

Structure Learning in Conditional Probability Models via an Entropic Prior and Parameter Extinction

Matthew Brand

Mitsubishi Electric Research Labs, Cambridge Research Center, Cambridge, MA 02139, U.S.A.

We introduce an entropic prior for multinomial parameter estimation problems and solve for its maximum a posteriori (MAP) estimator. The prior is a bias for maximally structured and minimally ambiguous models. In conditional probability models with hidden state, iterative MAP estimation drives weakly supported parameters toward extinction, effectively turning them off. Thus, structure discovery is folded into parameter estimation. We then establish criteria for simplifying a probabilistic model's graphical structure by trimming parameters and states, with a guarantee that any such deletion will increase the posterior probability of the model. Trimming accelerates learning by sparsifying the model. All operations monotonically and maximally increase the posterior probability, yielding structure-learning algorithms only slightly slower than parameter estimation via expectation-maximization and orders of magnitude faster than search-based structure induction. When applied to hidden Markov model training, the resulting models show superior generalization to held-out test data. In many cases the resulting models are so sparse and concise that they are interpretable, with hidden states that strongly correlate with meaningful categories.

1 Introduction

Probabilistic models are widely used to model and classify signals. There are efficient algorithms for fitting models to data, but the user is obliged to specify the structure of the model: How many hidden variables? Which hidden and observed variables interact? Which are independent? This is particularly important when the data are incomplete or have hidden structure, in which case the model's structure is a hypothesis about causal factors that have not been observed. Typically a user will make several guesses; each may introduce unintended assumptions into the model. Testing each guess is computationally intensive, and methods for comparing the results are still debated (Dietterich, 1998). The process is tedious but necessary. Structure is the primary determinant of a model's selectivity and speed of computation. Moreover, if one shares the view that science seeks to discover lawful

relations between hidden processes and observable effects, structure is the only part of the model that sheds light on the phenomenon that is being modeled.

Here we show how to fold structure learning into highly efficient parameter estimation algorithms such as expectation-maximization (EM). We introduce an entropic prior and apply it to multinomials, which are the building blocks of conditional probability models. The prior is a bias for sparsity, structure, and determinism in probabilistic models. Iterative maximum a posteriori (MAP) estimation using this prior tends to drive weakly supported parameters toward extinction, sculpting a lower-dimensional model whose structure comes to reflect that of the data. To accelerate this process, we establish when weakly supported parameters can be trimmed from the model. Each transform removes the model from a local probability maximum, simplifies it, and opens it to further training. All operations monotonically increase the posterior probability, so that training proceeds directly to a (locally) optimal structure and parameterization. All of the attractive properties of EM are retained: polynomial-time reestimation, monotonic convergence from any nonzero initialization, and maximal gains at each step.¹

In this article we develop an entropic prior, MAP estimator, and trimming criterion for models containing multinomial parameters. We demonstrate the utility of the prior in learning the structure of mixture models and hidden Markov models (HMMs). The resulting models are topologically simpler and show superior generalization on average, where generalization is measured by the prediction or classification of held-out data. Perhaps the most interesting property of the prior is that it leads to models that are interpretable; one can often discover something interesting about the deep structure of a data set just by looking at the learned structure of an entropically trained model.

We begin by deriving the main results in section 2. In section 3 we use mixture models to illustrate visually the difference between entropic and conventional estimation. In section 4 we develop a “train and trim” algorithm for the transition matrix of continuous-output HMMs and experimentally compare entropically and conventionally estimated HMMs. In section 5 we extend the algorithm to the output parameters of discrete-output HMMs and explore its ability to find meaningful structure in data sets of music and text. In section 6 we draw connections to the literatures on HMM model induction and maximum-entropy methods. In section 7 we discuss some open questions and potential weaknesses of our approach. Finally, we show that entropic MAP estimator solves a classic problem in graph theory and raise

¹ Bauer, Koller, and Singer (1997) have pointed out that it is possible to have larger gains from initializations near the solution at a cost of losing convergence guarantees from all initializations.

some interesting mathematical questions that arise in connection with the prior.

2 A Maximum-Structure Entropic Prior

Even if one claims not to have prior beliefs, there are compelling reasons to specify a prior probability density function. The likelihood function alone cannot be interpreted as a density without specifying a measure on parameter space; this is provided by the prior. If the modeler simply wants the data to speak for themselves, then the prior should be noninformative and invariant to the particular way the likelihood function is parameterized. It is common to follow Laplace and specify a uniform prior $P_u(\boldsymbol{\theta}) \propto 1$ on parameter values $\boldsymbol{\theta} = \{\theta_1, \theta_2, \theta_3, \dots\}$, as if one knows nothing about what parameter values will best fit as-yet-unobserved evidence (Laplace, 1812). The main appeal of this noninformative prior is that the estimation problem reduces to maximum likelihood (ML) equations that are often conveniently tractable. However, the uniform prior is not invariant to reparameterizations of the problem (e.g., $\theta'_i = \exp \theta_i$), and it probably underestimates one's prior knowledge. Even if one has no prior beliefs about the specific problem, there are prior beliefs about learning and what makes a good model.

In entropic estimation, we assert that parameters that do not reduce uncertainty are improbable. For example, in a multinomial distribution over K mutually exclusive kinds of events, a parameter at chance $\theta_i = \frac{1}{K}$ adds no information to the model and is thus a wasted degree of freedom. On the other hand, a parameter near zero removes a degree of freedom, making the model more selective and more resistant to overfitting. In this view, learning is a process of increasing the specificity of a model, or equivalently, minimizing entropy. We can capture this intuition in a simple expression² that takes on a particularly elegant form in the case of multinomials:

$$P_e(\boldsymbol{\theta}) \propto e^{-H(\boldsymbol{\theta})} = \exp \sum_i \theta_i \log \theta_i = \prod_i \theta_i^{\theta_i} = \boldsymbol{\theta}^{\boldsymbol{\theta}}. \quad (2.1)$$

$P_e(\cdot)$ is noninformative to the degree that it does not favor one parameter set over another provided they specify equally uncertain models. It is invariant insofar as our entropy measure $H(\boldsymbol{\theta})$ is a function of the model's distribution, not its parameterization.

In section 6.1 we will discuss how this prior can be derived mathematically. Here we will concentrate on its behavior. The bolded convex curve in Figure 1a shows that this prior is averse to chance values and favors parameters near the extremes of $[0,1]$.

² We will use lowercase p for probabilities, capital P for probability density functions (pdf) and subscripted P_e for pdf's having an entropy term.

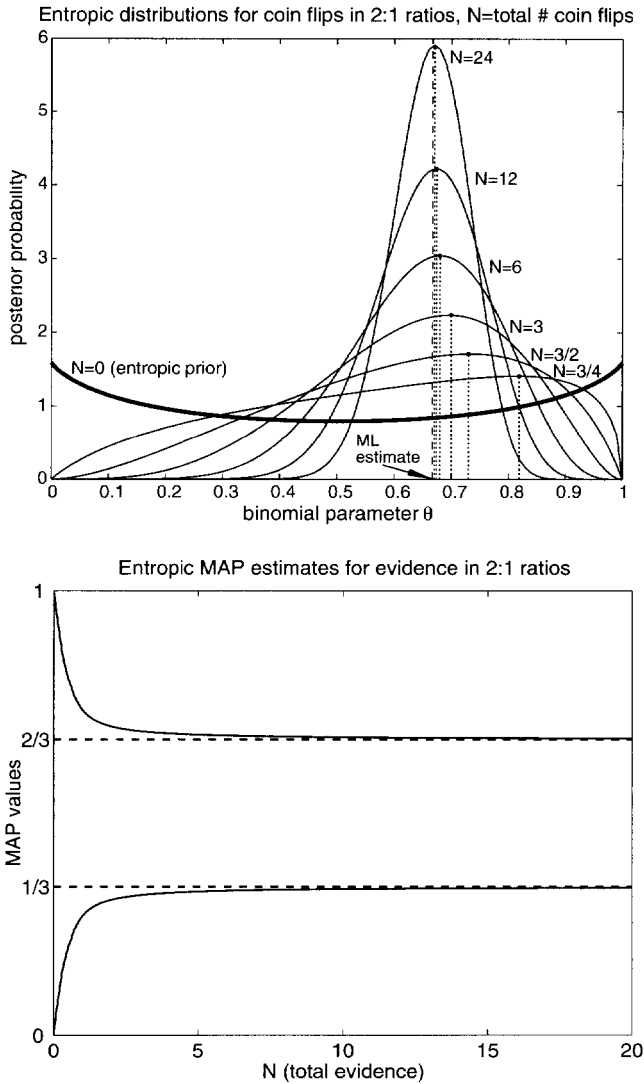


Figure 1: (a) Entropic posterior distributions of binomial models $\theta = \{\theta_h, \theta_t\}$, $\theta_h + \theta_t = 1$ for a weighted coin whose sample statistics $\omega = \{\omega_h, \omega_t\}$, $N = \omega_h + \omega_t$ indicate heads twice as often as tails ($\omega_h = 2\omega_t$). The mass of data is varied between curves. The boldface convex curve $P_e(\theta) \propto \exp(-H(\theta))$ shows how extremal values are preferred in the absence of evidence ($N = 0$). Dotted verticals show the MAP estimates. (b) MAP estimates as a function of the mass of data. As $N \rightarrow \infty$ the MAP estimates converge to the maximum likelihood (ML) estimates.

