Efficient Feature Learning Using Perturb-and-MAP

Ke Li, Kevin Swersky, Richard Zemel
Dept. of Computer Science, University of Toronto
{keli,kswersky,zemel}@cs.toronto.edu

Abstract

Perturb-and-MAP [1] is a technique for efficiently drawing approximate samples from discrete probabilistic graphical models. These samples are useful for both characterizing the uncertainty in the model, as well as learning its parameters. In this work, we show that this same technique is effective at learning features from images using graphical models with complex dependencies between variables. In particular, we apply this technique in order to learn the parameters of a latent-variable model, the restricted Boltzmann machine, with additional higher-order potentials. We also use it in a bipartite matching model to learn features that are specifically tailored to tracking image patches in video sequences. Our final contribution is the proposal of a novel method for generating perturbations.

1 Introduction

Probablistic graphical models provide a natural and powerful way to represent uncertainty about predictions; unfortunately, sampling from all but the simplest models is often computationally expensive and thus makes learning particularly challenging. The difficulty arises from the fact that directly computing the partition function is usually intractable. A common solution is to use Markov chain Monte Carlo methods; however, in practice, Markov chains may converge slowly, and as a result these methods may be too inefficient to be used during learning.

On the other hand, for many classes of models, efficient discrete optimization algorithms have been developed to perform exact MAP inference (i.e., finding the most probable configuration of random variables), even though computing the partition function is intractable. If such optimization algorithms can be leveraged to perform sampling, efficient sampling algorithms can be potentially obtained for many classes of models where efficient MAP inference can be performed. Motivated by this goal, Papandreou and Yuille [1] proposed a method called Perturb-and-MAP that can perform one-shot approximate sampling from discrete-label Markov random fields (MRFs) using existing MAP inference algorithms. It works by perturbing the entries in potential tables by random noise and then takes the MAP configuration based on the perturbed potentials as a sample.

Perturb-and-MAP has been successfully demonstrated to be effective for pairwise Markov random fields, however to our knowledge its application to other kinds of graphical models has been limited. There are two possible reasons for this: it is not always obvious how to design the perturbations and MAP inference is often itself quite difficult.

In this paper, we apply the Perturb-and-MAP framework to graphical models with latent variables and high-order potentials. Our motivating theme is using these to learn features from images. The first model we consider is the cardinality restricted Boltzmann machine (CaRBM) [2] and the second is a bipartite matching model for tracking image patches in video sequences. We further show how for these models and others, a new class of perturbations can be designed.

2 Background

Perturb-and-MAP is a sampling method that leverages existing optimization algorithms for performing MAP inference. If MAP inference can be performed efficiently, Perturb-and-MAP can leverage this to draw approximate samples. In the context of discrete-label MRFs, Perturb-and-MAP works by perturbing potentials with random noise, and then performing MAP inference on the model with perturbed potentials.

At its core, Perturb-and-MAP relies on the following fact:

Fact. *If* $\epsilon_1, \ldots, \epsilon_n \sim$ *i.i.d. Gumbel(0,1), then* $\mathrm{P}(a_k + \epsilon_k = \max_i\{a_i + \epsilon_i\}) = \frac{\exp(a_k)}{\sum_i \exp(a_i)}$.

Throughout this paper, $Gumbel(\mu, \beta)$ denotes the maximum form of the Gumbel distribution, whose cdf is given by $e^{-e^{-(x-\mu)/\beta}}$.

It follows from the above fact that if the negative energy of each joint configuration is perturbed with i.i.d. standard Gumbel noise, Perturb-and-MAP yields an exact sample from the MRF. Of course, doing this in practice is intractable, since the number of joint configurations scales exponentially in the number of random variables. As an approximation, all entries in unary potential tables and some entries in pairwise and higher-order potential tables may be perturbed with standard Gumbel noise independently. It has been shown empirically that this reduced-order perturbation produces qualitatively similar results as perturbing the negative energy of each joint configuration.

3 Cardinality Restricted Boltzmann Machines

The cardinality restricted Boltzmann machine [2] is an extension of the restricted Boltzmann machine (RBM) that enforces a sparsity constraint over hidden units. It has been shown that CaRBMs are able to extract features that are more interpretable than those extracted by standard RBMs.

The CaRBM adds a prior ψ_k on the hidden units to the probability distribution defined by the standard RBM:

$$\mathrm{P}(\mathbf{v}, \mathbf{h}) = \frac{1}{\mathcal{Z}} \exp(\mathbf{h}^T W \mathbf{v} + \mathbf{b}^T \mathbf{h} + \mathbf{c}^T \mathbf{v}) \psi_k(\sum_i h_i) \tag{1}$$

where $\psi_k(x)$ is defined to be 1 if $x \leq k$ and 0 otherwise, $\mathbf{v} \in \{0,1\}^D$ and $\mathbf{h} \in \{0,1\}^F$ are binary random variables representing visible and hidden units respectively, $W \in \mathbb{R}^{F \times D}, \mathbf{b} \in \mathbb{R}^F, \mathbf{c} \in \mathbb{R}^D$ are model parameters representing weights between all visible and hidden units, biases on hidden units and biases on visible units respectively, and $\mathcal{Z}$ is the partition function.

