Scaling up Structured Multi-Label Prediction using Discriminative Mean Field Networks

David Belanger
UMass Amherst
belanger@cs.umass.edu

Andrew McCallum
UMass Amherst
mccallum@cs.umass.edu

Abstract

Multi-label classification is an important task in many modern machine learning applications. Accurate methods model the correlations and relationships between labels, either by assuming a low-dimensional embedding of the labels or a graph structure of label dependencies. While such interactions can be achieved using feed-forward predictors, problems with tight coupling between labels are better posed as structured prediction problems. Unfortunately, prior applications of graphical models to multi-label classification scale poorly. In response, we introduce *discriminative mean field networks*, an iterative structured prediction technique applicable to substantially larger label sets. We employ a deep architecture to define an *energy function* of candidate labels, and form predictions using backpropagation to iteratively optimize the energy with respect to the labels. This deep architecture captures dependencies between labels that would lead to completely intractable graphical models, and enables a form of *structure learning* by automatically learning discriminative features of the structured output. The technique is effective on a variety of benchmarks, and generalizes easily to other structured prediction applications.

1 Introduction

Multi-label classification is an important task in a variety of machine learning applications. The data consist of $\{x, y\}$ pairs, where $y = \{y_1, \ldots, y_L\} \in \{0, 1\}^L$ is a set of multiple binary labels we seek to predict and x is a feature vector. In many cases, we are given no structure among the L labels a-priori, though the labels may be quite correlated.

The most simple multi-label classification approach is to independently predict each label y_i using a separate classifier, also known as the 'binary relevance model' (Tsoumakas & Katakis, 2006). This can perform poorly, particularly when certain labels are rare and the labels are highly-correlated. Improved training can be achieved using max-margin or ranking losses that directly address the multi-label structure (Elisseeff & Weston, 2001; Godbole & Sarawagi, 2004; Zhang & Zhou, 2006; Bucak et al., 2009). Alternatively, model capacity can be controlled, and correlated predictions can be achieved, by enforcing low-rank structure in the model's parameters, eg. (Ji & Ye, 2009; Cabral et al., 2011; Yu et al., 2014; Xu et al., 2014). While the prediction cost of such methods grows linearly with L, there is a limit to how expressive these low-rank models can be, particularly when there are strict structural constraints among labels, such as mutual exclusivity and implicature.

Therefore, it is natural to instead approach multi-label classification using *structured prediction* methods, which model interactions between prediction labels directly. The drawback of such techniques, however, is that their computational complexity typically either grows super-linearly in L (Finley & Joachims, 2008; Meshi et al., 2010; Petterson & Caetano, 2011) , or requires practitioners to impose strict assumption about the dependencies between labels (Read et al., 2011; Jasinska & Dembczyski, 2015; Niculescu-Mizil & Abbasnejad, 2015). This prevents scalability to large label spaces with complex interaction structure.

In response, this paper contributes a new structured prediction method, *discriminative mean field networks* (DMFNs) that scales linearly in L and makes no assumptions a-priori about the relationship between the labels, using a deep neural network to encode an *energy function* on candidate outputs. At test time, prediction is performed by approximately minimizing the energy with respect to the prediction variables using gradient descent, where gradients are obtained by backpropagation through the deep architecture. The parameters of the network are trained using an adaptation of a structured SVM (Taskar et al., 2004; Tsochantaridis et al., 2004). The deep network allows us to model high-arity interactions that would result in unmanageable treewidth if the problem was posed as a graphical model.

When practitioners choose among structured prediction techniques, they face a tradeoff between expressivity of the prediction function and algorithmic guarantees of the prediction and learning procedures. We embrace the first, sacrificing the latter. On a variety of benchmark multi-label classification tasks, we demonstrate that the expressivity of our deep energy function provides accuracy improvements against a variety of competitive baselines and discuss the latter considerations. Inspired by the success of neural networks to automatically discover salient features of the inputs, we apply DMFNs to automatically discover salient features of the input labels, providing a novel technique for structure learning. In general, we recommend further exploration of our technique, and its generalization to additional structured prediction problems.

2 Discriminative Mean-Field Networks

A fully-general way to specify the set of all $x \rightarrow y$ mappings is to pose y as the solution to a potentially non-linear combinatorial optimization problem, with parameters dependent on x:

$$\min_{y} \quad \mathrm{E}_{\mathrm{x}}(y) \quad \text{subject to} \qquad y \in \{0,1\}^L. \tag{1}$$

The structured prediction problem (1) could be rendered tractable by assuming certain specific structure for the 'energy function' $E_x()$, such as a tree-structured undirected graphical model. Instead, we consider general $E_x()$, but optimize over a convex relaxation of the constraint set:

$$\min_{y} \quad \mathrm{E}_{\mathrm{x}}(\bar{y}) \quad \text{subject to} \qquad \bar{y} \in [0,1]^L. \tag{2}$$

In general, $E_x(\bar{y})$ may be non-convex, so exactly solving (2) may be intractable. A reasonable approximate optimization procedure, however, is to minimize (2) via gradient descent, obtaining a local minimum. Optimization over the set $[0,1]^L$ can be performed using entropic mirror descent (aka exponentiated gradient) by normalizing over each coordinate (Beck & Teboulle, 2003).