Combining the prior with the multinomial yields the posterior,

$$P_e(\boldsymbol{\theta}|\boldsymbol{\omega}) \propto P(\boldsymbol{\omega}|\boldsymbol{\theta})P_e(\boldsymbol{\theta}) \propto \left(\prod_i^N \theta_i^{\omega_i}\right) \left(\prod_i^N \theta_i^{\theta_i}\right) = \prod_i^N \theta_i^{\theta_i+\omega_i}, \quad (2.2)$$

where nonnegative ω_i is evidence for event type i .

As Figure 1a shows, with ample evidence this distribution becomes sharply peaked around the maximum likelihood estimate, but with scant evidence it flattens and skews to stronger odds. This is the opposite behavior that one obtains from a Dirichlet prior $\text{Dir}(\boldsymbol{\theta}|\alpha_1, \dots, \alpha_N)$, often used in learning Bayes' net parameters from data (Heckerman, 1996). With $\alpha_i > 1$, the Dirichlet MAP estimate skews to weaker odds.

The prior $P_e(\boldsymbol{\theta})$ was initially formulated to push parameters as far as possible from their noninformative initializations. We subsequently discovered an interesting connection to maximum entropy (ME) methods. ME methods typically seek the weakest (most noncommittal) model that can explain the data. Here we seek the strongest (sparsest, most structured, and closest to deterministic) model that is compatible with the data. In Brand (1999b) we resolve this apparent opposition by showing that our minimum-entropy prior can be constructed directly from maximum-entropy considerations.

2.1 MAP Estimator. The MAP estimator yields parameter values that maximize the probability of the model given the data. When an analytic form is available, it leads to learning algorithms that are considerably faster and more precise than gradient-based methods. To obtain MAP estimates for the entropic posterior, we set the derivative of log posterior to zero, using a Lagrange multiplier to ensure $\sum \theta_i = 1$.

$$0 = \frac{\partial}{\partial \theta_i} \left[\log \prod_i^N \theta_i^{\omega_i+\theta_i} + \lambda \left(\sum_i^N \theta_i - 1 \right) \right] \quad (2.3)$$

$$= \sum_i^N \frac{\partial}{\partial \theta_i} (\omega_i + \theta_i) \log \theta_i + \lambda \sum_i^N \frac{\partial}{\partial \theta_i} \theta_i \quad (2.4)$$

$$= \frac{\omega_i}{\theta_i} + \log \theta_i + 1 + \lambda. \quad (2.5)$$

This yields a system of simultaneous transcendental equations. It is not widely known that nonalgebraic systems of mixed polynomial and logarithmic terms such as equation 2.5 can be solved. We solve for θ_i using the Lambert W function (Corless, Gonnet, Hare, Jeffrey, and Knuth, 1996), an inverse mapping satisfying $W(y)e^{W(y)} = y$ and therefore $\log W(y) + W(y) = \log y$. Setting $y = e^x$ and working backward toward equation 2.5,

$$0 = -W(e^x) - \log W(e^x) + x \quad (2.6)$$

$$= \frac{-1}{1/W(e^x)} - \log W(e^x) + x + \log z - \log z \quad (2.7)$$

$$= \frac{-z}{z/W(e^x)} + \log z/W(e^x) + x - \log z. \quad (2.8)$$

Setting $x = 1 + \lambda + \log z$ and $z = -\omega_i$, equation 2.8 simplifies to equation 2.5:

$$\begin{aligned} 0 &= \frac{\omega_i}{-\omega_i/W(e^{1+\lambda+\log -\omega_i})} + \log -\omega_i/W(e^{1+\lambda+\log -\omega_i}) \\ &\quad + 1 + \lambda + \log -\omega_i - \log -\omega_i \\ &= \omega_i/\theta_i + \log \theta_i + 1 + \lambda \end{aligned} \quad (2.9)$$

which implies that

$$\theta_i = \frac{-\omega_i}{W(-\omega_i e^{1+\lambda})}. \quad (2.10)$$

Equations 2.5 and 2.10 define a fix-point for λ , which in turn yields a fast iterative procedure for the entropic MAP estimator: calculate θ given λ ; normalize θ ; calculate λ given θ ; repeat. λ may be understood as a measure of how much the dynamic range increases from ω to θ . Convergence is fast; given an initial guess of $\lambda = -\sum \omega_i - \langle \log \omega \rangle$ or $\theta_i \propto \omega_i^{1-1/\sum \omega_i}$ iff $\forall_i \omega_i \geq 1$, it typically takes two to five iterations to converge to machine precision. Since many of these calculations involve adding values to their logarithms, some care must be taken to avoid loss of precision near branch points, infinitesimals, and at dynamic ranges greater than $\text{ulp}(1)^{-1}$. In the last case, machine precision is exhausted in intermediate values, and we polish the result via Newton-Raphson. In appendix A we present some recurrences for computing W .

2.2 Interpretation. The entropic MAP estimator strikes a balance that favors fair (ML) parameter values when data are extensive, and biases toward low-entropy values when data are scarce (see Figure 1b). Patterns in large samples are likely to be significant, but in small data sets, patterns may be plausibly discounted as accidental properties of the sample, for example, as noise or sampling artifacts. The entropic MAP estimator may be understood to select the strongest hypothesis compatible with the data, rather than fairest, or best unbiased model. One might say it is better to start out with strong opinions that are later moderated by experience; correct predictions garner more credibility, and incorrect predictions provide more diagnostic information for learning. Note that the balance is determined by the mass of evidence and may be artificially adjusted by scaling ω .

Formally, some manipulation of the posterior (see equation 2.2) allows us to understand the MAP estimate in terms of entropies:

$$-\max_{\theta} \log P_e(\theta|\omega) = \min_{\theta} -\log \prod_i^N \theta_i^{\theta_i + \omega_i} \quad (2.11)$$

$$= \min_{\theta} -\sum_i^N (\theta_i + \omega_i) \log \theta_i \quad (2.12)$$

$$= \min_{\theta} -\sum_i^N (\theta_i \log \theta_i + \omega_i \log \theta_i - \omega_i \log \omega_i + \omega_i \log(\omega_i)) \quad (2.13)$$

$$= \min_{\theta} -\sum_i^N \theta_i \log \theta_i + \sum_i^N \omega_i \log \frac{\omega_i}{\theta_i} - \sum_i^N \omega_i \log \omega_i \quad (2.14)$$

$$= \min_{\theta} H(\theta) + D(\omega\|\theta) + H(\omega). \quad (2.15)$$

In minimizing this sum of entropies, the MAP estimator reduces uncertainty in all respects. Each term in this sum has a useful interpretation. The entropy $H(\theta)$ measures ambiguity within the model. The cross-entropy $D(\omega\|\theta)$ measures divergence between the parameters θ and the data's descriptive statistics ω ; it is the lower bound on the expected number of bits needed to code aspects of the data set not captured by the model, such as noise. In problems with hidden variables, the expected sufficient statistics ω are computed relative to the structure of the model; thus $H(\omega)$ is a lower bound on the expected number of bits needed to specify which of the variations allowed by the model is instantiated by the data. As $H(\theta)$ declines, the model becomes increasingly structured and near-deterministic. As $H(\omega)$ declines, the model comes to agree with the underlying structure of the data. Finally, as $D(\omega\|\theta)$ declines, the residual aspects of the data not captured by the model become less and less structured, approaching pure normally distributed noise.

Alternatively, we can understand equation 2.15 to show that the MAP estimator minimizes the lower bound of the expected coding lengths of the model and of the data relative to it. In this light, entropic EM is a searchless and highly efficient form of structure learning under a minimum coding length constraint.

2.3 Training. The entropic posterior defines a distribution over all possible model structures and parameterizations within a class; small, accurate models having minimal ambiguity in their joint distribution are the most

probable. To find these models, we replace the M-step of EM with the entropic MAP estimator, with the following effect. First, the E-step distributes probability mass unevenly through the model, because the model is not in perfect accordance with the intrinsic structure of the training data. In the MAP-step, the estimator exaggerates the dynamic range of multinomials in improbable parts of the model. This drives weakly supported parameters toward zero and concentrates evidence on surviving parameters, causing their estimates to approach the ML estimate. Structurally irrelevant parts of the model gradually expire, leaving a skeletal model whose surviving parameters become increasingly well supported and accurate.