One of the main challenges in training a CaRBM is in computing $\mathrm{P}(\mathbf{h}|\mathbf{v})$; unlike a standard RBM, each hidden unit in a CaRBM is not conditionally independent of other hidden units given all visible units. Swersky et al. [2] proposed an algorithm for exactly computing $\mathrm{P}(\mathbf{h}|\mathbf{v})$ in $O(kF)$ time, which will be referred to as the exact marginalization algorithm. Using Perturb-and-MAP, we can devise more efficient approximate sampling algorithms by leveraging a plethora of existing selection algorithms. To draw an approximate sample, we first independently perturb the total input to each hidden unit with standard logistic noise and then find the hidden units with the k largest inputs using a selection algorithm. If the prune-and-search selection algorithm [3] is used to perform MAP inference, an approximate sample can be drawn in $O(F)$ time. This is significantly faster than the exact marginalization algorithm, especially for large k.

4 Bipartite Matching

Many problems in artificial intelligence involve finding the correct matching in a bipartite graph. For example, in computer vision, one often needs to find the correct matching between key points in a pair of related images. Solving this problem is central to many applications, such as image stitching, stereo reconstruction and video tracking. For this problem, we are given image patches around each key point in each image, and the ground truth matching between key points in each pair of images in the training set. The aim is to learn a good descriptor for image patches that will facilitate matching in a pair of test frames.

Consider a directed model ϕ parameterized by θ that maps the data to some feature space. Our model defines a probability distribution over match matrices based on the Euclidean distance between points in feature space:

$$\mathrm{P}(M;\theta) = \frac{1}{\mathcal{Z}} \exp(-\frac{1}{2N} \sum_{i,j} m_{ij} \left\|\phi(\mathbf{x}_i;\theta) - \phi(\mathbf{x}'_j;\theta)\right\|_2^2) \prod_j \psi(\sum_i m_{ij}) \prod_i \psi(\sum_j m_{ij}) \quad (2)$$

where $\psi(x)$ is defined to be 1 if $x = 1$ and 0 otherwise, $M \in \{0,1\}^{N\times N}$ are binary random variables representing a match matrix, m_{ij} is entry (i,j) in M representing if ith key point in frame 1 and the jth key point in frame 2 match, and $\mathbf{x}_i \in \mathbb{R}^D$ and $\mathbf{x}'_j \in \mathbb{R}^D$ are vectors representing input data associated with the ith key point in frame 1 and the jth key point in frame 2 respectively.

During training, the log probabilities of the ground truth match matrices are maximized with respect to θ using stochastic gradient ascent. In the case where ϕ is linear, i.e., $\phi(\mathbf{x};W) = W\mathbf{x}$, in order to compute an estimate of the gradient, we need to estimate $\mathbb{E}_M[\sum_{i,j} m_{ij}(W\mathbf{x}_i - W\mathbf{x}'_j)(\mathbf{x}_i - \mathbf{x}'_j)^T]$ with a sample from $\mathrm{P}(M;W)$. As computing the partition function of $\mathrm{P}(M;W)$ is known to be #P-hard, sampling from $\mathrm{P}(M;W)$ is challenging. On the other hand, finding the MAP configuration of M can be done in just $O(N^3)$ time using the Hungarian algorithm. Therefore, if the model is perturbed with noise from the right distribution, we can draw approximate samples in $O(N^3)$ time by leveraging this existing MAP inference algorithm.

5 Designing Perturbations for Sampling Match Matrices

Consider a more general form of the probability distribution defined in equation 2:

$$\mathrm{P}(M) = \frac{1}{\mathcal{Z}} \exp(\sum_{i,j} m_{ij} c_{ij}) \prod_j \psi(\sum_i m_{ij}) \prod_i \psi(\sum_j m_{ij}) \quad (3)$$

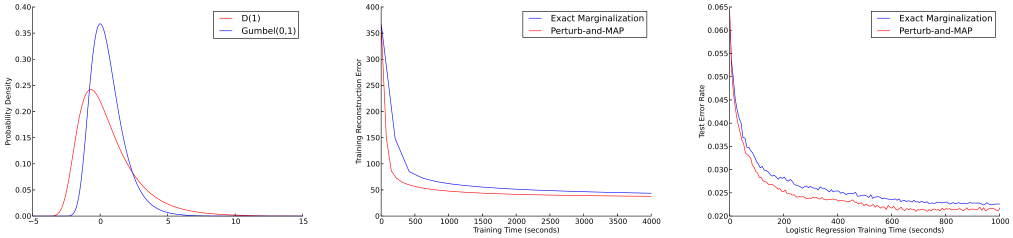
where c_{ij} denotes the compatibility score between the ith key point in frame 1 and the jth key point in frame 2 and all other symbols are as defined in equation 2. Intuitively, the compatibility score between a pair of data examples represents the degree of preference for matching the pair.