There are no guarantees that our output $\bar{y}$ values are nearly 0-1. To obtain outputs, we may round. In other applications, it is sometimes useful to have 'soft' predictions, eg. for detection problems, since we may want to threshold based on confidence.

We refer to the relaxation from y to $\bar{y}$ as a *mean-field* formulation of the problem, inspired by such factorizations in the approximate posterior inference literature, where $\bar{y}_i$ would be interpreted as the marginal probability that $y_i = 1$. We make no such probabilistic assumptions, however. We use the term 'discriminative' because a primary difference between this work and prior mean-field techniques is that we directly parametrize the objective that the mean field inference procedure optimizes, rather than parametrizing a probabilistic model for which inference is intractable, and inducing a mean-field objective when we seek to perform approximate variational inference.

This approach based on continuous optimization can be performed using black-box access to a gradient subroutine for $E_x(\bar{y})$. Therefore, it is natural to parametrize $E_x(\bar{y})$ using a deep architecture, since deep learning libraries provide modular tools to differentiate very complex functions with sophisticated parameter tying, etc.

3 Network Architecture

We parametrize $E_x(\bar{y})$ as a neural network that takes both x and $\bar{y}$ as inputs and returns a single number. In the following, we denote matrices in upper case and vectors in lower case. We use $g()$

to denote a coordinate-wise non-linearity function. We may use different functions, eg. sigmoid vs. rectifier, in different places.

The *feature network* takes x and returns a compressed F-dimensional feature representation. We employ a simple multi-layer perceptron:

$$f(x) = g(A_2 g(A_1 x)) \tag{3}$$

The *local energy network*, ignores any interaction between coordinates of $\bar{y}$ and scores $\bar{y}$ as if it was the sum of the scores of L linear models.

$$E_x^{\text{local}}(\bar{y}) = \sum_{i=1}^{L} \bar{y}_i b_i^\top f(x) \tag{4}$$

Here, each b_i is an F dimensional vector of parameters for every label. If $f(x) = x$, then (4) corresponds to L independent per-label generalized linear models. If $f(x) = Cx$ performs linear dimensionality reduction, then this energy corresponds to a generalized linear model with a low-rank weight matrix, a popular technique in multi-label classification (Ji & Ye, 2009; Cabral et al., 2011; Yu et al., 2014; Xu et al., 2014).

The total energy is the sum of the output of the local energy network and the *label energy network*, which scores configurations of $\bar{y}$ independent of x:

$$E_x^{\text{label}}(\bar{y}) = c_2^\top g(C_1 \bar{y}). \tag{5}$$

We interpret the product $C_1\bar{y}$ as a set of learned linear measurements of the output, which allow the practitioner to avoid imposing any structure a-priori on the interaction structure between the labels in y. Computing such a product is linear in L and provides a method of automatic structure learning. Such measurements also appear in compressed sensing approaches error-correcting code to multi-label classification (Hsu et al., 2009; Hariharan et al., 2010; Kapoor et al., 2012), which rely on assumptions about the sparsity of the true labels or prior knowledge about label interactions, and often do not learn the measurement matrix from data.

Note that the energy only depends on x via the value of $f(x)$. During iterative prediction, we improve efficiency by precomputing $f(x)$ and not back-propagating through f when differentiating the energy with respect to $\bar{y}$. Also note that certain choices of g result in a convex prediction problem. In practice, however, we found it was best to select g based on resulting model accuracy rather than any algorithmic guarantees resulting from convexity.

In future work, it would be natural to use a *conditional label energy network*, which would be similar to the label energy network, but concatenates $\bar{y}$ with the output of the feature network:

$$E_x^{\text{cond}}(\bar{y}) = d_2^\top g(D_1[\bar{y}; f(x)]) \tag{6}$$

There are important parallels between the above energy networks and the parametrization of a conditional random field (CRF) (Lafferty et al., 2001; Sutton & McCallum, 2011). For the sake of notational simplicity, consider a fully-connected pairwise CRF with local potentials that depend on x, but data-independent pairwise potentials. Let vec() flatten a matrix into a vector. The corresponding label energy net would be:

$$E_x^{\text{crf}}(\bar{y}) = s_2^\top \text{vec}(\bar{y}\bar{y}^\top), \tag{7}$$

In applications with large label spaces,(7) is troublesome in terms of both the statistical efficiency of parameter estimation and the computational efficiency of prediction because of the quadratic dependence on L. Statistical issues can be mitigated by imposing parameter tying of the CRF potentials, using a low-rank assumption, eg. (Srikumar & Manning, 2014; Jernite et al., 2015), or using a deep architecture to map x to a table of CRF potentials (LeCun et al., 2006). The computational concerns of a pairwise CRF, namely the quadratic dependence on L, can be mitigated by choosing a sparse graph. This is difficult for practitioners when they do not know the dependencies between labels a-priori. Furthermore, CRFs pose an even steeper computational burden when modeling high-order interactions than pairwise relationships between labels, while DMFNs do not.