2.4 Trimming. The MAP estimator increases the structure of a model by driving irrelevant parameters asymptotically to zero. Here we explore some conditions under which we can reify this behavior by altering the graphical structure of the model (i.e., removing dependencies between variables). The entropic prior licenses simple tests that identify opportunities to trim parameters *and* increase the posterior probability of the model. One may trim a parameter θ_i whenever the loss in the likelihood is balanced by a gain in the prior:

$$P_e(\boldsymbol{\theta} \setminus \theta_i | \mathbf{X}) \geq P(\boldsymbol{\theta} | \mathbf{X}) \quad (2.16)$$

$$P(\mathbf{X} | \boldsymbol{\theta} \setminus \theta_i) P_e(\boldsymbol{\theta} \setminus \theta_i) \geq P(\mathbf{X} | \boldsymbol{\theta}) P_e(\boldsymbol{\theta}) \quad (2.17)$$

$$\frac{P_e(\boldsymbol{\theta} \setminus \theta_i)}{P_e(\boldsymbol{\theta})} \geq \frac{P(\mathbf{X} | \boldsymbol{\theta})}{P(\mathbf{X} | \boldsymbol{\theta} \setminus \theta_i)} \quad (2.18)$$

$$\log P_e(\boldsymbol{\theta} \setminus \theta_i) - \log P_e(\boldsymbol{\theta}) \geq \log P(\mathbf{X} | \boldsymbol{\theta}) - \log P(\mathbf{X} | \boldsymbol{\theta} \setminus \theta_i) \quad (2.19)$$

$$H(\boldsymbol{\theta}) - H(\boldsymbol{\theta} \setminus \theta_i) \geq \log P(\mathbf{X} | \boldsymbol{\theta}) - \log P(\mathbf{X} | \boldsymbol{\theta} \setminus \theta_i). \quad (2.20)$$

If θ_i is small and positive, we can substitute the following differentials:

$$\theta_i \frac{\partial H(\boldsymbol{\theta})}{\partial \theta_i} \geq \theta_i \frac{\partial \log P(\mathbf{X} | \boldsymbol{\theta})}{\partial \theta_i}. \quad (2.21)$$

In sum, a parameter can be trimmed when it varies the entropy more than the log-likelihood. Any combination of the left and right terms in equations 2.20 and 2.21 will yield a trimming criterion. For example, we may substitute the entropic prior on multinomials into the left-hand side of equation 2.20 and set that against the right-hand side of equation 2.21, yielding,

$$h(\theta_i) \geq \theta_i \frac{\partial \log P(\mathbf{X} | \boldsymbol{\theta})}{\partial \theta_i}, \quad (2.22)$$

where $h(\theta_i) = -\theta_i \log \theta_i$. Dividing by $-\theta_i$ and exponentiating, we obtain

$$\theta_i \leq \exp \left[-\frac{\partial \log P(\mathbf{X} | \boldsymbol{\theta})}{\partial \theta_i} \right]. \quad (2.23)$$

Conveniently, the gradient of the log-likelihood $\partial \log P(\mathbf{X}|\boldsymbol{\theta})/\partial\theta_i$ will have already been calculated for reestimation in most learning algorithms.

Trimming accelerates training by removing parameters that would otherwise decay asymptotically to zero. Satisfying the trimming criterion is equivalent to discovering that specific values of two variables related by a parameter are incompatible; repeated trims may make the relationship between the two variables deterministic. Although the mathematics makes no recommendation when to trim, as a matter of practice we wait until the model is at or near convergence. Trimming then bumps the model out of the local probability maximum and into a parameter subspace of simpler geometry, thus enabling further training. Trimming near convergence also gives us confidence that further training would not resuscitate a nearly extinct parameter.

Note that if a model is to be used for life-long learning—periodic or gradual retraining on samples from a slowly evolving nonstationary process—then trimming is not advised, since nearly extinct parameters may be revived to model new structures that arise as the process evolves.

3 Mixture Models

Semiparametric distributions such as mixture or cluster models usually require iterative estimation of a single multinomial, the mixing parameters $\boldsymbol{\theta}$. In the E-step of EM, we calculate the expected sufficient statistic as usual:

$$\omega_i = \sum_n^N p(c_i|\mathbf{x}_n), \tag{3.1}$$

where $p(c_i|\mathbf{x}_n)$ is the probability of mixture component c_i given the n th data point. Dividing by N yields the MLE for the conventional M-step. For entropic estimation, we instead apply the entropic MAP estimator to $\boldsymbol{\omega}$ to obtain $\boldsymbol{\theta}$. The trimming criterion derives directly from equation 2.23:

$$\theta_i \leq \exp \left[-\frac{\partial \log P(\mathbf{X}|\boldsymbol{\theta})}{\partial\theta_i} \right] = \exp \left[-\sum_n^N \frac{p(\mathbf{x}_n|c_i)}{\sum_i^M p(\mathbf{x}_n|c_i)\theta_i} \right]. \tag{3.2}$$

The well-known annulus problem (Bishop, 1995, p. 68) affords a good opportunity to illustrate visually the qualitative difference between entropic and conventionally estimated models. We are given 900 random points sampled from an annular region and 30 gaussian components with which to form a mixture model. Figure 2 shows that entropic estimation is an effective procedure for discovering the essential structure of the data. All the components that might cause overfitting have been removed, and the surviving components provide good coverage of the data. The maximum likelihood model is captive of the accidental structure of the data (e.g., irregularities of the sampling). As is the case for all examples in the article,

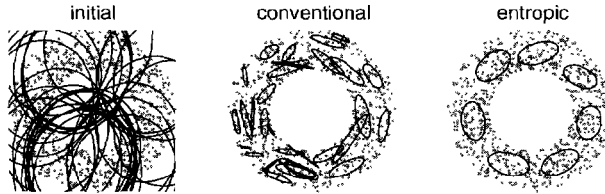


Figure 2: Mixture models estimated entropically (right) and conventionally (center) from identical initial conditions (left). Dots are data points sampled from the annular region; ellipses are isoproability contours of the gaussian mixture components.

entropic estimation took roughly half again as many iterations as conventional EM.

Like conventional EM, this method in theory can cause excess gaussian components to collapse on individual data points, leading to infinite likelihoods. This problem is ameliorated in the entropic framework because these components are typically trimmed before they collapse.

4 Continuous-Output HMMs

An HMM model is a dynamically evolving mixture model, where mixing probabilities in each time step are conditioned on those of the previous time step via a matrix of transition probabilities. In HMMs, the mixture components are known as states. The transition matrix is a stack of multinomials (e.g., the probability of state i given state j is the i th element of row j). For entropic estimation of HMM transition probabilities, we once again use a conventional E-step to obtain the probability mass for each transition:

$$\gamma_{j,i} = \sum_t \alpha_j(t) \theta_{ij} p(\mathbf{x}_{t+1} | s_i) \beta_i(t+1). \quad (4.1)$$

θ_{ij} is a transition probability from state j , $p(\mathbf{x}_{t+1} | s_i)$ is the probability of state i observing data point \mathbf{x}_{t+1} , and α, β are E-step statistics obtained from forward-backward analysis as per Rabiner (1989). For the MAP-step we calculate new estimates $\{\hat{P}_{ij}\}_i = \theta$ by applying the entropic MAP estimator to each $\omega = \{\gamma_{j,i}\}_i$. (For conventional Baum-Welch reestimation with a uniform prior, one simply sets $\hat{P}_{ij} = \gamma_{j,i} / \sum_i \gamma_{j,i}$.)

We compared entropically and conventionally estimated continuous-output HMMs on sign language gesture data provided by a computer vision lab (Starnier & Pentland, 1997). Experimental conditions for this and all subsequent tests are detailed in appendix B. Entropic estimation consistently yielded HMMs with simpler transition matrices having many parameters at

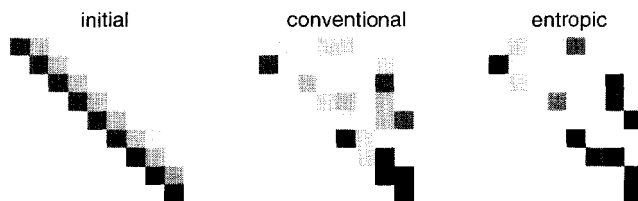


Figure 3: Initial, conventional, Baum-Welch, and entropically reestimated transition matrices. Each row depicts transition probabilities from a single state; white is zero. The first two matrices are fully upper-diagonal; the right-most is sparse.

or near zero (see Figure 3)—lower-entropy dynamical models. When tested on held-out sequences from the same source, entropically trained HMMs were found to overfit less in that they yielded higher log-likelihoods on held-out test data than conventionally trained HMMs. (Analysis of variance indicates that this result is significant at $p < 10^{-3}$; equivalently, this is the probability that the observed superiority of the entropic algorithm is due to chance factors.) This translated into improved classification: the entropically estimated HMMs also yielded superior generalization in a binary gesture classification task ($p < 10^{-2}$, measuring the statistical significance of the mean difference in correct classifications).

