Minimum Probability Flow Learning

Jascha Sohl-Dickstein[ab*] JASCHA@BERKELEY.EDU
Peter Battaglino[ac*] PBB@BERKELEY.EDU
Michael R. DeWeese[acd] DEWEESE@BERKELEY.EDU
[a]Redwood Center for Theoretical Neuroscience, [b] Biophysics Graduate Group, [c] Physics Department, [d] Helen Wills Neuroscience Institute, University of California, Berkeley, 94720 [*]*These authors contributed equally.*

Abstract

Fitting probabilistic models to data is often difficult, due to the general intractability of the partition function and its derivatives. Here we propose a new parameter estimation technique that does not require computing an intractable normalization factor or sampling from the equilibrium distribution of the model. This is achieved by establishing dynamics that would transform the observed data distribution into the model distribution, and then setting as the objective the minimization of the KL divergence between the data distribution and the distribution produced by running the dynamics for an infinitesimal time. Score matching, minimum velocity learning, and certain forms of contrastive divergence are shown to be special cases of this learning technique. We demonstrate parameter estimation in Ising models, deep belief networks and an independent component analysis model of natural scenes. In the Ising model case, current state of the art techniques are outperformed by at least an order of magnitude in learning time, with lower error in recovered coupling parameters.

1. Introduction

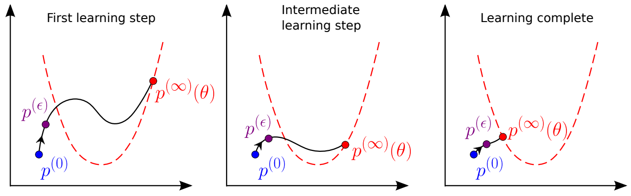
Estimating parameters for probabilistic models is a fundamental problem in many scientific and engineering disciplines. Unfortunately, most probabilistic learning techniques require calculating the normalization factor, or partition function, of the probabilistic model in question, or at least calculating its gradient. For the overwhelming majority of models there are no known analytic solutions. Thus, development of powerful new techniques for parameter estimation promises to greatly expand the variety of models that can be fit to complex data sets.

Many approaches exist for approximate learning, including mean field theory and its expansions, variational Bayes techniques and a variety of sampling or numerical integration based methods (Tanaka, 1998; Kappen & Rodríguez, 1997; Jaakkola & Jordan, 1997; Haykin, 2008). Of particular interest are contrastive divergence (CD), developed by Hinton, Welling and Carreira-Perpiñán (Welling & Hinton, 2002; Carreira-Perpiñán & Hinton, 2004), Hyvärinen's score matching (SM) (Hyvärinen, 2005), Besag's pseudolikelihood (PL) (Besag, 1975), and the minimum velocity learning framework proposed by Movellan (Movellan, 2008a;b; Movellan & McClelland, 1993).

Contrastive divergence (Welling & Hinton, 2002; Carreira-Perpiñán & Hinton, 2004) is a variation on steepest gradient descent of the maximum (log) likelihood (ML) objective function. Rather than integrating over the full model distribution, CD approximates the partition function term in the gradient by averaging over the distribution obtained after taking a few, or only one, Markov chain Monte Carlo (MCMC) steps away from the data distribution (Equation 17). Qualitatively, one can imagine that the data distribution is contrasted against a distribution that has evolved only a small distance towards the model distribution, whereas it would be contrasted against the true model distribution in traditional MCMC approaches. Although CD is not guaranteed to converge to the right answer, or even to a fixed point, it has proven to be an effective and fast heuristic for parameter estimation (MacKay, 2001; Yuille, 2005).

Score matching (Hyvärinen, 2005) is a method that learns parameters in a probabilistic model using only derivatives of the energy function evaluated over the data distribution (see Equation (19)). This sidesteps

Appearing in *Proceedings of the 28th International Conference on Machine Learning*, Bellevue, WA, USA, 2011.



Progression of Learning

Figure 1. An illustration of parameter estimation using minimum probability flow (MPF). In each panel, the axes represent the space of all probability distributions. The three successive panels illustrate the sequence of parameter updates that occur during learning. The dashed red curves indicate the family of model distributions $\mathbf{p}^{(\infty)}(\theta)$ parametrized by θ. The black curves indicate deterministic dynamics that transform the data distribution $\mathbf{p}^{(0)}$ into the model distribution $\mathbf{p}^{(\infty)}(\theta)$. Under maximum likelihood learning, model parameters θ are chosen so as to minimize the Kullback–Leibler (KL) divergence between the data distribution $\mathbf{p}^{(0)}$ and the model distribution $\mathbf{p}^{(\infty)}(\theta)$. Under MPF, however, the KL divergence between $\mathbf{p}^{(0)}$ and $\mathbf{p}^{(\epsilon)}$ is minimized instead, where $\mathbf{p}^{(\epsilon)}$ is the distribution obtained by initializing the dynamics at the data distribution $\mathbf{p}^{(0)}$ and then evolving them for an infinitesimal time ϵ. Here we represent graphically how parameter updates that pull $\mathbf{p}^{(\epsilon)}$ towards $\mathbf{p}^{(0)}$ also tend to pull $\mathbf{p}^{(\infty)}(\theta)$ towards $\mathbf{p}^{(0)}$.

the need to explicitly sample or integrate over the model distribution. In score matching one minimizes the expected square distance of the score function with respect to spatial coordinates given by the data distribution from the similar score function given by the model distribution. A number of connections have been made between score matching and other learning techniques (Hyvärinen, 2007; Sohl-Dickstein & Olshausen, 2009; Movellan, 2008a; Lyu, 2009).

Pseudolikelihood (Besag, 1975) approximates the joint probability distribution of a collection of random variables with a computationally tractable product of conditional distributions, where each factor is the distribution of a single random variable conditioned on the others. This approach often leads to surprisingly good parameter estimates, despite the extreme nature of the approximation.

Minimum velocity learning is an approach recently proposed by Movellan (Movellan, 2008a) that recasts a number of the ideas behind CD, treating the minimization of the initial dynamics away from the data distribution as the goal itself rather than a surrogate for it. Rather than directly minimize the difference between the data and the model, Movellan's proposal is to introduce system dynamics that have the model as their equilibrium distribution, and minimize the initial flow of probability away from the data under those dynamics. If the model looks exactly like the data there will be no flow of probability, and if model and data are similar the flow of probability will tend to be minimal. Movellan applies this intuition to the specific case of distributions over continuous state spaces evolving via diffusion dynamics, and recovers the score matching objective function.

Two additional recent techniques deserve mention. Minimum KL contraction (Lyu, 2011) involves applying a contraction mapping to both data and model distributions, and minimizing the amount by which this contraction mapping shrinks the KL divergence between data and model distributions. Like minimum probability flow, it appears to be a generalization of a number of existing parameter estimation techniques based on "local" information about the model distribution. Noise contrastive estimation (Gutmann & Hyvärinen, 2010) estimates model parameters and the partition function by training a classifier to distinguish between the data distribution and a noise distribution carefully chosen to resemble the data distribution.

Here we propose a consistent parameter estimation framework called minimum probability flow learning (MPF), applicable to *any* parametric model without latent variables. Minimum velocity learning, SM and certain forms of CD are all special cases of MPF, which is in many situations more powerful than any of these other algorithms. We demonstrate that learning under this framework is effective and fast in a number of cases: Ising models (Brush, 1967; Ackley et al., 1985), deep belief networks (Hinton et al., 2006), and independent component analysis (Bell AJ, 1995).

2. Minimum Probability Flow

Our goal is to find the parameters that cause a probabilistic model to best agree with a list $\mathcal{D}$ of (assumed iid) observations of the state of a system. We will do this by introducing deterministic dynamics that guarantee the transformation of the data distribution into the model distribution, and then minimizing the KL divergence between the data distribution and the distribution that results from running those dynamics for a short time ϵ (see Figure 1).

