \hat{P}^{(i)}_{(t,s,r,l)} \left[\lambda_t \cdot S_{t,s,r,l}(Y_i^{s,r,l}) + \frac{n_{s,r,l}}{2} \cdot (S_{t,s,r,l}(Y_i^{s,r,l}))^2 \right] \\
&= \sum_{i,s,r,l} \hat{P}^{(i)}_{(t,s,r,l)} \left[\lambda_t \cdot \frac{(Y_i^{s,r,l})^\top M_{s,r,l} T_t}{\sqrt{T_t^\top M_{s,r,l}^\top M_{s,r,l} T_t}} + \frac{n_{s,r,l}}{2} \cdot \left(\frac{(Y_i^{s,r,l})^\top M_{s,r,l} T_t}{\sqrt{T_t^\top M_{s,r,l}^\top M_{s,r,l} T_t}} \right)^2 \right]
\end{aligned}$$

At this moment, its becomes clear that the matrix decomposition in Section 4.2.2 does not work for the maximization problem above. That is due to the non-linear denominators $\sqrt{T_t^\top M_{s,r,l}^\top M_{s,r,l} T_t}$, resulted from the projection of T_t. As we mentioned earlier, the gradient ascent method is taken to attack this problem instead. First take the derivative of $\hat{B}$ w.r.t T_t,

$$\frac{\partial \hat{B}}{\partial T_t} = \sum_{i,s,r,l} \left(\lambda_t + n_{s,r,l} \cdot S_{t,s,r,l}(Y_i^{s,r,l}) \right) \cdot \frac{\partial S_{t,s,r,l}(Y_i^{s,r,l})}{\partial T_t},$$

where

$$\begin{aligned}
\frac{\partial S_{t,s,r,l}(Y_i^{s,r,l})}{\partial T_t} &= \frac{M_{s,r,l}^\top Y_i^{s,r,l} \sqrt{T_t^\top M_{s,r,l}^\top M_{s,r,l} T_t} - (Y_i^{s,r,l})^\top M_{s,r,l} T_t \frac{M_{s,r,l}^\top M_{s,r,l} T_t}{\sqrt{T_t^\top M_{s,r,l}^\top M_{s,r,l} T_t}}}{T_t^\top M_{s,r,l}^\top M_{s,r,l} T_t} \\
&= \frac{M_{s,r,l}^\top Y_i^{s,r,l}}{\sqrt{T_t^\top M_{s,r,l}^\top M_{s,r,l} T_t}} - \frac{(Y_i^{s,r,l})^\top M_{s,r,l} T_t}{\sqrt{T_t^\top M_{s,r,l}^\top M_{s,r,l} T_t}} \cdot \frac{M_{s,r,l}^\top M_{s,r,l} T_t}{T_t^\top M_{s,r,l}^\top M_{s,r,l} T_t} \\
&= \frac{M_{s,r,l}^\top}{\sqrt{T_t^\top M_{s,r,l}^\top M_{s,r,l} T_t}} \cdot Y_i^{s,r,l} - \frac{M_{s,r,l}^\top M_{s,r,l} T_t}{T_t^\top M_{s,r,l}^\top M_{s,r,l} T_t} \cdot S_{t,s,r,l}(Y_i^{s,r,l}).
\end{aligned}$$

Combining two derivatives above,

$$\frac{\partial \hat{B}}{\partial T_t} = \sum_{i,s,r,l} \frac{M_{s,r,l}^{\top}}{\sqrt{T_t^{\top} M_{s,r,l}^{\top} M_{s,r,l} T_t}} \cdot \hat{P}_{(t,s,r,l)}^{(i)} \left(\lambda_t + n_{s,r,l} \cdot S_{t,s,r,l}(Y_i^{s,r,l})\right) \; Y_i^{s,r,l}$$

$$- \sum_{i,s,r,l} \frac{M_{s,r,l}^{\top} M_{s,r,l} T_t}{T_t^{\top} M_{s,r,l}^{\top} M_{s,r,l} T_t} \cdot \hat{P}_{(t,s,r,l)}^{(i)} \left(\lambda_t + n_{s,r,l} \cdot S_{t,s,r,l}(Y_i^{s,r,l})\right) \; S_{t,s,r,l}(Y_i^{s,r,l}) \qquad (4.27)$$

A note on Model Simplification. The model given by (4.25) accommodates the case of shifting a template by a fraction of a pixel, i.e., $Q_{s,r}$ can includes a fraction, besides an integer. Compared to pure integer shifts, the fraction shifts give the model more flexibility to better accommodate the image data Y. However, on the other hand, this results in a high computation and memory cost in parameter learning, especially in updating N_t templates through gradient ascent given in (4.27). If we only consider integer pixel shifts, the computation and memory cost can be reduced to a great extent, since $M_{s,r,l}$ and $n_{s,r,l}$ will be independent of l. And (4.27) will become

$$\begin{aligned} \frac{\partial \hat{B}}{\partial T_t} &= \sum_{i,s,r} \frac{M_{s,r}^{\tau}}{\sqrt{T_t^{\top} M_{s,r}^{\top} M_{s,r} T_t}} \sum_{l \in Q_{s,r}} \hat{P}_{(t,s,r,l)}^{(i)} \left(\lambda_t + n_{s,r} \cdot S_{t,s,r,l}(Y_i^{s,r,l})\right) \; Y_i^{s,r,l} \\ &- \sum_{i,s,r} \frac{M_{s,r}^{\top} M_{s,r} T_t}{T_t^{\top} M_{s,r}^{\top} M_{s,r} T_t} \sum_{l \in Q_{s,r}} \hat{P}_{(t,s,r,l)}^{(i)} \left(\lambda_t + n_{s,r} \cdot S_{t,s,r,l}(Y_i^{s,r,l})\right) \; S_{t,s,r,l}(Y_i^{s,r,l}) \end{aligned}$$

4.3 Experiments on Learning Facial Part Templates and Applications on Ethnicity Classification

4.3.1 Facial Part Templates

We implemented 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 15 facial landmarks manually labeled in advance. Figure 4.1 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. Two sets of experiments will be described in this section. The first set sought to learn templates from training image patches that belong to the same facial part. We used left eye image patches as the training data for a sequence of experiments (Experiment 1 through Experiment 4). We will later show how the learned template changed with respect to different models. The second set of experiments sought to learn templates from training image patches that belong to different facial parts. The purpose of the second set of experiments was to check whether different types of facial templates will pop up automatically and be associated with the right weights. Here we used a mixture of mouth image patches and nose image patches as the training data, and presented the implementation result in Experiment 5.

1. Experiment for the model with mixtures only over multiple templates, (4.14).

The training data was composed of 499 left eye image patches, each with size 12×19. Starting from random initialization, the EM algorithm learned 16 templates each with size 12×19. By random initialization, we mean each pixel of the initial template

Figure 4.1: 12 face images from Feret Face database, each with 17 landmarks labeled manually.

is i.i.d. random noise. In the parameter initialization for the EM, all the λs were set to 3, and all the ϵs were set to be the same (i.e. each initial template was given the same initial weight). Figure 4.2 shows 70 training left eye images. Figure 4.3 shows the evolution of the first 8 templates as the EM algorithm ran. The first row shows the 8 random initial templates, and the $(i+1)^{th}$ row shows the updated 8 templates after i runs of the EM algorithm. As shown in Figure 4.3, the EM algorithm converged quickly. Figure 4.4 shows the evolution of the other 8 templates from random noise as the EM algorithm ran.

2. Experiments for the model with mixtures only over multiple templates and spatial shifts.

The training dataset was composed of 499 left eye images where the left eye was not necessarily centered in the middle, each with size 15×23. The EM algorithm learned 16 templates each with size 12×19 from random initialization. λs and ϵs were initialized in the same way as in the previous experiment. Figure 4.5 shows 70 training image patches and the 16 learned templates.

To demonstrate that the model with mixtures over multiple templates and spatial shifts yields a better result than the basic model with mixtures only over multiple

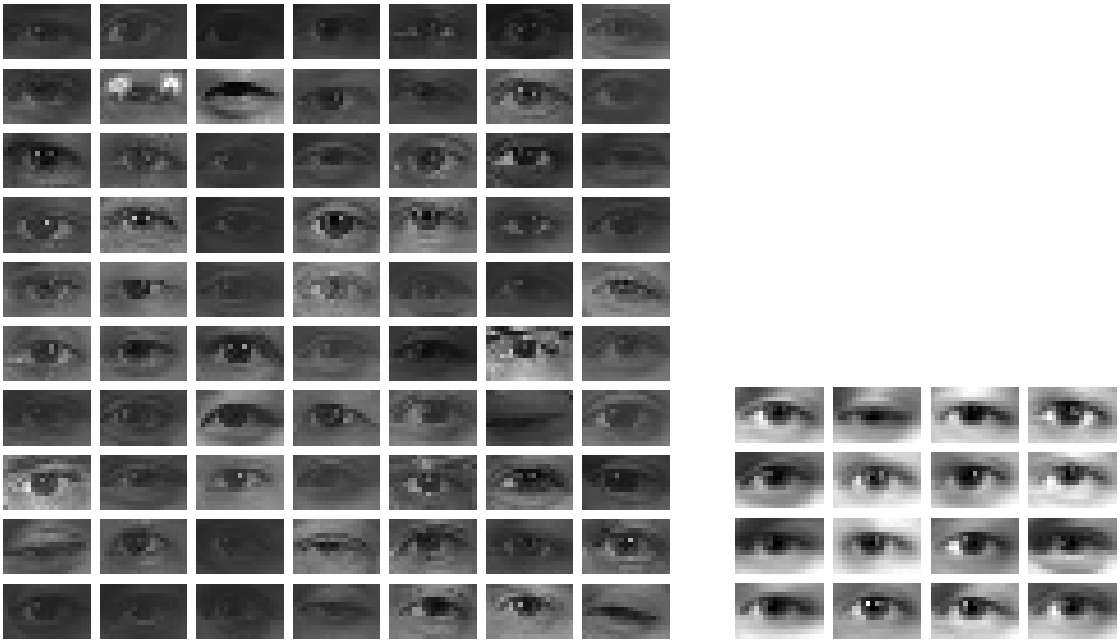

Figure 4.2: The left panel shows 70 training left eye images. The right panel shows the 16 learned templates.

templates (4.14), we did another experiment under the basic model, to learn 16 templates each with the same size, 12×19. To cut down the training image patches to size 12×19 for the basic model, we trimmed each of the original training images (15×23) by a few pixels on the margin. Based on the trimmed training images, the EM algorithm learned 16 templates each with size 12×19 under the basic model (4.14). Figure 4.6 compares the two sets of 16 learned templates side by side. It is obvious that the model with mixtures over both multiple templates and spatial shifts yielded sharper templates with greater divergence, than the basic model with mixtures only over multiple templates (4.14).

3. Experiment for the model with mixtures over multiple templates, spatial shifts and scales.

The dataset was composed of 499 left eye images, each with a random size ranging

Figure 4.3: The evolution of the first 8 templates as the EM algorithm ran.