4 Learning

In Section 2, we described a technique for producing predictions by performing continuous optimization in the space of outputs. Now, we discuss a gradient-based technique for learning the parameters of the deep architecture $E_x(\bar{y})$.

In many structured prediction applications, the practitioner is able to interact with the model in only two ways: (1) evaluate the model's energy on a given value of y, and (2) minimize the energy with respect to the y. This occurs, for example, when predicting combinatorial structures such as bipartite matchings and graph cuts. A very popular technique in these settings is the structured support vector machine (SSVM) (Taskar et al., 2004; Tsochantaridis et al., 2004).

If we assume (incorrectly) that our prediction procedure is not subject to optimization errors, then (1) and (2) apply to our model and it is straightforward to train using an SSVM. This ignores errors resulting from the potential non-convexity of $E_x(\bar{y})$ or the relaxation from y to $\bar{y}$. However, such an assumption is a reasonable way to construct an approximate learning procedure.

Define $\Delta(y_p, y_g)$ to be an error function between a prediction y_p and the ground truth y_g, such as the Hamming loss. Let Ψ denote the parameters of E_x. Let $[\cdot]_+ = \max(0, \cdot)$. The SSVM minimizes the training objective

$$L(\Psi) = \sum_{\{x_i, y_i\}} \max_y \left[\Delta(y_i, y) - E_{x_i}(y) + E_{x_i}(y_i)\right]_+ . \tag{8}$$

Note that the signs in (8) differ from convention because here prediction minimizes $E_x()$. We minimize our loss with respect to the parameters of the deep architecture E_x using mini-batch stochastic gradient descent. For a given $\{x_i, y_i\}$, the subgradient of 8 is:

$$\nabla_\Psi L(\Psi) = I\left[\Delta(y_i, y_\text{p}) - E_{x_i}(y_\text{p}) + E_{x_i}(y_i) > 0\right]\left(-\nabla_\Psi E_{x_i}(y_\text{p}) + \nabla_\Psi E_{x_i}(y_i)\right) \tag{9}$$

Here, $I[\cdot]$ is an indicator function for a predicate, and y_p is the output of *loss-augmented inference*:

$$y_p = \arg\min_y \left(-\Delta(y_i, y) + E_{x_i}(y)\right). \tag{10}$$

With this, (9) can be computed using back-propagation through E_x.

We perform loss-augmented inference by again using gradient descent on the relaxation $\bar{y}$, rather than performing combinatorial optimization over y. Since Δ is a discrete function such as the Hamming loss, we need to approximate it with a differentiable surrogate, such as the squared loss. Any surrogate loss used for training a feed-forward predictor with gradient descent can be used here. Note that the objective (8) only considers the energy values of the ground truth and the prediction, ensuring that they're separated by a margin, not the actual ground truth and predicted labels (10). Therefore, we do not round the output of (10) in order to approximate a subgradient of (8); instead, we evaluate the energy directly on the $\bar{y}$ obtained by approximately minimizing (10).

Finally, we found that it was useful to initialize the parameters of the feature network by first training them using a simple binary classification loss, ignoring any interactions between coordinates of y. For problems with limited training data, we keep the parameters of the feature network fixed when optimizing the label energy network parameters.

See Section 9 for a discussion of various implementation-level details used to improve the efficiency of DMFNs in practice.

5 Related Work

Our use of backprogation to perform prediction, by iteratively changing the inputs to the network, differs from most deep learning applications, where backpropagation is used to update the network parameters. However, such an approach has been useful in a variety of deep learning applications, including *siamese networks* (Bromley et al., 1993), methods for generating adversarial examples (Szegedy et al., 2014; Goodfellow et al., 2014), methods for embedding documents as dense vectors (Le & Mikolov, 2014), and successful techniques for image generation and texture synthesis (Mordvintsev et al., 2015; Gatys et al., 2015a,b).

Missing Data	Yes		No			
Method	SP	PR	BR	LR	MLP	DMFN
Bibtex	43.9	44.2	37.2	39.0	38.9	41.6
Delicious	29.0	33.3	26.5	35.3	37.0	35.2
Bookmarks	34.6	34.9	30.7	31.0	33.8	34.4

Table 1: Comparison of various methods on 3 standard datasets in terms of F1 measure (larger is better). The left 2 methods model learning as a missing data problem, and do not assume that un-annoted labels are negative. The right 4 do no such modeling.