2.1. Distributions

The data distribution is represented by a vector $\mathbf{p}^{(0)}$, with $p_i^{(0)}$ the fraction of the observations $\mathcal{D}$ in state i. The superscript (0) represents time $t = 0$ under the system dynamics (which will be described in more detail in Section 2.2). For example, in a two variable

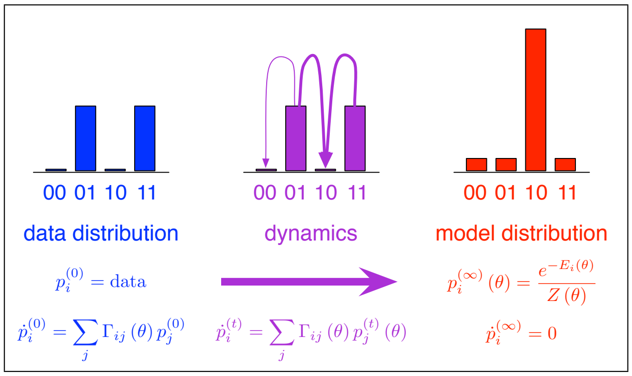

00 01 10 11
data distribution
$p_i^{(0)} = \text{data}$
$\dot{p}_i^{(0)} = \sum_j \Gamma_{ij}(\theta)\, p_j^{(0)}$
00 01 10 11
dynamics
$\dot{p}_i^{(t)} = \sum_j \Gamma_{ij}(\theta)\, p_j^{(t)}(\theta)$
00 01 10 11
model distribution
$p_i^{(\infty)}(\theta) = \frac{e^{-E_i(\theta)}}{Z(\theta)}$
$\dot{p}_i^{(\infty)} = 0$

Figure 2. Dynamics of minimum probability flow learning. Model dynamics represented by the probability flow matrix $\mathbf{\Gamma}$ (*middle*) determine how probability flows from the empirical histogram of the sample data points (*left*) to the equilibrium distribution of the model (*right*) after a sufficiently long time. In this example there are only four possible states for the system, which consists of a pair of binary variables, and the particular model parameters favor state 10 whereas the data falls on other states.

binary system, $\mathbf{p}^{(0)}$ would have four entries representing the fraction of the data in states 00, 01, 10 and 11 (Figure 2).

Our goal is to find the parameters θ that cause a model distribution $\mathbf{p}^{(\infty)}(\theta)$ to best match the data distribution $\mathbf{p}^{(0)}$. The superscript (∞) on the model distribution indicates that this is the equilibrium distribution reached after running the dynamics for infinite time. Without loss of generality, we assume the model distribution is of the form

$$p_i^{(\infty)}(\theta) = \frac{\exp\left(-E_i(\theta)\right)}{Z(\theta)}, \tag{1}$$

where $\mathbf{E}(\theta)$ is referred to as the energy function, and the normalizing factor $Z(\theta)$ is the partition function,

$$Z(\theta) = \sum_i \exp\left(-E_i(\theta)\right) \tag{2}$$

(this can be thought of as a Boltzmann distribution of a physical system with $k_B T$ set to 1).

2.2. Dynamics

Most Monte-Carlo algorithms rely on two core concepts from statistical physics, the first being conservation of probability as enforced by the master equation for the time evolution of a distribution $\mathbf{p}^{(t)}$ (Pathria, 1972):

$$\dot{p}_i^{(t)} = \sum_{j\neq i} \Gamma_{ij}(\theta)\, p_j^{(t)} - \sum_{j\neq i} \Gamma_{ji}(\theta)\, p_i^{(t)}, \tag{3}$$

where $\dot{p}_i^{(t)}$ is the time derivative of $p_i^{(t)}$. Transition rates $\Gamma_{ij}(\theta)$, for $i \neq j$, give the rate at which probability flows from a state j into a state i. The first term of Equation (3) captures the flow of probability out of other states j into the state i, and the second captures flow out of i into other states j. The dependence on θ results from the requirement that the chosen dynamics cause $\mathbf{p}^{(t)}$ to flow to the equilibrium distribution $\mathbf{p}^{(\infty)}(\theta)$. For readability, explicit dependence on θ will be dropped except where necessary. If we choose to set the diagonal elements of $\mathbf{\Gamma}$ to obey $\Gamma_{ii} = -\sum_{j\neq i}\Gamma_{ji}$, then we can write the dynamics as

$$\dot{\mathbf{p}}^{(t)} = \mathbf{\Gamma}\mathbf{p}^{(t)} \tag{4}$$

(see Figure 2). The unique solution for $\mathbf{p}^{(t)}$ is given by[1]

$$\mathbf{p}^{(t)} = \exp\left(\mathbf{\Gamma}t\right)\mathbf{p}^{(0)}, \tag{5}$$

where $\exp\left(\mathbf{\Gamma}t\right)$ is a matrix exponential.

2.3. Detailed Balance

The second core concept is detailed balance,

$$\Gamma_{ji}\, p_i^{(\infty)}(\theta) = \Gamma_{ij}\, p_j^{(\infty)}(\theta), \tag{6}$$

which states that at equilibrium the probability flow from state i into state j equals the probability flow from j into i. When satisfied, detailed balance guarantees that the distribution $\mathbf{p}^{(\infty)}(\theta)$ is a fixed point of the dynamics. Sampling in most Monte Carlo methods is performed by choosing $\mathbf{\Gamma}$ consistent with Equation 6 (and the added requirement of ergodicity), then stochastically running the dynamics of Equation 3. Note that there is no need to restrict the dynamics defined by $\mathbf{\Gamma}$ to those of any real physical process, such as diffusion.

Equation 6 can be written in terms of the model's energy function $\mathbf{E}(\theta)$ by substituting in Equation 1 for $\mathbf{p}^{(\infty)}(\theta)$:

$$\Gamma_{ji}\exp\left(-E_i(\theta)\right) = \Gamma_{ij}\exp\left(-E_j(\theta)\right). \tag{7}$$

$\mathbf{\Gamma}$ is underconstrained by the above equation. Introducing the additional constraint that Γ be invariant to the addition of a constant to the energy function (as the model distribution $\mathbf{p}^{(\infty)}(\theta)$ is), we choose the following form for the non-diagonal entries in $\mathbf{\Gamma}$

$$\Gamma_{ij} = g_{ij}\exp\left[\frac{1}{2}\left(E_j(\theta) - E_i(\theta)\right)\right] \qquad (i \neq j), \tag{8}$$

[1] The form chosen for $\mathbf{\Gamma}$ in Equation (4), coupled with the satisfaction of detailed balance and ergodicity introduced in section 2.3, guarantees that there is a unique eigenvector $\mathbf{p}^{(\infty)}$ of $\mathbf{\Gamma}$ with eigenvalue zero, and that all other eigenvalues of $\mathbf{\Gamma}$ have negative real parts.

where the connectivity function

$$g_{ij} = g_{ji} = \begin{cases} 0 & \text{unconnected states} \\ 1 & \text{connected states} \end{cases} \quad (i \neq j) \tag{9}$$

determines which states are allowed to directly exchange probability with each other[2]. g_{ij} can be set such that $\mathbf{\Gamma}$ is *extremely* sparse (see Section 2.5). Theoretically, to guarantee convergence to the model distribution, the non-zero elements of $\mathbf{\Gamma}$ must be chosen such that, given sufficient time, probability can flow between any pair of states (ergodicity).