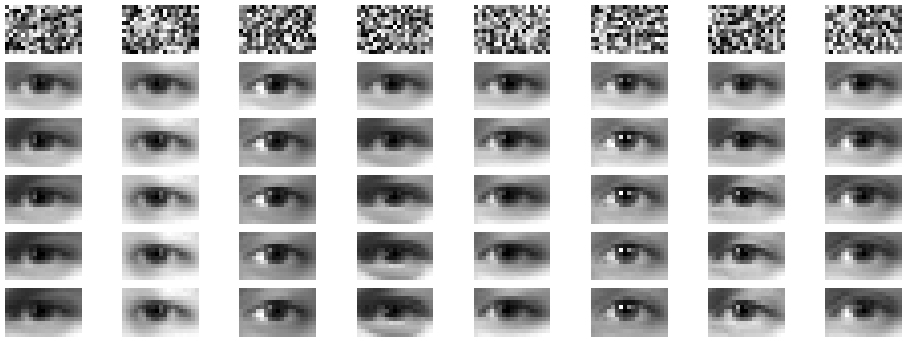

Figure 4.4: The evolution of the last 8 templates as the EM algorithm ran.

from 12×18 to 18×27. The eye was not necessarily located in the center of the training image patch. 16 templates each with size 12×19 were learned after 8 runs of the EM algorithm. Only two discrete scales (0.9 and 1.1) were considered for each template, hence, $N_s = 2, N_r = 1$ in the model (4.25). Due to the fact that two discrete scales are not sufficient to cover a continuous size range of training images, the learned templates were not as sharp as in previous experiments. The quality of learned templates can be improved simply by adding more discrete scales for each template, i.e., by increasing N_s. In Figure 4.8 , we show how the 16 templates were updated from random noise after each run of the EM algorithm. The first row is the initialization, and the $(i+1)^{th}$ row shows the updated templates after i runs of the

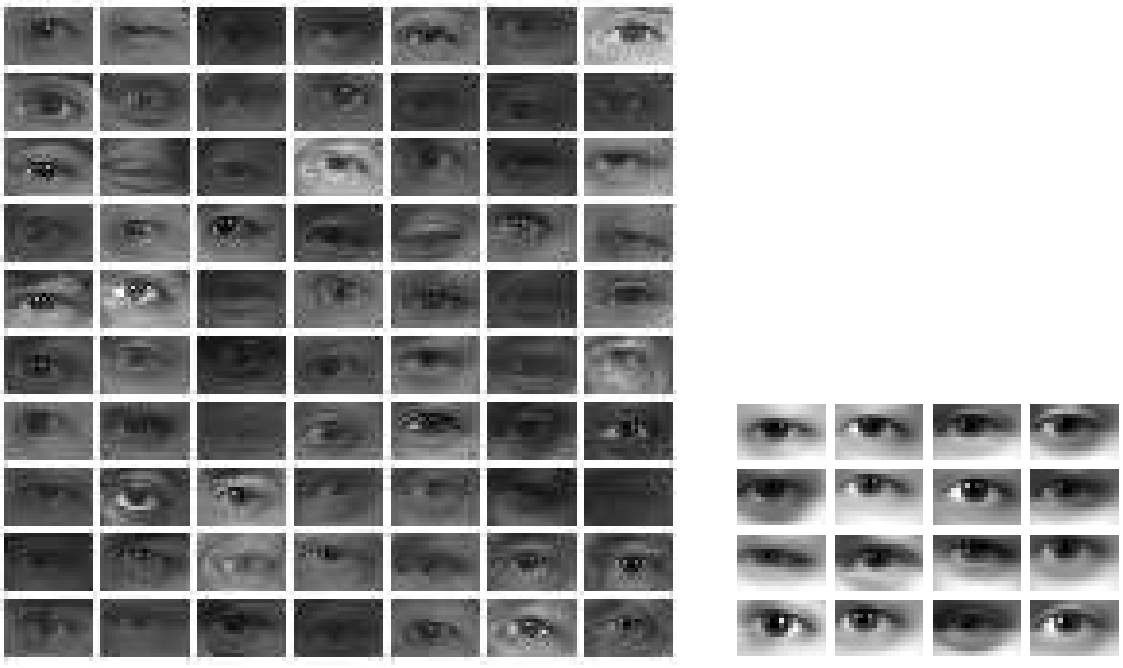

Figure 4.5: The left panel shows 70 training left eye images, each with size 15×23). The right panel shows the 16 learned templates, each with size 12×19, under the model with mixtures over multiple templates and spatial shifts.

EM algorithm. As it shows in the figures, the EM algorithm converged quickly for our case.

4. Experiment for the full model with mixtures over multiple templates, spatial shifts, scales and rotations (4.25).

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 not always in the center; the size of each image patch ranged from 16×18 to 30×33. We implemented the fully generalized model, given in (4.25), on this training dataset. 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 $\{-6.7^o,\ 0^o,\ 6.7^o\}$,

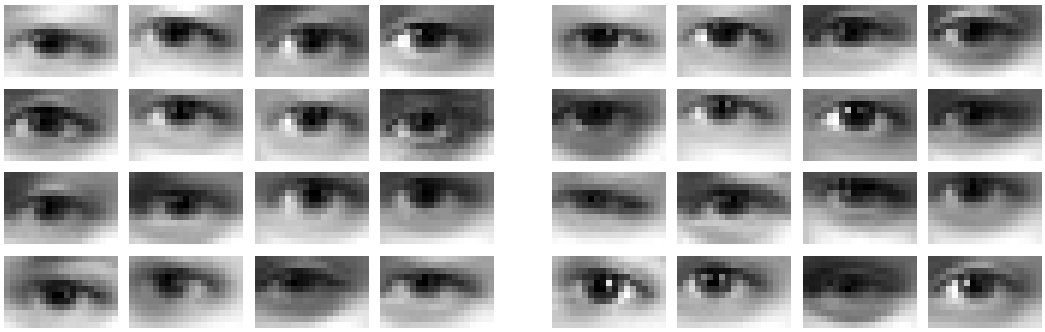

Figure 4.6: The left panel shows the 16 learned templates from the model not considering spatial shifts. The right panel shows the 16 learned templates from the model considering spatial shifts templates.

hence $N_t = 16$, $N_s = N_r = 3$ in (4.25). Figure 4.9 shows 120 training data and the 16 learned templates through the EM algorithm. Figure 4.10 shows the evolution of the 16 templates as the EM algorithm ran.

5. Experiment on a training dataset composed of two types of facial parts, under the model with mixtures over multiple templates and spatial shifts.

The training dataset 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 4.11 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 moustache, smiling or not smiling. Hence, as expected, the learned templates from the EM algorithm revealed this variation of facial features and expressions, besides distinguishing noses from mouths. In addition, both the summation of the weights associated with nose templates and the summation of the weights associated with mouth templates were very close to 0.5, which indicated that our model was properly weighted. Figure 4.12 and 4.13 shows the evolution of the first 16 templates and the last 16 templates as the EM algorithm ran. Again,

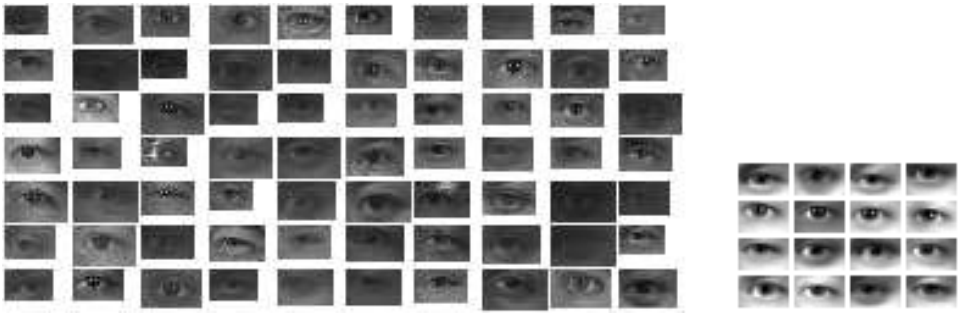

Figure 4.7: The left panel shows 70 training left eye images, with different sizes ranging from 12by18 to 18by27; eyes are not in the center. The right panel shows the 16 learned templates, each with size 12 by 19, under the model considering spatial shifts and mixtures over two scales 0.9 and 1.1.

the first row shows the 16 random initial templates, and the $(i+1)^{th}$ row shows the updated templates after i runs of the EM algorithm. The EM algorithm converged quickly.

4.3.2 Ethnicity Classification of Face Images

The learned templates of facial parts in the previous section can be useful building blocks for a complete face model. For example, one way to achieve this complete model would be to combine these learned templates with information on the relative positions of facial parts. However, these templates are also directly useful by themselves in fulfilling certain computer vision tasks. In this section, we will take the example of ethnicity classification of face images to show the direct usefulness of these learned templates of facial parts. We had a dataset composed of 352 East Asian male faces and 272 South Asian male faces. 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 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 an South Asian

Figure 4.8: The evolution of the 16 templates during 8 runs of the EM algorithm, starting from a random initialization.

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 an decision about the original face images. The classification of eye image patches was done 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 height, 10 pixels. Now we had two sets of image patches, 352 East Asian eyes (call it set A_e) and 272 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 half (selected randomly) of the image patches from A_i as training data for the South Asian group. The other half from A_e and the other half from A_i were merged together, and played the role of testing data. We implemented the model with mixtures over 8 templates (each with size 8×20), spatial shifts, and 4 discrete scales (1.2, 1.1, 1, and 0.9). The EM algorithm was performed on both the East Asian training data and South Asian training data, and two models were learned: $P(Y|\text{East Asian eyes})$ and $P(Y|\text{South Asian eyes})$, where Y represented an image patch. Figure 4.14 shows 70 East Asian eyes from set A_e and the corresponding 8 learned templates. As for testing the performance of

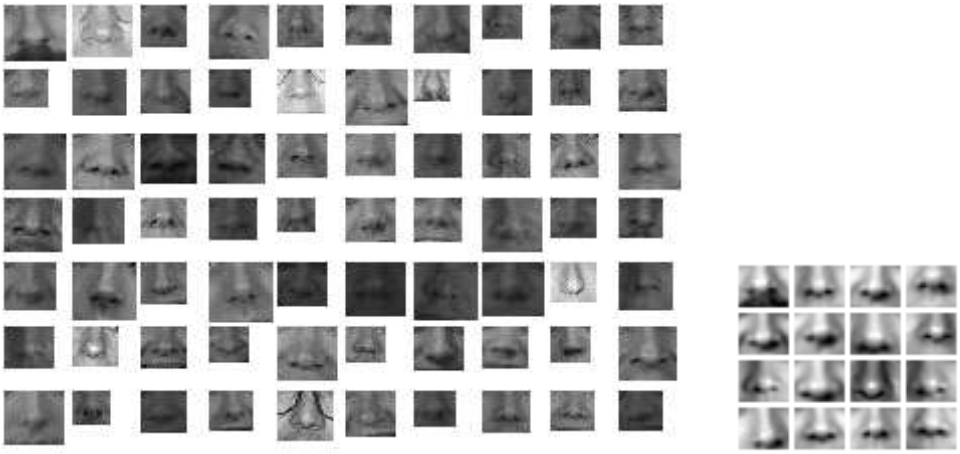

Figure 4.9: The left panel shows 120 training image patches. The right panel shows the 16 learned templates, each with size 15×18, from the fully generalized model, with mixtures over multiple templates, spatial shifts, scales, and rotations.

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 $P(Y|\text{East Asian eyes}) < P(Y|\text{South Asian eyes})$. Figure 4.15 shows 70 South Asian eyes from set A_i and the corresponding 8 learned templates. As shown in the figures, East Asian eye templates and South Asian eye templates looked very different from each other, and captured well the facial features associated with the corresponding ethnic group.

Since we only had a few hundred face images, in order to achieve a less biased result, we performed 50 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, $i \in \{1, \ldots, 50\}$. Finally, after 50 cross-validations were finished, we averaged all the Rs (R_1 through R_{50}), giving us a classification rate of 97 percent.

Figure 4.10: The evolution of the 16 templates during 8 runs of the EM algorithm, starting from a random initialization.

4.4 Background Template Learning

The need for prior models of natural image structures occurs in many machine vision problems including denoising, optical flow, super-resolution, stereo etc., even in object detection. In their attempts at object detection, many researchers have considered only object-class models for object detection while neglecting background models, but mathematically this is essentially equivalent to assuming that all images under the background model have the same probability, i.e. are made of i.i.d. uniform random noise (white noise). It is obvious that the probability of generating anything resembling a natural scene from images with random pixel intensities is extremely low. This suggests that in this state-space of all possible scenes, the region of the space occupied by natural scenes is also extremely low, hence the white noise background model is improper.

