Discovering Compositional Structure

by
Matthew T. Harrison
B.A., University of Virginia, 1998
Sc.M., Brown University, 2000

Doctoral dissertation
Ph.D. Advisor: Stuart Geman

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 2005

Chapter 2

Model building by perturbation

2.1 Introduction

In this chapter we describe an unsupervised learning heuristic for incrementally building hierarchical, parts-based models. The hierarchy is built from the bottom up by adding new parts. In principle, the process of detecting and adding a new part is recursive, i.e., agnostic to the size of the hierarchy. Furthermore, it is not specific to vision problems, but could be potentially useful for other sensory modalities. The main idea is that dependencies are only incorporated into the model through the introduction of new parts.

A major deficiency of the setup here is that we do not address computation, only representation. We partially remedy this by formulating the problem in terms of probabilistic graphical models. Computation using these types of representations has been studied extensively and is still an active area of research. Nevertheless, efficient computation in compositional systems is currently an unsolved problem and it is not clear that the learning heuristic used here will create representations that can actually be used for computation. This becomes evident in Chapters 3 and 4 where we can only partially experiment with the learning heuristic.

The framework focuses on building an internal model that can represent certain aspects of the external world. The hope is that a good statistical model of the world that also happened to be compositional would likely be a useful representation for many visual tasks like segmentation and recognition. We do not directly address this issue here.

2.2 Learning heuristic

We want to interpret some data Y in terms of various discrete parts X. Y might be an image and X might be lines, T-junctions and L-junctions of various lengths, scales, positions and orientations. X is our internal representation of the external data Y.

One way to approach this problem is to define a generative model that specifies a prior distribution P_X over the internal states X and a conditional distribution $P_{Y|X}$ for the data given the internal states. Interpretation then proceeds in Bayesian manner using the posterior distribution $P_{X|Y}$ or some related quantity. We are not concerned here with the specifics of interpretation, but rather how to choose a good generative model.

2.2.1 Evaluating the model

The generative model implicitly defines a marginal distribution on the data:

$$P_Y(A) = \sum_x P_{X,Y}(x, A) = \sum_x P_{Y|X}(A|x)P_X(x).$$

This can be quite different from the "world's distribution" $\mathbb{P}_Y$ on the data. From a probabilistic point of view, the ideal goal is to have our distribution on Y match the world's distribution on Y, that is

$$P_Y \overset{\text{goal}}{=} \mathbb{P}_Y.$$

This is a difficult goal to affirm; typically Y takes values in a high-dimensional space. Collecting evidence for failure is somewhat easier.

Just as the model implicitly defines a distribution on Y, the world implicitly defines a distribution on X by reversing the model:

$$\mathbb{P}_X(x) = \int_y P_{X|Y}(x|y)\mathbb{P}_Y(dy) = \mathbb{E}_Y\big[P_{X|Y}(x|Y)\big],$$

and similarly for any statistic $S = S(X)$ of the internal states:

$$\mathbb{P}_S(s) = \mathbb{E}_Y\big[P_{S|Y}(s|Y)\big],$$

where $\mathbb{E}_Y$ denotes expectation with respect to $\mathbb{P}_Y$. Note that since X is discrete, S is also discrete.

If $P_Y = \mathbb{P}_Y$, then $E_Y = \mathbb{E}_Y$ and

$$\mathbb{P}_X(x) = \mathbb{E}_Y\big[P_{X|Y}(x|Y)\big] \overset{\text{goal}}{=} E_Y\big[P_{X|Y}(x|Y)\big] = P_X(x).$$

Similarly,

$$\mathbb{P}_S(s) \overset{\text{goal}}{=} P_S(s).$$

If S is simple, then this final implication of $P_Y = \mathbb{P}_Y$ might be a good place to look for problems with the model. In particular, approximating $\mathbb{E}_Y$ with an empirical distribution $\hat{\mathbb{E}}_Y$ over a collection of data $y_1, \ldots, y_n$, gives

$$\hat{\mathbb{P}}_S(s) = \hat{\mathbb{E}}_Y\big[P_{S|Y}(s|Y)\big] = \frac{1}{n}\sum_{k=1}^{n} P_{S|Y}(s|y_k) \approx \mathbb{E}_Y\big[P_{S|Y}(s|Y)\big] = \mathbb{P}_S(s) \overset{\text{goal}}{=} P_S(s).$$

A significant departure from this goal, namely

$$\hat{\mathbb{P}}_S \not\approx P_S, \tag{1}$$

suggests that $P_Y \not\approx \mathbb{P}_Y$ and indicates a problem with the model.

2.2.2 Improving the model

Suppose that for some statistic $S = S(X)$ we see the failure mode described in (1). One way to possibly improve the model is to modify the distribution of S while leaving the rest of the model unchanged. In particular, since $S = S(X)$ is a function of X, we can express P_X as

$$P_X(x) = P_{X,S}(x, S(x)) = P_{X|S}(x|S(x))P_S(S(x))$$

and then modify P_S to get a new prior distribution on X. The data model $P_{Y|X}$ remains unchanged.