In concurrent work, (Carreira et al., 2015) propose an iterative structured prediction method for human pose estimation, where $E_x(y)$, doesn't return a number, but instead an increment $\Delta(x, y)$. Predictions are constructed by incrementally stepping as $y_{t+1} = y_t + \Delta(x, y_t)$. The network for Δ is trained as a multi-variate regression task, by defining a ground truth trajectory for target values for intermediate y_t. It is unclear how to best generalize this construction of intermediate target values to our application.

There is a rich body of work on using models with low-rank parameters matrices for multi-label classification, eg. (Ji & Ye, 2009; Cabral et al., 2011; Yu et al., 2014; Xu et al., 2014). By using a multi-layer perceptron (MLP) for the feature network with hidden layers of lower dimensionality than the input, we are able to capture similar low-dimensional structure. In our experiments, an MLP is a very competitive baseline.

Many successful multi-label classification methods approach learning as a missing data problem. Here, the training labels y are assumed to be correct only when $y_i = 1$. When $y_i = 0$, this is treated as missing data, whose values can be imputed using assumptions about the rank (Lin et al., 2014) or sparsity (Bucak et al., 2011; Agrawal et al., 2013) of the matrix of training labels. It is possible that some of these insights could be applied to our approach, and we leave this for future work.

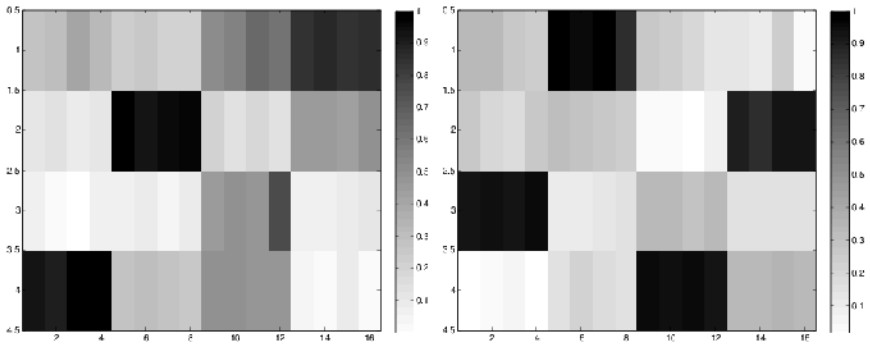
A natural alternative to DMFNs for multi-label prediction is to encode $E_x(y)$ as a conditional random field (CRF) (Ghamrawi & McCallum, 2005; Finley & Joachims, 2008; Meshi et al., 2010). CRF inference is exponential in the treewidth of the graph, whereas the measurements employed by DMFNs can extract information from arbitrarily many labels at once. Consequently, previous applications of CRFS to multi-label classification have only considered very small label spaces. While the per-iteration complexity of DMFN prediction is superior to CRFs of comparable expressivity, it is difficult to analyze its overall cost compared to CRF inference, eg. using belief propagation, because both perform non-convex optimization.

Training CRFs using an SSVM loss is conceptually more attractive than training DMFNs, however. In loopy graphical models, it is tractable to solve the LP relaxation of MAP inference, using graph-cuts or message passing techniques. Solving the LP relaxation, instead of performing exact MAP inference, in the inner loop of SSVM learning is fairly benign, since it is guaranteed to over-generate margin violations in (8). A chief concern, in both theory and practice, when training a DMFN with an SSVM is that the non-convex optimization in the inner loop of learning will find poor local minima such that no margin violations in (8) are discovered (Kulesza & Pereira, 2007; Finley & Joachims, 2008). Since parameter updates (9) only occur when margin violations are discovered, this halts the learning process.

6 Experiments

6.1 Evaluation on Multi-Label Classification Benchmarks

Table 1 compares DMFNs to a variety of high-performing baselines on a selection of standard multi-label classification tasks (Tsoumakas & Katakis, 2006). Dataset sizes, etc. are described in Table 4. We compare **BR**: independent per-label logistic regression, ie. the 'binary relevance model' Tsoumakas & Katakis (2006). **MLP**: multi-layer perceptron with ReLU non-linearities trained with per-label logistic loss, ie the 'feature network' coupled with the local energy network (4) of Section 3. **LR**: the low-rank-weights method of Yu et al. (2014). **SP**: sparsity-based technique for handling negative training labels, along the lines of Bucak et al. (2011) and Agrawal et al. (2013).



(a) ReLU (b) HardTanh

Figure 1: Learned DMFN measurement matrices on synthetic data containing mutual exclusivity of labels within size-4 blocks, for two different choices of nonlinearity in the label energy network. 16 Labels on horizontal axis and 4 hidden units on vertical axis. Measurements that place equal weight on every element in a block can be used to detect violations of the data's mutual exclusivity constraints. When using ReLU, which acts as the identity for positive activations, violations of the data constraints can be detected by taking linear combinations of the measurements (a). This obfuscates our ability to perform structure learning by investigating the measurement matrix. On the other hand, since applying HardTanh to measurements saturates from above, the network learns to utilize each measurement individually, yielding substantially more accurate structure learning.

