Unlabeled Data Improves Word Prediction

Nicolas Loeff, Ali Farhadi, Ian Endres and David A. Forsyth
Department of Computer Science
University of Illinois at Urbana-Champaign
{loeff,afarhad2,iendres2,daf}@uiuc.edu

Abstract

Labeling image collections is a tedious task, especially when multiple labels have to be chosen for each image. In this paper we introduce a new framework that extends state of the art models in word prediction to incorporate information from unlabeled examples, using manifold regularization. To the best of our knowledge this is the first semi-supervised multi-task model used in vision problems. The new model can be solved using gradient descent and is fast and efficient. We show remarkable improvements for cases with few labeled examples for challenging multi-task learning problems in vision (predicting words for images and attributes for objects).

1. Introduction

Semi-supervised learning is a special form of classification that deals with the problem of learning from data when not all the labels are available. Situations like this arise when obtaining data is significantly less expensive than manually labeling it. This can be due to the time consumed labeling the examples, or because manual tagging requires expertise in the problem. The explosion of search engines on the Internet has also provided an inexpensive source of images with no labels or low-quality labels; the issue then is how can this trove of information be used to improve vision algorithms.

Despite significant progress in the machine learning community, current semi-supervised algorithms make strong assumptions about the distribution of the data: if these are wrong then the algorithms will not benefit from the unlabeled examples. In fact, it is common under these circumstances that the performance of the algorithm *decreases* with the unlabeled data. There is no free lunch [32]: good performance requires a matching of the model assumptions with the problem structure, and usually a higher effort in designing features and/or similarity functions than in regular supervised learning.

So, how does semi-supervised learning perform its magic? Most models use the distribution of the unlabeled data to *regularize* the classifier. In other words, the geometry of the unlabeled data may provide significant clues of the structure of the classifier boundary. This can be formalized by assuming the classifier "output" $p(y|x)$ (the conditional distribution) is influenced by the distribution of the marginal $p(x)$. There are far too numerous ways of translating this intuition into a learning algorithm to describe them in this paper so we will refer the interested reader to a recent survey [32]. As a summary, we will group many of these algorithms as in [7], according to the assumption they make about the geometry of the data:

Cluster assumption : Data points "in the same cluster" tend to share the same label. Several classifiers are based on this assumption, for instance Transductive SVMs [16], Cluster-Kernel [8] or Low Density Separation (LDS) [9].

Manifold assumption : Even though the data lives in a high-dimensional space, it is supported on a *manifold* of much lower intrinsic dimensionality. ISOMAP [28], LLE [26] and others estimate this structure explicitly, while algorithms like LapSVM [27], ManifoldBoost [20] and others do it implicitly, via a penalty term that imposes smoothness conditions on functions restricted to the manifold.

Semi-supervised learning also has attracted attention recently in vision ([4],[13] and others)

Our model differs from most of these algorithms that have been used to enhance "single"-task classifiers; it combines manifold regularization in a multi-task learning setting. It is also a significant improvement with respect to [18], that introduces a semisupervised multitask learning model by first using the unlabeled data to define neighborhoods that are then used to learn classifiers that share a common prior. Our model does not separate the stages, and is more scalable: we present results with one order of magnitude more datapoints and tasks.

The model we are going to use is a simple stack of linear classifiers, one for each word we are going to predict.

As in [19] we use the tracenorm regularization to prevent overfitting by constructing an internal representation that encourages sharing of features. It has produced state of the art results in word prediction and scene discovery on the Corel dataset [10]. As in implicit manifold learning algorithms we use the Laplacian regularization, that encourages sharing of label information by penalizing variability in the classifier among neighbors, but we extend it to the multi-task case.

Our contributions are manifold: We combine two powerful frameworks, multi-task learning and semi-supervised learning in a simple formulation by adding the required regularization terms. The terms encourage (a) sharing of features among the classifiers to improve generalization and (b) propagation of label information among neighbors in the manifold to learn from unlabeled examples. Moreover, this formulation does not change the properties of the original model: the cost is still convex and thus simple gradient descent can be performed to arrive to the global optimum. The new model both shares information from different labels and uses unlabeled data, with remarkable results on challenging problems, specially for cases with few labeled samples.