2.4. Objective Function

Maximum likelihood parameter estimation involves maximizing the likelihood of some observations $\mathcal{D}$ under a model, or equivalently minimizing the KL divergence between the data distribution $\mathbf{p}^{(0)}$ and model distribution $\mathbf{p}^{(\infty)}$,

$$\hat{\theta}_{\mathrm{ML}} = \underset{\theta}{\operatorname{argmin}}\, D_{KL}\left(\mathbf{p^{(0)}}||\mathbf{p}^{(\infty)}\left(\theta\right)\right) \tag{10}$$

Rather than running the dynamics for infinite time, we propose to minimize the KL divergence after running the dynamics for an infinitesimal time ϵ,

$$\hat{\theta}_{\mathrm{MPF}} = \underset{\theta}{\operatorname{argmin}}\, K\left(\theta\right) \tag{11}$$

$$K\left(\theta\right) = D_{KL}\left(\mathbf{p^{(0)}}||\mathbf{p}^{(\epsilon)}\left(\theta\right)\right). \tag{12}$$

For small ϵ, $D_{KL}\left(\mathbf{p^{(0)}}||\mathbf{p}^{(\epsilon)}\left(\theta\right)\right)$ can be approximated by a first order Taylor expansion,

$$K\left(\theta\right) \approx D_{KL}\left(\mathbf{p^{(0)}}||\mathbf{p^{(t)}}\left(\theta\right)\right)\Big|_{t=0} + \epsilon \frac{\partial D_{KL}\left(\mathbf{p^{(0)}}||\mathbf{p^{(t)}}\left(\theta\right)\right)}{\partial t}\Big|_{t=0}. \tag{13}$$

Further algebra (see Appendix A) reduces $K\left(\theta\right)$ to a measure of the flow of probability, at time $t = 0$ under the dynamics, out of data states $j \in \mathcal{D}$ into non-data states $i \notin \mathcal{D}$,

$$K\left(\theta\right) = \frac{\epsilon}{|\mathcal{D}|} \sum_{i\notin\mathcal{D}} \sum_{j\in\mathcal{D}} \Gamma_{ij} \tag{14}$$

$$= \frac{\epsilon}{|\mathcal{D}|} \sum_{j\in\mathcal{D}} \sum_{i\notin\mathcal{D}} g_{ij} \exp\left[\frac{1}{2}\left(E_j\left(\theta\right) - E_i\left(\theta\right)\right)\right] \tag{15}$$

with gradient

$$\frac{\partial K\left(\theta\right)}{\partial \theta} = \frac{\epsilon}{|\mathcal{D}|} \sum_{j\in\mathcal{D}} \sum_{i\notin\mathcal{D}} \left[\frac{\partial E_j\left(\theta\right)}{\partial \theta} - \frac{\partial E_i\left(\theta\right)}{\partial \theta}\right] g_{ij} \exp\left[\frac{1}{2}\left(E_j\left(\theta\right) - E_i\left(\theta\right)\right)\right], \tag{16}$$

where $|\mathcal{D}|$ is the number of observed data points. Note that Equations (14) and (16) do not depend on the partition function $Z\left(\theta\right)$ or its derivatives.

$K\left(\theta\right)$ is uniquely zero when $\mathbf{p}^{(0)}$ and $\mathbf{p}^{(\infty)}\left(\theta\right)$ are equal. This implies consistency, in that if the data comes from the model class, in the limit of infinite data $K\left(\theta\right)$ will be minimized by exactly the right θ. In addition, $K\left(\theta\right)$ is convex for all models $\mathbf{p}^{(\infty)}\left(\theta\right)$ in the exponential family - that is, models whose energy functions $\mathbf{E}\left(\theta\right)$ are linear in their parameters θ (Macke & Gerwinn, 2009) (see Appendix B).

2.5. Tractability

The dimensionality of the vector $\mathbf{p}^{(0)}$ is typically huge, as is that of $\mathbf{\Gamma}$ (*e.g.*, 2^d and $2^d \times 2^d$, respectively, for a d-bit binary system). Naïvely, this would seem to prohibit evaluation and minimization of the objective function. Fortunately, we need only visit those columns of Γ_{ij} corresponding to data states, $j \in \mathcal{D}$. Additionally, g_{ij} can be populated so as to connect each state j to only a small fixed number of additional states i. The cost in both memory and time to evaluate the objective function is thus $\mathcal{O}(|\mathcal{D}|)$, and does not depend on the number of system states, only on the (much smaller) number of observed data points.

2.6. Continuous State Spaces

Although we have motivated this technique using systems with a large, but finite, number of states, it generalizes to continuous state spaces. Γ_{ji}, g_{ji}, and $p_i^{(t)}$ become continuous functions $\Gamma\left(\mathbf{x}_j, \mathbf{x}_i\right)$, $g\left(\mathbf{x}_j, \mathbf{x}_i\right)$, and $p^{(t)}\left(\mathbf{x}_i\right)$. $\Gamma\left(\mathbf{x}_j, \mathbf{x}_i\right)$ can be populated stochastically and extremely sparsely (see Appendix D), preserving the $\mathcal{O}(|\mathcal{D}|)$ cost. A specific scheme (similar to CD with Hamiltonian Monte Carlo) for estimating parameters in a continuous state space via MPF is described in Appendix E.

2.7. Choosing the Connectivity Function g

Qualitatively, the most informative states to connect data states to are those that are most probable under the model. In discrete state spaces, nearest neighbor connectivity schemes for g_{ji} work extremely well (eg Equation 21 below). This is because, as learning converges, the states that are near data states become the

[2] The non-zero $\mathbf{\Gamma}$ may also be sampled from a proposal distribution rather than set via a deterministic scheme, in which case g_{ij} takes on the role of proposal distribution - see Appendix D.

states that are probable under the model.

In continuous state spaces, the estimated parameters are much more sensitive to the choice of $g\left(\mathbf{x}_j, \mathbf{x}_i\right)$. One effective form for $g\left(\mathbf{x}_j, \mathbf{x}_i\right)$ is described in Appendix E, but theory supporting different choices of $g\left(\mathbf{x}_j, \mathbf{x}_i\right)$ remains an area of active exploration.

3. Connection to Other Learning Techniques

3.1. Contrastive Divergence

The contrastive divergence update rule can be written in the form

$$\Delta\theta_{CD} \propto -\sum_{j\in\mathcal{D}}\sum_{i\notin\mathcal{D}}\left[\frac{\partial E_j\left(\theta\right)}{\partial\theta} - \frac{\partial E_i\left(\theta\right)}{\partial\theta}\right] T_{ij}, \qquad (17)$$

where T_{ij} is the probability of transitioning from state j to state i in a single Markov chain Monte Carlo step (or k steps for CD-k). Equation 17 has obvious similarities to the MPF learning gradient in Equation 16. Thus, steepest gradient descent under MPF resembles CD updates, but with the MCMC sampling/rejection step T_{ij} replaced by a weighting factor $g_{ij}\exp\left[\frac{1}{2}\left(E_j\left(\theta\right) - E_i\left(\theta\right)\right)\right]$.

Note that this difference in form provides MPF with a well-defined objective function. One important consequence of the existence of an objective function is that MPF can readily utilize general purpose, off-the-shelf optimization packages for gradient descent, which would have to be tailored in some way to be applied to CD. This is part of what accounts for the dramatic difference in learning time between CD and MPF in some cases (see Fig. 3).

3.2. Score Matching

For a continuous state space, MPF reduces to score matching if the connectivity function $g\left(\mathbf{x}_j, \mathbf{x}_i\right)$ is set to connect all states within a small distance r of each other,

$$g(\mathbf{x}_i, \mathbf{x}_j) = g(\mathbf{x}_j, \mathbf{x}_i) = \left\{\begin{array}{ll} 0 & d(\mathbf{x}_i, \mathbf{x}_j) > r \\ 1 & d(\mathbf{x}_i, \mathbf{x}_j) \leq r \end{array}\right., \qquad (18)$$

where $d(\mathbf{x}_i, \mathbf{x}_j)$ is the Euclidean distance between states $\mathbf{x}_i$ and $\mathbf{x}_j$. In the limit as r goes to 0 (within an overall constant and scaling factor),

$$\begin{aligned} \lim_{r\to 0} K\left(\theta\right) &\sim K_{\text{SM}}\left(\theta\right) \\ &= \sum_{\mathbf{x}\in\mathcal{D}}\left[\frac{1}{2}\nabla E(\mathbf{x})\cdot\nabla E(\mathbf{x}) - \nabla^2 E(\mathbf{x})\right], \end{aligned} \qquad (19)$$

where $K_{\text{SM}}\left(\theta\right)$ is the SM objective function (see Appendix C). Unlike SM, MPF is applicable to any parametric model, including discrete systems, and it does not require evaluating a third order derivative, which can result in unwieldy expressions.

4. Experimental Results

Matlab code implementing MPF for several cases is available at https://github.com/Sohl-Dickstein/Minimum-Probability-Flow-Learning.

All minimization was performed using minFunc (Schmidt, 2005).

4.1. Ising Model

The Ising model has a long and storied history in physics (Brush, 1967) and machine learning (Ackley et al., 1985) and it has recently been found to be a surprisingly useful model for networks of neurons in the retina (Schneidman et al., 2006; Shlens et al., 2006).