Most interesting, the dynamic range of surviving transition parameters was far greater than that obtained from conventional training. This remedies a common complaint about continuous-output HMMs: that model selectivity is determined mainly by model structure, then by output distributions, and finally by transition probabilities, because they have the smallest dynamic range (Bengio, 1997). (Historically some users have found structure so selective that parameter values can be ignored; (Sakoe & Chiba, 1978)).

4.1 Transition Trimming. To obtain a trimming criterion for HMM transition parameters, we substitute the E-step statistics into equation 2.23, yielding,

$$\theta_{ij} \leq \exp \left[-\frac{\partial \log P(\mathbf{X}|\theta)}{\partial \theta_{ij}} \right] \quad (4.2)$$

$$= \exp \left[-\frac{\sum_{t=1}^{T-1} \alpha_j(t) p(\mathbf{x}_{t+1}|s_j) \beta_i(t+1)}{\sum_k \alpha_k(T)} \right] \quad (4.3)$$

This test licenses a deletion when the transition is relatively improbable and the source state is seldom visited. Note that θ_{ij} must indeed be quite small,

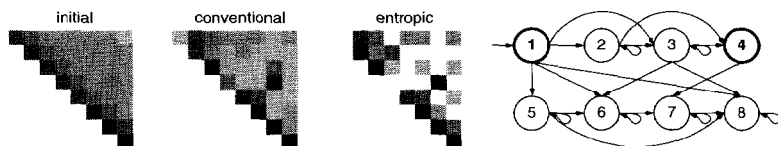


Figure 4: Entropic training reserves some states for purely transition logic. In the graphical view at right, gating state 1 forks to several subpaths; gating state 4 collects two of the branches and forwards to state 7.

since the gradient of the log-likelihood can be quite large. Fortunately, the MAP estimator brings many or most parameter values within trimming range.

Equation 4.3 is conservative. We may also consider the gain obtained from redistributing the trimmed probability to surviving parameters, in particular the parameter $\theta_{k|j}$ that maximizes $\partial P_e(\theta|x)/\partial\theta_{k|j}$. This leads to a more aggressive trimming test:

$$h(\theta_{i|j}) - \theta_{i|j} \frac{\partial H(\boldsymbol{\theta})}{\partial\theta_{k|j}} \geq \theta_{i|j} \left[\frac{\partial \log P(\mathbf{X}|\boldsymbol{\theta})}{\partial\theta_{i|j}} - \frac{\partial \log P(x|\theta)}{\partial\theta_{k|j}} \right] \quad (4.4)$$

$$\log \theta_{i|j} - 1 - \log \theta_{k|j} \leq - \left[\frac{\partial \log P(x|\theta)}{\partial\theta_{i|j}} - \frac{\partial \log P(x|\theta)}{\partial\theta_{k|j}} \right] \quad (4.5)$$

$$\theta_{i|j} \leq \theta_{k|j} \exp \left[1 + \frac{\partial \log P(x|\theta)}{\partial\theta_{k|j}} - \frac{\partial \log P(\mathbf{X}|\boldsymbol{\theta})}{\partial\theta_{i|j}} \right]. \quad (4.6)$$

The gesture data experiments were repeated with deletion using the trimming criterion of equation 4.3. We batch-deleted one exit transition per state between reestimations. There was a small but statistically significant improvement in generalization ($p < 0.02$), which is to be expected since deletions free the model from local maxima. The resulting models were simpler and faster, removing 81% of transitions on average for 15-state models, 42% from 10-state models, and 6% from 5-state models (thought to be the ideal state count for the gesture data set). Since complexity is linear in the number of transitions, this can produce a considerable speed-up.

In continuous-output HMMs, entropic training appears to produce two kinds of states: *data modeling*, having output distributions tuned to subsets of the data, and *gating*, having near-zero durations ($\theta_{ij} \approx 0$) and often having

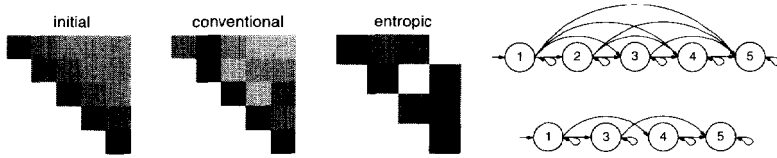


Figure 5: Even when beginning with a near-optimal number of states, entropic training will occasionally pinch off a state by deleting all incoming transitions. In this problem, state 2 was removed. Graphical views are shown at right.

highly nonselective output probabilities. Gating states appear to serve as branch points in the transition graph, bracketing alternative subpaths (see Figure 4). Their main virtue is that they compress the transition graph by summarizing common transition patterns.

One benefit of trimming is that sparsified HMMs are much more likely to encode long-term dependencies successfully. Dense transition matrices cause diffusion of credit, thus learning a long-term dependency gets exponentially harder with time (Bengio & Frasconi, 1995). Sparsity can dramatically reduce diffusion. Bengio and Frasconi suggested handcrafted sparse transition matrices or discrete optimization over the space of all sparse matrices as remedies. Entropic training with trimming essentially incorporates this discrete optimization into EM.

4.2 State Trimming. One of the more interesting properties of entropic training is that it tends to reduce the occupancy rate of states that do little to direct the flow of probability mass, whether by vice of broad output distributions or nonselective exit transitions. As a result, their incoming transitions become so attenuated that such states are virtually pinched off from the transition graph (see Figure 5). As with transitions, one may detect a trimmable state s_i by balancing the prior probability of all of its incoming and exit transitions against the probability mass that flows through it (see equation 2.18):

$$\frac{P(\mathbf{X}|\boldsymbol{\theta} \setminus s_i)}{P(\mathbf{X}|\boldsymbol{\theta})} \geq \theta_{i|i}^{\theta_{i|}} \prod_{j \neq i} \theta_{j|i}^{\theta_{j|}} \theta_{i|j}^{\theta_{i|}}. \tag{4.7}$$

Table 1: Average State Deletions and Conversions as a Function of Initial State Counts.

N	Deleted + Gated	Δ Log-Likelihood		Δ Error		Perplexity
5	0.08 + 0.24	-0.00003412,	$p > 2$	-0.02,	$p < 1$	2.74
10	0.76 + 1.45	0.1709,	$p < 0.03$	-1.02,	$p < 0.3$	2.90
15	1.36 + 2.79	0.2422,	$p < 0.008$	-1.85,	$p < 0.02$	2.93
20	1.87 + 3.91	1.249,	$p < 10^{-5}$	-2.39,	$p < 10^{-3}$	2.84

Note: Δ Log-likelihood is the mean advantage over conventionally trained models in recognizing held-out data, in nats/data point; p is the statistical significance of this mean. Δ error, measuring the mean difference in errors in binary classification, shows that the entropically estimated models were consistently better.

$P(\mathbf{X}|\theta \setminus s_i)$ can be computed in a modified forward analysis in which we set the output probabilities of one state to zero ($\forall_i p(x_t|s_i) \leftarrow 0$). However, this is speculative computation, which we wish to avoid. We propose a nonspeculative heuristic that we found equally effective: we bias transition trimming to zero self-transitions first. Continued entropic training then drives an affected state’s output probabilities to extremely small values, often dropping the state’s occupancy low enough to lead to its being pinched off. In experiments measuring the number of states removed and the resulting classification accuracy, we found no statistically significance difference between the two methods.

We ran the gesture data experiments again with the addition of state trimming. The average number of states deleted was quite small; the algorithm appears to prefer to keep superfluous states for use as gating states (see Table 1). Clearly the algorithm did not converge to an “ideal” state count, even discounting gating states. Given that the data record continuous-motion trajectories, it is not clear that there is any such ideal. Note however, that models of various initial sizes do appear to converge to a constant perplexity (in conventional HMMs perplexity is typically proportional to the state count). This strongly suggests that entropic training is finding a *dynamically* simplest model of the data rather than a *statically* simplest model.