Here, negative annotated labels are treated as missing data, with a sparsity-inducing penalty on their values. **PR**: posterior-regularization technique for handling negative training labels by placing a low-rank regularizer on the matrix of inferred training labels (Lin et al., 2014) . All results besides **MLP** and **DMFN**, are taken from Lin et al. (2014). We report the 'example averaged' F1 measure. For Bibtex and Delicious, we tuned parameters by first jack-knifing a separate train-test split. For Bookmarks, there is no official train-dev-test split, but we used the same one as Lin et al. (2014). For DMFNs, we obtained predictions by rounding $\bar{y}_i$ above a threshold, which was tuned on held-out data.

We split the approaches in Table 1 into those that explicitly model learning as a missing data problem and those that do not. Since PR is effectively the LR model, but with extra modeling of the missing training data, this suggests that such modeling is important for the Bibtex and Bookmarks datasets. Such an approach is somewhat orthogonal to the modeling of DMFNs, and ideally one would combine missing data techniques with DMFNs.

There are a few key results in Table 1. First, DMFNs are very competitive, particularly against techniques that do not model missing data. Second, MLP, a technique that has not been treated as a baseline in recent literature, performs quite well. Finally, the MLP outperformed DMFN on the Delicious dataset. We found that this is because the MLP, trained with logistic regression, is better at predicting soft predictions, to be combined with a confidence threshold, than DMFNs. To obtain competitive results with DMFNs, we actually needed to smooth the test-time prediction problem with extra entropy terms to obtain softer predictions, which we could then threshold.

6.2 Performing Structure Learning Using DMFNs

Next, we perform experiments on synthetic data designed to demonstrate that the label measurement matrix, C_1 in the label energy network (5), provides a useful tool for analyzing the structure of dependencies between labels. The experiments also show that DMFNs excel in regimes of limited training data, due to their superior parsimony compared to analogous feed-forward approaches.

To generate data, we first draw a design matrix X with 64 features, with each entry drawn from $N(0, 1)$. Then, we generate a 64 x 16 weights matrix A, again with entries from $N(0, 1)$. Then, we construct $Z = XA$ and split the 16 columns of Z into 4 consecutive blocks. For each block, we set $Y_{ij} = 1$ if Z_{ij} is the maximum entry in its row-wise block, and 0 otherwise. We seek to learn a model with predictions that reliably obey these within-block mutual exclusivity constraints.

# train examples	Linear	3-Layer MLP	DMFN w/ Linear Local Energy
1.5k	80.0	81.6	**91.5**
15k	81.8	96.3	**96.7**

Table 2: Comparing F1 performance on the synthetic task with block-strucutred mutual exclusivity between labels. Due to its parsimonious parametrization, the DMFN succeeds with limited data. With more data, the MLP performs comparably to the DMFN, suggesting that even rigid structural constraints among labels can be predicted in a feed-forward fashion using a sufficiently expressive architecture.

In the caption of Fig 1 we analyze the ability to automatically the data's constraint structure by analyzing the measurement matrix of the DMFN.

Next, in Table 2 we compare: a linear classifier, a 3-Layer ReLU MLP with hidden units of size 64 and 16, and a DMFN with a simple linear local energy network and a 2-layer label energy network with HardTanh activations and 4 hidden units. Using fewer hidden units in the MLP resulted in substantially poorer performance. We avoid using a non-linear local energy network in the DMFN because we want to force the label energy network to capture all interactions between labels, in order to improve the interpretability of the label measurement matrix for structure learning.

The table provides a variety of illustrative results. First, note that the DMFN consistently outperforms the MLP, particularly when training on only 1.5k examples. This is partly because the MLP has 5x more parameters, and partly because we injected domain knowledge about the constraint structure when designing the label energy network's architecture. In the appendix, Figure 1 demonstrates that we can perform the same structure learning as in Figure 1 on this small training data. Next, observe that for 15k examples the performance of the MLP and DMFN are comparable . Initially, we hypothesized that the mutual exclusivity constraints of the labels could not be satisfied by a feed-forward predictor, and reconciling their interactions would require an iterative procedure. However, it seems that a large, expressive MLP can learn an accurate predictor when presented with lots of examples. Going forward, a useful we would like to investigate the parsimony vs. expressivity tradeoffs of DMFNs and MLPs.