In section 2 we describe the model and introduce the manifold regularization term for semi-supervised learning. We discuss the datasets used and the experiments ran in section 3. Finally, in section 4 we present our conclusions.

2. Model

Let $\{x_i\}$ denote a set of N (d-dimensional) vectors that represent the features extracted from the image. The problem consists of learning to predict M words (tasks) from these feature vectors. Each word can take one of three values for each image: 1 if the word describes the image, -1 if the word does not describe the image, and 0 if the label is not provided ($y_i^m \in \{-1, 0, 1\}$).

In this paper we will use M linear classifiers $w_m^T x_i$ to predict the words from the images. In order to do so, we will minimize a cost C

$$\begin{aligned} \arg\min_W C(W) &= \underbrace{L(Y; W^T X)}_{\text{Supervised loss}} + \underbrace{\lambda_{mt} R_{mt}(W)}_{\text{Multitask regularization}} \\ &+ \underbrace{\lambda_{ss} R_{ss}(W, X)}_{\text{Semi-supervised (manifold) regularization}} \end{aligned} \quad (1)$$

where $W \in \mathbb{R}^{d \times M}$ is a matrix where each column in the classifier w_m, $Y \in \{0, \pm 1\}^{M \times N}$ is the matrix of labels in which each column is a different image and each row a word/task, $X \in \mathbb{R}^{d \times N}$ is the observation matrix, λ_{mt} the weight of the Multitask regularization and λ_{ss} the weight of the semi-supervised regularization.

The first term in eq. 1 represents the expected loss for predicted labels; it will be described in detail in section 2.1. In section 2.2 we introduce the multitask regularization term that promotes sharing features among word classifiers. The manifold regularization to propagate label information is described in section 2.3. In section 2.4 we put together these components and finally we show how to minimize the cost.

2.1. Loss

We want to produce a max-margin classifier W, so an appropriate loss is the hinge function $h(z) = \max(0, 1-z)$. The loss L can be expressed as the empirical risk

$$L(Y; W, X) = \frac{1}{NM} \sum_{i=1}^{N} \sum_{m=1}^{M} \Delta(y_i^m) h(y_i^m \cdot (w_m^T x_i)) \quad (2)$$

where Δ is a slack re-scaling term that deals with the imbalance between positive and negative examples. Most words are not present in a given image (the average in the datasets we use is around 4 words per image) so Δ is introduced to penalize errors differently: false negatives $\Delta(1) = \frac{n}{n+p}$ and false positives $\Delta(-1) = \frac{p}{n+p}$ where n is the number of negative examples for a word and p the number of positive examples. If the datapoint is unlabeled, then $\Delta(0) = 0$.

2.2. Multitask Regularization

Regularization is needed to prevent overfitting. One of the simplest terms we can add to L is a matrix norm to control the complexity of the classifier. If we wanted to learn independent SVMs for each word / task, then the L_2 (or Frobenium norm) regularization $\sum_m ||w_m||_2^2 = ||W||_F^2$ is suitable.

$$R_{mt}(W) = \frac{1}{2} ||W||_F^2$$

The **tracenorm regularization** is an alternative that takes advantage of the natural correlation between words. For instance, consider the labels "beach" and "sand": these labels tend to co-occur and the features needed to classify one should help for the other task. One way to aid feature sharing between the word classifiers is to control the rank of W. Unfortunately, minimizing this constraint is too difficult so we choose the tracenorm, a proxy for rank minimization. The tracenorm can be used as

$$R_{mt}(W) = \frac{1}{2} ||W||_\Sigma = \min_{W=FG} \frac{1}{2} \left(||F||_F^2 + ||G||_F^2 \right) \quad (3)$$

This interpretation of the tracenorm is natural when tracenorm is shown as equivalent to $||W||_\Sigma = \sum_l |\gamma_l|$ (where γ_l is the l^{th} singular value). Then the tracenorm minimization is equivalent to minimizing the L_1 norm of the singular values of W, an approximation to minimizing the L_0 norm