The particular suggestion here is to perturb P_S by slightly mixing it with another distribution P_S^1 to get

$$P_S^\epsilon = (1-\epsilon)P_S + \epsilon P_S^1,$$

This creates new distributions on X and Y, namely,

$$P_X^\epsilon(x) = P_{X|S}(x|S(x))P_S^\epsilon(S(x)) \quad \text{and} \quad P_Y^\epsilon(A) = \sum_x P_{Y|X}(A|x)P_X^\epsilon(x),$$

for $\epsilon \in [0,1]$. As long as P_S^1 does not put positive probability on any impossible (i.e., $P_S(s) = 0$) values for $S(X)$, everything makes sense. Note that $\epsilon = 0$ corresponds to no perturbation: $P_S^0 = P_S$, $P_X^0 = P_X$ and $P_Y^0 = P_Y$.

The perturbation improves the model if P_Y^ϵ is an improved approximation of $\mathbb{P}_Y$. We can quantify this with relative entropy:

$$D\big(\mathbb{P}_Y \big\| P_Y^\epsilon\big) \overset{\text{goal}}{<} D\big(\mathbb{P}_Y \big\| P_Y\big). \tag{2}$$

Since $D\big(\mathbb{P}_Y \big\| P_Y^\epsilon\big)$ is convex in ϵ, it should be possible to choose a good ϵ. (For convexity, see Lemma 2.2.2 below. To avoid technicalities, we will always assume that $D\big(\mathbb{P}_Y \big\| P_Y\big) < \infty$.) The hard part is finding an appropriate statistic S and distribution P_S^1. The next theorem gives a potential search criterion. A proof can be found at the end of this section.

Theorem 2.2.1. Equation (2) is achievable for some ϵ if and only if

$$\mathbb{E}_S\left[\frac{P_S^1(S)}{P_S(S)}\right] > 1, \tag{3}$$

where $\mathbb{E}_S$ denotes expectation with respect to $\mathbb{P}_S$. If (3) holds, then $\mathbb{P}_S \neq P_S$.

The proof also shows that the larger the left side of (3) then the larger the improvement in relative entropy, at least for small perturbations (ϵ near 0).

Theorem 2.2.1 essentially says that we can improve the model if we can find a statistic S and a distribution P_S^1 such that *on average* $S(X)$ is more likely under P_S^1 than P_S. The key is that *on average* means we average over internal states X driven by the world's distribution on Y and the current posterior distribution $P_{X|Y}$. Interpretation will typically be based on some approximation to the posterior, so the computations needed for interpretation under the current model should also produce the relevant statistics needed to evaluate potential improvements to the model.

All of this suggests looking for statistics S and distributions P_S^1 such that

$$\hat{\mathbb{E}}_S\left[\frac{P_S^1(S)}{P_S(S)}\right] \gg 1, \tag{4}$$

where $\hat{\mathbb{E}}_S$ is expectation with respect to $\hat{\mathbb{P}}_S$.

Proof of Theorem 2.2.1. P_X^ϵ and P_Y^ϵ are also additive mixtures:

$$P_X^\epsilon = (1-\epsilon)P_X + \epsilon P_X^1 \quad \text{and} \quad P_Y^\epsilon = (1-\epsilon)P_Y + \epsilon P_Y^1.$$

Relative entropy is convex in both arguments, so $D(\epsilon) := D\big(\mathbb{P}_Y \big\| P_Y^\epsilon\big)$ is convex in ϵ on $[0,1]$. Since $D(0) = D\big(\mathbb{P}_Y \big\| P_Y\big)$, (2) is achievable if and only if $D'(0) < 0$.

Lemmas 2.2.2 and 2.2.3 compute $D'(0)$. The first gives

$$D'(0) = 1 - \mathbb{E}_Y\left[\frac{dP_Y^1}{dP_Y}(Y)\right]$$

and the second gives

$$\frac{dP_Y^1}{dP_Y}(Y) = E_{S|Y}\left[\frac{dP_S^1}{dP_S}(S)\middle| Y\right].$$

To apply the second lemma we take $(X,Y) = (S,Y)$ under the current model and $(X',Y') = (S,Y)$ under the $\epsilon = 1$ altered model with $P_S = P_S^1$. Recall that we assumed that P_S^1 puts probability one on the support of P_S, so $P_S^1 \ll P_S$, $P_X^1 \ll P_X$ and $P_Y^1 \ll P_Y$.

In the discrete setting here $[dP_S^1/dP_S](s) = P_S^1(s)/P_S(s)$. Also, by definition $\mathbb{E}_S[\cdot] = \mathbb{E}_Y[E_{S|Y}[\cdot|Y]]$. These give

$$D'(0) = 1 - \mathbb{E}_S\big[P_S^1(S)/P_S(S)\big],$$

which we need to be negative. Note that $\mathbb{P}_S = P_S$ gives

$$D'(0) = 1 - E_S\big[P_S^1(S)/P_S(S)\big] = 1 - \textstyle\sum_{s:P_S(s)>0} P_S^1(s) \geq 1 - 1 = 0$$

for any P_S^1, so (3) cannot hold. □

Lemma 2.2.2. Let Q, P^0 and P^1 be probability measures with $D(Q\|P^0) < \infty$ and $P^1 \ll P^0$. Define $P^\epsilon = (1-\epsilon)P^0 + \epsilon P^1$. Then $D(\epsilon) := D(Q\|P^\epsilon)$ is real valued on $[0,1)$ and convex on $[0,1]$ with $D'(0) = 1 - E_Q[dP^1/dP^0]$. (The derivative is only evaluated from the right.)