6.3 Analyzing the Effect of Search Errors on SSVM Training

Due to scalability considerations, prior applications of CRFs to multi-label classification have been restricted to substantially smaller L than those considered in Table 1. In Table 3, we consider the 14-label yeast dataset (Elisseeff & Weston, 2001), which is the largest label space fit using a CRF in Finley & Joachims (2008) and Meshi et al. (2010). Finley & Joachims (2008) analyze the effects of inexact prediction on SSVM training and on test-time prediction. Table 3 considers greedy prediction, loopy belief propagation, exact prediction using an ILP solver, solving the LP relaxation, and DMFNs, where the same technique is used at train and test time. All results, besides DMFNs, are from Finley & Joachims (2008), which also considers cases where different methods are used in train vs. test. We report hamming error, using 10-fold cross validation.

A key argument of Finley & Joachims (2008) is that SSVM training is more effective when the train-time inference method will not under-generate margin violations. Here, LBP and DMFN, which both approximately minimize a non-convex inference objective, have such a vulnerability, whereas LP does not, since solving the LP relaxation provides a lower bound on the true solution to the value of (10). Since DMFN performs similarly to EXACT and LP, this suggests that perhaps the effect of inexact prediction is more benign for DMFNs than for LBP. However, DMFNs exhibit alternative expressive power to pairwise CRFs, and thus it is difficult to isolate the effect of SSVM training on accuracy. In future work, we will perform additional experiments to test this.

7 Conclusion and Future Work

Our experiments explore values of L that are very large compared to prior work using structured prediction for multi-label classification. These are not large-scale classification tasks, however. Fortunately, there are ample opportunities for further scaling of DMFNs. First, note that the compu-

GREEDY	LBP	EXACT	LP	DMFN
$21.6 \pm .56$	$24.3 \pm .61$	$20.23 \pm .53$	$20.49 \pm .54$	$20.88 \pm .19$

Table 3: Comparing different prediction methods, which are used both during SSVM training and at test time, using the setup of Finley & Joachims (2008) on Yeast dataset. We report hamming error (smaller is better). DMFNs perform comparably to prediction methods that provide stronger guarantees when used in SSVM training.

tational cost of iterative prediction is currently overkill because much of it is wasted on labels that could have been ruled out using simpler methods. In future work, we will consider a cascade approach, where a preliminary model is used to filter high-confidence positive and negative predictions. Training is computationally expensive because we need to run iterative prediction until convergence for every training example. We would like to leverage methods from the SSVM literature to improve efficiency by interleaving inference and learning, eg. Meshi et al. (2010). Furthermore, since we perform prediction in parallel on GPUs in minibatches, we are subject to the 'curse of the last reducer,' where unnecessary gradient computation is performed on easy examples while we wait for inference on difficult examples to converge. This can be mitigated using smarter low-level code. Alternatively, we can explore special-case prediction procedures that exploit the piecewise-linear nature of energy networks with ReLU activations.

DMFNs parametrize an $x \rightarrow y$ mapping implicitly, through an energy function and a prediction-time optimization procedure. For the sake of prediction speed, it would be much more attractive if such a mapping was specified in a purely feed-forward manner. It is unclear if such a mapping exists, however, when there are tight dependencies between labels, such as mutual exclusivity, or if they need to be reconciled by 'inference.' Lately, iterative variational approaches for posterior inference have been replaced by feed-forward 'inference networks' that directly predict the parameters of a variational distribution(Kingma & Welling, 2014; Rezende et al., 2014). It would be very useful overall to understand the strengths and limitations of iterative vs. feed-forward procedures, as this would illuminate problems where the iterative prediction of DMFNs will be most effective compared to feed-forward approaches.

Finally, we found that DMFN predictions were nearly always spiked at either 0 or 1, despite optimizing a non-convex energy over the set $[0, 1]$. We expect that this results from the energy function being fit to data that is always 0 or 1. We would like to further develop DMFN architectures that fit the data well and also encourage integral predictions in practice.

8 Acknowledgement

This work was supported in part by the Center for Intelligent Information Retrieval and in part by NSF grant #CNS-0958392. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the sponsor.

References

Agrawal, Rahul, Gupta, Archit, Prabhu, Yashoteja, and Varma, Manik. Multi-label learning with millions of labels: Recommending advertiser bid phrases for web pages. In *Proceedings of the 22nd international conference on World Wide Web*, pp. 13–24. International World Wide Web Conferences Steering Committee, 2013.

Beck, Amir and Teboulle, Marc. Mirror descent and nonlinear projected subgradient methods for convex optimization. *Operations Research Letters*, 31(3):167–175, 2003.

Bromley, Jane, Bentz, James W, Bottou, Léon, Guyon, Isabelle, LeCun, Yann, Moore, Cliff, Säckinger, Eduard, and Shah, Roopak. Signature verification using a siamese time delay neural network. *International Journal of Pattern Recognition and Artificial Intelligence*, 7(04):669–688, 1993.

Bucak, Serhat S, Mallapragada, Pavan Kumar, Jin, Rong, and Jain, Anil K. Efficient multi-label ranking for multi-class learning: application to object recognition. In *Computer Vision, 2009 IEEE 12th International Conference on*, pp. 2098–2105. IEEE, 2009.