4.3 Ambient Video. In Brand (1997) we used the full algorithm to learn a concise probabilistic automaton (HMM) modeling human activity in an office setting from a motion time-series extracted from a half-hour of video (see appendix B for details). We compared the generalization and discrimination of entropically trained HMMs, conventionally trained HMMs, and entropically trained HMMs with transition parameters subsequently flattened to chance. Four data sets were employed: train, test, test reversed, and altered behavior (the video subject had large amounts of coffee). Figure 6 shows that the entropically trained HMM did best in discriminating out-of-class sequences. The conventional HMM shows more overfitting of

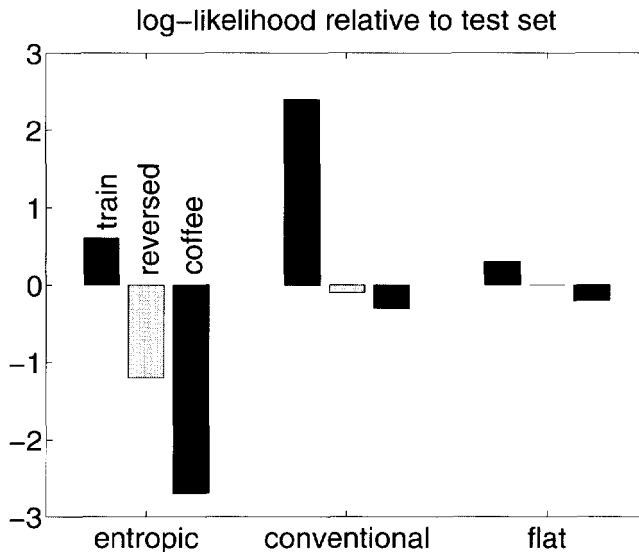


Figure 6: Log-likelihoods of three different classes of video, normalized to sequence length and compared to those of the test class.

the training set and little ability to distinguish the dynamics of the three test data sets. The flattened case shows that the classes do not differ substantially in the static distribution of points, only in their dynamics.

5 Discrete-Output HMMs

Discrete-output HMMs are composed entirely of cascaded multinomials. In the following experiments we entropically estimate both transition and output probabilities. In both cases we simply replace the M-step with the MAP estimator. We liberalize the state-pinning criterion by also considering the gain in the prior obtained by discarding the state's output parameters.

5.1 Bach Chorales. The "chorales" is a widely used dataset containing melodic lines from 100 of J. S. Bach's 371 surviving chorales. Modeling this data set with HMMs, we seek to find an underlying dynamics that accounts for the melodic structure of the genre. We expected this data set to be especially challenging because entropic estimation is predicated on noisy data, but the chorales are noiseless. In addition, the chorales are sampled from a nonstationary process: Bach was highly inventive and open to influences; his composing style evolved considerably even in his early years at Leipzig (Breig, 1986).

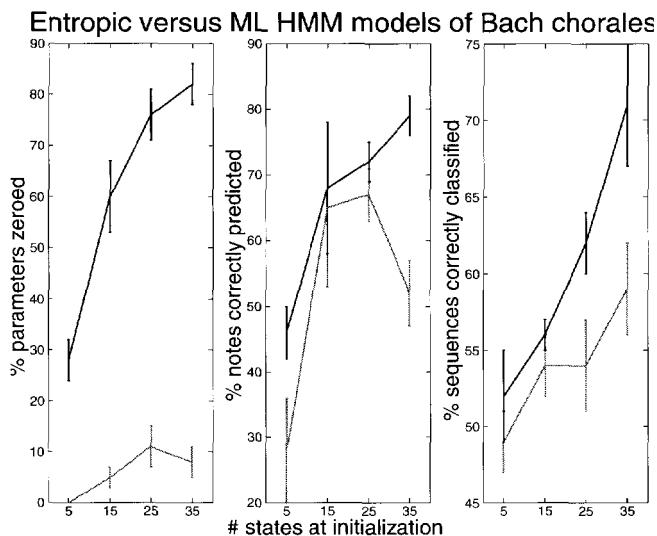


Figure 7: Entropic modeling of the Bach chorales. Lines indicate mean performance over 10 trials; error bars are 2 standard deviations.

We compared entropically and conventionally estimated HMMs in prediction and classification tasks, using a variety of different initial state counts. Figure 7 illustrates the resulting influences. Despite substantial loss of parameters to sparsification, the entropically estimated HMMs were, on average, better predictors of notes in the test set than the conventionally estimated HMMs. They also were better at discriminating between test chorales and temporally reversed test chorales—challenging because Bach famously employed melodic reversal as a compositional device. On the other hand, the entropically estimated HMMs also showed greater divergence between the per-note likelihoods of training and test sequences. This raises the possibility that the estimator does pay a price for assuming noise where there is none. Another possibility is that the entropically estimated models are indeed capturing more of the dynamical structure of the training melodies, and therefore are able to make deeper distinctions among melodies in different styles. This accords with our observation that six chorales in particular had low likelihoods when rotated into the test set.³

Perhaps the most interesting difference is while the conventionally estimated HMMs were wholly uninterpretable, one can discern in the entropically estimated HMMs several basic musical structures (see Figure 8), in-

³ Unfortunately, the data set is unlabeled, and we cannot relate this observation to the musicology of Bach's 371 chorales.

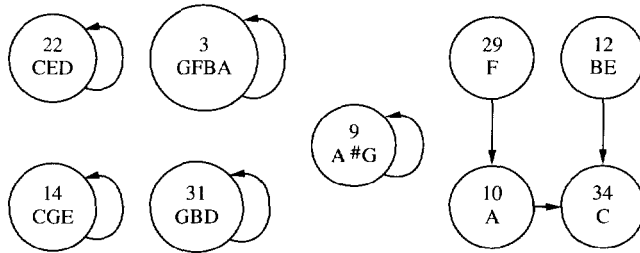


Figure 8: High-probability states and subgraphs of interest from a 35-state chorale HMM, with tones output by each state listed in order of probability. Extraneous arcs are removed for clarity.

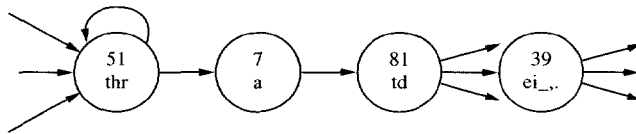


Figure 9: A nearly deterministic subgraph from a text-modeling HMM. Nodes show state number and symbols output by that state, in order of probability.

cluding self-transitioning states that output only tonic (C-E-G) or dominant (G-B-D) triads, lower- or upper-register diatonic tones (C-D-E or F-G-A-B), and trills and mordents (A-♯G-A). Dynamically, we found states that lead to the tonic (C) via the mediant (E) or the leading tone (B), as well as chordal state sequences (F-A-C). Generally these patterns were easier to discern in larger, sparser HMMs. We explore this theme briefly in the modeling of text.

5.2 Text. Human signals such as music and language have enormous amounts of hidden state. Yet interesting patterns can be discovered by entropic training of HMMs having modest numbers of states. For example, we entropically and conventionally trained 100-state, 30-symbol discrete-output HMMs on the abstract and introduction of the original version of this article. Entropic training pinched off 4 states and trimmed 94% of the transition parameters and 91% of the output parameters, leaving states that output an average of 2.72 symbols. Some states within the HMM formed near-deterministic chains; for example, Figure 9 shows a subgraph that can output the word fragments *rate*, *that*, *rotation*, and *tradition*, among others. When used to predict the next character given random text fragments taken from the body of the article, the entropic HMM scored 27% correct, and the conventional HMM scored 12%. The subgraph in Figure 9 probably accounts for the entropic HMM’s correct prediction given the word fragment

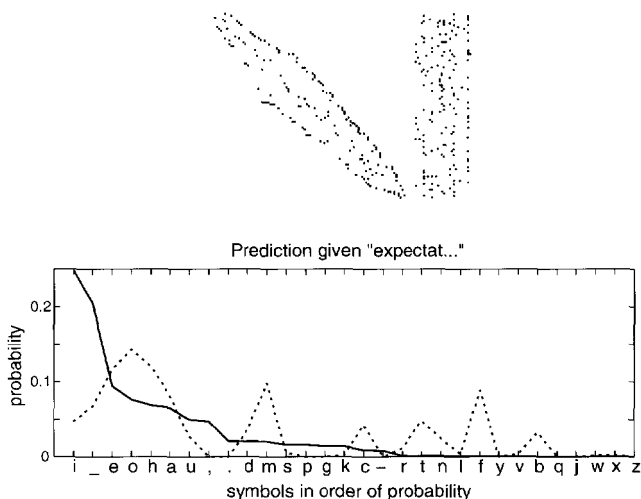


Figure 10: (Top) Nonzero entries of the transition and output matrices for a 96-state text model. (Bottom) Prediction of entropic and conventional HMMs for the letter following *expectat* (- = white space).

expectat. Figure 10 shows that the entropic model correctly predicts *i* and a range of less likely but plausible continuations. The conventionally trained model makes less specific predictions and errs in favor of typical first-order effects, for example, *h* often follows *t*. In predicting *i* over whitespace, *h*, and *e*, the entropic model is using context going back at least three symbols, since *expectation*, *demonstrate*, *motivated*, *automaton*, and *patterns* all occurred in the training sequence.