We estimated parameters for an Ising model (sometimes referred to as a fully visible Boltzmann machine or an Ising spin glass) of the form

$$p^{(\infty)}(\mathbf{x};\mathbf{J}) = \frac{1}{Z(\mathbf{J})}\exp\left[-\mathbf{x}^{\mathrm{T}}\mathbf{J}\mathbf{x}\right], \qquad (20)$$

where the coupling matrix $\mathbf{J}$ only had non-zero elements corresponding to nearest-neighbor units in a two-dimensional square lattice, and bias terms along the diagonal. The training data $\mathcal{D}$ consisted of $20,000$ d-element iid binary samples $\mathbf{x}\in\{0,1\}^d$ generated via Swendsen-Wang sampling (Swendsen & Wang, 1987) from a spin glass with known coupling parameters. We used a square 10×10 lattice, $d = 10^2$. The non-diagonal nearest-neighbor elements of $\mathbf{J}$ were set using draws from a normal distribution with variance $\sigma^2 = 10$. The diagonal (bias) elements of $\mathbf{J}$ were set in such a way that each column of $\mathbf{J}$ summed to 0, so that the expected unit activations were 0.5. The transition matrix $\mathbf{\Gamma}$ had $2^d\times 2^d$ elements, but for learning we populated it sparsely, setting

$$g_{ij} = g_{ji} = \left\{\begin{array}{ll} 1 & \text{states } i,j \text{ differ by single bit flip} \\ 0 & \text{otherwise} \end{array}\right. . \qquad (21)$$

Figure 3 shows the mean square error in the estimated $\mathbf{J}$ and the mean square error in the corresponding pairwise correlations as a function of learning time for MPF and four competing approaches: mean field theory with TAP corrections (Tanaka, 1998), CD with both one and ten sampling steps per iteration, and

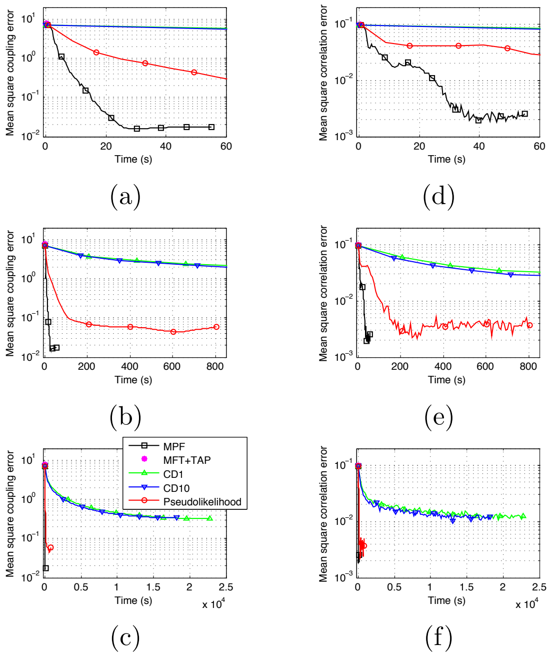


Figure 3. A demonstration of Minimum Probability Flow (MPF) outperforming existing techniques for parameter recovery in an Ising model. **(a)** Time evolution of the mean square error in the coupling strengths for 5 methods for the first 60 seconds of learning. Note that mean field theory with second order corrections (MFT+TAP) actually increases the error above random parameter assignments in this case. **(b)** Mean square error in the coupling strengths for the first 800 seconds of learning. **(c)** Mean square error in coupling strengths for the entire learning period. **(d)**–**(f)** Mean square error in pairwise correlations for the first 60 seconds of learning, the first 800 seconds of learning, and the entire learning period, respectively. In every comparison above MPF finds a better fit, and for all cases but MFT+TAP does so in a shorter time (see Table 1).

pseudolikelihood. Using MPF, learning took approximately 60 seconds, compared to roughly 800 seconds for pseudolikelihood and upwards of 20,000 seconds for 1-step and 10-step CD. Note that given sufficient training samples, MPF would converge exactly to the right answer, as learning in the Ising model is convex (see Appendix B), and has its global minimum at the true solution. Table 1 shows the relative performance at convergence in terms of mean square error in recovered weights, mean square error in the resulting model's correlation function, and convergence time. MPF was dramatically faster to converge than any of the other models tested, with the exception of MFT+TAP, which failed to find reasonable parameters. MPF fit the model to the data substantially better than any of the other models.

Table 1. Mean square error in recovered coupling strengths (ϵ_J), mean square error in pairwise correlations (ϵ_{corr}) and learning time for MPF versus mean field theory with TAP correction (MFT+TAP), 1-step and 10-step contrastive divergence (CD-1 and CD-10), and pseudolikelihood (PL).

Technique	ϵ_J	ϵ_{corr}	Time (s)
MPF	0.0172	0.0025	∼60
MFT+TAP	7.7704	0.0983	0.1
CD-1	0.3196	0.0127	∼20000
CD-10	0.3341	0.0123	∼20000
PL	0.0582	0.0036	∼800

4.2. Deep Belief Network

As a demonstration of learning on a more complex discrete valued model, we trained a 4 layer deep belief network (DBN) (Hinton et al., 2006) on MNIST handwritten digits. A DBN consists of stacked restricted Boltzmann machines (RBMs), such that the hidden layer of one RBM forms the visible layer of the next. Each RBM has the form

$$p^{(\infty)}(\mathbf{x}_{\mathrm{vis}}, \mathbf{x}_{\mathrm{hid}}; \mathbf{W}) = \frac{\exp\left[\mathbf{x}_{\mathrm{hid}}^T \mathbf{W} \mathbf{x}_{\mathrm{vis}}\right]}{Z(\mathbf{W})}, \tag{22}$$

$$p^{(\infty)}(\mathbf{x}_{\mathrm{vis}}; \mathbf{W}) = \frac{\exp\left[\sum_k \log\left(1 + \exp\left[\mathbf{W}_k \mathbf{x}_{\mathrm{vis}}\right]\right)\right]}{Z(\mathbf{W})}. \tag{23}$$

Sampling-free application of MPF requires analytically marginalizing over the hidden units. RBMs were trained in sequence, starting at the bottom layer, on 10,000 samples from the MNIST postal hand written digits data set. As in the Ising case, the transition matrix $\mathbf{\Gamma}$ was populated so as to connect every state to all states that differed by only a single bit flip (Equation 21). Training was performed by both MPF and single step CD (note that CD turns into full ML learning as the number of steps is increased, and that many step CD would have produced a superior, more computationally expensive, answer).

Confabulations were generated by Gibbs sampling from the top layer RBM, then propagating each sample back down to the pixel layer by way of the conditional distribution $p^{(\infty)}(\mathbf{x}_{\mathrm{vis}} | \mathbf{x}_{\mathrm{hid}}; \mathbf{W}^k)$ for each of the intermediary RBMs, where k indexes the layer in the stack. 1,000 sampling steps were taken between each confabulation. As shown in Figure 4, MPF learned a good model of handwritten digits.

4.3. Independent Component Analysis

As a demonstration of parameter estimation in continuous state space probabilistic models, we trained the receptive fields $\mathbf{J} \in R^{K \times K}$ of a K dimensional in-

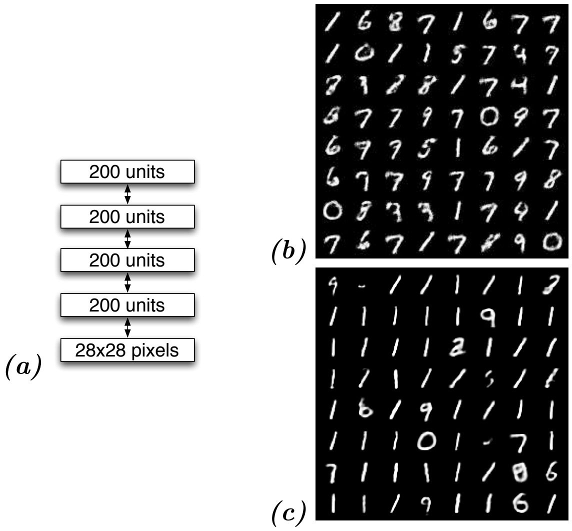


Figure 4. A deep belief network trained using minimum probability flow learning (MPF). *(a)* A four layer deep belief network was trained on the MNIST postal hand written digits dataset by MPF and single step contrastive divergence (CD). *(b)* Confabulations after training via MPF. A reasonable probabilistic model for handwritten digits has been learned. *(c)* Confabulations after training via CD. The uneven distribution of digit occurrences suggests that CD-1 has learned a less representative model than MPF.