The challenge with applying Perturb-and-MAP to this model is in designing perturbations to c_{ij}'s so that the distribution of MAP configurations approximates the model distribution. As noted in [1] and section 2 on the preceding page, perturbing the negative energy of all configurations of M with i.i.d. standard Gumbel noise would yield a procedure that draws exact samples from the model distribution. For Perturb-and-MAP to be tractable, perturbations must be applied to a subset of the $N!$ configurations. One approach is to apply it to the N^2 compatibility scores; however, this introduces dependencies among the perturbed energies of different configurations. Here we consider if it is nevertheless possible to design perturbations so that the negative perturbed energy of each configuration follows the standard Gumbel distribution.

Observe that for any configuration with a positive probability mass, the negative energy can be expressed as $\sum_i c_{i,m(i)}$, where $m(i) = j$ such that $m_{ij} = 1$ (j exists and is unique because any configuration with $\sum_j m_{ij} \neq 1$ has a probability mass of zero). The negative perturbed energy is therefore $\sum_i (c_{i,m(i)} + \epsilon_i)$. Our goal is to find distributions that ϵ_i should be drawn from so that $\sum_i \epsilon_i \sim \mathit{Gumbel}(0,1)$.

We construct the distributions by first finding a distribution $\mathcal{D}(1)$ such that if $X \sim \mathit{Gumbel}(0,1)$, $Y \sim \mathcal{D}(1)$ and $X \perp Y$, $X + Y \sim \mathit{Gumbel}(0,2)$. Since the probability density function (pdf) of a sum of independent random variables is simply the convolution of the pdf's of the individual random variables, we found the pdf of $\mathcal{D}(1)$ numerically by deconvolving the pdf of $\mathit{Gumbel}(0,1)$ out of the pdf of $\mathit{Gumbel}(0,2)$. The pdf of $\mathcal{D}(1)$ is shown in Figure 1a on the next page along with the pdf of $\mathit{Gumbel}(0,1)$ for comparison.

$\mathcal{D}(s)$ is defined in terms of $\mathcal{D}(1)$, with s denoting scale, so that if $Z \sim \mathcal{D}(1)$, $sZ \sim \mathcal{D}(s)$. Since if $X \sim \mathit{Gumbel}(0,1)$, $sX \sim \mathit{Gumbel}(0,s)$, it follows that if $X \sim \mathit{Gumbel}(0,s)$, $Y \sim \mathcal{D}(s)$ and $X \perp Y$, $X + Y \sim \mathit{Gumbel}(0,2s)$. By applying this fact recursively, it is easy to see that if $X \sim \mathit{Gumbel}(0, 2^{-(N-1)})$, $Y_i \sim \mathcal{D}(2^{-i})$ for $i \in \{1,\ldots,N-1\}$ and $X \perp Y_1 \perp \ldots \perp Y_{N-1}$, $X + \sum_i Y_i \sim \mathit{Gumbel}(0,1)$.



(a) (b) (c)

Figure 1: (a) The pdfs of $\mathcal{D}(1)$ and $Gumbel(0, 1)$. (b) Reconstruction errors while training CaRBMs using exact marginalization vs. Perturb-and-MAP. (c) Prediction errors of logistic regression on features extracted by CaRBMs trained using exact marginalization vs. Perturb-and-MAP.

Therefore, if we independently perturb each c_{ij} with noise from $\mathcal{D}(2^{-i})$ for $i \in \{1, ..., N-1\}$ and each c_{Nj} with noise from $Gumbel(0, 2^{-(N-1)})$, the negative perturbed energy of each positive-mass configuration is guaranteed to follow a standard Gumbel distribution. We are currently exploring this approach compared to commonly used logistic or normal perturbations.

6 Experiments

6.1 Cardinality Restricted Boltzmann Machine

We trained CaRBMs with 500 hidden units and a sparsity of 10% (i.e., $k = 50$) on the MNIST handwritten digit dataset using both the exact marginalization method and Perturb-and-MAP to sample the hidden units given the visible units. They are trained using the one-step Contrastive Divergence algorithm [4]. As shown in Figure 1b, training with Perturb-and-MAP is faster and achieves lower reconstruction error than training the same model with exact marginalization.