Proof. It is well known that relative entropy is always nonnegative, so $D(\epsilon) \geq 0$. Let $\tilde{P}^\epsilon$

represent the absolutely continuous component of P^ϵ with respect to Q. We have

$$\begin{aligned} D(\epsilon) &= E_Q \log \frac{dQ}{dP^\epsilon} = -E_Q \log \frac{d\tilde{P}^\epsilon}{dQ} = -E_Q \log \left[(1-\epsilon) \frac{d\tilde{P}^0}{dQ} + \epsilon \frac{d\tilde{P}^1}{dQ} \right] \\ &= -E_Q \log \left[(1-\epsilon) \frac{d\tilde{P}^0}{dQ} + \epsilon \frac{d\tilde{P}^1}{d\tilde{P}^0} \frac{d\tilde{P}^0}{dQ} \right] = -E_Q \log \left[1 + \epsilon \big(\frac{d\tilde{P}^1}{d\tilde{P}^0} - 1 \big) \right] - E_Q \log \frac{d\tilde{P}^0}{dQ} \\ &= D(Q \| P^0) - E_Q \log \left[1 + \epsilon \big(\frac{dP^1}{dP^0} - 1 \big) \right] = D(0) - E_Q \log \left[1 + \epsilon f \right], \end{aligned}$$

where the next to last equality holds because $d\tilde{P}^1/d\tilde{P}^0 = dP^1/dP^0$ a.s. Q and where we define $f = dP^1/dP^0 - 1$. Note that $f \geq -1$, so we have the trivial bound $0 \leq D(\epsilon) \leq D(0) - \log(1-\epsilon) < \infty$ for $\epsilon \in [0,1)$. Note also that the concavity of the logarithm implies that $D(\epsilon)$ is convex on $[0,1]$.

Using the previous calculations gives

$$D'(0) = \lim_{\epsilon \downarrow 0} \frac{D(\epsilon) - D(0)}{\epsilon} = \lim_{\epsilon \downarrow 0} \frac{-E_Q \log \left[1 + \epsilon f \right]}{\epsilon}.$$

If we move the limit inside the expectation, then we get $D'(0) = -E_Q f = 1 - E_Q[dP^1/dP^0]$ as claimed. So we need only justify exchanging the limit and the integration.

Let $h_\epsilon = \epsilon^{-1} \log(1 + \epsilon f)$. Since $f \geq -1$, $h_\epsilon \geq \epsilon^{-1} \log(1-\epsilon) \uparrow -1$ as $\epsilon \downarrow 0$. On the set $\{f \leq 0\}$, $h_\epsilon \leq 0$ so the dominated convergence theorem can be applied here. On the set $\{f > 0\}$, $h_\epsilon > 0$ and we can write

$$h_\epsilon = \left[\frac{\log(1 + \epsilon f)}{\epsilon f} \right] f,$$

which is increasing as $\epsilon \downarrow 0$. The monotone convergence theorem completes the proof (even in the case where $E_Q[dP^1/dP^0] = \infty$). □

Lemma 2.2.3. Let (X, Y) and (X', Y') be random elements with regular conditional distributions $P_{Y'|X'} = P_{Y|X}$ and with $P_{X'} \ll P_X$. Then $P_{Y'} \ll P_Y$ and

$$\frac{dP_{Y'}}{dP_Y}(Y) = E\left[\frac{dP_{X'}}{dP_X}(X) \middle| Y \right] \text{ a.s.}$$

Proof. For absolute continuity, note that $P_Y(B) = 0$ implies $\int_B P_{Y|X}(dy, x) = 0$ a.s. P_X and thus a.s. $P_{X'}$. Replacing $P_{Y|X}$ with $P_{Y'|X'}$ and integrating w.r.t. $P_{X'}$ shows that $P_{Y'}(B) = 0$.

For the main result, we will show that the left side satisfies the definition of the conditional expectation on the right side. Clearly the left side is $\sigma(Y)$-measurable. If $A \in \sigma(Y)$, then

$A = \{Y \in B\}$ for some B and we can compute

$$\begin{aligned}\int_A \frac{dP_{X'}}{dP_X}(X)dP &= \int_{\mathcal{X}\times B} \frac{dP_{X'}}{dP_X}(x)P_{X,Y}(d(x\times y)) \\ &= \int_{\mathcal{X}} \frac{dP_{X'}}{dP_X}(x)P_X(dx)\int_B P_{Y|X}(x,dy) = \int_{\mathcal{X}} P_{X'}(dx)\int_B P_{Y'|X'}(x,dy) \\ &= \int_{\mathcal{X}\times B} P_{X',Y'}(d(x\times y)) = P_{Y'}(B) = \int_B \frac{dP_{Y'}}{dP_Y}(y)P_Y(dy) = \int_A \frac{dP_{Y'}}{dP_Y}(Y)dP. \qquad \square\end{aligned}$$

2.2.3 Updating the model

Once we have found a candidate statistic $S = S(X)$ and a distribution P_S^1 that can improve the model via a perturbation, we need to update the model to incorporate this improvement. We will do this by introducing a new part Z to the representation X in such a way that the new distribution on S is exactly P_S^ϵ.