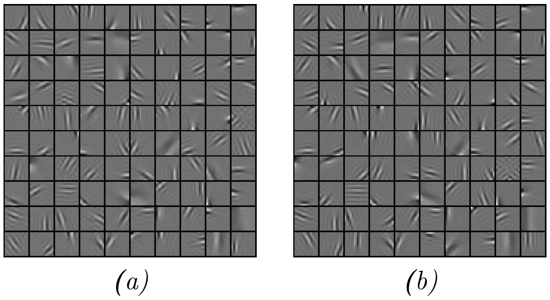


Figure 5. A continuous state space model fit using minimum probability flow learning (MPF). Learned 10 × 10 pixel independent component analysis receptive fields **J** trained on natural image patches via *(a)* MPF and *(b)* maximum likelihood learning (ML). The average log likelihood of the model found by MPF (−120.61 nats) was nearly identical to that found by ML (−120.33 nats), consistent with the visual similarity of the receptive fields.

dependent component analysis (ICA) (Bell AJ, 1995) model with a Laplace prior,

$$p^{(\infty)}\left(\mathbf{x};\mathbf{J}\right) = \frac{e^{-\sum_k \left|\mathbf{J}_k \mathbf{x}\right|}}{2^K \left|\mathbf{J}^{-1}\right|}, \tag{24}$$

on 100, 000 10 × 10 whitened natural image patches from the van Hateren database (Hateren & Schaaf, 1998). Since the log likelihood and its gradient can be calculated analytically for ICA, we solved for **J** via both maximum likelihood learning and MPF, and compared the resulting log likelihoods. Both training techniques were initialized with identical Gaussian noise, and trained on the same data, which accounts for the similarity of individual receptive fields found by the two algorithms. The average log likelihood of the model after parameter estimation via MPF was −120.61 nats, while the average log likelihood after estimation via maximum likelihood was −120.33 nats. The receptive fields resulting from training under both techniques are shown in Figure 5. MPF minimization was performed by alternating steps of updating the connectivity function $g\left(\mathbf{x}_j, \mathbf{x}_i\right)$ using a Hamiltonian dynamics based scheme, and minimizing the objective function in Equation 15 via LBFGS for fixed $g\left(\mathbf{x}_j, \mathbf{x}_i\right)$. This is described in more detail in Appendix E.

5. Summary

We have presented a novel, general purpose framework, called minimum probability flow learning (MPF), for parameter estimation in probabilistic models that outperforms current techniques in both learning time and accuracy. MPF works for any parametric model without hidden state variables, including those over both continuous and discrete state space systems, and it avoids explicit calculation of the partition function by employing deterministic dynamics in place of the slow sampling required by many existing approaches. Because MPF provides a simple and well-defined objective function, it can be minimized quickly using existing higher order gradient descent techniques. Furthermore, the objective function is convex for models in the exponential family, ensuring that the global minimum can be found with gradient descent in these cases. MPF was inspired by the minimum velocity approach developed by Movellan, and it reduces to that technique as well as to score matching and some forms of contrastive divergence under suitable choices for the dynamics and state space. We hope that this new approach to parameter estimation will enable probabilistic modeling for previously intractable problems.

Acknowledgments

We would like to thank Javier Movellan, Tamara Broderick, Miroslav Dudík, Gašper Tkačik, Robert E. Schapire, William Bialek for sharing work in progress and data; Ashvin Vishwanath, Jonathon Shlens, Tony Bell, Charles Cadieu, Nicole Carlson, Christopher Hillar, Kilian Koepsell, Bruno Olshausen and the rest of the Redwood Center for many useful discussions; and the James S. McDonnell Foundation (JSD, PB, JSD) and the Canadian Institute for Advanced Research - Neural Computation and Perception Program (JSD) for financial support.

Appendices Available at

HTTP://REDWOOD.BERKELEY.EDU/JASCHA/.

References

Ackley, D H, Hinton, G E, and Sejnowski, T J. A learning algorithm for Boltzmann machines. *Cognitive Science*, 9 (2):147–169, 1985.

Bell AJ, Sejnowski TJ. An information-maximization approach to blind separation and blind deconvolution. *Neural Computation 1995; vol. 7:1129-1159*, 1995.

Besag, J. Statistical analysis of non-lattice data. *The Statistician, 24(3), 179-195*, 1975.

Brush, S G. History of the Lenz-Ising model. *Reviews of Modern Physics*, 39(4):883–893, Oct 1967.

Carreira-Perpiñán, M A and Hinton, G E. On contrastive divergence (CD) learning. *Technical report, Dept. of Computer Science, University of Toronto*, 2004.

Gutmann, M and Hyvärinen, A. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. *Proc. Int. Conf. on Artificial Intelligence and Statistics (AISTATS2010)*, 2010.

Hateren, J. H. van and Schaaf, A. van der. Independent component filters of natural images compared with simple cells in primary visual cortex. *Proceedings: Biological Sciences*, 265(1394):359–366, Mar 1998.

Haykin, S. *Neural networks and learning machines; 3rd edition*. Prentice Hall, 2008.

Hinton, Geoffrey E, Osindero, Simon, and Teh, Yee-Whye. A fast learning algorithm for deep belief nets. *Neural Computation*, 18(7):1527–1554, Jul 2006. doi: 10.1162/neco.2006.18.7.1527.

Hyvärinen, A. Estimation of non-normalized statistical models using score matching. *Journal of Machine Learning Research*, 6:695–709, 2005.

Hyvärinen, A. Connections between score matching, contrastive divergence, and pseudolikelihood for continuous-valued variables. *IEEE Transactions on Neural Networks*, Jan 2007.

Jaakkola, T and Jordan, M. A variational approach to Bayesian logistic regression models and their extensions. *Proceedings of the Sixth International Workshop on Artificial Intelligence and Statistics*, Jan 1997.

Kappen, H and Rodríguez, F. Mean field approach to learning in Boltzmann machines. *Pattern Recognition Letters*, Jan 1997.

Lyu, S. Interpretation and generalization of score matching. *The proceedings of the 25th conference on uncertainty in artificial intelligence (UAI*90)*, 2009.

Lyu, S. Personal communication. 2011.

MacKay, D. Failures of the one-step learning algorithm. *Failures of the one-step learning algorithm*, Jan 2001.

Macke, J and Gerwinn, S. Personal communication. 2009.

Movellan, J R. A minimum velocity approach to learning. *unpublished draft*, Jan 2008a.

Movellan, J R. Contrastive divergence in gaussian diffusions. *Neural Computation*, 20(9):2238–2252, 2008b.

Movellan, J R and McClelland, J L. Learning continuous probability distributions with symmetric diffusion networks. *Cognitive Science*, 17:463–496, 1993.

Pathria, R. *Statistical Mechanics*. Butterworth Heinemann, Jan 1972.

Schmidt, M. minfunc. *http://www.cs.ubc.ca/ schmidtm/Software/minFunc.html*, 2005.

Schneidman, E, 2nd, M J Berry, Segev, R, and Bialek, W. Weak pairwise correlations imply strongly correlated network states in a neural population. *Nature*, 440(7087):1007–12, 2006.

Shlens, J, Field, G D, Gauthier, J L, Grivich, M I, Petrusca, D, Sher, A, Litke, A M, and Chichilnisky, E J. The structure of multi-neuron firing patterns in primate retina. *J. Neurosci.*, 26(32):8254–66, 2006.

Sohl-Dickstein, J and Olshausen, B. A spatial derivation of score matching. *Redwood Center Technical Report*, 2009.

Swendsen, R.H. and Wang, J.S. Nonuniversal critical dynamics in Monte Carlo simulations. *Physical Review Letters*, 58(2):86–88, 1987. ISSN 1079-7114.

Tanaka, T. Mean-field theory of Boltzmann machine learning. *Physical Review Letters E*, Jan 1998.

Welling, M and Hinton, G. A new learning algorithm for mean field Boltzmann machines. *Lecture Notes in Computer Science*, Jan 2002.

Yuille, A. The convergence of contrastive divergences. *Department of Statistics, UCLA. Department of Statistics Papers.*, 2005.