Far from white noise, the world is delicately structured. When we slide a small window across a natural image, most of what we see through the window is edges and uniform colors. These structured patches are the bricks building up the world.

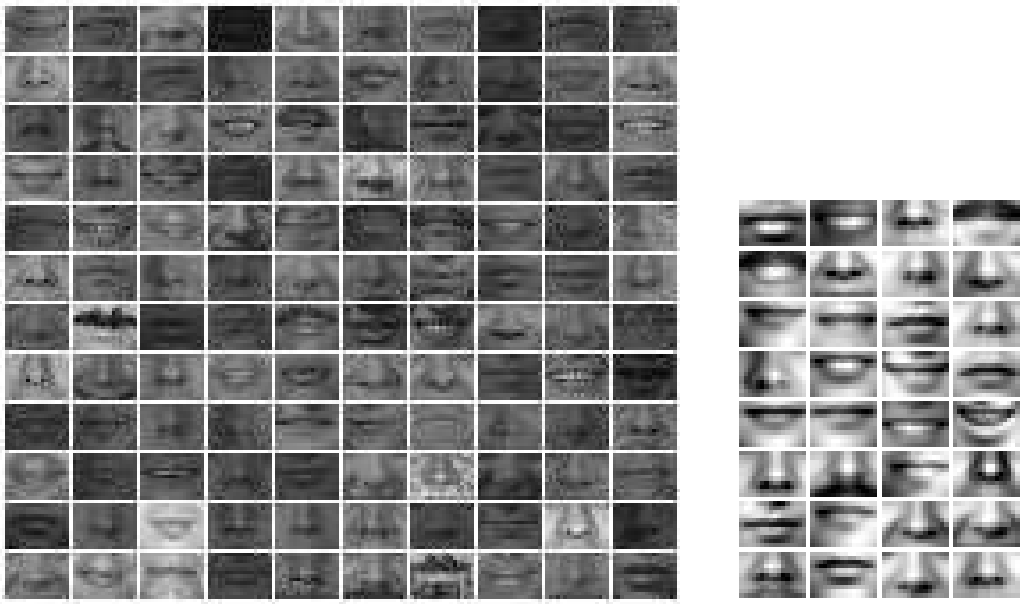

Figure 4.11: The left panel shows 70 training 13×18 nose or mouth image patches. The right panel shows the 16 learned templates each with size 11×16.

Hence, the statistics of small image patches have received extensive attention in the literature. In particular, sparse coding approaches attempt to model structural properties of images in terms of a set of linear filter responses. Olshausen and Field [44] have represented an image patch in terms of a liner combination of learned filters. Welling et al. [46] built a probabilistic model for natural image patches based on the Product-of-Experts (POE) framework [45], taking the product of student-t distributions on linear filter responses, and learned all the unknown parameters including the filter themselves by the principle of maximum likelihood.

One of the influential statistical models for natural images is the Markov Random Field (MRF). MRFs define a distribution over images that is based on simple and local interactions between pixels. Roth and Black [47, 48] introduced the Fields of Experts (FOE) model under the MRF framework, taking as the potentials the POE model from [46]. They used the principle of maximum likelihood, as well, to find the optimum filters. However, as in previous models of images based on filter outputs

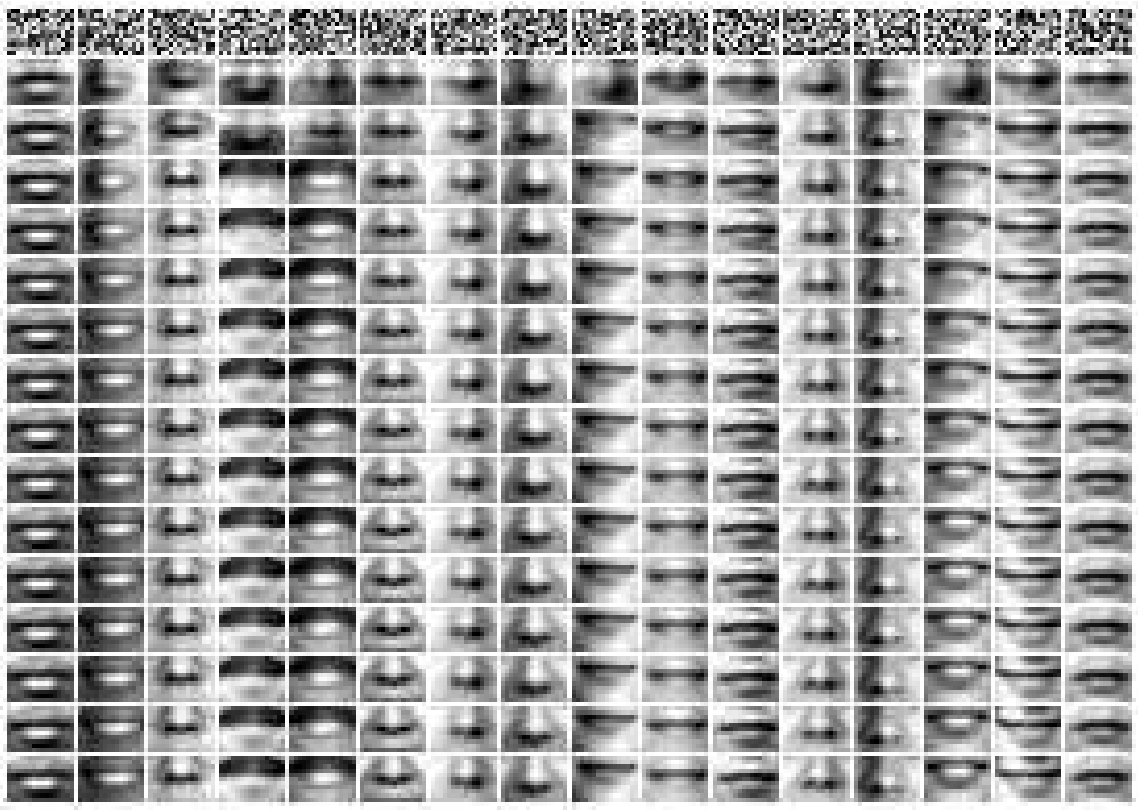

Figure 4.12: The evolution of the first 16 templates as the EM algorithm ran, starting from a random initialization.

[49], the probability of an image given the model involves an intractable partition function. This makes learning extremely slow, since MCMC sampling has to be performed before every step of gradient ascent of the log likelihood.

(**Note:** From this point until the end of this chapter, we will use "background" and "natural" interchangeably. By "background" image patches, we mean image patches cropped randomly from random natural images.)

As in our modeling of the image patches of object parts, we model background image patches based on templates. Our world is structured, hence we expect the background templates to be structured with similar patterns. We will again only consider gray-scale background images. And we will follow the framework of learning

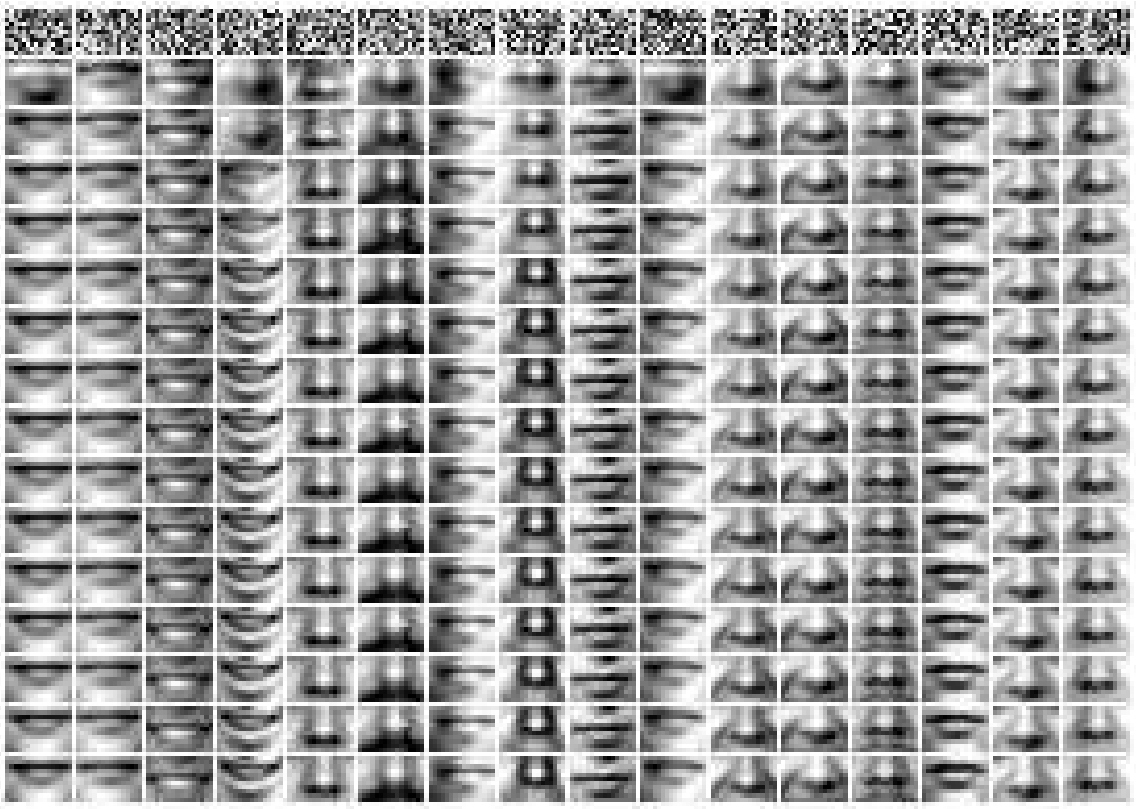

Figure 4.13: The evolution of the last 16 templates as the EM algorithm ran, starting from a random initialization.

object templates to learn background templates. However, the correlation between a uniform-color template and an image patch is not defined due to the fact that the variance of a uniform-color template is 0. Hence, certain modification needs to be made in order to accommodate the characteristics of the background image patches.

4.4.1 A Basic Probabilistic Model of Background Image Patches

Let $Y = (y_1, \ldots, y_n)$ be a background image patch with n pixels. Let $P(Y|\text{bg})$ represent the distribution of Y conditioned on it is a background image patch. We start by giving $P(Y|\text{bg})$ the same expression as in 4.14, which is a mixture over

Figure 4.14: The left panel shows 70 East Asian eyes, each with height of 10 pixels. The right panel shows the 8 templates, each with size 8×20, learned from the model with mixtures over 4 scales: 1.2, 1.1, 1, and 0.9.

normalized (mean 0, variance 1) N_t templates each with the same size as Y:

$$\begin{aligned} P(Y|\text{bg}) &= \sum_{t=1}^{N_t} \epsilon_t \cdot P(Y|T_t) \\ &= \sum_{t=1}^{N_t} \epsilon_t \cdot \frac{\frac{1}{1-e^{-2\lambda_t}} \lambda_t \, e^{-\lambda_t \, (1-S_t(Y))}}{Q_t \cdot (256)^n \frac{\sqrt{n}}{\sqrt{2\pi}} e^{-\frac{nS_t(Y)^2}{2}}}, \end{aligned} \tag{4.28}$$

where S_t is the normalized correlation between Y and T_t, $\forall t \in \{1, \ldots, N_t\}$. To capture the big portion of uniform-color background image patches mentioned earlier, we add one more mixture into (4.28), and this new mixture involves a new template, T_0, that is uniformly colored. We will define this $(N_t + 1)^{th}$ mixture in the following paragraph and afterwards we will see that T_0 is actually an "abstract" template.