Entropic estimation of undersized models seeks hidden states that optimally compress the context; therefore we should expect to see some interesting categories in the finished model. Using the same data and a five-state initialization, we obtained the model shown in Figure 11. The hidden states in this HMM are highly correlated with phonotactic categories—regularities of *spoken* language that give rise, indirectly, to the patterns of written language:

1. Consonants that begin words and consonant clusters (e.g., *str*)
2. Vowels and semivowels
3. White space and punctuation (interword symbols)
4. Common word endings (e.g., the plural *s*)
5. Consonants in all other contexts

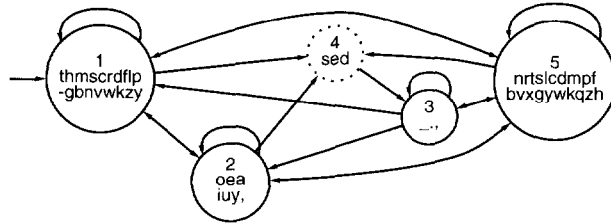


Figure 11: Graphical model of a five-state HMM trained on this text.

We identified these categories by using forward-backward analysis to assign most probable states to each character in a text—for example:

The cross-entropy statistics are
 1 1 4 3 1 5 2 5 5 1 2 5 1 5 2 5 2 3 1 5 2 5 2 1 5 2 5 4 3 2 5 4

We stress that our interpretation is correlative; the true genesis of the states is probably as follows: Having discovered the statistically most salient categories (vowels versus consonants versus interword symbols), entropic estimation went on to identify phenomena that reliably happen at the boundaries between these categories (word endings and consonant cluster beginnings).

6 Related Work

Extensive literature searches suggest that the entropic prior, MAP estimator, and trimming criteria are novel. However, the prior does have antecedents in the maximum entropy literature, which we turn to now.

6.1 Maximum Entropy and Geometric Priors. Maximum entropy (ME) refers to the set of methods for constructing probability distributions from prior knowledge without introducing unintended assumptions (Jaynes, 1996). These “ignorance-preserving” distributions have maximum entropy with regard to unknowns and will avoid modeling patterns that have inadequate evidentiary support. Classic ME deals with assertions about the expectations of a random variable rather than about samples from it. Probabilistic modelers typically deal only with samples. For this, one uses Bayesian ME, in which ignorance-preserving considerations lead to the construction of a prior. Although there is no unique ME prior, in the ME community the phrase *entropic prior* has come to connote the form (Skilling, 1989; Rodriguez, 1991, 1996):

$$P_{ME}(d\theta|\alpha, \theta_0) \propto e^{-\alpha D(\theta||\theta_0)} \sqrt{|J(\theta)|} d\theta, \tag{6.1}$$

where $D(\cdot)$ is the cross-entropy between the current parameter set and a reference model θ_0 , α is a positive constant indicating confidence in θ_0 , and $J(\theta)$ is the Fisher information matrix of the model parameterized by θ .

The exponential term is not applicable in our setting, as we typically have no reference model. The second term, $\sqrt{|J(\theta)|}$, is Jeffreys's noninformative prior (Jeffreys, 1961). It is typically motivated from differential geometry as a uniform prior in the space of distributions, and therefore invariant to changes of parameterization. It has an interesting relation to our minimum entropy prior $e^{-H(\theta)}$: Given a distribution specified by θ , Jeffreys's prior divides the posterior by the volume of parameterizations that would yield equivalent distributions (given infinite data) (Balasubramanian, 1997); the ME prior divides the posterior by the volume of the distribution's typical set (small typical sets have few equivalent parameterizations). Both priors measure specificity; the Jeffreys prior is actually a stronger bias. In some cases Jeffreys felt that $\sqrt{|J(\theta)|}$ was problematic, and he recommended other noninformative priors such as $1/\sigma$ for $\mathcal{N}(\mu, \sigma^2)$ one-dimensional gaussian variance estimation (Jeffreys, 1961); this can be derived from the general form of the entropic prior $e^{-H(\theta)}$. In Brand (1999b) we show that our prior can be derived directly from a classical maximum entropy treatment of the assertion, "The expected unpredictability of the process being modeled is finite."

6.2 HMM Induction. The literature of HMM structure induction is almost wholly based on generate-and-test searches over the space of discrete-output HMMs, using state splitting or merging to perturb the model followed by parameter estimation to test for improvement. For example, Vasko, El-Jaroudi, and Boston (1996) proposed a heuristic scheme in which a set of randomly pruned HMMs is compared, looking for a model that combines a small loss of likelihood and a large number of prunings. Stolcke and Omohundro (1994) begin with the disjunction of all possible samples (a maximally overfit model) and iteratively merged states using a Dirichlet prior and Bayesian posterior probability criterion to test for success, failure, and completion. Takami and Sagayama (1991, 1994) took an opposite approach, beginning with a single state and heuristically splitting states and adding transitions. Ikeda (1993) presented a similar scheme with an objective function built around Aikake's Information Criterion to limit overfitting. The speech recognition literature now contains numerous variants of this strategy, including maximum likelihood criteria for splitting (Ostendorf & Singer, 1997), search by genetic algorithms (Yada, Ishikawa, Tanaka, & Asai, 1996; Takara, Higa, & Nagayama, 1997), and splitting to describe exceptions (Fujiwara, Asogawa, & Konagaya, 1995; Valtchev, Odell, Woodland, & Young, 1997). Nearly all of these algorithms use beam search (generate-and-test with multiple heads) to compensate for dead-ends and

declines in posterior probability; most of the computation is squandered. Reported run times are typically in hours or days, and discrete-output HMMs are computational lightweights compared to continuous-output HMMs. In contrast, our hill-climbing algorithm applies to any kind of state-space Markov model and takes only slightly longer than classic EM; the examples in this article required only a few seconds of CPU time.

Other proposals include two-stage methods in which data are statically clustered to yield a state-space and transition topology (Falaschi & Pucci, 1991; Wolfertstetter & Ruske, 1995). The second stage is conventional training. Minimum description length (MDL) methods can be applied to prevent overfitting in the first stage. However, it is fairly easy to construct problems that will thwart two-stage methods, such as uniformly distributed samples that have structure only by virtue of their patterns through time.

Entropic estimation is similar in spirit to neural network pruning schemes, particularly weight elimination, in which a heuristic regularization term in the objective function causes small weights to decay toward zero (Hanson & Pratt, 1989; Lang & Hinton, 1990). In fact, weights decay to *near* zero (Bishop, 1995); it is then necessary to add a pruning step at a cost of some increase in error (LeCun, Denker, & Solla, 1990), although the damage can be minimized by small adjustments to the surviving weights (Hassibi & Stork, 1993). All of these schemes require one or more hand-chosen regularization parameters. In Brand (1999a) we propose entropic training and trimming rules for nonlinear dynamical systems, including recurrent neural networks.

Outside of probabilistic modeling, there is a small but growing combinatorial optimization literature that embeds discrete problems in continuous functions having hidden indicator variables. Gradient descent on a combined entropy and error function forces the system to explore broadly the search space and then settle into a syntactically valid and near-optimal state. Stolorz (1992) gave traveling salesman problems (TSP) this treatment; in Brand (1999b) we show that TSP can be reformulated as an iterative MAP problem.

7 Limitations and Open Questions

We believe entropic estimation is best used to sculpt a well-fit model out of an overfit model. Given an underfit or a structurally "correct" model, we have no reason to believe that entropically estimated parameters, being biased, are superior to maximum likelihood parameters, except perhaps as a correction to noise. Indeed, it might be advisable to polish an entropically estimated model with a few cycles of maximum likelihood reestimation.

Our framework is currently agnostic and thus vulnerable to ambiguities with regard to choice of entropy measure. For example, with sequence data, one may choose the entropy or the entropy rate (entropy per symbol). The

case of continuous distributions is complicated by the fact that differential entropy $(-\int P(x) \log P(x)dx)$ has some pathologies that can lead to absurdities such as infinitely negative entropy. Finally, for many kinds of complex models, there is no analytically tractable form for the entropy $H(\theta)$. In cases such as these, we decompose the model into simpler distributions whose entropies are known. By the subadditivity principle, the sum of these entropies will upper-bound the true entropy; hence, the MAP estimator will always reduce entropies. In this scheme the sum of entropies in equation 2.15 has a clear interpretation as a description length. In sections 4 and 5 we upper-bounded the entropy rate of the HMM in this manner. Alternatively, we could use conditional entropies in the prior—in the case of Markov models conditional entropy and entropy rate are asymptotically equal:

$$P_{er}(\theta) \propto \exp \sum_j p_j \sum_i \theta_{ij} \log \theta_{ij} = \prod_j \left(\prod_i \theta_{ij}^{\theta_{ij}} \right)^{p_j} \tag{7.1}$$

In the context of HMMs, p_j is the stationary probability of state j , which can be estimated from the data. The MAP estimate can easily be obtained by scaling $\omega_j = \{\omega_{1j}, \omega_{2j}, \omega_{3j}, \dots\}$ and λ by $1/p_j$ in equations 2.10 and 2.5.