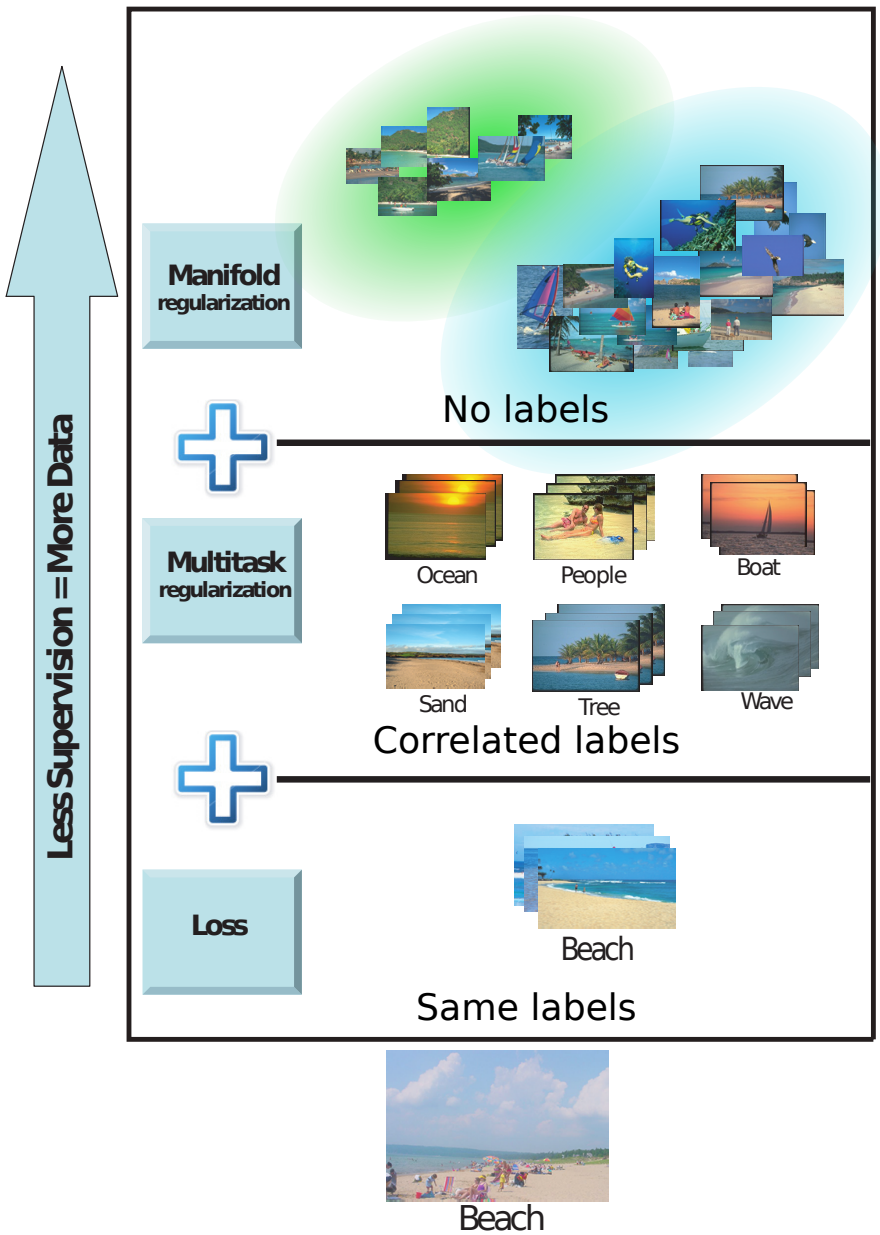

Figure 1. This figure summarizes three components of our model: loss, multitask regularization, and manifold regularization. The supervised Loss term requires fully labeled data which is the most expensive to obtain. For this reason we might not be able to provide the learner with enough training examples. The multitask regularization helps the learner to share information across correlated words. If we want to learn "Beach" and the algorithm will realize "Beach" and "Sand" are highly correlated, and thus it will use the "Sand" examples to improve the "Beach" classifier. This provides us more training examples to obtain better generalization. The manifold regularization term goes one step further by making use of images with no labels. This type of data is cheap to obtain. This term penalizes variability in the prediction of images that are close in the manifold. As we move up we require less supervision. This means we can use more training data with improved results.

of the singular values. This leads to a low-rank solution, in which correlated words share features.