The $(N_t + 1)^{th}$ mixture is also based on a sufficient statistics S_0, a function of Y. Different from the definition of $\{S_t\}_{t=1}^{N_t}$, S_0 is defined to be uniquely determined by the variance of Y:

$$S_0(Y) = \sum_k (y_k - \overline{y})^2 / (255^2 \cdot n),$$

where $\overline{y} = (1/n) \cdot \sum_k y_k$. From the definition, S_0 does not involve T_0 explicitly. But it is connected to T_0 in the following sense: The smaller S_0 is, the more Y looks like

Figure 4.15: The left panel shows 70 Indian eyes, each with height of 10 pixels. The right panel shows the 8 templates, each with size 8×20, learned from the model with mixtures over 4 scales: 1.2, 1.1, 1, and 0.9.

a uniform-color image patch generated from T_0, due to the fact that the variance of T_0 is equal to zero. This special template T_0 is our trick to handle the specialty of background image patches. After adding the $(N_t+1)^{th}$ mixture into (4.28), we have

$$\begin{aligned} P(Y|\text{bg}) &= \sum_{t=0}^{N_t} \epsilon_t \ \cdot \ P(Y|T_t) \\ &= \sum_{t=1}^{N_t} \epsilon_t \ \cdot \ \frac{\frac{1}{1-e^{-2\lambda_t}} \ \lambda_t \ e^{-\lambda_t \ (1-S_t(Y))}}{Q_t \cdot (256)^n \frac{\sqrt{n}}{\sqrt{2\pi}} e^{-\frac{nS_t(Y)^2}{2}}} \ + \ P(Y|T_0) \\ &= \sum_{t=1}^{N_t} \epsilon_t \ \cdot \ \frac{\frac{1}{1-e^{-2\lambda_t}} \ \lambda_t \ e^{-\lambda_t \ (1-S_t(Y))}}{Q_t \cdot (256)^n \frac{\sqrt{n}}{\sqrt{2\pi}} e^{-\frac{nS_t(Y)^2}{2}}} \ + \ \frac{P_0(S_0(Y))}{\#\{\hat{Y} : S_0(\hat{Y}) = S_0(Y)\}}, \end{aligned} \tag{4.29}$$

where $\#\{\hat{Y} : S_0(\hat{Y}) = S_0(Y)\}$ is the counting measure of the set $\{\hat{Y} : S_0(\hat{Y}) = S_0(Y)\}$. (4.29) also indicates the sampling procedure of a background image patch Y: we first randomly select t^* from the set $\{0, 1, \ldots, N_t\}$ according to their weight $\{\epsilon_t\}_{t=0}^{N_t}$, followed by sampling the corresponding sufficient statistics $S_{t^*} = s$ from its marginal distribution, and finally uniformly sample Y from the set $\{Y \in \{0, 1, \ldots, 255\}^n : S_{t^*}(Y) = s\}$. If Y is generated from the uniform-color template T_0, $S_0(Y)$ will be more likely to be close to zero. Hence it is reasonable to model $P_0(S_0(Y) = s)$ as a

decreasing function of s. Here we model it as a discretized exponential distribution with parameter λ_0. As for $\#\{\hat{Y} : S_0(\hat{Y}) = S_0(Y)\}$, we again approximate it through CLT similarly as in Section 4.2.1. Hence,

$$P(Y|T_0) = \frac{P_0(S_0(Y))}{\#\{\hat{Y} : S_0(\hat{Y}) = S_0(Y)\}} = \frac{\lambda_0 e^{-\lambda_0 S_0(Y)}}{Q_0 \cdot \frac{\sqrt{n}}{\sqrt{2\pi}\sigma_0} e^{-\frac{n}{2\sigma_c^2}(S_0-\mu_0)^2}}, \tag{4.30}$$

where the numerator is from the exponential marginal distribution of $S_0(Y)$, and the denominator is due to the fact that if $\hat{Y}$ is a random vector with length n and each of it element is a i.i.d. random sample from a uniform distribution on $\{0, 1, \ldots, 255\}$, then $S_0(\hat{Y})$ converges in distribution to a normal distribution $N(\mu_0, \sigma_0^2/n)$ as n goes to infinity, where $\mu_0 = 5461.25/(255^2)$ and $\sigma_0 = 4884.6/(255^2)$. (See the Appendix for the detailed derivation of the asymptotic distribution of $S_0(\hat{Y})$ through CLT.) After plugging (4.30) into (4.29), we get

$$P(Y|\text{bg}) = \sum_{t=1}^{N_t} \epsilon_t \cdot \frac{\frac{1}{1-e^{-2\lambda_t}}\ \lambda_t\ e^{-\lambda_t\ (1-S_t(Y))}}{Q_t \cdot (256)^n \frac{\sqrt{n}}{\sqrt{2\pi}} e^{-\frac{nS_t(Y)^2}{2}}} + \epsilon_0 \cdot \frac{\lambda_0 e^{-\lambda_0 S_0(Y)}}{Q_0 \cdot \frac{\sqrt{n}}{\sqrt{2\pi}\sigma_0} e^{-\frac{n}{2\sigma_c^2}(S_0(Y)-\mu_0)^2}}, \tag{4.31}$$

where Q_t is close to 1 when n is large, $\forall t \in \{0, 1, \ldots, N_t\}$.

It is straightforward to generalize the model given in (4.31) one step further, by adding in one layer of mixtures over spatial shifts. We only consider here spatial shifts in the unit of integer pixels. Since scales and rotations are not considered for this moment, $\{T_t\}_{t=0}^{N_t}$ have to be no bigger than Y in order to shift around within the image patch Y. Let n_T be the pixel number of each template T_t, and n be the number of pixels in Y. $n_T \leq n$. Let Q be the set of all the discretized spatial shifts. Note here Q only contains integers. We assume each spatial shift is equally likely, i.e. $P(l) = \frac{1}{\|Q\|}, \forall l \in Q$, where $\|Q\|$ is the counting measure of set Q. Let Y^l represent the sub-region of Y covered by a template, and $(Y^l)^c$ represent the rest area of Y that is not covered by a template. Each pixel of $(Y^l)^c$ is modeled as i.i.d. random noise from $\{0, 1, \ldots, 255\}$, hence $P((Y^l)^c) = (\frac{1}{256})^{n-n_T}$. The generalized background

model becomes:

$$
\begin{aligned}
& P(Y|\text{bg}) \\
= & \sum_{l\in Q} \frac{1}{\|Q\|} \sum_{t=0}^{N_t} \epsilon_t \cdot P((Y^l)^c) \cdot P(Y^l|T_t) \\
= & \sum_{l\in Q} \frac{1}{\|Q\|} \cdot (\frac{1}{256})^{n-n_T} \cdot \sum_{t=1}^{N_t} \epsilon_t \cdot \frac{\frac{1}{1-e^{-2\lambda_t}}\ \lambda_t\ e^{-\lambda_t\ (1-S_t(Y^l))}}{Q_t \cdot (256)^{n_T} \cdot \frac{\sqrt{n_T}}{\sqrt{2\pi}} e^{-\frac{n_T S_t(Y^l)^2}{2}}} \\
+ & \sum_{l\in Q} \frac{1}{\|Q\|} \cdot (\frac{1}{256})^{n-n_T} \cdot \epsilon_0 \cdot \frac{\lambda_0 e^{-\lambda_0 S_0(Y^l)}}{Q_0 \cdot (256)^{n_T} \cdot \frac{\sqrt{n_T}}{\sqrt{2\pi\sigma_0}} e^{-\frac{n_T}{2\sigma_c^2}(S_0(Y^l)-\mu_0)^2}} \\
= & \sum_{l\in Q} \frac{1}{\|Q\|} \cdot (\frac{1}{256})^{n} \cdot \\
& \left(\sum_{t=1}^{N_t} \epsilon_t \ \cdot\ \frac{\frac{1}{1-e^{-2\lambda_t}}\ \lambda_t\ e^{-\lambda_t\ (1-S_t(Y^l))}}{Q_t \cdot \frac{\sqrt{n_T}}{\sqrt{2\pi}} e^{-\frac{n_T S_t(Y^l)^2}{2}}}\ +\ \epsilon_0\ \cdot\ \frac{\lambda_0 e^{-\lambda_0 S_0(Y^l)}}{Q_0 \cdot \frac{\sqrt{n_T}}{\sqrt{2\pi\sigma_0}} e^{-\frac{n_T}{2\sigma_c^2}(S_0(Y^l)-\mu_0)^2}} \right), \qquad (4.32)
\end{aligned}
$$

where Q_t is close to 1 when n_T is large, $\forall t \in \{0, 1, \ldots, N_t\}$. Let $Y^l = (y_1^l, \ldots, y_{n_T}^l)$, $S_0(Y^l)$ is defined as

$$S_0(Y^l) = \sum_{k=1}^{n_T} \frac{\left(y_k^l - \overline{y^l}\right)^2}{255^2 \cdot n_T},$$

where $\overline{y^l} = (1/n_T) \cdot \sum_{k=1}^{n_T} y_k^l$. And $S_t(Y^l)$ is defined to be the normalized correlation between Y^l and T_t, for all $t \in \{1, \ldots, N_t\}$.

Parameter Learning. We again apply the EM algorithm to learn all the unknown parameters. As we indicated earlier, there is no T_0, just a parameter λ_0 governing the distribution on the variance of Y^l. Hence only N_t templates $\{T_t = (\tau_1, \ldots, \tau_{n_T})\}_{t=1}^{N_t}$ need to be learned, besides the other parameters $\{\lambda_t\}_{t=0}^{N_t}$ and $\{\epsilon_t\}_{t=1}^{N_t}$, with $\sum_{t=0}^{N_t} \epsilon = 1$ and $\epsilon_t, \lambda_t \geq 0$. We also require that $\{T_t\}_{t=1}^{N_t}$ are normalized, with mean 0 and variance 1. Suppose that we have N background image patches $\{Y_i = (y_1^{(i)}, y_2^{(i)}, \ldots, y_n^{(i)})'\}_{i=1}^{N}$ for training. Assume that they are i.i.d. samples from the model $P(Y|\text{bg})$ in (4.32),

then the likelihood function is:

$$
\begin{aligned}
& P(\vec{Y}|\text{bg}) = \prod_i P(Y_i|\text{bg}) \\
= & \prod_i \sum_{l\in Q} \frac{1}{\|Q\|} \cdot \left(\frac{1}{256}\right)^n \cdot \\
& \left(\sum_{t=1}^{N_t} \epsilon_t \cdot \frac{\frac{1}{1-e^{-2\lambda_t}} \lambda_t \, e^{-\lambda_t \, (1-S_t(Y_i^l))}}{Q_t \cdot \frac{\sqrt{n_T}}{\sqrt{2\pi}} e^{-\frac{n_T S_t(Y_i^l)^2}{2}}} + \epsilon_0 \cdot \frac{\lambda_0 e^{-\lambda_0 S_0(Y_i^l)}}{\tilde{Q}_t \cdot \frac{\sqrt{n_T}}{\sqrt{2\pi\sigma_0}} e^{-\frac{n_T}{2\sigma_c^2}(S_0(Y_i^l)-\mu_0)^2}} \right).
\end{aligned}
$$

Since Q_t is close to 1 $\forall t \in \{0, \ldots, N_t\}$ when n_T is large, we replace it with 1 in the likelihood above to simplify our computation. Let $\vec{\theta}$ represent the vector of all unknown parameters.