Let $Z \in \{0,1\}$. Define the prior on the new state space $X' = (X,Z)$ as

$$P_{X'}(x,z) = P_{X|S}(x|S(x))\big[(1-z)(1-\epsilon)P_S(S(x)) + z\epsilon P_S^1(S(x))\big],$$

and define the data model as

$$P_{Y|X'}(A|(x,z)) = P_{Y|X}(A|x).$$

The marginal of X under $P_{X'}$ is exactly P_X^ϵ. Z indicates which of the two mixture components is present with $Z = 1$ corresponding to the new component P_X^1. The data model remains the same, so the marginal of Y under the new model is P_Y^ϵ which is the intended perturbation of P_Y.

Suppose X has multiple components (parts) $X = (X_1,\ldots,X_N)$ and suppose P_X respects the dependency graph G with vertices corresponding to the X_i's, i.e., P_X factors into a product of functions defined only on the cliques of G. Then $X' = (X_1,\ldots,X_N,Z)$ and $P_{X'}$ respects the graph G' which adds a vertex Z to G and adds a clique (X_S,Z), where X_S are those components of X that S depends on, i.e., $S(x) = S(x_S)$. This follows from the alternative representation

$$\begin{aligned}P_{X'}(x,z) &= P_{X|S}(x|S(x))\big[(1-\epsilon)P_S(S(x))\big]^{(1-z)}\big[\epsilon P_S^1(S(x))\big]^z \\ &= \frac{P_X(x)}{P_S(S(x))}\big[(1-\epsilon)P_S(S(x))\big]^{(1-z)}\big[\epsilon P_S^1(S(x))\big]^z \\ &= P_X(x)(1-\epsilon)\left(\frac{\epsilon P_S^1(S(x_S))}{(1-\epsilon)P_S(S(x_S))}\right)^z.\end{aligned}$$

If G facilitates computation under the original model and if X_S is small, then G' will likely facilitate computation under the new perturbed model.

2.2.3.1 Parameter estimation

The perturbed model depends on a parameter $\epsilon = P_Z(1)$. We need to choose an appropriate value of ϵ in order to guarantee that the perturbed model is better than the original, that is, to guarantee (2). (Recall that P_Y^ϵ is the perturbed model's distribution on Y.) Ideally, we want to choose ϵ that minimizes $D\big(\mathbb{P}_Y \big\| P_Y^\epsilon\big)$.

The minimizing ϵ satisfies the fixed point equation

$$\mathbb{P}_Z(1) = P_Z(1) = \epsilon, \tag{5}$$

where $\mathbb{P}_Z(z) = \mathbb{E}_Y\big[P_{Z|Y}(z|Y)\big]$ and where P_Z and $P_{Z|Y}$ refer to the perturbed model with parameter ϵ. Note that each term in (5) depends on ϵ. A proof is given at the end of this section, where we also show that the minimizing ϵ exists and is unique.

The fixed point (5) makes sense intuitively. When we introduce a new variable Z into the model, we want its distributions under the model and under the world to be the same.

Finding the fixed point (approximately) is more or less straightforward. $\mathbb{P}_Z$ can be approximated by $\hat{\mathbb{P}}_Z$ which should arise naturally from using the model for interpretation. The parameter ϵ can then be adjusted up or down appropriately, perhaps with a simple neural network-like learning rule or perhaps by recursively setting $\epsilon = \hat{\mathbb{P}}_Z(1)$. This latter strategy is a stochastic version of the EM algorithm.

Proof of (5). We assume that $P_Y^1 \neq P_Y$, which, for example, is implied by Theorem 2.2.1 when (3) holds. (If they are equal then ϵ has no effect on the model, i.e., $P_Y^\epsilon = P_Y$, and (2) cannot hold.) So $D\big(\mathbb{P}_Y \big\| P_Y^\epsilon\big)$ is strictly convex in ϵ on $[0,1]$ and it has a unique minimizer ϵ^*.

Consider the perturbed model with $\epsilon = \epsilon^*$. In order to derive a contradiction, suppose that $\mathbb{P}_Z(1) \neq P_Z(1) = \epsilon^*$. Fixing the perturbed model, we can use the setup from Section 2.2.2 to imagine perturbing the perturbed model. In particular, we will substitute Z for S, $\mathbb{P}_Z$ for P_S^1 and δ for ϵ. We will also use $P_{Y'}$ and $P_{Y'}^\delta$ to denote the new marginals on Y. Since $\mathbb{E}_Z\big[\mathbb{P}_Z(Z)/P_Z(Z)\big] > 1$, Theorem 2.2.1 implies that $D\big(\mathbb{P}_Y \big\| P_{Y'}^\delta\big) < D\big(\mathbb{P}_Y \big\| P_{Y'}\big)$. But $P_{Y'} = P_Y^{\epsilon^*}$ and $P_{Y'}^\delta = P_Y^\epsilon$ for some $\epsilon \neq \epsilon^*$, which contradicts the fact that ϵ^* is the unique minimizer of $D\big(\mathbb{P}_Y \big\| P_Y^\epsilon\big)$.