Bucak, Serhat Selcuk, Jin, Rong, and Jain, Anil K. Multi-label learning with incomplete class assignments. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pp. 2801–2808. IEEE, 2011.

Cabral, Ricardo S, Torre, Fernando, Costeira, João P, and Bernardino, Alexandre. Matrix completion for multi-label image classification. In *Advances in Neural Information Processing Systems*, pp. 190–198, 2011.

Carreira, Joao, Agrawal, Pulkit, Fragkiadaki, Katerina, and Malik, Jitendra. Human pose estimation with iterative error feedback. *arXiv preprint arXiv:1507.06550*, 2015.

Elisseeff, André and Weston, Jason. A kernel method for multi-labelled classification. In *Advances in neural information processing systems*, pp. 681–687, 2001.

Finley, Thomas and Joachims, Thorsten. Training structural svms when exact inference is intractable. In *Proceedings of the 25th international conference on Machine learning*, pp. 304–311. ACM, 2008.

Gatys, Leon A., Ecker, Alexander S., and Bethge, Matthias. A neural algorithm of artistic style. *CoRR*, abs/1508.06576, 2015a. URL `http://arxiv.org/abs/1508.06576`.

Gatys, Leon A., Ecker, Alexander S., and Bethge, Matthias. Texture synthesis using convolutional neural networks. In *Advances in Neural Information Processing Systems 28 (NIPS)*. 2015b.

Ghamrawi, Nadia and McCallum, Andrew. Collective multi-label classification. In *Proceedings of the 14th ACM international conference on Information and knowledge management*, pp. 195–200. ACM, 2005.

Godbole, Shantanu and Sarawagi, Sunita. Discriminative methods for multi-labeled classification. In *Advances in Knowledge Discovery and Data Mining*, pp. 22–30. Springer, 2004.

Goodfellow, Ian J, Shlens, Jonathon, and Szegedy, Christian. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

Hariharan, Bharath, Zelnik-Manor, Lihi, Varma, Manik, and Vishwanathan, Svn. Large scale max-margin multi-label classification with priors. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pp. 423–430, 2010.

Hsu, Daniel, Kakade, Sham, Langford, John, and Zhang, Tong. Multi-label prediction via compressed sensing. In *NIPS*, volume 22, pp. 772–780, 2009.

Jasinska, Kalina and Dembczyski, Krzysztof. Consistent label tree classifiers for extreme multi-label classification. In *ICML 2015 Workshop on Extreme Classification*, 2015.

Jernite, Yacine, Rush, Alexander M., and Sontag, David. A fast variational approach for learning markov random field language models. In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, pp. 2209–2217, 2015.

Ji, Shuiwang and Ye, Jieping. Linear dimensionality reduction for multi-label classification. In *IJCAI*, volume 9, pp. 1077–1082, 2009.

Kapoor, Ashish, Viswanathan, Raajay, and Jain, Prateek. Multilabel classification using bayesian compressed sensing. In *Advances in Neural Information Processing Systems*, pp. 2645–2653, 2012.

Kingma, Diederik P and Welling, Max. Auto-encoding variational bayes. *International Conference on Learning Representations 2014*, 2014.

Kulesza, Alex and Pereira, Fernando. Structured learning with approximate inference. In *Advances in neural information processing systems*, pp. 785–792, 2007.

Lafferty, John D, McCallum, Andrew, and Pereira, Fernando CN. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, pp. 282–289, 2001.

Le, Quoc and Mikolov, Tomas. Distributed representations of sentences and documents. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pp. 1188–1196, 2014.

LeCun, Yann, Chopra, Sumit, Hadsell, Raia, Ranzato, M, and Huang, F. A tutorial on energy-based learning. *Predicting structured data*, 1:0, 2006.

Lin, Victoria (Xi), Singh, Sameer, He, Luheng, Taskar, Ben, and Zettlemoyer, Luke. Multi-label learning with posterior regularization. In *NIPS Workshop on Modern Machine Learning and Natural Language Processing*, 2014.

Meshi, Ofer, Sontag, David, Globerson, Amir, and Jaakkola, Tommi S. Learning efficiently with approximate inference via dual losses. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pp. 783–790, 2010.

Mordvintsev, Alexander, Olah, Christopher, and Tyka, Mike. Inceptionism: Going deeper into neural networks, June 2015. URL `http://googleresearch.blogspot.com/2015/06/inceptionism-going-deeper-into-neural.html`.

Niculescu-Mizil, Alexandru and Abbasnejad, Ehsan. Label filters for large scale multilabel classification. In *ICML 2015 Workshop on Extreme Classification*, 2015.

Petterson, James and Caetano, Tibério S. Submodular multi-label learning. In *Advances in Neural Information Processing Systems*, pp. 1512–1520, 2011.