Expectation Step. $\forall i$, $\forall t \in \{0, \ldots, N_t\}$, we define

$$
\begin{aligned}
\hat{P}_{(t,l)}^{(i)} &= P(X_i = (t,l)|Y_i, \vec{\theta}^{(c)}) \\
&= \frac{\epsilon_t^{(c)} \cdot \frac{1}{\|Q\|} \cdot P_{(t,l)}^{(c)}(Y_i)}{\sum_{t=1}^{N_t} \sum_{l\in Q} \epsilon_t^{(c)} \cdot \frac{1}{\|Q\|} \cdot P_{(t,l)}^{(c)}(Y_i)},
\end{aligned}
$$

where $\theta^{(c)}$ stands for the "current" guess of θ and $P_{(t,l)}^{(c)}(Y_i)$ is defined as

$$
\begin{aligned}
P_{(t,l)}^{(c)}(Y_i) &= \left(\frac{1}{256}\right)^n \cdot \frac{\frac{1}{1-e^{-2\lambda_t^{(c)}}} \lambda_t^{(c)} \, e^{-\lambda_t^{(c)} \, (1-S_t(Y_i^l))}}{\frac{\sqrt{n_T}}{\sqrt{2\pi}} e^{-\frac{n_T S_t(Y_i^l)^2}{2}}}, \qquad \forall t \in \{1, \ldots, N_t\}; \\
P_{(0,l)}^{(c)}(Y_i) &= \left(\frac{1}{256}\right)^n \cdot \frac{\lambda_0^{(c)} e^{-\lambda_0^{(c)} S_0(Y_i^l)}}{\frac{\sqrt{n_T}}{\sqrt{2\pi\sigma_0}} e^{-\frac{n_T}{2\sigma_c^2}(S_0(Y_i^l)-\mu_0)^2}}
\end{aligned}
$$

Maximization Step. Define

$$
\begin{aligned}
B &= \sum_{t=0}^{N_t}\sum_{l\in Q}\sum_{i}\hat{P}^{(i)}_{(t,l)}\cdot\log\left[\epsilon_t\cdot\frac{1}{\|Q\|}\cdot P(Y_i|X_i=(t,l),\vec{\theta})\right] \\
&= \sum_{t=0}^{N_t}\sum_{l,i}\hat{P}^{(i)}_{(t,l)}\cdot\log(\epsilon_t)+\sum_{t=1}^{N_t}\sum_{l,i}\hat{P}^{(i)}_{(t,l)}\cdot\log\left((\frac{1}{256})^n\cdot\frac{\frac{1}{1-e^{-2\lambda_t}}\ \lambda_t\ e^{-\lambda_t\ (1-S_t(Y_i^l))}}{\frac{\sqrt{n_T}}{\sqrt{2\pi}}e^{-\frac{n_T S_t(Y_i^l)^2}{2}}}\right) \\
&\quad +\sum_{l,i}\hat{P}^{(i)}_{(0,l)}\cdot\log\left((\frac{1}{256})^n\cdot\frac{\lambda_0 e^{-\lambda_0 S_0(Y_i^l)}}{\frac{\sqrt{n_0}}{\sqrt{2\pi\sigma_0}}e^{-\frac{n_0}{2\sigma_c^2}(S_0(Y_i^l)-\mu_0)^2}}\right).
\end{aligned}
$$

We want to maximize B w.r.t. $\{\epsilon_t\}_t, \{\lambda_t\}_t, \{T_t\}_t$ subject to:

$$
\lambda_t\geq 0;\quad \sum_{t=0}^{N_t}\epsilon_t=1;\quad \sum_{k=1}^{n}\tau_k^{(t)}=0;\quad \sum_{k=1}^{n}(\tau_k^{(t)})^2=1.
$$

It is straightforward to solve out $\{\lambda_t\}$ and $\{\epsilon_t\}$ as follows

$$
\begin{aligned}
\epsilon_t &= \frac{1}{N}\sum_{i,l}\hat{P}^{(i)}_{t,l},\quad \forall t\in\{0,1,\ldots,N_t\}, \\
\lambda_0 &= \frac{\sum_{i,l}\hat{P}^{(i)}_{t}}{\sum_{i,l}\hat{P}^{(i)}_{t}\cdot S_0(Y_i^l)}, \\
\lambda_t &= \frac{2\lambda_t e^{-2\lambda_t}+e^{-2\lambda_t}-1}{e^{-2\lambda_t}-1}\cdot\frac{\sum_{i,l}\hat{P}^{(i)}_{(t,l)}}{\sum_{i,l}\hat{P}^{(i)}_{(t,l)}\cdot(1-S_t(Y_i^l))},\quad \forall t\in\{1,\ldots,N_t\},
\end{aligned}
$$

where λ_t can be identified by a simple numerical searching method, e.g. Newton's method or Binary search. As for updating the unknown templates $\{T_t=(\tau_1,\ldots,\tau_{n_T})\}_{t=1}^{N_t}$, $\forall t\in\{1,2,\ldots,N_t\}$, maximizing B over T_t is equivalent to

$$
\max_{\{\tau_k^{(t)}\}_{k=1}^{n_T}}\hat{B},\quad \text{s.t. } \sum_{k=1}^{n_T}\tau_k^{(t)}=0,\ \sum_{k=1}^{n_T}(\tau_k^{(t)})^2=1,
$$

where

$$\hat{B} = \sum_{i,l} \hat{P}^{(i)}_{(t,l)} \left[\lambda_t \cdot (Y_i^l)^\top T_t + \frac{n}{2} \cdot ((Y_i^l)^\top T_t)^2 \right].$$

This is a constrained quadratic maximization problem, and can be solved through matrix decomposition and changing variables similarly as in the M-step in Section 4.2.2. Let

$$V = \sum_{i,l} \lambda_t \cdot \hat{P}^{(i)}_{(t,l)} \cdot Y_i^l \qquad A = \sum_{i,l} \frac{n}{2} \cdot \hat{P}^{(i)}_{(t,l)} \cdot Y_i^l \cdot (Y_i^l)^\top.$$

V is a column vector with length n_T, and A is a $n_T \times n_T$ semi-positive definite matrix. The original maximization problem of $\hat{B}$ becomes

$$\max_{\{\tau_k^{(t)}\}_{k=1}^{n_T}} \hat{B}, \quad \text{s.t. } \sum_{k=1}^{n_T} \tau_k^{(t)} = 0, \ \sum_{k=1}^{n_T} (\tau_k^{(t)})^2 = 1,$$

where

$$\hat{B} = V^\top \cdot T_t + T_t^\top \cdot A \cdot T_t.$$

The following steps to solve this maximization problem above are exactly the same as the M-step for learning eye templates in Section 4.2.2, where matrix eigen decomposition was applied.

4.4.2 Experiment for the Basic Background Model

We collected 59 natural images from Internet for training, as shown in Figure 4.16. These images have the same width, 240 pixels, and different heights (ranging from 160 pixels to 360 pixels). We tried to cover a good diversity of background images, from man-made objects to animals to human, from indoor offices to outdoor urban scenes. We cropped out 15×15 non-overlapping image patches from these 59 images, and that yielded 12753 training image patches, i.e., $N = 12753, n = 15 \cdot 15$. Under the model given by (4.32), the EM algorithm learned from this dataset 32 templates

$\{T_t\}_{t=1}^{32}$ each with size 10×10, starting from a random initialization. By the end of the EM algorithm, ϵ_0 was equal to 0.9092, and λ_0 was equal to 174.8316, that means the mean of the exponential distribution of S_0 was equal to 0.0057. This indicates that most of the natural image patches look very like uniform-color patches. Figure 4.17 shows 600 training image patches and the 32 learned templates.

Figure 4.16: 59 natural images collected from Internet.

4.4.3 Improvement on the Background Model

From the learned the background templates in Section 4.4.2, we observe that most of the time, templates came in pairs. For example, both the left-white/right-black

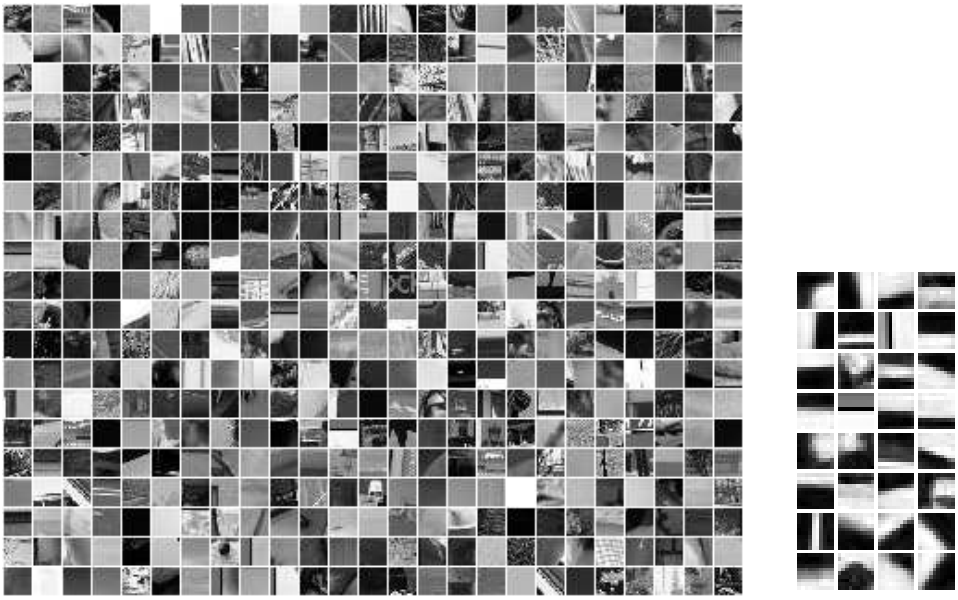

Figure 4.17: The left panel shows 500 randomly selected training image patches (15×15). The right panel shows the 32 learned templates, each with size 10×10, learned from the basic background model.

template and the left-black/right-white template showed up, and either of the two can be seen as a flip of the other. Since all the templates are normalized, flipping a template is equivalent to multiplying the original by '-1'. Another observation is that some templates flipped during the EM iterations, from the 2nd iteration to the 3rd iteration, or from the 3rd iteration to the 4th iteration, as shown in Figure 4.18. This indicates that templates tried to flip themselves back and forth, in order to better fit the training data. Both "Templates come in pairs" and "Templates flipped while the EM algorithm ran" are not surprising, since most of our world is symmetric and well structured. Upon realizing the automatic template flipping, we want to improve the background model by encoding the flipping pattern into the model, to gain better representation of our world. We re-formalize the generating procedure of Y from $\{T_t\}_{t=0}^{N_t}$. To sample a background image patch Y, we first select t^* randomly from the set $\{0, 1, \ldots, N_t\}$ and that determines which template to use.

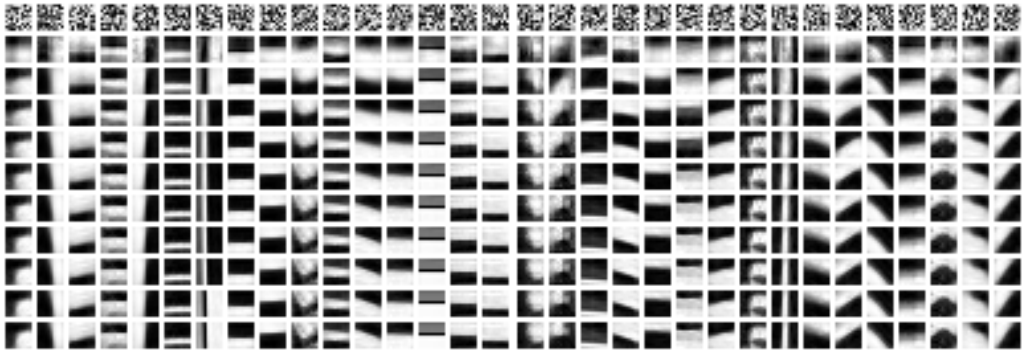

Figure 4.18: The evolution of the 32 templates during 20 runs of the EM algorithm, under the basic background model, where the first row shows the random initialization of the templates.

Independently we also select a random spatial shift l and that determines where to put down this template in Y. Now, if $t^* \neq 0$, we do the following: With probability 0.5 the template T_{t^*} generates Y, and with probability 0.5 the flip of T_{t^*} (i.e. $-T_{t^*}$) generates Y. If $t^* = 0$, then no flipping is involved, and Y is generated from T_0 directly.