We can derive a similar contradiction in the other direction by supposing that $\mathbb{P}_Z(1) = P_Z(1) \neq \epsilon^*$. Using the same changes in notation, except now substituting $(1-\epsilon^*)^{(1-z)}\epsilon^{*z}$ for P_S^1, Theorem 2.2.1 implies that $D\big(\mathbb{P}_Y \big\| P_{Y'}^1\big) \geq D\big(\mathbb{P}_Y \big\| P_{Y'}\big)$. But this is impossible since $P_{Y'}^1 = P_Y^{\epsilon^*}$ and $P_{Y'} = P_Y^\epsilon$ for $\epsilon \neq \epsilon^*$. □

2.2.4 Recursive model building

The above method begins with a candidate statistic S and distribution P_S^1. If these can improve the model, say (4) holds, then the model is perturbed slightly by adding a new variable Z. Z interacts with and modifies the joint distribution of the components of X that S depends on. The perturbation is local: the state space, its distribution and presumably computation are all slightly modified, but only in the neighborhood of Z.

In principle, this model perturbation strategy can be applied recursively to incrementally grow a large graphical model in an unsupervised fashion. We interpret each new categorical

variable as a feature, or a part. A new part modifies the distribution of a certain subset of previous parts, which we can think about as the subparts of the new part. This introduces a natural hierarchy. Since the subsets can overlap, the parts are reusable.

Mathematically, each new part adds a new mixture component to the model's implicit distribution on the data. Viewed in this way, the learning strategy approximates the true distribution on the data with a large mixture model. The criterion is likelihood. In practice, all of the typical issues that arise from using large mixture models, such as overfitting and robustness, are likely to be of great importance.

2.3 Suspicious coincidences

Another major issue is how to generate candidate statistics S and distributions P_S^1. One possibility is to look for departures from independence, that is, each statistic S depends on a subset of components of X that are independent under the current model and each distribution P_S^1 introduces dependencies among these components. If these are the only type of statistics entertained by the model, then all dependencies arise from parts. In a sense, dependencies are parts.

Suppose that the current model $X = (X_1, \ldots, X_N)$ has two parts X_i and X_j that are independent under the model and suppose that their respective distributions have been tuned to fit the world's distribution: $\mathbb{P}_{X_k} = P_{X_k}$, $k = i, j$. Consider the candidate statistic $S(X) = S(X_i, X_j) = \mathbb{1}\{X_i \in A_i, X_j \in A_j\}$ for $i \neq j$. Since X_i and X_j are independent under the model, the distribution of S is easy to compute, namely,

$$P_S(1) = P_{X_i,X_j}(A_i \times A_j) = P_{X_i}(A_i)P_{X_j}(A_j) = \mathbb{P}_{X_i}(A_i)\mathbb{P}_{X_j}(A_j).$$

If we detect evidence that

$$\mathbb{P}_S(1) = \mathbb{P}_{X_i,X_j}(A_i \times A_j) \neq \mathbb{P}_{X_i}(A_i)\mathbb{P}_{X_j}(A_j) = P_S(1),$$

then X_i and X_j are dependent under the world's distribution and we would like to incorporate this into the model.

When $\mathbb{P}_S(1) > P_S(1)$, the distribution $P_S^1(s) = \mathbb{1}\{s = 1\}$, which is the point mass at 1, satisfies

$$\mathbb{E}_S\left[\frac{P_S^1(S)}{P_S(S)}\right] = \mathbb{P}_S(1)\frac{1}{P_S(1)} + 0 > 1,$$

and Theorem 2.2.1 implies that we can improve the model with S and P_S^1. Note that

$$\mathbb{P}_S(1)\frac{1}{P_S(1)} = \frac{\mathbb{P}_{X_i,X_j}(A_i \times A_j)}{\mathbb{P}_{X_i}(A_i)\mathbb{P}_{X_j}(A_j)},$$

so the criterion of interest becomes

$$\frac{\mathbb{P}_{X_i,X_j}(A_i \times A_j)}{\mathbb{P}_{X_i}(A_i)\mathbb{P}_{X_j}(A_j)} > 1. \tag{6}$$

If X_i and X_j are modeled as independent, then (6) is called a *suspicious coincidence*.

Clearly this generalizes to candidate statistics of the form

$$S(X) = S(X_{i_1}, \ldots, X_{i_m}) = \mathbb{1}\{X_{i_1} \in A_{i_1}, \ldots, X_{i_m} \in A_{i_m}\}$$

for independent X_{i_k}'s whose marginal distributions are the same under the model and under the world. The criterion then becomes an mth-order suspicious coincidence,

$$\frac{\mathbb{P}_{X_{i_1},\ldots,X_{i_m}}(A_{i_1} \times \cdots \times A_{i_m})}{\mathbb{P}_{X_{i_1}}(A_{i_1}) \cdots \mathbb{P}_{X_{i_m}}(A_{i_m})} > 1. \tag{7}$$