Much work remains to be done on the mathematical characterization of the entropic prior, including a closed-form solution for the Lagrangian term λ and normalization terms for multivariate priors:

$$\int_0^1 \theta_1^{\theta_1} \left(\int_0^{1-\theta_1} \theta_2^{\theta_2} \left(\dots \int_0^{1-\sum_{i=1}^{k-1} \theta_i} \theta_k^{\theta_k} \left(1 - \sum_i^k \theta_i \right)^{(1-\sum_i^k \theta_i)} d\theta_k \dots \right) d\theta_2 \right) d\theta_1 \tag{7.2}$$

A normalized posterior will also be necessary for comparing different models of the same data.

Now we turn to questions about the prior that open connections to related fields.

7.1 Graph Theory. Readers of combinatorics will recognize in equation 2.10 the tree function $T(x) = -W_{-1}(-x)$, used in the enumeration of trees and graphs on sets of labeled vertices (Wright, 1977; Janson, Knuth, Luczak, & Pittel, 1993) and in computing the distribution of cycles in random mappings (Flajolet & Soria, 1990). Connections to dynamical stability via the W function and to sparse graph enumeration via the T function are very intriguing and may lead to arguments as to whether the entropic prior is optimal for learning concise sparse models.

We offer a tantalizing clue, reworking and solving a partial result from midcentury work on the connectivity of neural tissue (Solomonoff & Rapoport, 1951) and random graphs (Erdős & Rényi, 1960). If n people (vertices)

have an average of a acquaintances (edges) apiece and one individual starts a rumor, the probability that a randomly selected individual has not heard this rumor (is not connected) is approximately $p = e^{-a(1-p)}$ (Landau, 1952). Solving for p via W , we can now obtain $p = [-a/W(-ae^{-a})]^{-1}$. Note that the bracketed term is essentially the MAP estimator of equation 2.10 with the Lagrange multiplier set to $\lambda = -\omega - 1$. We may thus understand the MAP estimator as striking a compromise between extreme graph sparsity and minimally adequate graph reachability; the compromise is governed by the statistics of the training set. Since a is essentially the perplexity of the rumor-mongering population, we see here the glimmerings of a formal connection among the entropic MAP estimate, the connectedness of the transition graph, and the perplexity of the data.

7.2 Optimization. Entropic estimation pushes the model toward one of the corners of the infinite-dimensional hypercube containing the parameter space. Typically many of the local optima in different corners will be identical, modulo permutations of the parameter matrix and hidden variables. This is why we must work with the MAP estimator and not the posterior mean, a meaningless point in the center of the hypercube. We seek the region of the posterior having the greatest probability mass, yet the posterior is multimodal and very spiky. (This is a hallmark of discrete optimization problems.) Unfortunately, initial conditions determine the particular optimum found by EM (indeed, by any finite-resource optimizer). One approach is to improve the quality of the local optimum found by a single trial of EM; in particular we have found a simple generalization of the entropic MAP estimator that automatically folds deterministic annealing into EM (Brand, 1998b). An open question of some relevance here is whether EM can be generalized to yield successively better optima given additional compute time.

Finally, the general prior $e^{-H(\theta)}$ has a specific form for virtually any probability density function; it remains to solve for the MAP estimators and trimming criteria. In a forthcoming article, we extend the entropic structure-discovery framework with similar results for a variety of other parameter types and demonstrate applications to other models of interest including generalized recurrent neural networks and radial basis function networks (Brand, 1999a).

8 Conclusion

We have presented a mathematical framework for simultaneously estimating parameters and simplifying model structure in probabilistic models containing hidden variables and multinomial parameters, such as hidden Markov models. The key is an entropic prior that prefers low entropy estimates to fair estimates when evidence is limited, on the premise that small data sets are less representative of generating process and more profoundly

contaminated by noise and sampling artifacts. Our main result is a solution for the MAP estimator, which drives weakly supported parameters toward extinction, effectively turning off excess parameters. We augment the extinction process with explicit tests and transforms for parameter deletion; these sparsify the model, accelerate learning, and rescue EM from local probability maxima. In HMMs, entropic estimation gradually zeroes superfluous transitions and pinches off nonselective states, sparsifying the model. Sparsity provides protection against overfitting; experimentally, this translates into superior generalization in prediction and classification tasks. In addition, entropic estimation converts some data-modeling states into gating states, which effectively have no output distributions and serve only to compress the transition graph. Perhaps most interesting, the structure discovered by entropic estimation can often shed light on the hidden process that generated the data.

Entropic estimation monotonically and maximally hill-climbs in posterior probability; there is no wasted computation as in backtracking or beam search. Consequently we are able to “train and trim” HMMs and related models in times comparable to conventional EM, yet produce simpler, faster, and better models.

Appendix A: Computing W

W is multivalued, having an infinite number of complex branches and two partly real branches, W_0 and W_{-1} . $W_{-1}(-e^x)$ is real on $x \in (-\infty, -1]$ and contains the solution of equation 2.5. All branches of the W function can be computed quickly using Halley’s method, a third-order generalization of Newton’s method for finding roots. The recurrence equations are

$$\delta_j = w_j e^{w_j} - z \tag{A.1}$$

$$w_{j+1} = w_j - \frac{\delta_j}{e^{w_j}(w_j + 1) - \frac{\delta_j(w_j+2)}{2(w_j+1)}}. \tag{A.2}$$

See Corless et al. (1996) for details on selecting an initial value w_0 that leads to the desired branch.

We found it is sometimes necessary to compute $W(z)$ for $z = -e^{-x}$ that are well outside the range of values of digital floating-point representations. For such cases we observe that $W(e^x) + \log W(e^x) = x$, which licenses the swiftly converging recurrence for $W_{-1}(-e^{-x})$:

$$w_{j+1} = -x - \log |w_j| \tag{A.3}$$

$$w_0 = -x \tag{A.4}$$

Appendix B: Experimental Details

B.1 Gesture Data. One hundred trials with a database of sign language gestures were obtained from computer vision. One-third of the sequences for a particular gesture were taken randomly for training in each trial. The remaining two-thirds were used for testing. Identical initial conditions were provided to the entropic and conventional training algorithms. Transition matrices were initialized with $\theta_{ij} = \{2^{j-1}/(2^j - 1)$ if $j \geq i$ else 0), which imposes a forward topology with skip-ahead probabilities that decline exponentially with the size of the jump. This topology was mainly for ease of analysis; our results were generally more pronounced when using full transition matrices. To make sure results were not an artifact of the data set, we checked for similar outcomes in a duplicate set of experiments with synthetic data $y_t = \{\sin((t+k_1)/100), \sin((t+k_1)/(133-k_2))\}$, k_1, k_2 random, corrupted with gaussian noise ($\sigma = \frac{1}{2}$).

B.2 Office Activity Data. Roughly a half-hour of video was taken at five frames per second randomly over the course of three days. Adaptive statistical models of background pixel variation and foreground motion were used to identify a foreground figure in each frame. The largest set of connected pixels in this foreground figure was modeled with a single two-dimensional gaussian. The isoprobability contour of this gaussian is an ellipse; we recorded the five parameters necessary to describe the ellipse of each frame, plus their derivatives, as the time series for each video episode. Roughly two-thirds of the time series were used for training. Training was initialized with fully dense matrices of random parameters.

B.3 Bach Chorales. The data set was obtained from the UCI machine-learning repository at the University of California, Irvine (Merz & Murphy, 1998). Each sequence tracks the pitch, duration, key, and time signature of one melody. We combined the pitch and key information to obtain a 12-symbol time series representing pitch relative to the tonic. We compared entropically and conventionally estimated HMMs by training with 90 of the chorales and testing with remaining 10. In 10 trials, all chorales were rotated into the test set. Prior to experimentation, the full data set was randomly reordered to minimize nonstationarity due to changes in Bach's composing style. HMMs were estimated entropically and conventionally from identical initial conditions, with fully dense random transition and output matrices. For the note prediction task, each test sequence was truncated to a random length, and the HMMs were used to predict the first missing note.

B.4 Text. The first 2000 readable characters of this article (as originally submitted) were used for training. The original character set was condensed to 30 symbols: 26 letters, a white space symbol, and three classes of punctuation. HMMs were estimated entropically and conventionally from identical

initial conditions, with fully dense random transition and output matrices. After training, the prior probabilities of hidden states were set to their average occupancy probabilities, so that the HMMs could be tested on any text sequence that did not start with the first symbol of the training set. For the prediction task, 100 test sequences of 20 symbols each were taken randomly from the body of the text.