The tracenorm regularization has been used successfully in the past in problems like collaborative filtering [25], multiclass classification [1] and multitask learning [20].

2.3. Manifold Regularization

To impose the manifold regularity on our classifier, we will assume that over regions of high data density, the classifier varies slowly. In other words the output of the classifier $f_m(x) = w_m^T x$ should not change abruptly between neighbors in the manifold. This means that if two images look similar under a *suitable* metric, then the words predicted should not be too different. Thus if one of the images is labeled this will "propagate" the labels to the other image.

One way to do this is to penalize the norm of the gradient of $f_m(x)$. We further assume the support of the marginal $p(x)$ lies on a domain $\mathcal{M} \subset \mathcal{R}^d$ (i. e. the "manifold").

$$R_{ss} = \frac{1}{M} \sum_{m=1}^{M} \int_{\mathcal{M}} ||\nabla_{\mathcal{M}} f_m(x)||^2 p(x) dx \qquad (4)$$

where $\nabla_{\mathcal{M}}$ is the gradient operator over the manifold.

Assuming sufficient regularity, and using the first Green identity,

$$\int ||\nabla_{\mathcal{M}} f_m(x)||^2 p(x) dx \quad = \quad \int f_m(x)^T \nabla^2_{\mathcal{M}} f_m(x) p(x) dx \qquad (5)$$

where $\nabla^2_{\mathcal{M}}$ is the Laplace-Beltrami operator (i. e. negative Laplacian).

Our regularization term is similar to that of [3], but it operates in several tasks simultaneously, encouraging smoothness of each word classifier in regions of high probability density. This term tries to make the predictions in each "connected" component as smooth as possible for each word.

Discrete approximation. During learning we usually do not know the distribution $p(x)$, so eq. 5 has to be discretized and integrals over x become summations over the datapoints. The Laplacian operator in equation 5 also has to be discretized. A usual approximation for this regularization term is the graph Laplacian $\mathcal{L}$ (the definition is beyond the scope of this paper, [3] describes it in detail). It consists of a weighted difference between the function at a point and its K nearest neighboring points; it is a generalization of the square lattice Laplacian discretization commonly used in numerical analysis and image processing.

$$R_{ss} = \frac{1}{MNK} \sum_{m=1}^{M} \sum_{i,j=1}^{N} f_m(x_i) \mathcal{L}_{i,j} f_m(x_j) \qquad (6)$$

where K is the number of neighbors for each datapoint. In our experiments, we chose to use Euclidean distance to compute nearest neighbors, and we used binary weights on the edges of the graph Laplacian. We also chose to use the *normalized* Laplacian [3] formulation.

2.4. Putting it all together

The hinge loss in eq. 2, the tracenorm in eq. 3 and the manifold loss in eq. 6 are all convex in the parameters of the classifier W. Therefore any local minimum of eq. 1 will be global. It is tempting then to solve for W using gradient descent techniques. Unfortunately neither the hinge loss or the tracenorm are differentiable. Thus, we follow [19] and use *smoothed* approximations of the hinge loss and absolute value.

We will consider a smooth approximation $h_\rho(z)$ of the hinge loss $h(z)$ that is exact for $|1 - z| \geq \rho$, and is twice differentiable everywhere. Likewise, For the tracenorm we use $||W||_\Sigma \approx ||W||_S = \sum_l a_\sigma(\gamma_l)$, where the smoothed absolute value $a_\sigma(x) = |x|$ for $|x| \geq \sigma$ and is twice differentiable everywhere. In our experiments we use $\rho = \sigma = 1$.