Besides this modification involving flipping of templates, we make another improvement by generalizing the distribution of $S_0(Y)$ from an exponential distribution to a more flexible distribution, a gamma distribution with parameter λ_0 and k. Therefore, the new probability density function of a background image patch Y becomes (Note that $S_t(Y, -T_t) = -S_t(Y, T_t), \forall t \in \{1, 2, \ldots, N_t\}$)

$$
\begin{aligned}
& P(Y|\text{bg}) \\
= & \sum_{l \in Q} \frac{1}{\|Q\|} \cdot (\frac{1}{256})^n \cdot \sum_{t=1}^{N_t} \frac{\epsilon_t}{2} \cdot \left[\frac{\frac{1}{1-e^{-2\lambda_t}} \ \lambda_t \ e^{-\lambda_t \ (1-S_t(Y^l))}}{Q_t \cdot \frac{\sqrt{n_T}}{\sqrt{2\pi}} e^{-\frac{n_T S_t(Y^l)^2}{2}}} + \frac{\frac{1}{1-e^{-2\lambda_t}} \ \lambda_t \ e^{-\lambda_t \ (1+S_t(Y^l))}}{\tilde{Q}_t \cdot \frac{\sqrt{n_T}}{\sqrt{2\pi}} e^{-\frac{n_T S_t(Y^l)^2}{2}}} \right] \\
+ & \sum_{l \in Q} \frac{1}{\|Q\|} \cdot (\frac{1}{256})^n \cdot \epsilon_0 \cdot \frac{S_0(Y^l)^{k-1} \cdot \lambda_0^k \cdot e^{-\lambda_0 S_0(Y^l)} \ / \ \Gamma(k)}{Q_0 \cdot \frac{\sqrt{n_T}}{\sqrt{2\pi\sigma_0}} e^{-\frac{n_T}{2\sigma_c^2}(S_0(Y^l)-\mu_0)^2}}.
\end{aligned} \tag{4.33}
$$

The set of unknown parameters is composed of $\{T_t = (\tau_1, \ldots, \tau_{n_T})\}_{t=0}^{N_t}$, $\{\lambda_t\}_{t=0}^{N_t}$, k,

and $\{\epsilon_t\}_{t=0}^{N_t}$, subject to $\sum_{t=0}^{N_t}\epsilon = 1$, $\epsilon_t, \lambda_t \geq 0$, $\sum_k \tau_k = 0$, and $\sum_k \tau_k^2 = 1$. The EM algorithm is applied again to learn the unknown parameters. Its updating procedure is very similar as that of the model (4.32) in Section 4.4.1. In M-step, ϵ_t, λ_t, and k are solved out directly, and matrix eigen decomposition is applied to get the closed form solution for T_t.

4.4.4 Experiment for the Improved Background Model

Compared with the experiment section 4.4.2, here we used the same dataset and learned the same amount of templates (32 templates), each with the same size 10×10. The only difference is that we used the improved background model given by (4.33). The learned ϵ_0 was equal to 0.9057, and $\lambda_0 = 81.0450$, $k = 0.4831$. This implies that the mean of the gamma distribution of $S_0(Y)$ was equal to 0.0060 (compared to 0.0057 in Section 4.4.2). Figure 4.19 shows 600 training image patches and the 32 learned templates. For the purpose of comparison between the basic background model and the improved background model, Figure 4.19 also shows in the third panel the 32 templates learned from the basic background model given in Section 4.4.2). By comparing two sets of learned templates in Figure 4.19, it is obvious that the improved model produced a bigger diversity among the same amount of templates. For example, under the improved model, templates with texture showed up as well. It is also worthy to notice that the flipping phenomena that appeared from the 2nd to 4th row in Figure 4.17 (the template evolution under the basic model) did not appear in Figure 4.20) (the template evolution under the improved model).

4.5 Discussion

In this chapter, we developed a generative probabilistic framework to model image patches, based on deformable representative templates. It can be used to model image patches from both foreground objects and background scenes. The templates

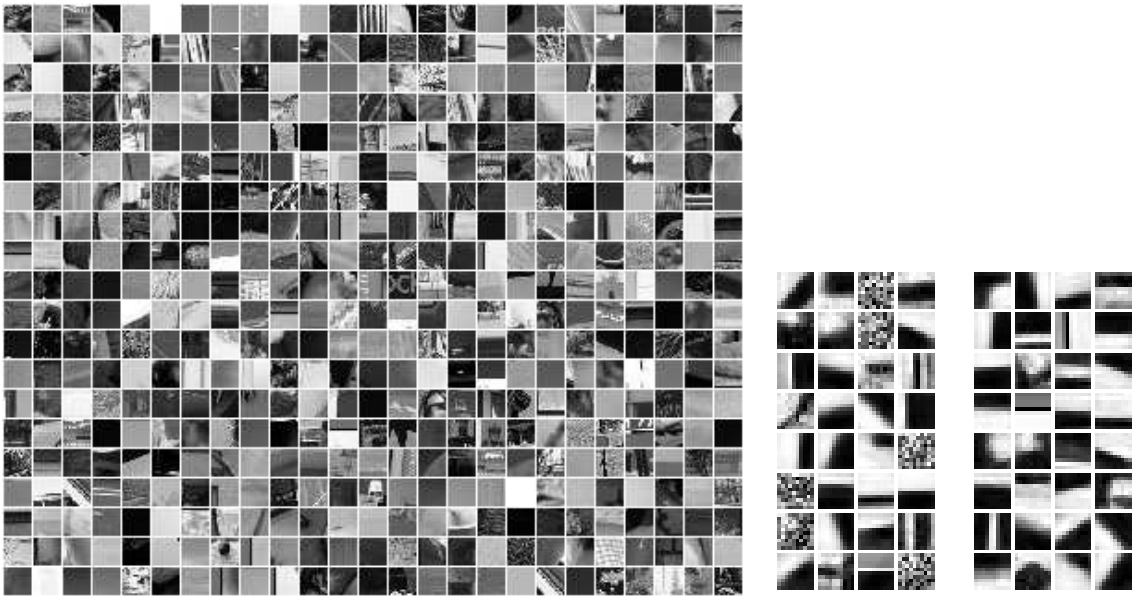

Figure 4.19: The left panel shows 500 randomly selected training image patches (15×15). The middle panel shows the 32 learned templates from the improved background model, each with size 10×10. The right panel shows the 32 learned templates from the basic background model, each with size 10×10.

learned from the model are useful building blocks in developing models for object detection and recognition.

There are a few aspects of this model that can be improved. First, the marginal distribution of the sufficient statistics was modeled as backwards truncated exponential distribution, each with a single parameter λ. This distribution approximated the empirical histogram of the sufficient statistics of the real image data relatively well, but not perfectly. The empirical histogram looked more like a gamma distribution. Its peak was close to $+1$, but not exactly at $+1$. Hence it will better fit the training data if the marginal distribution is modeled in a more flexible form – for example, a gamma distribution with two parameters instead of exponential distribution with a single parameter. And we can learn its unknown parameters through the EM algorithm, much as what we did with the current model. Second, the shape of the

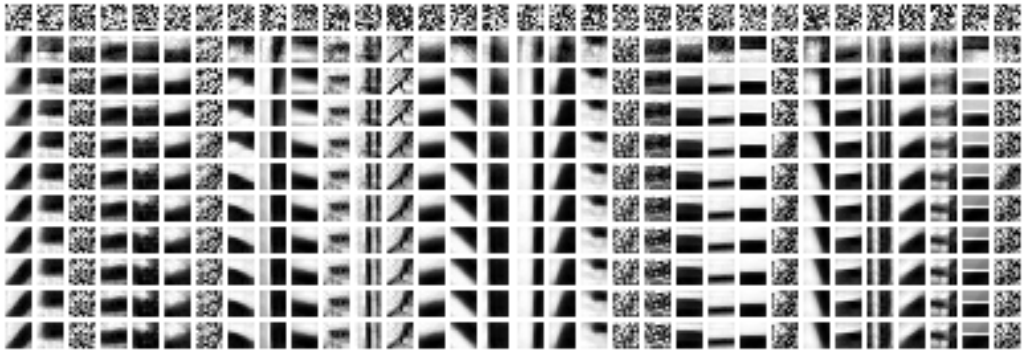

Figure 4.20: The evolution of the 32 templates during 20 runs of the EM algorithm under the improved background model, where the first row shows the random initialization of the templates.

templates was modeled as a rectangle. An important generalization would be to use non-rectangular templates. For example, we might use oval-shaped templates to model a face and almond-shaped templates to model an eye. One way to achieve this is to manually design the shape of the templates, i.e., to give binary weights, either 0 or 1, to each single pixel of the rectangular templates. Alternatively, a better approach would be to associate a weight with each pixel of the rectangular templates. These weights could then be learned automatically along with the pixel values of templates. Third, the number of templates in our model was manually selected. It has not been studied how to pick an optimal number of templates and how this number influences the model performance. There are a few questions left to answer, e.g., does it always improve performance to have more templates? At this point one possible practical solution would be to compare the performance of several models with different numbers of templates and to pick the best one. Certainly, this type of model selection would be very time consuming. It would be better to do a theoretical analysis on the relation between the number of the templates and the model's performance, and to select the optimal number of templates accordingly.

Another aspect worth attention is data collection. In this chapter, all the facial

image patches used in our experiments were cropped from Feret face images with the help of their pre-labeled landmarks. Without these landmarks, we would have to manually collect the facial image patches one by one, and that would be labor intense. The bad news is that in reality useful landmarks are not available in general, let alone good training image patches that are ready to use. However, the good news is that we may be able to take advantage of existing methods to ease the data collection for certain applications. For example, face detection has been thoroughly studied in computer vision; one successful method of face detection has been the Viola-Jones boosting method, [43]. The existing methods work well for face detection, but not face recognition. These face detectors can be used to collect datasets for our model as follows: First we can apply the face detector to target the position of faces in images, then crop the facial parts according to the holistic spatial constraints of the parts layout. These cropped image patches will be a dataset for our model. Certainly, most of the collected image patches will not be calibrated – the interesting region will be off center; objects will have different sizes and rotations. But they are still good enough to provide training and serve the goal of face recognition, due to the fact that our model is invariant to spatial shifts, scales and rotations.

4.6 Appendix

1. How to sample random image uniformly given its normalized correlation with a fixed template, in Section 4.2.1.

Question: Given gray-scale template T with n pixels, $T = \{t_1, t_2, \ldots, t_n\}$, and a real number $\rho_0 \in [-1, 1]$, how would we sample a gray-scale image $Y = \{y_1, y_2, \ldots, y_n\}$ uniformly from the set $\{Y \in \{0, \ldots, 255\}^n : S(Y, T) = \rho_0\}$? Here $S(Y, T)$ is the normalized correlation between Y and T, and is defined as

$$S(Y,T) = \frac{\sum_{i=1}^{n}(t_i - \frac{1}{n}\sum_{j=1}^{n} t_j)\sum_{i=1}^{n}(y_i - \frac{1}{n}\sum_{j=1}^{n} y_j)}{\sqrt{\sum_{i=1}^{n}(t_i - \frac{1}{n}\sum_{j=1}^{n} t_j)^2}\sqrt{\sum_{i=1}^{n}(y_i - \frac{1}{n}\sum_{j=1}^{n} y_j)^2}}$$

Solution: Let $T = \vec{t}$, $Y = \vec{y}$. Without loss of generality, the original question is equivalent to: given $T = \vec{t} \in R^n$, with $\sum_{i=1}^n t_i = 0$ and $\sum_{i=1}^n t_i^2 = 1$, we want to sample $Y = \vec{y} \in R^n$ with $\sum_{i=1}^n y_i = 0$ and $\sum_{i=1}^n y_i^2 = 1$ such that $\sum_{i=1}^n t_i y_i = \rho_0$. Observe that $\vec{t}, \vec{y} \in (n-1)$ dimension unit sphere and $\cos \angle(\vec{t}, \vec{y}) =< \vec{t}, \vec{y} >= S(T, Y)$, hence $\|OA\| = \rho_0$, $\|AB\| = \sqrt{1-\rho_0^2}$. If we can sample $\vec{AB}$, denoted by $\vec{z}$, then a desirable $\vec{y}$ will be $\vec{y} = \rho_0 \cdot \vec{t} + \vec{z}$. Now the question comes down to how to sample $\vec{z}$. To guarantee that $\sum_{i=1}^n y_i = 0$ and $\sum_{i=1}^n y_i^2 = 1$, $\vec{z}$ should satisfy

$$\begin{cases} \sum_{i=1}^n z_i = 0 \\ \vec{z} \perp \vec{t} \\ \|z\| = \sqrt{1-\rho_0^2}. \end{cases}$$

Let $\vec{e} = (1, 1, \ldots, 1) \in R^n$, observe the first two conditions above is equivalent to $\vec{z} \perp \alpha(\vec{t}, \vec{e})$, where $\alpha(\vec{t}, \vec{e})$ is the space spanned by $\vec{t}$ and $\vec{e}$. From this observation, if we can sample $\vec{x}$ uniformly from the $(n-1)$ dimension unit sphere, and project it onto $\alpha(\vec{t}, \vec{e})^c$, the complement space of $\alpha(\vec{t}, \vec{e})$, we will get a desirable $\vec{z}$ as $\vec{z} = \vec{x_p} \cdot \frac{\sqrt{1-\rho_0^2}}{\|\vec{x_p}\|}$, where $\vec{x_p}$ denotes the projected vector from $\vec{x}$.