Acknowledgments

Robert Corless provided several useful series for calculating the W function, as well as an excellent review of its applications in other fields (Corless et al., 1996). Thad Starner provided the computer vision sign language data, also used in Starner and Pentland (1997). Many thanks to local and anonymous reviewers for pointing out numerous avenues of improvement.

References

- Balasubramanian, V. (1997). Statistical inference, Occam's razor and statistical mechanics on the space of probability distributions. *Neural Computation*, 9(2), 349–368.
- Bauer, E., Koller, D., & Singer, Y. (1997). Update rules for parameter estimation in Bayesian networks. In *Proc. Uncertainty in Artificial Intelligence* (Providence, RI).
- Bengio, Y. (1997). *Markovian models for sequential data* (Technical Rep.). Montreal: University of Montreal.
- Bengio, Y., & Frasconi, P. (1995). Diffusion of credit in Markovian models. In G. Tesauro, D. S. Touretzky, & T. Leen (Eds.), *Advances in neural information processing systems*, 7 (pp. 553–560). Cambridge, MA: MIT Press.
- Bishop, C. (1995). *Neural networks for pattern recognition*. New York: Oxford University Press.
- Brand, M. (1997). *Learning concise models of human activity from ambient video* (Tech. Rep. No. 97-25). Cambridge, MA: Mitsubishi Electric Research Labs.
- Brand, M. (1999a). *Entropy estimation blends continuous and discrete optimization* (Technical Rep.). Cambridge, MA: Mitsubishi Electric Research Labs.
- Brand, M. (1999b). Pattern discovery via entropy optimization. In D. Heckerman and J. Whittaker (Eds.) *Proceedings of the 7th International Conference on Artificial Intelligence and Statistics*. San Francisco, CA: Morgan Kaufmann.
- Breig, W. (1986). J. S. Bach as organist: His instruments, music and performance practices. In G. Stauffer & E. May (Eds.), *The "Great Eighteen" Chorales: Bach's Revisional Process and the Genesis of the Work* (pp. 102–120). Bloomington, IN: Indiana University Press.
- Corless, R. M., Gonnet, G. H., Hare, D. E. G., Jeffrey, D. J., & Knuth, D. E. (1996). On the Lambert W function. *Advances in Computational Mathematics*, 5, 329–359.
- Dietterich, T. G. (1998). Approximate statistical tests for comparing supervised classification learning algorithms. *Neural Computation*, 10(7), 1895–1923.

- Erdős, P., & Rényi, A. (1960). On the evolution of random graphs. *MTA Mat. Kut. Int. Közl.*, 5, 17–61.
- Falaschi, A., & Pucci, M. (1991). Automatic derivation of HMM alternative pronunciation network topologies. In *Proc., 2nd European Conference on Speech Communication and Technology* (Vol. 2, pp. 671–674).
- Flajolet, P., & Soria, M. (1990). Gaussian limiting distributions for the number of components in combinatorial structures. *Journal of Combinatorial Theory, Series A*, 53, 165–182.
- Fujiwara, Y., Asogawa, M., & Konagaya, A. (1995). Motif extraction using an improved iterative duplication method for HMM topology learning. In *Pacific Symposium on Biocomputing '96* (pp. 713–714).
- Hanson, S. J., & Pratt, L. Y. (1989). Comparing biases for minimal network construction with back-propagation. In D. S. Touretzky (Ed.), *Advances in neural information processing systems*, 1 (pp. 177–195). San Mateo, CA: Morgan Kaufman.
- Hassibi, B., & Stork, D. (1993). Second order derivatives for network pruning: Optimal Brain Surgeon. In S. Hanson, J. Cowan, & C. Giles (Eds.), *Advances in neural information processing systems*, 5 (pp. 177–185). Cambridge, MA: MIT Press.
- Heckerman, D. (1996). *A tutorial on learning with Bayesian networks* (Tech. Rep. No. MSR-TR-95-06). Seattle: Microsoft Research. Available online at: <ftp://ftp.research.microsoft.com/pub/tr/TR-95-06a.html>.
- Ikeda, S. (1993). Construction of phoneme models—Model search of hidden Markov models. In *International Workshop on Intelligent Signal Processing and Communication Systems*. Sendai.
- Janson, S., Knuth, D. E., Luczak, T., & Pittel, B. (1993). The birth of the giant component. *Random Structures and Algorithms*, 4, 233–358.
- Jaynes, E. T. (1996). Probability theory: The logic of science. Fragmentary edition of March 1996. Available online via: <ftp://bayes.wustl.edu/pub/Jaynes/book.probability.theory/>.
- Jeffreys, H. (1961). *Theory of probability*. Oxford: Oxford University Press.
- Landau, H. G. (1952). On some problems of random nets. *Bulletin of Mathematical Biophysics*, 14, 203–212.
- Lang, K., & Hinton, G. (1990). Dimensionality reduction and prior knowledge in E-set recognition. In D. Touretzky (Ed.), *Advances in neural information processing*, 2 (pp. 178–185). San Mateo, CA: Morgan Kaufman.
- Laplace, P. S. (1812). *Theorie analytique des probabilités*. Paris: Courcier.
- LeCun, Y., Denker, J., & Solla, S. (1990). Optimal Brain Damage. In D. Touretzky (Ed.), *Advances in neural information processing*, 2 (pp. 178–185). San Mateo, CA: Morgan Kaufman.
- Merz, C. J., & Murphy, P. M. (1998). UCI repository of machine learning databases. University of California, Irvine: Dept. of Information and Computer Sciences.
- Ostendorf, M., & Singer, H. (1997). HMM topology design using maximum likelihood successive state splitting. *Computer Speech and Language*, 11(1), 17–41.
- Rabiner, L. R. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2), 257–286.

- Rodriguez, C. C. (1991). *Entropic priors* (Tech. Rep.). Albany: State University of New York at Albany, Department of Mathematics and Statistics.
- Rodriguez, C. C. (1996). Bayesian robustness: A new look from geometry. In G. Heidbreder (Ed.), *Maximum entropy and Bayesian methods*. Norwell, MA: Kluwer.
- Sakoe, H., & Chiba, C. (1978). Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing, ASSP-26*, 43–49.
- Skilling, J. (1989). Classical MaxEnt data analysis. In J. Skilling (Ed.), *Maximum entropy and Bayesian methods*. Norwell, MA: Kluwer.
- Solomonoff, R., & Rapoport, A. (1951). Connectivity of random nets. *Bulletin of Mathematical Biophysics*, 13, 107–117.
- Starner, T., & Pentland, A. P. (1997). A wearable-computer based American sign language recognizer. In *International Symposium on Wearable Computing* (Vol. 1). New York: IEEE Press.
- Stolcke, A., & Omohundro, S. (1994). *Best-first model merging for hidden Markov model induction* (Tech. Rep. No. TR-94-003). Berkeley: International Computer Science Institute.
- Stolorz, P. (1992). *Recasting deterministic annealing as constrained optimization* (Tech. Rep. No. 92-04-019). Santa Fe: Santa Fe Institute.
- Takami, J.-I., & Sagayama, S. (1991). *Automatic generation of the hidden Markov model by successive state splitting on the contextual domain and the temporal domain* (Tech. Rep. No. SP91-88). Tokyo: IEICE.
- Takami, J., & Sagayama, S. (1994). Automatic generation of hidden Markov networks by a successive state splitting algorithm. *Systems and Computers in Japan*, 25(12), 42–53.
- Takara, T., Higa, K., & Nagayama, I. (1997). Isolated word recognition using the HMM structure selected by the genetic algorithm. In *IEEE International Conference on Acoustics, Speech, and Signal Processing* (Vol. 2, pp. 967–970).
- Valtchev, V., Odell, J., Woodland, P., & Young, S. (1997). MMIE training of large vocabulary recognition systems. *Speech Communication*, 22(4), 303–314.
- Vasko, Jr., R., El-Jaroudi, A., & Boston, J. (1996). An algorithm to determine hidden Markov model topology. In *IEEE International Conference on Acoustics, Speech, and Signal Processing Conference* (Vol. 6, pp. 3577–3580).
- Wolfertstetter, F., & Ruske, G. (1995). Structured Markov models for speech recognition. In *International Conference on Acoustics, Speech, and Signal Processing* (Vol. 1, pp. 544–547).
- Wright, E. M. (1977). The number of sparsely connected edged graphs. *Journal of Graph Theory*, 1, 317–330.
- Yada, T., Ishikawa, M., Tanaka, H., & Asai, K. (1996). Signal pattern extraction from DNA sequences using hidden Markov model and genetic algorithm. *Transactions of the Information Processing Society of Japan*, 37(6), 1117–1129.

Received December 12, 1997; accepted August 24, 1998.