The final problem is approximated by

$$\begin{aligned} C(W) \approx C^S_{\sigma,\rho}(W) &= L^S(Y; W^T X) + \lambda_{mt} R^S_{mt}(W) \\ &+ \lambda_{ss} R^S_{ss}(W; X) \end{aligned} \qquad (7)$$

where the loss is

$$L^S(W; Y, X) = \frac{1}{N} \sum_{i=1}^{N} \sum_{m=1}^{M} \Delta(y_i^m) h_\rho(y_i^m \cdot (w_m^t x_i)) \qquad (8)$$

the multitask regularization,

$$R_{mt}(W) = ||W||_S \qquad (9)$$

and the manifold regularization,

$$R^S_{ss} = \frac{1}{MNK} \sum_{m=1}^{M} \sum_{i,j=1}^{N} (x_i^T w_m) \mathcal{L}_{i,j} (w_m^T x_j) \qquad (10)$$

Using the SVD decomposition $W = UDV^T$,

$$\frac{\partial R^S_{mt}}{\partial W} = U a'_\sigma(D) V^T \qquad (11)$$

For the manifold regularization term,

$$\frac{\partial R^S_{ss}}{\partial W} = X \mathcal{L} X^T W \qquad (12)$$

and the gradient of the data loss term is,

$$\frac{\partial L^S}{\partial W} = -X(\Delta(Y) \cdot h'_\rho(Y \cdot W^T X) \cdot Y)^T \qquad (13)$$

where $(A \cdot B)$ is the Hadamard or element-wise product: $(A \cdot B)_{ij} = a_{ij} b_{ij}$.

We used limited-memory BFGS for minimization. The algorithm is very efficient and the only step with $O(N^2)$ complexity is the computation of the graph Laplacian that is completed before learning starts.

3. Experiments and discussion

The main goal of this paper is to show the benefits of using unlabeled data with few labeled examples in cases when we have multiple correlated labels for each example. We evaluate our method in two challenging problem in computer vision which suffer from the limitations of labeled examples and have multiple labels per example; predicting word annotations in Corel dataset and predicting attributes for Pascal08 objects.

3.1. Annotation prediction for Corel

Corel dataset: This dataset [10] has been extensively used as a standard benchmark dataset for annotation prediction tasks. This subset of Corel images consists of 5000 images grouped in 50 different sets (CDs). These images are separated into 4500 training and 500 test images. The vocabulary size of this dataset is 374, out of which 371 appear in train and 263 in test set. The annotation length varies from 1 to 5 words per image.

Features: We employ features used in the PicSOM [30] image content analysis framework. These features convey image information using 10 different, but not necessarily uncorrelated, feature extraction methods. Feature vector components include: DCT coefficients of average color in 20x20 grid (analogous to MPEG-7 ColorLayout feature), CIE LAB color coordinates of two dominant color clusters, 16×16 FFT of Sobel edge image, MPEG-7 EdgeHistogram descriptor, Haar transform of quantized HSV color histogram, three first central moments of color distribution in CIE LAB color space, average CIE LAB color, co-occurrence matrix of four Sobel edge directions, histogram of four Sobel edge directions and texture feature based on relative brightness of neighboring pixels. The final image descriptor is a 682 dimensional vector. We append a constant value 1 to each vector to learn a threshold for our linear classifiers. [1]

Procedure: The task is to predict word annotations for images. To be able to test the benefits of unlabeled data in our model we remove ground truth annotations from images. In particular, for images in some subset of the training set, we remove all labels. In this setting we observe some portion of the training images and their corresponding annotation labels. However, we still use all of the training images, without their labels for the manifold term. The rest of the experimental settings mirror those of [19] so that results are comparable. Table 2 shows the gain achieved by considering the manifold term in optimization 2. In this table the first column shows the percentage of labeled training examples considered. The second column shows the F1 measure of predicting word annotations when we are not using the manifold term, and the third column is the same F1 measure with a manifold term incorporated. For example, if we use only 1% of the training labels, the F1 measure of predicting annotations without using the manifold is 2.8%, compared to F1 measure of 5.1% when we use the manifold term. This is an extremely challenging task. We are only using 45 training examples to train word predictors for more than 300 words. The manifold term offers a signifi-

[1] Note: some of the results (e.g. PicSOM) are not directly comparable as they limit the annotation length to be at most five (we do not place this limit as we aim to complete the annotations for each image. See [19] for details.