Minimum Probability Flow Learning

APPENDICES

A Taylor Expansion of KL Divergence

$$
\begin{aligned}
K(\theta) \approx & D_{KL}\left(\mathbf{p^{(0)}}||\mathbf{p^{(t)}}(\theta)\right)\Big|_{t=0} \\
& + \epsilon \frac{\partial D_{KL}\left(\mathbf{p^{(0)}}||\mathbf{p^{(t)}}(\theta)\right)}{\partial t}\Big|_{t=0} & \text{(A-1)} \\
& = 0 + \epsilon \frac{\partial D_{KL}\left(\mathbf{p^{(0)}}||\mathbf{p^{(t)}}(\theta)\right)}{\partial t}\Big|_{t=0} & \text{(A-2)} \\
& = \epsilon \frac{\partial}{\partial t}\left(\sum_{i\in\mathcal{D}} p_i^{(0)} \log \frac{p_i^{(0)}}{p_i^{(t)}}\right)\Bigg|_0 & \text{(A-3)} \\
& = -\epsilon \sum_{i\in\mathcal{D}} \frac{p_i^{(0)}}{p_i^{(0)}} \frac{\partial p_i^{(t)}}{\partial t}\Bigg|_0 & \text{(A-4)} \\
& = -\epsilon \sum_{i\in\mathcal{D}} \frac{\partial p_i^{(t)}}{\partial t}\Bigg|_0 & \text{(A-5)} \\
& = -\epsilon \left(\frac{\partial}{\partial t}\sum_{i\in\mathcal{D}} p_i^{(t)}\right)\Bigg|_0 & \text{(A-6)} \\
& = -\epsilon \frac{\partial}{\partial t}\left(1 - \sum_{i\notin\mathcal{D}} p_i^{(t)}\right)\Bigg|_0 & \text{(A-7)} \\
& = \epsilon \sum_{i\notin\mathcal{D}} \frac{\partial p_i^{(t)}}{\partial t}\Bigg|_0 & \text{(A-8)} \\
& = \epsilon \sum_{i\notin\mathcal{D}} \sum_{j\in\mathcal{D}} \Gamma_{ij} p_j^{(0)} & \text{(A-9)} \\
& = \frac{\epsilon}{|\mathcal{D}|} \sum_{i\notin\mathcal{D}} \sum_{j\in\mathcal{D}} \Gamma_{ij}, & \text{(A-10)}
\end{aligned}
$$

where we used the fact that $\sum_{i\in\mathcal{D}} p_i^{(t)} + \sum_{i\notin\mathcal{D}} p_i^{(t)} = 1$. This implies that the rate of growth of the KL divergence at time $t = 0$ equals the total initial flow of probability from states with data into states without.

B Convexity

As observed by Macke and Gerwinn (Macke & Gerwin, 2009), the MPF objective function is convex for models in the exponential family.

We wish to minimize

$$K = \sum_{i \in D} \sum_{j \in D^C} \Gamma_{ji} p_i^{(0)}. \tag{B-1}$$

K has derivative

$$\begin{aligned}
\frac{\partial K}{\partial \theta_m} &= \sum_{i \in D} \sum_{j \in D^c} \left(\frac{\partial \Gamma_{ij}}{\partial \theta_m} \right) p_i^{(0)} && \text{(B-2)} \\
&= \frac{1}{2} \sum_{i \in D} \sum_{j \in D^c} \Gamma_{ij} \left(\frac{\partial E_j}{\partial \theta_m} - \frac{\partial E_i}{\partial \theta_m} \right) p_i^{(0)}, && \text{(B-3)}
\end{aligned}$$

and Hessian

$$\begin{aligned}
\frac{\partial^2 K}{\partial \theta_m \partial \theta_n} = &\frac{1}{4} \sum_{i \in D} \sum_{j \in D^c} \Gamma_{ij} \left(\frac{\partial E_j}{\partial \theta_m} - \frac{\partial E_i}{\partial \theta_m} \right) \left(\frac{\partial E_j}{\partial \theta_n} - \frac{\partial E_i}{\partial \theta_n} \right) p_i^{(0)} \\
&+ \frac{1}{2} \sum_{i \in D} \sum_{j \in D^c} \Gamma_{ij} \left(\frac{\partial^2 E_j}{\partial \theta_m \partial \theta_n} - \frac{\partial^2 E_i}{\partial \theta_m \partial \theta_n} \right) p_i^{(0)}. && \text{(B-4)}
\end{aligned}$$

The first term is a weighted sum of outer products, with non-negative weights $\frac{1}{4}\Gamma_{ij} p_i^{(0)}$, and is thus positive semidefinite. The second term is 0 for models in the exponential family (those with energy functions linear in their parameters).

Parameter estimation for models in the exponential family is therefore convex using minimum probability flow learning.

C Score matching

Score matching, developed by Aapo Hyvärinen [Hyvärinen(2005)], is a method that learns parameters in a probabilistic model using only derivatives of the energy function evaluated over the data distribution (see Equation (C-5)). This sidesteps the need to explicitly sample or integrate over the model distribution. In score matching one minimizes the expected square distance of the score function with respect to spatial coordinates given by the data distribution from the similar score function given by the model distribution. A number of connections have been made between score matching and other learning techniques [Hyvärinen(2007a), Sohl-Dickstein & Olshausen(2009)Sohl-Dickstein and Olshausen, Movellan(2008), Lyu(2009)]. Here we show that in the correct limit, MPF also reduces to score matching.

For a d-dimensional, continuous state space, we can write the MPF objective function as

$$\begin{aligned}
K_{\text{MPF}} &= \frac{1}{N} \sum_{x \in \mathcal{D}} \int \mathrm{d}^d y \, \Gamma(y, x) \\
&= \frac{1}{N} \sum_{x \in \mathcal{D}} \int \mathrm{d}^d y \, g(y, x) e^{(E(y|\theta) - E(x|\theta))}, && \text{(C-1)}
\end{aligned}$$

where the sum $\sum_{x\in\mathcal{D}}$ is over all data samples, and N is the number of samples in the data set $\mathcal{D}$. Now we assume that transitions are only allowed from states x to states y that are within a hypercube of side length ϵ centered around x in state space. (The master equation will reduce to Gaussian diffusion as $\epsilon \to 0$.) Thus, the function $g(y,x)$ will equal 1 when y is within the x-centered cube (or x within the y-centered cube) and 0 otherwise. Calling this cube C_ϵ, and writing $y = x + \alpha$ with $\alpha \in C_\epsilon$, we have

$$K_{\mathrm{MPF}} = \frac{1}{N}\sum_{x\in\mathcal{D}}\int_{C_\epsilon} \mathrm{d}^d\alpha\ e^{(E(x+\alpha|\theta)-E(x|\theta))}. \tag{C-2}$$

If we Taylor expand in α to second order and ignore cubic and higher terms, we get

$$\begin{aligned} K_{\mathrm{MPF}} \approx{}& \frac{1}{N}\sum_{x\in\mathcal{D}}\int_{C_\epsilon} \mathrm{d}^d\alpha\ (1) \\ & - \frac{1}{N}\sum_{x\in\mathcal{D}}\int_{C_\epsilon} \mathrm{d}^d\alpha\ \frac{1}{2}\sum_{i=1}^{d}\alpha_i\nabla_{x_i}E(x|\theta) \\ & + \frac{1}{N}\sum_{x\in\mathcal{D}}\int_{C_\epsilon} \mathrm{d}^d\alpha\ \frac{1}{4}\Bigg(\frac{1}{2}\Bigg[\sum_{i=1}^{d}\alpha_i\nabla_{x_i}E(x|\theta)\Bigg]^2 \\ & \quad - \sum_{i,j=1}^{d}\alpha_i\alpha_j\nabla_{x_i}\nabla_{x_j}E(x|\theta)\Bigg). \end{aligned} \tag{C-3}$$

This reduces to

$$\begin{aligned} K_{\mathrm{MPF}} \approx{}& \frac{1}{N}\sum_{x\in\mathcal{D}}\Bigg[\epsilon^d + \frac{1}{4}\Bigg(\frac{1}{2}\frac{2}{3}\epsilon^{d+2}\sum_{i=1}^{d}\Big[\nabla_{x_i}E(x|\theta)\Big]^2 \\ & \quad - \frac{2}{3}\epsilon^{d+2}\sum_{i=1}^{d}\nabla_{x_i}^2 E(x|\theta)\Bigg)\Bigg], \end{aligned} \tag{C-4}$$

which, removing a constant offset and scaling factor, is exactly equal to the score matching objective function,

$$K_{\mathrm{MPF}} \sim \frac{1}{N}\sum_{x\in\mathcal{D}}\Bigg[\frac{1}{2}\nabla E(x|\theta)\cdot\nabla E(x|\theta) - \nabla^2 E(x|\theta)\Bigg] \tag{C-5}$$

$$= K_{\mathrm{SM}}. \tag{C-6}$$

Score matching is thus equivalent to MPF when the connectivity function $g(y,x)$ is non-zero only for states infinitesimally close to each other. It should be noted that the score matching estimator has a closed-form solution when the model distribution belongs to the exponential family [Hyvärinen(2007b)], so the same can be said for MPF in this limit.