The candidate distribution is still $P_S^1(s) = \mathbb{1}\{s = 1\}$. A further generalization is

$$S(X) = S(X_{i_1}, \ldots, X_{i_m}) = \mathbb{1}\{(X_{i_1}, \ldots, X_{i_m}) \in A\},$$

in which case the criterion becomes

$$\frac{\mathbb{P}_{X_{i_1},\ldots,X_{i_m}}(A)}{\big(\mathbb{P}_{X_{i_1}} \times \cdots \times \mathbb{P}_{X_{i_m}}\big)(A)} > 1.$$

If the model has many components that are presumed to be independent, then this provides a large class of potential statistics that can be used to search for ways to improve the model. The search involves looking for suspicious coincidences. When one is detected, a new part, say Z, is added to the model that better models the dependency. In particular, if the statistic is $S(X_i, X_j) = \mathbb{1}\{X_i \in A_i, X_j \in A_j\}$, then $Z = 1$ in the updated model implies $X_i \in A_i$ and $X_j \in A_j$. That is, the presence of the new part implies a particular configuration of its constituent subparts. When $Z = 0$, the constituent parts are independent again. During interpretation, if there is evidence that $X_i \in A_i$ and $X_j \in A_j$, then the interpretation algorithm will have to decide if this happened "by chance" or if it happened "because" $Z = 1$. In the latter case, we say that X_i and X_j are composed into Z.

Iteratively detecting suspicious coincidences seems like a nice method for growing the hierarchical structures of a composition system. Imagine some sort of compositional algorithm that interprets an image by identifying parts (or features) and their relationships. These are then composed into larger parts and the algorithm iterates. At the highest or final level the algorithm gives an interpretation of the scene and behaves as if there are no more parts that should be composed into larger objects. Looking for suspicious coincidences among these "high-level" objects would indicate whether or not new compositions are required.

For example, suppose the current system only knows about small lines or edges and builds an interpretation of an image out of these basic parts. Let A be the occurrence of a small vertical line in one part of an image and let B be the occurrence of a small vertical line just above A. It seems reasonable that A and B are strongly correlated events because they will both occur whenever there is a larger line that encompasses them both. In particular $\text{Prob}(AB) \gg \text{Prob}(A)\,\text{Prob}(B)$. Over time we can detect this as a suspicious coincidence and create a new feature C which is the composition of A and B – a longer line. Now that the algorithm knows about C it can be used to create simpler interpretations of images. This process will iterate to grow larger and larger features.

In Chapters 3 and 4 we will focus exclusively on model building by detecting suspicious coincidences. The notion of using suspicious coincidences to learn about the world is not new.

2.3.1 Finding suspicious coincidences

One of the major roles of the brain is detecting associations. There has been some speculation that this is the primary goal of the cerebral cortex. Barlow has advanced the idea that the particular associations of interest are suspicious coincidences [15, 2, 5, 6]. The general idea is nearly identical to the one here: find suspicious coincidences and learn to anticipate them so that they are no longer suspicious. This brings up the issue of how to find suspicious coincidences.

For any collection of N features, there are N^2 possible binary compositions and 2^N possible compositions of all orders. Already these numbers are unmanageable; we cannot consider every possible composition. Adding in several different types of composition relationships (to the right of, to the left of, etc.) only makes things worse. Barlow and colleagues immediately recognized this [15] and several mechanisms have been postulated for dealing with it [6], including:

- Only look for coincidences that are likely to occur or be of importance, for example, coincidences whose components have similar spatial locations and scales.
- Only look for coincidences whose components occur frequently. If the components rarely occur, then the combination of them will hardly ever happen.
- Use sparse representations. Densely distributed and/or highly repetitive representations make it difficult to determine when a coincidence is suspicious.

The utility of the first two principles is relatively straightforward, although it is certainly not clear exactly how they should be implemented in an actual algorithm. The notion of sparse coding is somewhat less intuitive. Field [7] cites at least three separate considerations that have led people to suggest sparseness as an important coding principle: improved signal-to-noise ratio, simpler and more reliable detection of statistical dependencies, and higher capacity in associative memory networks. In a distributed representation, sparse coding means that a given item is represented by only a few of the many units. Barlow [4, 6] points out that sparsity is important for detecting suspicious coincidences and also uses physiological evidence to argue that visual cortex uses a sparse code [1, 3]. In a densely distributed representation a given feature of interest will be represented by a complex activity pattern over many (neural) elements. Detecting a suspicious coincidence among high-level features requires keeping track of complex higher-order statistical dependencies. This will be memory intensive, inefficient and error-prone. Fortunately, compositionality does not lend itself to densely distributed representations. We think of each element in a representation as being a feature or a part which can stand alone. We do not necessarily need to know the state of all the other elements in order to interpret a single element. In this sense, a compositional representation is ideal for detecting suspicious coincidences.

Sparsity is also important to ensure that a collection of features is well-separated. If it is not, but includes a lot of similar or identical repetitions, then detecting suspicious

coincidences will become inefficient. These similar elements will certainly be highly correlated and the new feature created by composing them together will again be quite similar. Nothing much has been gained by this composition. Also, for every composition in which a certain feature plays a role, all of its similar features will play roles in similar compositions. Both of these things cause the number of compositions detected by suspicious coincidences to explode. Avoiding this type of departure from sparsity is crucial for compositional learning.