Method	P	R	F1	Ref
Co-occ	0.03	0.02	0.02	[23]
Trans	0.06	0.04	0.05	[10]
CMRM	0.10	0.09	0.10	[14]
TSIS	0.10	0.09	0.10	[6]
MaxEnt	0.09	0.12	0.10	[15]
CRM	0.16	0.19	0.17	[17]
CT-3×3	0.18	0.21	0.19	[31]
CRM-rect	0.22	0.23	0.23	[12]
InfNet	0.17	0.24	0.23	[22]
Independent SVMs	0.22	0.25	0.23	[19]
MBRM	0.24	0.25	0.25	[12]
MixHier	0.23	0.29	0.26	[5]
MatFact (Linear)	0.27	0.27	0.27	[19]
This work (Linear)	0.21	0.40	0.28	
MatFact (Kernel)	0.29	0.29	0.29	[19]
Label Transfer	0.27	0.32	0.29	[21]
PicSOM	0.35	0.35	0.35	[30]

Table 1. Comparison of the performance on the Corel dataset using all 4500 training examples with that of Co-occurrence model (Co-occ), Translation Model (Trans), Cross-Media Relevance Model (CMRM), Text space to image space (TSIS), Maximum Entropy model (MaxEnt), Continuous Relevance Model (CRM), 3×3 grid of color and texture moments (CT-3×3), Inference Network (InfNet), independent SVMs on the PicSOM features, Multiple Bernoulli Relevance Models (MBRM), Mixture Hierarchies model (MixHier), Matrix Factorization (MatFact) with linear and kernelized classifiers, Greedy Label transfer and PicSOM with global features[1].

Labeled ratio	1.00	0.50	0.10	0.01
Frob. norm no Manif.	0.233	0.183	0.126	0.028
Tracenorm no Manif.	0.278	0.228	0.138	0.028
Tracenorm w/ Manif.	0.278	0.227	0.171	0.051

Table 2. F1 scores on Corel dataset word prediction. The first row is the ratio of images labeled (from 100% to 1%), the second row are the F1 of independent linear SVMs (Frobenius norm and no manifold term), the third row are the F1 scores of the multitask model without the manifold term, and the last row is with this term. It is clear the **manifold term increases performance** especially when there are very few labels (for 1% there are less than 50 images to learn almost 400 words - hence the low F1 scores). Unlabeled data is useful for word prediction.

cant boost in the F1 measure of predicting annotations. As expected, if both methods observe enough training data the manifold term doesn't help significantly. The gap between using and not using the manifold term become smaller as we add more training examples. As in [19], the effect of the tracenorm is clear in this case.

3.2. Attribute prediction for objects in Pascal08

Another challenging task which is well suited to demonstrate our method is to predict visual attributes of objects. Recently a new dataset of object attributes was made available [11]. This dataset is collected for a different purpose, but it is well defined for our task as well. In particular, there are multiple correlated attribute labels per object in this dataset.

Attribute dataset: This dataset provides a list of 64 attributes annotated for the objects in the Pascal VOC 2008 dataset. These attributes describe parts, shapes, or materials, for example "has head", "is cylindrical", or "is furry", respectively. The object classes in Pascal VOC 2008 are: people, bird, cat, cow, dog, horse, sheep aeroplane, bicycle, boat, bus, car, motorbike, train, bottle, chair, dining table, potted plant, sofa, and tv/monitor. The number of objects from each category ranges from 150 to 1000, along with over 5000 instances of people.

Features: We employ features similar to [11]. We use color and texture, which are good for materials; visual words, which are useful for parts; and edges which are useful for shapes. Each of these four features are collected in a bag of words feature. Texture descriptors [29] are computed for each pixel, and quantized to the nearest 128 kmeans centers. The texture descriptor is extracted with a texton filterbank. Visual words are constructed with an HOG spatial pyramid, using 8x8 blocks, a 4 pixel step size, and 2 scales per octave. HOG descriptors are quantized to 256 kmeans centers. Edges are found using a standard canny edge detector and their orientations are quantized into 8 unsigned bins. Finally, color descriptors are densely sampled for each pixel, and quantized to the nearest 64 kmeans centers. The color descriptor consists of the LAB values.