D Sampling the connectivity function Γ_{ij}

Here we extend MPF to allow the connectivity function Γ_{ij} to be sampled rather than set via a deterministic scheme. Since Γ is now sampled, we modify detailed balance to demand that, averaging over the choices for Γ, the net flow between pairs of states is 0,

$$\begin{aligned}\left\langle \Gamma_{ji}\ p_i^{(\infty)}(\theta) \right\rangle &= \left\langle \Gamma_{ij}\ p_j^{(\infty)}(\theta) \right\rangle && \text{(D-1)}\\ \langle \Gamma_{ji} \rangle\ p_i^{(\infty)}(\theta) &= \langle \Gamma_{ij} \rangle\ p_j^{(\infty)}(\theta), && \text{(D-2)}\end{aligned}$$

where the ensemble average is over the connectivity scheme for Γ. We describe the connectivity scheme via a proposal distribution g_{ij}, such that the probability of there being a connection from state j to state i at any given moment is g_{ij}. We also introduce a function F_{ij}, which provides the value Γ_{ij} takes on when a connection occurs from j to i. That is, it is the probability flow rate when flow occurs -

$$\langle \Gamma_{ij} \rangle = g_{ij} F_{ij}. \tag{D-3}$$

Detailed balance now becomes

$$g_{ji} F_{ji}\ p_i^{(\infty)}(\theta) = g_{ij} F_{ij}\ p_j^{(\infty)}(\theta). \tag{D-4}$$

Solving for $\mathbf{F}$ we find

$$\frac{F_{ij}}{F_{ji}} = \frac{g_{ji}}{g_{ij}} \frac{p_i^{(\infty)}(\theta)}{p_j^{(\infty)}(\theta)} = \frac{g_{ji}}{g_{ij}} \exp\left[E_j(\theta) - E_i(\theta)\right]. \tag{D-5}$$

$\mathbf{F}$ is underconstrained by the above equation. Motivated by symmetry, we choose as the form for the (non-zero, non-diagonal) entries in $\mathbf{F}$

$$F_{ij} = \left(\frac{g_{ji}}{g_{ij}}\right)^{\frac{1}{2}} \exp\left[\frac{1}{2}\left(E_j(\theta) - E_i(\theta)\right)\right]. \tag{D-6}$$

$\mathbf{\Gamma}$ is now populated as

$$\begin{aligned} r_{ij} &\sim \text{rand}\,[0,1) && \text{(D-7)}\\ \Gamma_{ij} &= \left\{ \begin{array}{cc} -\sum_{k\neq i}\Gamma_{ki} & i=j \\ F_{ij} & r_{ij} < g_{ij} \text{ and } i \neq j \\ 0 & r_{ij} \geq g_{ij} \text{ and } i \neq j \end{array}\right. . && \text{(D-8)}\end{aligned}$$

Similarly, its average value can be written as

$$\begin{aligned}\langle \Gamma_{ij} \rangle &= g_{ij} \left(\frac{g_{ji}}{g_{ij}}\right)^{\frac{1}{2}} \exp\left[\frac{1}{2}\left(E_j(\theta) - E_i(\theta)\right)\right] && \text{(D-9)}\\ &= \left(g_{ij} g_{ji}\right)^{\frac{1}{2}} \exp\left[\frac{1}{2}\left(E_j(\theta) - E_i(\theta)\right)\right]. && \text{(D-10)}\end{aligned}$$

So, we can use any connectivity scheme $\mathbf{g}$ in learning. We just need to scale the non-zero, non-diagonal entries in $\mathbf{\Gamma}$ by $\left(\frac{g_{ji}}{g_{ij}}\right)^{\frac{1}{2}}$ so as to compensate for the biases introduced by the connectivity scheme.

The full MPF objective function in this case is

$$K = \sum_{j \in \mathcal{D}} \sum_{i \notin \mathcal{D}} g_{ij} \left(\frac{g_{ji}}{g_{ij}}\right)^{\frac{1}{2}} \exp\left[\frac{1}{2}\left(E_j - E_i\right)\right] \tag{D-11}$$

where the inner sum is found by averaging over samples from g_{ij}.

E Continuous state space learning with the connectivity function set via Hamiltonian Monte Carlo

Choosing the connectivity matrix g_{ij} for Minimum Probability Flow Learning is relatively straightforward in systems with binary or discrete state spaces. Nearly any nearest neighbor style scheme seems to work quite well. In continuous state spaces $\mathbf{q} \in \mathbb{R}^d$ however, connectivity functions $g\left(\mathbf{q}_i, \mathbf{q}_j\right)$ based on nearest neighbors prove insufficient. For instance, if the non-zero entries in $g\left(\mathbf{q}_i, \mathbf{q}_j\right)$ are drawn from an isotropic Gaussian centered on $\mathbf{q}_j$, then several hundred non-zero $g\left(\mathbf{q}_i, \mathbf{q}_j\right)$ are required for every value of $\mathbf{q}_j$ in order to achieve effective parameter estimation in some fairly standard problems, such as receptive field estimation in Independent Component Analysis [Bell AJ(1995)].

Qualitatively, we desire to connect every data state $\mathbf{q}_j \in \mathcal{D}$ to the non data states $\mathbf{q}_i$ which will be most informative for learning. The most informative states are those which have high probability under the model distribution $p^{(\infty)}\left(\mathbf{q}\right)$. We therefore propose to populate $g\left(\mathbf{q}_i, \mathbf{q}_j\right)$ using a Markov transition function for the model distribution. Borrowing techniques from Hamiltonian Monte Carlo [Neal(2010)] we use Hamiltonian dynamics in our transition function, so as to effectively explore the state space.

E.1 Extending the state space

In order to implement Hamiltonian dynamics, we first extend the state space to include auxiliary momentum variables.

The initial data and model distributions are $p^{(0)}\left(\mathbf{q}\right)$ and

$$p^{(\infty)}\left(\mathbf{q}; \theta\right) = \frac{\exp\left(-E\left(\mathbf{q}; \theta\right)\right)}{Z\left(\theta\right)}. \tag{E-1}$$

with state space $\mathbf{q} \in \mathbb{R}^d$. We introduce auxiliary momentum variables $\mathbf{v} \in \mathbb{R}^d$ for each state variable $\mathbf{q}$, and call the extended state space including the momentum variables $\mathbf{x} = \{\mathbf{q}, \mathbf{v}\}$. The momentum variables are given an isotropic gaussian distribution,

$$p\left(\mathbf{v}\right)=\frac{\exp\left(-\frac{1}{2}\mathbf{v}^{T}\mathbf{v}\right)}{\sqrt{2\pi}},\tag{E-2}$$

and the extended data and model distributions become

$$\begin{aligned}
p^{(0)}\left(\mathbf{x}\right) &= p^{(0)}\left(\mathbf{q}\right)p\left(\mathbf{v}\right) &\text{(E-3)}\\
&= p^{(0)}\left(\mathbf{q}\right)\frac{\exp\left(-\frac{1}{2}\mathbf{v}^{T}\mathbf{v}\right)}{\sqrt{2\pi}} &\text{(E-4)}\\
p^{(\infty)}\left(\mathbf{x};\theta\right) &= p^{(\infty)}\left(\mathbf{q};\theta\right)p\left(\mathbf{v}\right) &\text{(E-5)}\\
&= \frac{\exp\left(-E\left(\mathbf{q};\theta\right)\right)}{Z\left(\theta\right)}\frac{\exp\left(-\frac{1}{2}\mathbf{v}^{T}\mathbf{v}\right)}{\sqrt{2\pi}} &\text{(E-6)}\\
&= \frac{\exp\left(-H\left(\mathbf{x};\theta\right)\right)}{Z\left(\theta\right)\sqrt{2\pi}} &\text{(E-7)}\\
H\left(\mathbf{x};\theta\right) &= E\left(\mathbf{q};\theta\right)+\frac{1}{2}\mathbf{v}^{T}\mathbf{v}. &\text{(E-8)}
\end{aligned}$$

The initial (data) distribution over the joint space $\mathbf{x}$ can be realized by drawing a momentum $\mathbf{v}$ from a uniform Gaussian distribution for every observation $\mathbf{q}$ in the dataset $\mathcal{D}$.