Read, Jesse, Pfahringer, Bernhard, Holmes, Geoff, and Frank, Eibe. Classifier chains for multi-label classification. *Machine learning*, 85(3):333–359, 2011.

Rezende, Danilo Jimenez, Mohamed, Shakir, and Wierstra, Daan. Stochastic backpropagation and approximate inference in deep generative models. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, pp. 1278–1286, 2014.

Srikumar, Vivek and Manning, Christopher D. Learning distributed representations for structured output prediction. In *Advances in Neural Information Processing Systems*, pp. 3266–3274, 2014.

Sutton, Charles and McCallum, Andrew. An introduction to conditional random fields. *Machine Learning*, 4(4):267–373, 2011.

Szegedy, Christian, Zaremba, Wojciech, Sutskever, Ilya, Bruna, Joan, Erhan, Dumitru, Goodfellow, Ian, and Fergus, Rob. Intriguing properties of neural networks. In *International Conference on Learning Representations*, 2014. URL `http://arxiv.org/abs/1312.6199`.

Taskar, B., Guestrin, C., and Koller, D. Max-margin Markov networks. *NIPS*, 2004.

Tsochantaridis, Ioannis, Hofmann, Thomas, Joachims, Thorsten, and Altun, Yasemin. Support vector machine learning for interdependent and structured output spaces. In *Proceedings of the twenty-first international conference on Machine learning*, pp. 104. ACM, 2004.

Tsoumakas, Grigorios and Katakis, Ioannis. Multi-label classification: An overview. *Dept. of Informatics, Aristotle University of Thessaloniki, Greece*, 2006.

Xu, Linli, Wang, Zhen, Shen, Zefan, Wang, Yubo, and Chen, Enhong. Learning low-rank label correlations for multi-label classification with missing labels. In *Data Mining (ICDM), 2014 IEEE International Conference on*, pp. 1067–1072. IEEE, 2014.

Yu, Hsiang-Fu, Jain, Prateek, Kar, Purushottam, and Dhillon, Inderjit S. Large-scale multi-label learning with missing labels. In *International Conference on Machine Learning (ICML)*, volume 32, jun 2014.

Zhang, Min-Ling and Zhou, Zhi-Hua. Multilabel neural networks with applications to functional genomics and text categorization. *Knowledge and Data Engineering, IEEE Transactions on*, 18 (10):1338–1351, 2006.

9 Appendix

9.1 Details

Various tricks of the trade from the deep learning literature, such as momentum, can be applied to improve the prediction-time optimization performance of our entropic mirror descent approach described in Section 2, which are particularly important because $E_x(\bar{y})$ is generally non-convex.

We perform inference in minibatches in parallel on GPUs.

When 'soft' predictions are useful, it can be useful to augment $E_x(\bar{y})$ with an extra term for the entropy of $\bar{y}$. This can be handled at essentially no computational cost, by simply normalizing the

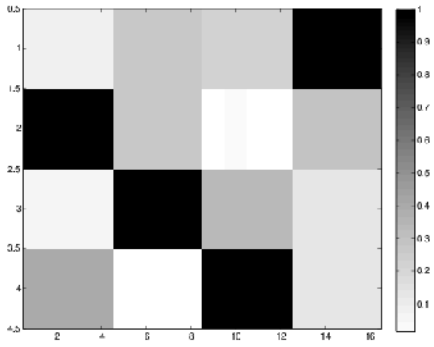


Figure 2: Structure learning on synthetic task using 10% of the data. The measurement matrix still recovers interactions between the labels characteristic of the data generating process

iterates in entropic mirror descent at a certain 'temperature.' This is only done at test time, not in the inner loop of learning.

Typically, backpropagation computes the gradient of output with respect to the input and also computes the gradient of the output with respect to any parameters of the network. For us, however, we only care about gradients with respect to the inputs $\bar{y}$ during inference. Therefore, we can obtain a considerable speedup by avoiding computation of the parameter gradients.

9.2 Hyperparameters

For prediction, both at test time and in the inner loop of learning, we ran gradient descent with momentum = 0.95, a learning rate of 0.1, and no learning rate decay. We terminated prediction when either the relative change in the objective was below a tolerance or the l_∞ change between iterates was below an absolute tolerance.

For training, we used sgd with momentum 0.9 with learning rate and learning rate decay tuned on development data. We use l2 regularization both when pre-training the features and net and during SSVM training, with l2 weights tuned on development data.

We used ReLU nonlinearities in all experiments. We did not tune the sizes of the hidden layers for the feature network and label energy network. These were set based on intuition and the size of the data, the number of training examples, etc.

	#labels	#features	# train	% true labels
Bibtex	159	1836	4880	2.40
Delicious	983	500	12920	19.02
Bookmarks	208	2150	60000	2.03
Yeast	14	103	2417	30.3

Table 4: Properties of the datasets.