In Chapter 3 we use some simple heuristics to keep our representations sparse by making sure that features are well separated within each level of the compositional hierarchy. Practically, this helps to alleviate the problems we just discussed. More theoretically, sparsity is important because it helps to justify the independence assumption in the suspicious coincidence criterion. We discuss this further in the next section.

2.3.2 Sparse coding and independence

An implicit assumption for the general framework here is that we can generate a large class of statistics whose distributions are well understood under the current model. This seems like a strong assumption. One of the key ideas behind using coincidence detectors is that we only need the current model to have many independent components. But even this is likely to be too strong of an assumption. Or if it is a valid assumption, then it might be difficult to determine which components should be independent under the current model.

A possible remedy is to relax the strict independence requirement. If X_i and X_j are nearly independent under the current, then $\mathbb{P}_{X_i,X_j}(A_i \times A_j) \gg \mathbb{P}_{X_i}(A_i)\mathbb{P}_{X_j}(A_j)$ would still indicate a deficiency in the model. Interestingly, the simple requirement of sparsity goes a long way toward the ideal of independence and may even be better than a truly independent code (a factorial code) because the latter could be densely distributed, which is problematic for detecting associations (not to mention the conceptual difficulties of creating a compositional factorial code).

That sparsity tends towards independence is easy to see from a well-known heuristic argument. If $X = (X_1, \ldots, X_n)$ is a distributed signal (random vector), then its entropy $H(X)$ quantifies (inversely) how much statistical structure exists in the signal. If we recode the signal more sparsely, say $Y = f(X)$, $Y = (Y_1, \ldots, Y_n)$, then we preserve the total amount of entropy, $H(Y) = H(X)$, but we reduce the entropy of the individual components, $H(Y_k) \leq H(X_k)$, because a sparse random variable has low entropy. This implies that

$$\sum_{k=1}^{n} H(Y_k) - H(Y) \leq \sum_{k=1}^{n} H(X_k) - H(X).$$

Since each side of this equation is a measure of the amount of statistical dependency that exists among the elements, we have moved toward independence by sparse coding. Furthermore, in a certain sense we can nearly achieve independence with the sparsest possible code (sometimes called grandmother cells), which although far from independent will nevertheless have no more than a bit or so of higher-order redundancy. This simple calculation is detailed in the next section.

2.3.2.1 Grandmother cells are nearly independent

X is a signal (random variable) that takes at most N values, labeled $1, \ldots, N$. For example, if $X = (X_1, \ldots, X_n)$ is a distributed signal and each $X_k \in \{0, 1\}$ is binary valued, then we can take N to be 2^n. If we recode the signal as sparsely as possible, say $Y = f(X)$, $Y = (Y_1, \ldots, Y_N)$, $Y_K = \mathbb{1}\{X = K\}$, so that each possible value of X excites a unique element of Y (grandmother cells), then this new representation has at most $\log_2 e \approx 1.44$ bits of higher-order redundancy. In this sense the Y_K's are nearly independent. They cannot be completely independent because only a single Y_K is active at any given time.

Let $p_K := \text{Prob}(X = K)$. The higher order redundancy in Y is

$$\begin{aligned}
\sum_{K=1}^{N} H(Y_K) - H(Y) &= \sum_{K=1}^{N} H(Y_K) - H(X) \\
&= -\sum_{K=1}^{N} [p_K \log p_K + (1 - p_K) \log(1 - p_K)] + \sum_{K=1}^{N} p_K \log p_K \\
&= -\sum_{K=1}^{N} (1 - p_K) \log(1 - p_K) \\
&\leq -\sum_{K=1}^{N} \left(1 - \frac{1}{N}\right) \log \left(1 - \frac{1}{N}\right) = \log \left(1 + \frac{1}{N-1}\right)^{N-1} \leq \log e.
\end{aligned}$$

The first inequality is easy to derive from the fact that the uniform distribution maximizes entropy once we note that $\sum_{K=1}^{N}(1 - p_K) = N - 1$ is fixed.

2.4 Other related work

Unsupervised learning of hierarchical representations is not a well understood problem [17], although there is an extensive literature describing various supervised (or semi-supervised) hierarchical learning procedures, especially within the neural network community. There is also a variety of work attempting to create learning procedures that can be roughly mapped onto the hierarchical organization of visual cortex, for example, [9, 18]. For a brief review of general (i.e., not necessarily hierarchical) unsupervised learning, see [10].

Perhaps the work most closely related to the ideas here are the various feature pursuit algorithms for sequential learning of additive random field models [16, 21, 20, 22] and several closely related algorithms for projection pursuit density estimation [8, 11] and independent components analysis [12, 19]. These algorithms build increasingly better approximations for a complicated, high-dimensional distribution by successively modeling (or matching) various lower-dimensional statistics. Typically, however, there is no notion of hierarchy. New features are not built out of previous ones, but instead capture statistics that were not captured by previous features. (Although, the higher-order ICA algorithm in [13] seems like it could be iterated hierarchically.) Also, the features tend not to be categorical, i.e., either present or absent, but instead can be present in varying degrees.