E.2 Defining the connectivity function $g\left(\mathbf{x}_i,\mathbf{x}_j\right)$

We connect every state $\mathbf{x}_j$ to all states which satisfy one of the following 2 criteria,

1. All states which share the same position $\mathbf{q}_j$, with a quadratic falloff in $g\left(\mathbf{x}_i,\mathbf{x}_j\right)$ with the momentum difference $\mathbf{v}_i-\mathbf{v}_j$.

2. The state which is reached by simulating Hamiltonian dynamics for a fixed time t on the system described by $H\left(\mathbf{x};\theta_H\right)$, and then negating the momentum. Note that the parameter vector θ_H is used only for the Hamiltonian dynamics.

More formally,

$$\begin{aligned}
g\left(\mathbf{x}_i,\mathbf{x}_j\right) = {}& \delta\left(\mathbf{q}_i-\mathbf{q}_j\right)\exp\left(-\left|\left|\mathbf{v}_i-\mathbf{v}_j\right|\right|_2^2\right)\\
&+\delta\left(\mathbf{x}_i-\mathrm{HAM}\left(\mathbf{x}_j;\theta_H\right)\right)
\end{aligned}\tag{E-9}$$

where if $\mathbf{x}'=\mathrm{HAM}\left(\mathbf{x};\theta_H\right)$, then $\mathbf{x}'$ is the state that results from integrating Hamiltonian dynamics for a time t and then negating the momentum. Because of the momentum negation, $\mathbf{x}=\mathrm{HAM}\left(\mathbf{x}';\theta_H\right)$, and $g\left(\mathbf{x}_i,\mathbf{x}_j\right)=g\left(\mathbf{x}_j,\mathbf{x}_i\right)$.

E.3 Discretizing Hamiltonian dynamics

It is generally impossible to **exactly** simulate the Hamiltonian dynamics for the system described by $H(\mathbf{x};\theta_H)$. However, if $\mathrm{HAM}(\mathbf{x};\theta_H)$ is set to simulate Hamiltonian dynamics via a series of leapfrog steps, it retains the important properties of reversibility and phase space volume conservation, and can be used in the connectivity function $g(\mathbf{x}_i,\mathbf{x}_j)$ in Equation E-9. In practice, therefore, $\mathrm{HAM}(\mathbf{x};\theta_H)$ involves the simulation of Hamiltonian dynamics by a series of leapfrog steps.

E.4 MPF objective function

The MPF objective function for continuous state spaces and a list of observations $\mathcal{D}$ is

$$K(\theta;\mathcal{D},\theta_H) = \sum_{\mathbf{x}_j\in\mathcal{D}} \int g(\mathbf{x}_i,\mathbf{x}_j) \exp\left(\frac{1}{2}\left[H(\mathbf{x}_j;\theta) - H(\mathbf{x}_i;\theta)\right]\right) d\mathbf{x}_i. \tag{E-10}$$

For the connectivity function $g(\mathbf{x}_i,\mathbf{x}_j)$ given in Section E.2, this reduces to

$$\begin{aligned} K(\theta;\mathcal{D},\theta_H) = & \\ & \sum_{\mathbf{x}_j\in\mathcal{D}} \int \exp\left(-\left|\left|\mathbf{v}_i-\mathbf{v}_j\right|\right|_2^2\right) \\ & \qquad \exp\left(\frac{1}{2}\left[\frac{1}{2}\mathbf{v}_j^T\mathbf{v}_j - -\frac{1}{2}\mathbf{v}_i^T\mathbf{v}_i\right]\right) d\mathbf{v}_i \\ & + \sum_{\mathbf{x}_j\in\mathcal{D}} \exp\left(\frac{1}{2}\left[H(\mathbf{x}_j;\theta) - H(\mathrm{HAM}(\mathbf{x}_j;\theta_H);\theta)\right]\right). \end{aligned} \tag{E-11}$$

Note that the first term does not depend on the parameters θ, and is thus just a constant offset which can be ignored during optimization. Therefore, we can say

$$\begin{aligned} K(\theta;\mathcal{D},\theta_H) \sim & \\ & \sum_{\mathbf{x}_j\in\mathcal{D}} \exp\left(\frac{1}{2}\left[H(\mathbf{x}_j;\theta) - H(\mathrm{HAM}(\mathbf{x}_j;\theta_H);\theta)\right]\right). \end{aligned} \tag{E-12}$$

Parameter estimation is performed by finding the parameter vector $\hat{\theta}$ which minimizes the objective function $K(\theta;\mathcal{D},\theta_H)$,

$$\hat{\theta} = \underset{\theta}{\mathrm{argmin}}\, K(\theta;\mathcal{D},\theta_H). \tag{E-13}$$

E.5 Iteratively improving the objective function

The more similar θ_H is to θ, the more informative $g(\mathbf{x}_i, \mathbf{x}_j)$ is for learning. If θ_H and θ are dissimilar, then many more data samples will be required in $\mathcal{D}$ to effectively learn. Therefore, we iterate the following procedure, which alternates between finding the $\hat{\theta}$ which minimizes $K(\theta; \mathcal{D}, \theta_H)$, and improving θ_H by setting it to $\hat{\theta}$,

1. Set $\hat{\theta}^{t+1} = \text{argmin}_\theta K(\theta; \mathcal{D}, \theta_H^t)$
2. Set $\theta_H^{t+1} = \hat{\theta}^{t+1}$

$\hat{\theta}^t$ then represents a steadily improving estimate for the parameter values which best fit the model distribution $p^{(\infty)}(\mathbf{q}; \theta)$ to the data distribution $p^{(0)}(\mathbf{q})$, described by observations $\mathcal{D}$. Practically, step 1 above will frequently be truncated early, perhaps after 10 or 100 L-BFGS gradient descent steps.

References

[Bell AJ(1995)] Bell AJ, Sejnowski TJ. An information-maximization approach to blind separation and blind deconvolution. *Neural Computation 1995; vol. 7:1129-1159*, 1995.

[Hyvärinen(2005)] Hyvärinen, A. Estimation of non-normalized statistical models using score matching. *Journal of Machine Learning Research*, 6:695–709, 2005.

[Hyvärinen(2007a)] Hyvärinen, A. Connections between score matching, contrastive divergence, and pseudolikelihood for continuous-valued variables. *IEEE Transactions on Neural Networks*, Jan 2007a.

[Hyvärinen(2007b)] Hyvärinen, A. Some extensions of score matching. *Computational statistics & data analysis*, 51(5):2499–2512, 2007b. ISSN 0167-9473.

[Lyu(2009)] Lyu, S. Interpretation and generalization of score matching. *The proceedings of the 25th conference on uncerrtainty in artificial intelligence (UAI*90)*, 2009.

[Movellan(2008)] Movellan, J R. A minimum velocity approach to learning. *unpublished draft*, Jan 2008.

[Neal(2010)] Neal, Radford M. Mcmc using hamiltonian dynamics. *Handbook of Markov Chain Monte Carlo*, Jan 2010. sections 5.2 and 5.3 for langevin dynamics.

[Sohl-Dickstein & Olshausen(2009)Sohl-Dickstein and Olshausen] Sohl-Dickstein, J and Olshausen, B. A spatial derivation of score matching. *Redwood Center Technical Report*, 2009.