Having quantized these values, local texture, HOG, edge, and color descriptors inside the bounding box are binned into individual histograms. To represent shapes and locations, we also generate histograms for each feature type for each cell in a grid of two vertical blocks. These seven histograms are stacked together resulting in a 1371-dimensional feature; we appended a 1 to learn the linear classifier threshold.

Procedure: throughout our experiments, similar to [11], we are assuming that the bounding boxes are provided for objects in Pascal08. This means that we are not addressing the problem of object localization or detection. We are instead interested in predicting attributes for objects. Similar to the Corel experiment, we remove attribute labels from training examples to see how manifold term helps in case of insufficient labeled training examples. Table 3 shows F1 measures for predicting attributes for Pascal objects. As expected, the manifold term offers a considerable gain in F1 measure when there is not enough labeled training data. For example, when learning with only 1% of the labeled data, we get an F1 measure of 17.4% without the manifold term which increases to 22.9% with the manifold term. This is a challenging task. As observed in the Corel experiment, as more training data becomes available the effects of the manifold term diminish.

Labeled ratio	1.00	0.50	0.10	0.01
Frob. norm no Manif.	0.317	0.286	0.239	0.178
Tracenorm no Manif.	0.337	0.322	0.265	0.174
Tracenorm w/ Manif.	0.337	0.329	0.294	0.229

Table 3. Attribute prediction results for Pascal08 objects. Again, the influence of the manifold term is clear, mainly for low numbers of labeled examples.

4. Conclusions

We have introduced a new framework for learning correlated tasks in the presence of unlabeled data. As far as we know this is the first model used in the vision community that combines these two powerful approaches. Our max-margin formulation shares features between tasks and also propagates label information for learning from unlabeled examples. Our experiments show that unlabeled data makes a large contribution to performance of the classifier, especially when the ratio of labeled examples is low.

Acknowledgments. This work was supported in part by the National Science Foundation under IIS -0803603 and in part by the Office of Naval Research under N00014-01-1-0890 as part of the MURI program. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect those of the National Science Foundation or the Office of Naval Research.

References

[1] Y. Amit, M. Fink, N. Srebro, and S. Ullman. Uncovering shared structures in multiclass classification. In *Proceedings of the International Conference on Machine Learning*, pages 17–24, 2007.

[2] R. K. Ando and T. Zhang. A high-performance semi-supervised learning method for text chunking. In *Proceedings of the Annual Meeting of Association for Computational Linguistics*, pages 1–9, 2005.

[3] M. Belkin, P. Niyogi, and V. Sindhwani. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *Journal of Machine Learning Research*, 7:2399–2434, Nov. 2006.

[4] Y. Bengio and Y. Lecun. Scaling learning algorithms towards AI. In L. Bottou, O. Chapelle, D. Decoste, and J. Weston, editors, *Large-Scale Kernel Machines*. MIT Press, Cambridge, MA, 2007.

[5] G. Carneiro and N. Vasconcelos. Formulating semantic image annotation as a supervised learning problem. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 163–168, 2005.

[6] E. Celebi and A. Alpkocak. Combining textual and visual clusters for semantic image retrieval and auto-annotation. In *Proceedings of the European Workshop on the Integration of Knowledge, Semantics and Digital Media Technology*, pages 219–225, 2005.

[7] O. Chapelle, B. Schölkopf, and A. Zien. *Semi-Supervised Learning*. MIT Press, Cambridge, MA, 2006.

[8] O. Chapelle, J. Weston, and B. Schölkopf. Cluster kernels for semi-supervised learning. In *Advances in Neural Information Processing Systems*, pages 585–592, 2003.

[9] O. Chapelle and A. Zien. Semi-supervised classification by low density separation. In *Proceedings of the International Workshop on Artificial Intelligence and Statistics*, pages 57–64, 2005.