A notable exception is the work by Della Pietra, Della Pietra and Lafferty (1997) [16]

which has many similarities to the work here. The features are categorical, newer features are composed entirely out of older features and the parameters of the current model are fixed before adding a new feature. They apply their model to unsupervised learning of English spellings. After incorporating 1500 features, sampling from the model produces "words" that show many of the statistical properties of English words. An important difference is that they do not interpret a new feature as adding a new vertex to the underlying graphical model. This keeps the size of the graph small, but allows the connectivity to become quite dense.

2.5 Discussion

Motivated by compositionality, we derived an iterative, unsupervised learning heuristic that can build hierarchical, parts-based, probabilistic graphical models. The general theme is that unexplained dependencies in the data become new parts in the generative model. This framework leads naturally to Barlow's notion of detecting suspicious coincidences and to other unsupervised learning principles like sparse coding. It also embeds Barlow's suspicious coincidence ideas firmly within a probabilistic modeling framework, something that they have been criticized for lacking [14].

In the next two chapters we partially experiment with this algorithm. In particular, we investigate whether or not suspicious coincidences can be used to create hierarchies of increasing selectivity and invariance from natural images. As will be evident from these experiments, many details need to be worked out, not the least of which is computation. Another important issue involves sparsity, which is evidently important, but which we have incorporated in an ad hoc manner. We hope to address some of these issues in future work. We also hope to draw tighter connections between this work and existing unsupervised learning algorithms.

Bibliography

[1] H. B. Barlow. Single units and sensation: A neuron doctrine for perceptual psychology? *Perception*, 1:371–394, 1972.

[2] H. B. Barlow. Cerebral cortex as model builder. In David Rose and Vernon G. Dobson, editors, *Models of the visual cortex*, pages 37–46. John Wiley & Sons, Chichester, 1985.

[3] H. B. Barlow. The Twelfth Bartlett Memorial Lecture: The role of single neurons in the psychology of perception. *The Quarterly Journal of Experimental Psychology*, 37A:121–145, 1985.

[4] H. B. Barlow. Unsupervised learning. *Neural Computation*, 1:295–311, 1989.

[5] H. B. Barlow. Vision tells you more than "what is where". In Andrei Gorea, editor, *Representation of Vision: Trends and tacit assumptions in vision research*, pages 319–329. Cambridge University Press, Cambridge, 1991.

[6] Horace Barlow. What is the computational goal of the neocortex? In Christof Koch and Joel L. Davis, editors, *Large-Scale Neuronal Theories of the Brain*, pages 1–22. MIT Press, Cambridge, 1994.

[7] D. J. Field. What is the goal of sensory coding? *Neural Computation*, 6:559–601, 1994.

[8] J. Friedman, W. Stuetzle, and A. Schroeder. Projection pursuit density estimation. *Journal of the American Statistical Association*, 79:599–608, 1984.

[9] Karl Friston. Learning and inference in the brain. *Neural Networks*, 16:1325–1352, 2003.

[10] Colin Fyfe. Trends in unsupervised learning. In *Proceedings of the European Symposium on Artificial Neural Networks (ESANN)*, volume 21–23, pages 319–326, Bruges, Belgium, April 1999.

[11] Peter Huber. Projection pursuit. *Annals of Statistics*, 13(2):435–475, 1985.

[12] A. Hyvärinen. Fast and robust fixed-point algorithms for independent component analysis. *IEEE Transactions on Neural Networks*, 10(3):626–634, 1999.

[13] Yan Karklin and Michael S. Lewicki. Learning higher-order structures in natural images. *Network: Computation in Neural Systems*, 14:483–499, 2003.

[14] David Mumford. Neuronal architectures for pattern-theoretic problems. In Christof Koch and Joel L. Davis, editors, *Large-Scale Neuronal Theories of the Brain*, pages 125–152. MIT Press, Cambridge, 1994.

[15] C. G. Phillips, S. Zeki, and H. B. Barlow. Localization of function in the cerebral cortex: past, present and future. *Brain*, 107(1):327–361, March 1984.

[16] Stephen Della Pietra, Vincent Della Pietra, and John Lafferty. Inducing features of random fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(4):380–393, April 1997.

[17] Tomaso Poggio and Steve Smale. The mathematics of learning: Dealing with data. *Notices of the American Mathematical Society*, 50(5):537–544, May 2003.

[18] Maximilian Riesenhuber and Tomaso Poggio. Are cortical models really bound by the "binding problem". *Neuron*, 24:87–93, September 1999.

[19] Max Welling, Richard S. Zemel, and Geoffrey E. Hinton. Probabilistic independent components analysis. *IEEE Transactions on Neural Networks*, 15(4):838–849, July 2004.

[20] Song Chun Zhu and David Mumford. Prior learning and gibbs reaction-diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(11):1236–1250, November 1997.

[21] Song Chun Zhu, Ying Nian Wu, and David Mumford. Minimax entropy principle and its application to texture modeling. *Neural Computation*, 9(8):1627–1660, November 1997.

[22] Song Chun Zhu, Yingnian Wu, and David Mumford. Filters, random fields and maximum entropy (FRAME): Towards a unified theory for texture modeling. *International Journal of Computer Vision*, 27(2):107–126, 1998.