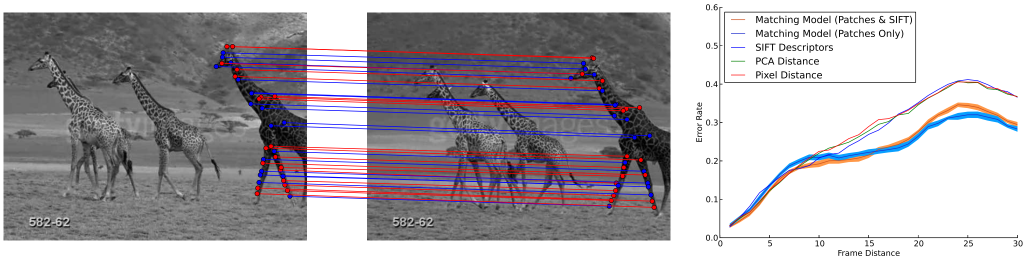
We compare the discriminative capability of the features learned by each method by applying them to a logistic regression classifier. For the exact method, we use the marginals, while for the Perturb-and-MAP approach we approximate the marginals with 50 samples. As shown in Figure 1c, logistic regression achieves lower prediction error when features extracted using the CaRBM trained with Perturb-and-MAP are used, indicating that better features are extracted when the CaRBM is trained with Perturb-and-MAP.

6.2 Bipartite Matching

We trained our linear bipartite matching model on a dataset [5] consisting of frames from a video of giraffes, locations of key points in each frame, and the ground truth matching between the key points in each pair of frames. The task is to predict matching between key points in arbitrary (i.e., not necessarily consecutive) pairs of frames.

Figure 2a on the following page shows two frames from the video along with the key points and ground truth matching. As shown, the patches around different key points are visually similar to each other; as a result, predicting the match from patches is challenging. Furthermore, patches around some key points, like those on the left and right front legs, can be easily confused with one another.

We trained two linear bipartite matching models, one on 65×65 patches and SIFT descriptors centered at key points, and one on patches only. Both models have 2000 features corresponding to patches, and the former also has an additional 128 features corresponding to SIFT descriptors. In the former model, we assume that there are no weight connections between features for SIFT descriptors and the input patches, and vice versa. We initialized the weights with the first 2000 principal components of the patches for both models and all 128 principal components of the SIFT descriptors for the former model. In each iteration of training, we uniformly sampled all frame distances from 5 to 30 and then randomly picked a pair of frames given the sampled frame distance



(a) (b)

Figure 2: (a) Two frames from the giraffe video dataset with the locations of key points and the ground truth matching. (b) Comparison of average test error rates achieved by our linear bipartite matching models, pixel distance, SIFT descriptors and PCA distance on the giraffe video dataset. The shaded areas show three standard deviations from the mean error rates estimated from 20 runs.

as training examples. The gradient was estimated using one approximate sample drawn using the procedure described in section 5 on page 3.

We compare our models to three benchmarks, pixel distance, SIFT descriptors, and PCA distance. For pixel distance and SIFT descriptors, matching is predicted based on the Euclidean distance between patches and SIFT descriptors of pairs of key points respectively. For PCA distance, patches of key points are projected onto the first 2000 principal components and their corresponding SIFT descriptors are projected onto all 128 principal components. Matching is predicted based on the Euclidean distance in this new space.

As shown in Figure 2b, our linear bipartite matching model achieves significantly lower error rates than all benchmarks for pairs of frames that are more than 12 frames apart. This indicates that unlike the benchmarks which mostly predict matching based on visual similarity of entire patches, our models learned which parts of the patches contain important features for matching. For pairs of frames that are less than 8 frames apart, it is not surprising that all methods performed similarly, as visual similarity between entire patches of the same key point is quite high. As a result, even a benchmark as simple as pixel distance can achieve relatively good performance for nearby frames.

References

[1] George Papandreou and Alan L. Yuille. Perturb-and-MAP random fields: Using discrete optimization to learn and sample from energy models. In *International Conference on Computer Vision*, pages 193–200, 2011.

[2] Kevin Swersky, Danny Tarlow, Ilya Sutskever, Ruslan Salakhutdinov, Rich Zemel, and Ryan Adams. Cardinality restricted boltzmann machines. In *Advances in Neural Information Processing Systems 25*, pages 3302–3310, 2012.

[3] Manuel Blum, Robert W. Floyd, Vaughan R. Pratt, Ronald L. Rivest, and Robert Endre Tarjan. Time Bounds for Selection. *Journal of Computer and System Sciences*, 7:448–461, 1973.

[4] Geoffrey E. Hinton. Training Products of Experts by Minimizing Contrastive Divergence. *Neural Computation*, 14:1771–1800, 2002.

[5] David A. Ross, Daniel Tarlow, and Richard S. Zemel. Learning Articulated Structure and Motion. *International Journal of Computer Vision*, 88:214–237, 2010.