Detail:

(1). How to get a random sample from the $(n-1)$ dimensional unit sphere: Let $\vec{0} = (0, \ldots, 0) \in R^n$ and I be the unit matrix with size n. Sample a n dimensional vector $\vec{v} = (v_1, v_2, \ldots, v_n) \sim N(\vec{0}, I)$, the n-D Gaussian distribution. Let $\vec{x} = \frac{\vec{v}}{\|v\|}$, then $\vec{x}$ is a random sample from $(n-1)$ dimension unit sphere.

(2). How to project $\vec{x}$ onto $\alpha(\vec{t},\vec{e})^c$: Define a matrix

$$A = \begin{pmatrix} \vec{t} \\ \vec{e} \\ \vec{0} \\ \vdots \\ \vec{0} \end{pmatrix}_{n\times n}.$$

Let $\vec{\mu_1},\ldots,\vec{\mu_k}$ be all the eigenvectors corresponding to the "0" eigenvalue of A. Define a size n-by-k matrix $B=(\vec{\mu_1},\ldots,\vec{\mu_k})$ and a size n-by-n matrix $P = B\cdot(B^*\cdot B)^{-1}\cdot B^*$, then P is our projection matrix, i.e., $P\vec{x} = \vec{x_p}$.

2. Approximate $P(S(Y))$ by Central Limit Theorem, in Section 4.2.1.

Let $Y=(y_1,y_2,\ldots,y_n)$, $T=(t_1,t_2,\ldots,t_n)$, where, $\{y_i\}$ are i.i.d. random variable with uniform distribution on $\{0,1,\ldots,255\}$; $\{t_i\}$ are constants with $\sum_i t_i = 0$ and $\sum_i t_i^2 = 1$. Denote $m=E(y_i)$, $\sigma^2 = Var(y_i)$, $\overline{y} = (1/n)\sum y_i$, and

$$S(Y) = \text{corrcoef}(Y,T) = \frac{\sum(y_i-\overline{y})t_i}{\sqrt{\sum(y_i-\overline{y})^2}} = \frac{1}{\sqrt{n}}\cdot\frac{\sum(t_iy_i/\sigma)}{\sqrt{\frac{1}{n\sigma^2}\cdot\sum(y_i-\overline{y})^2}}.$$

By Strong Law of Large numbers,

$$\frac{\sum(y_-m)^2}{n} \to\ E[(y_1-m)^2]=\sigma^2; \qquad \frac{\sum(y_i-m)}{n} = \to\ E[y_1]-m=0,$$

hence

$$\frac{1}{n\sigma^2}\cdot\sum(y_i-\overline{y})^2 = \frac{1}{n\sigma^2}\cdot\sum(y_i-m)^2 - (\frac{1}{n\sigma}\cdot\sum(y_i-m))^2 \longrightarrow 1 \text{ a.s., as } n\to\infty.$$

By generalized Central Limit Theorem,

$$\sum_i \frac{t_i y_i}{\sigma} \xrightarrow{D} N(0,1), \text{ as } n \to \infty,$$

if $\forall \epsilon > 0$,

$$\sum_{i=1}^{n} E\left[(t_i y_i - t_i m)^2 \cdot 1_{\{|t_i y_i - t_i m| > \epsilon\}}\right] \longrightarrow 0, \text{ as } n \to \infty. \tag{4.34}$$

Condition (1) is satisfied if

$$\max_i |t_i| \longrightarrow 0, \text{ as } n \to \infty. \tag{4.35}$$

Under our settings, T stands for the normalization for an meaningful image template, hence (2) is true except for the trivial templates, for example T only has a fixed number of pixels with non-zero values as $n \to \infty$. But we don't consider such trivial templates in our model. Therefore, when n is large, we can approximate the distribution of $S(Y)$ by $N(0, 1/n)$.

3. Proof for "$v_m \cdot s_m = 0, \ \forall m = 1, 2, \ldots, n$," Eqn. (4.21).

Recall that

$$V = \sum_i \lambda_l \hat{P}_l^{(i)} Y_i \qquad A = \sum_i \frac{n}{2} \hat{P}_l^{(i)} Y_i (Y_i)^\top = \sum_m a_m e_m e_m^\top,$$

n-D column vector $e = (1, 1, \ldots, 1)^\top$, and $s_m = e^\top e_m$. By the definition of A and that Y_i has been normalized, i.e. $e^\top Y_i = 0$, we have

$$\begin{aligned} e^\top A e = 0 &\implies \sum a_m e^\top e_m e_m^\top e = 0 \\ &\implies \sum a_m (e^\top e_m)^2 = 0. \end{aligned}$$

Since $a_m \geq 0, (e^\top e_m)^2 \geq 0$, we have $\forall m$, either $e^\top e_m = 0$ or $a_m = 0$. If $e^\top e_m = 0$,

then $s_m = 0$, hence $v_m s_m = 0$ is proved. If $a_m = 0$, since a_m is the eigen value of A,

$$\begin{aligned} Ae_m = 0 &\implies \sum_i \hat{P}_l^{(i)} Y_i (Y_i)^\top e_m = 0 \\ &\implies e_m^\top \sum_i \hat{P}_l^{(i)} Y_i (Y_i)^\top e_m = 0 \\ &\implies \sum_i \hat{P}_l^{(i)} (e_m^\top Y_i)^2 = 0. \end{aligned}$$

Since $\hat{P}_l^{(i)} \geq 0$, it must be that $\forall i$, either $\hat{P}_l^{(i)} = 0$ or $e_m^\top Y_i = 0$, hence

$$v_m = e_m^\top V = \sum_i \hat{P}_l^{(i)} e_m^\top Y_i = 0,$$

and $v_m s_m = 0$ is true automatically.

4. Approximate $P_0(S_0(Y))$ by Central Limit Theorem, in Section 4.4.1.

Let $Y = (y_1, y_2, \ldots, y_n)$, where $\{y_k\}$ are i.i.d. random variable with uniform distribution on $\{0, 1, \ldots, 255\}$. Denote $m = E(y_k)$, $\sigma^2 = Var(y_k)$, $\overline{y} = (1/n)\sum y_k$ and

$$S_0(Y) = \frac{1}{n}\sum_k (y_k - \overline{y})^2 = \frac{1}{n}\sum_k (y_k - m)^2 - \frac{1}{n^2}(\sum_k (y_k - m))^2.$$

Observe that

$$\begin{aligned} E\left[\frac{1}{n}\sum_k (y_k - m)^2\right] &= \mathrm{Var}(y_1), \\ E\left[\frac{1}{n^2}\,(\sum_k (y_k - m))^2\right] &= \frac{1}{n^2}\,E\left[\sum_k (y_k - m)^2 + \sum_{k \neq j} (y_k - m)(y_j - m)\right] \\ &= \frac{1}{n^2}[\; n\mathrm{Var}(y_1) + 0\;] = \frac{1}{n}\mathrm{Var}(y_1), \end{aligned}$$

hence,

$$E[S_0(Y)] \approx \mathrm{Var}(y_1) = 5461.25.$$

As for the variance of S_0, from the derivation of $E[S_0]$ above we can see that as $n \to +\infty$,

$$S_0 = \frac{1}{n}\sum_k (y_k - \overline{y})^2 \approx \frac{1}{n}\sum_k (y_k - m)^2.$$

Let $z_k = (y_k - m)^2$, then $\{z_1, \ldots, z_n\}$ are i.i.d. uniformly distributed from the set $\{0.5^2, 1.5^2, \ldots, 126.5^2, 127.5^2\}$. Then

$$S_0 = \frac{1}{n}\sum_k (y_k - \overline{y})^2 \approx \frac{1}{n}\sum_k (y_k - m)^2 = \frac{1}{n}\sum z_k.$$

By Central limit theorem,

$$\frac{1}{\sqrt{n}}\sum (z_k - E[z_1]) \xrightarrow{D} N(0,\ \mathrm{Var}(z_1)), \quad \text{as } n \to +\infty.$$

Since

$$S_0 \approx \frac{1}{n}\sum z_k = \frac{1}{\sqrt{n}}\left(\frac{1}{\sqrt{n}}\sum (z_k - E[z_1])\right) + E[z_1],$$

when n is big enough, we can approximate the distribution of S_0 by a Gaussian distribution with mean $E[z_1] = \mathrm{Var}(y_1) = 5461.25$ and variance $\mathrm{Var}(z_1)/n = \frac{4884.6^2}{n}$.

5. Derivation of $M_{s,r,l}$, the projection matrix of T_t onto the coordinate of image patch Y, under scale s, rotation r, and location shift l, in Section 4.2.3.

$g_{s,r,l}(T_t) = M_{s,r,l} \cdot T_t$, is the projected template on the coordinate of Y. The projection from T_t to $g_{s,r,l}(T_t)$ is composed of two steps. Step 1 scales, rotates, and shifts the pixel grid of T_t under s, r, and l. It moves the (i, j) pixel of T_t to a new location $(\hat{i}, \hat{j})$ w.r.t. the coordinate of Y. Note that $i, j \in \mathcal{Z}$, $\hat{i}, \hat{j} \in \mathcal{R}$. For each pixel (k_1, k_2) of $g_{s,r,l}(T_t)$, step 2 computes its value by averaging the values of pixels of T_t whose transformed location $(\hat{i}, \hat{j})$ falls in a small neighborhood of (k_1, k_2).

Step 1: $(i, j) \xrightarrow{(s,r,l)} (\hat{i}, \hat{j})$. Step 2: We take the Blackman window method, [42].

For each pixel (k_1, k_2) of $g_{s,r,l}(T_t)$, its pixel value is equal to

$$\sum_{(\hat{i},\hat{j})} 1_{\{d_{ij}\leq 1\}} \cdot [0.42 + 0.5\cos(\pi \cdot d_{ij}) + 0.08\cos(2\pi \cdot d_{ij})] \cdot T(i,j),$$

where

$$d_{ij} = \begin{cases} \frac{\|(k_1,k_2)-(\hat{i},\hat{j})\|_2}{R\cdot b_s}, & \text{if } \|(k_1,k_2)-(\hat{i},\hat{j})\|_2 \leq R\cdot b_s, \\ 0, & \text{otherwise.} \end{cases},$$

where $\|\cdot\|_2$ above stands for L-2 norm on the Y coordinate; b_s is the s^{th} scaling factor for T_t; R is a constant, in charge of the size of the neighborhood of (k_1, k_2), and we pick $R = 1.6$.