[10] P. Duygulu, K. Barnard, J. F. G. de Freitas, and D. A. Forsyth. Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary. In *Proceedings of the European Conference on Computer Vision*, pages 97–112, 2002.

[11] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth. Learning to describe objects. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 137–145, 2009.

[12] S. L. Feng, R. Manmatha, and V. Lavrenko. Multiple bernoulli relevance models for image and video annotation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1002–1009, 2004.

[13] X. He, S. Yan, Y. Hu, P. Niyogi, and H.-J. Zhang. Face recognition using laplacianfaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(3):328–340, March 2005.

[14] J. Jeon, V. Lavrenko, and R. Manmatha. Automatic image annotation and retrieval using cross-media relevance models. In *Proceedings of the Annual International ACM Conference on Research and Development in Information Retrieval*, pages 119–126, 2003.

[15] J. Jeon and R. Manmatha. Using maximum entropy for automatic image annotation. In *Proceedings of the International Conference on Image and Video Retrieval*, pages 24–32, 2004.

[16] T. Joachims. Transductive inference for text classification using support vector machines. In *Proceedings of the International Conference on Machine Learning*, pages 200–209, 1999.

[17] V. Lavrenko, R. Manmatha, and J. Jeon. A model for learning the semantics of pictures. In *Advances in Neural Information Processing Systems*, pages 251–259, 2003.

[18] Q. Liu, X. Liao, and L. Carin. Semi-supervised multitask learning. In *Advances in Neural Information Processing Systems*, pages 937–944, 2007.

[19] N. Loeff and A. Farhadi. Scene discovery by matrix factorization. In *Proceedings of the European Conference on Computer Vision*, pages 451–464, 2008.

[20] N. Loeff, D. Forsyth, and D. Ramachandran. ManifoldBoost: Stagewise function approximation for fully-, semi- and unsupervised learning. In *Proceedings of the International Conference on Machine Learning*, pages 600–607, 2008.

[21] A. Makadia, V. Pavlovic, and S. Kumar. A new baseline for image annotation. In *Proceeding of the European Conference on Computer Vision (3)*, pages 316–329, 2008.

[22] D. Metzler and R. Manmatha. An inference network approach to image retrieval. In *Proceedings of the International Conference on Image and Video Retrieval*, pages 42–50, 2004.

[23] Y. Mori, H. Takahashi, and R. Oka. Image-to-word transformation based on dividing and vector quantizing images with words. In *Proceedings of the International Workshop on Multimedia Intelligent Storage and Retrieval Management*, pages 9–16, 1999.

[24] A. Quattoni, M. Collins, and T. Darrell. Learning visual representations using images with captions. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2007.

[25] J. D. M. Rennie and N. Srebro. Fast maximum margin matrix factorization for collaborative prediction. In *Proceedings of the International Conference on Machine Learning*, pages 713–719, 2005.

[26] S. T. Roweis and L. K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, Dec. 2000.

[27] V. Sindhwani, M. Belkin, and P. Niyogi. The geometric basis of semi-supervised learning. In O. Chapelle, B. Schölkopf, and A. Zien, editors, *Semi-Supervised Learning*, pages 217–235. MIT Press, Cambridge, MA, 2006.

[28] J. B. Tenenbaum, V. de Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, Dec. 2000.

[29] M. Varma and A. Zisserman. A statistical approach to texture classification from single images. *International Journal of Computer Vision*, 62(1):61–81, April 2005.

[30] V. Viitaniemi and J. Laaksonen. Evaluating the performance in automatic image annotation: Example case by adaptive fusion of global image features. *Image Communication*, 22(6):557–568, July 2007.

[31] A. Yavlinsky, E. Schofield, and S. Rüger. Automated image annotation using global features and robust nonparametric density estimation. In *Proceedings of the International Conference on Image and Video Retrieval*, pages 507–517, 2005.

[32] X. Zhu. Semi-supervised learning literature survey. Technical Report 1530, Computer Sciences, University of Wisconsin-Madison, Madison, WI, 2005.