$M_{s,r,l}$ is uniquely determined by the two steps above.

Chapter 5

Conclusion and Future Directions

Conclusion

We believe the "hierarchy of reusabe parts" moves us closer to bridging the ROC gap which persistently occurs in comparison of human and machine performance in vision. As we know, with quite a bit of training, current machines can achieve a reasonable detection rate at an allowable false positive rate. Nevertheless, every percentage of improvement towards perfect detection (i.e. no missing target) is overwhelmed by massive false positive targets. The underlying reason is that there was never a proper background model. In some sense, people all try to separate background from foreground. They model the generative background as simple as white noise or as complicated as MRF depending on the assumption, while modeling the foreground object in a totally different manner, one not compatible with the background model. We all know the mistakes (false positive) occur at the most ambiguous region (the cluttered region). In this region, the image patch looks like everything including the target to the machine if you have to set one threshold for a test of object versus non-object. It is easy to see that those cluttered regions are made of the same parts as the foreground. This prevailing sharing phenomena paired with predominant false detection in this region suggests an object equipped with its own background model, that is, a model that accommodate both objects

and background which consist of the same reusable parts, in a uniform way. The composition system embodies this idea, and we have shown that theoretically the ROC gap can be narrowed more than ever. The maximum likelihood template gives a solution to how to model image data given the upper level image interpretations, and also provides a way to model objects and background on a same platform.

Future Directions

1. Continuous study of maximum likelihood templates.

The joining of constituents to make a composition (e.g. a left eye, a right eye, and a forehead to make a top half of a face) will in general depend on the poses (e.g. position, scale, rotation) of the constituents. This dependency can be conveniently formulated in terms of a likelihood ratio: the likelihood of an interpretation involving the constituents composed into a single entity divided by the likelihood of the same interpretation in which the constituents are related only by chance. The entire distribution on image interpretations can be stitched together from the specification of all such likelihood ratios. These likelihood ratios, and their dependence on the poses of constituent parts, raise the important question of coordinate systems. A compelling argument can be made for using only *relational* coordinate systems, so as to define an object independent of it's particular presentation in three-dimensional space. But the formulation of a suitable relational coordinate system is not straightforward, due mostly to the nonlinearities introduced by scale.

2. Applications in medical imaging.

In medical imaging, the term atlas usually refers to a (probabilistic) model of a population of images, with the parameters learned from a training dataset. In its simplest form, an atlas is a mean intensity image. Atlases are used for various purposes including normalization of new subjects for structure and function locations, segmentation or parcellation of certain structures of interest and group analyses to identify pathology related changes or developmental trends. We will explore the applications of our maximum likelihood templates in these areas, especially in the

study of MRI images. The learned templates will play the role of atlases. Since MRI images are 3D, we need to generalize our model from 2D to 3D, and the unit concept from "pixel" to "voxel". Since our model is not constrained by the dimensionality of the image data, this generalization is straightforward. Another issue that we need to address to is the specialty of medical images compared to natural images. Minor local differences between two MRI images can lead to opposite diagnosis results. Currently our models only consider the rigid global transformation of templates, e.g. scales and rotations, and this is not enough for modeling medical images. We have to consider local deformation in addition to the current global transformation, e.g. 3D nonlinear transformation model parameterized via B-splines [50].

3. Video Analysis.

One advantage for using generative (Bayesian) models is the availability of a mathematically coherent extension to multi-source data. Thus, there is nothing in principle that confines a compositional model to a single frame. This generative framework can be adapted to solve, for example, video-related tasks. Given a compositional hierarchy for a face or body, and a "state equation" for a trajectory of parts, the generative form of the model generates image sequences (video) rather than single images. The "inverse" problem, the problem of identifying high-likelihood interpretations under the posterior distribution, becomes a version of the tracking problem.

Bibliography

[1] V. N. Vapnik. *The nature of Statistical Learning Theory* Springer-Verlag, Berlin, 1995.

[2] C. M. Bishop. *Neural networks for pattern recognition.* Oxford University Press, Oxford, UK, 1995.

[3] Y. Freund and R. E. Schapire. *A decision-theoretic generalization of on-line learning and an application to boosting.* Journal of Computer and System Sciences, 55(1): 119-139, 1997.

[4] F. Fleuret and D. Geman. *Coarse-to-fine face detection.* Inter. Journal of Computer Vision, 41, 85-107, 2001.

[5] U. Grenander. *General pattern theory: a study of regular structures.* Oxford University Press, 1993.

[6] S. Geman, K. Manbeck, and E. McClure. *Coarse-to-fine search and rank-sum statistics in object recognition.* Technical report, Division of Applied Mathematics, Brown University, 1995.

[7] S. Geman and M. Johnson. *Article title Probability and statistics in computational linguistics, a brief review* . Mathematical foundations of speech and language processing. M. Johnson; S. P. Khudanpur; M. Ostendorf; R. Rosenfeld (Eds.), X, ISBN: 0-387-20326-5, pp. 1-26, 2004.

[8] Y. Amit, D. Geman, and X. Fan. *A coarse-to-fine strategy for multi-class shape detection.* IEEE Trans. PAMI, 2004.

[9] B. Ommer and J. M. Buhmann. *Learning the compositional nature of visual objects.* In CVPR'07, IEEE, 2007.

[10] R. Fergus, P. Perona, and A. Zisserman. *Object Class Recognition by Unsupervised Scale-Invariant Learning.* CVPR'03, IEEE, Vol. 2 (2003), pp. 264-271, 2003.

[11] E. Bienenstock, S. Geman, and D. Potter. *Compositionality, MDL priors, and object recognition.* M.C. Mozer, M.I. Jordan, and T. Petsche, editors, Advances in Neural Information Processing Systems, Vol. 9, pp. 838-844, MIT Press, 1997

[12] S. H. Huang. *Compositional approach to recognition using multi-scale computations.* Ph.D. thesis, Division of Applied Mathematics, Brown University, 2001.

[13] M. Harrison *Discovering compositional structure.* Ph.D. thesis, Division of Applied Mathematics, Brown University, 2005.

[14] S. Geman, D. F. Potter, and Z. Chi. *Composition Systems.* Quarterly of Applied mathematics, LX: 707-736, 2002.

[15] Ya Jin. *Non-Markovian Hierarchy Vision System.* PhD thesis, Brown University, Division of Applied Mathematics, 2006.

[16] Y. Jin and S. Geman. *Context and hierarchy in a probabilistic image model.* In CVPR'06, volume (2), pp. 2145-2152. IEEE, 2006.

[17] W. T. Freeman, J. Yedidia, and Y. Weiss. *Understanding belief propagation and its generalizations.* International Joint Conference on Artificial Intelligence, 2001.

[18] M. Riesenhuber and T. Poggio. *Models of object recognition.* Nature Neuroscience, supplement, volume 3: 1199-1204, 2000.

[19] Y. Amit and A. Trouve. *Pop: Patchwork of parts models for object recognition.* Technical report, University of Chicago, 2004.

[20] M. Weber, M. Welling, and P. Perona. *Unsupervised learning of models for recognition.* Proc. Sixth European Conf. computer Vision, pp. 18-32, 2000.

[21] 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.[pdf]

[22] 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.

[23] L. Zhu, Y. Chen, X. Ye, and A. Yuille. *Structure-Perceptron Learning of a Hierarchical Log-Linear Model.* In CVPR, 2008.

[24] David M. Green & John A. Swets (1966). *Signal Detection Theory and Psychophysics.* New York: Wiley.

[25] Thomas M. Cover and Joy A. Thomas (1991). *Elements of Information Theory, pp. 300-301.* New York: Wiley.

[26] E. Sali and S. Ullman. *Combining class-specific fragments for object classification.* In proc. 10th British Machine Vision Conference, volume 1, 203-213, 1999.

[27] S. Ullman, E. Sali, and M. Vidal-Niquet. *A fragment-based approach to object representation and classification.* In IWVF, pp. 85-100, 2001.

[28] E. Borenstein and S. Ullman. *Class-specific, top-down segmentation*, In ECCV02, LNCS 2353, pp. 109-122, 2002.

[29] B. Epshtein and S. Ullman. *Feature hierarchies for object classification.* In ICCV, 2005.

[30] B. Heisele, T. Serre, M. Pontil, T. Vetter, and T. Poggio. *Categorization by learning and combining object parts.* In NIPS, 2001.

[31] B. Heisele, T. Serre, and T. Poggio. *A component-based framework for face detection and identification.* Int. J. of Comp. Vision 74(2), pp. 167-181, 2007.

[32] T. Serre, A. Oliva, and T. Poggio. *A feedforward architecture accounts for rapid categorization.* Proceedings of the National Academy of Science, 104(15), pp. 6424-6429, April 2007.

[33] D. H. Hubel and T. N. Wiesel. *Receptive fields and functional architecture of monkey striate cortex.* J. Physiol., 195, pp. 215-243, 1968.

[34] T. Serre, A. Oliva, T. Poggio. *A feedforward architecture accounts for rapid categorization.* Proceedings of the National Academy of Sciences, vol. 104, no. 15, 2007.

[35] S. Geman. *Invariance and selectivity in the ventral visual pathway.* Journal of Physiology-Paris, Vol. 100, No. 4., pp. 212-224, October 2006.

[36] R. Fergus, P. Perona, and A. Zisserman. *Object class recognition by unsupervised scale-invariant learning.* In CVPR 2003.

[37] B. Leibe and B. Schiele. *Interleaved object categorization and segmentation.* Proceedings of British Machine Vision Conference (BMVC), 2003.

[38] S. Agarwal, A. Awan, and D. Roth. *Learning to detect objects in images via a sparse part-based representation.* IEEE TPAMI, 26(11): 1475-1490, 2004.

[39] 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, 2007, vol. 69, issue 1, pp. 3-29.

[40] M. R. Sabuncu, S. K. Balci, and P. Golland. *Discovering Modes of an Image Population through Mixture Modeling.* Proceedings of the International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI), LNCS 5242, p. 381-389, 2008.

[41] Z. Si, H. Gong, S.C. Zhu, and Y.N. Wu. *Learning Active Basis Models by EM-Type Algorithms.* Statistical Science, 2009.

[42] R. B. Blackman and J. W. Tukey. *The measurement of power spectra.* Dover Publications, New York, 1958.

[43] P. Viola and M. Jones. *Rapid object detection using a boosted cascade of simple features.* In IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2001.

[44] B. A. Olshausen and D. J. Field. *Sparse coding with an overcomplete basis set: A strategy employed by V1?.* Vision Research, 37(23): 3311-3325, Dec. 1997.

[45] G.E. Hinton. *Products of experts.* In Int. Conf. on Art. Neur. Netw. (ICANN), vol. 1, pp. 1-6, Sept. 1999.

[46] M. Welling, G.E. Hinton, and S. Osindero. *Learning Sparse Topographic Representationswith Products of Student-t Distributions.* In Adv. in Neur. Inf. Proc. Sys. (NIPS), vol 15, pp. 1359-1366, 2003.

[47] S. Roth and M. J. Black. *Fields of experts: A framework for learning image priors.* In IEEE Conf. on Computer Vision and Pattern Recognition, vol. II, pp. 860-867, June 2005.

[48] S. Roth and M. J. Black. *Fields of experts*. In International Journal of Computer Vision, 82(2): 205-229, Apr. 2009.

[49] S. C. Zhu and D. Mumford. *Prior learning and Gibbs reaction-diffusion*. IEEE Trans. Patter Anal. Mach. Intell., 19(11): 1236-1250, Nov. 1997.

[50] D. Rueckert, et al.. *Nonrigid registration using free-form deformations: application to breast MR images*. IEEE Transaction of Medical Imaging 18, 712-721, 1999.