Learning with Locally Linear Feature Regularization

Ted Sandler, John Blitzer, Lyle Ungar
Department of Computer and Information Science
University of Pennsylvania, Philadelphia, PA 19104
(tsandler, blitzer, ungar)@seas.upenn.edu

Machine learning models are successfully being used for problems in language, vision, and biology that have millions or tens of millions of features. A common approach to alleviating the complexity of high dimensional feature spaces is to penalize the L_1 or L_2 norm of the parameter vector. We may be able to design more effective regularizers, though, if we possess external information about which features should behave similarly. For example, word co-occurrence statistics or thesauri such as Wordnet can indicate similarities which are useful for predicting the topic or sentiment of a document, and biological databases of gene pathways give similarities of genes which can predict disease based on gene expression levels. We present a simple framework in which similarities between features are encoded as a graph on features and a regression model is learned whose feature coefficients are similar for neighboring nodes. Our regularization criterion is closely related to locally linear embedding, a method for learning low dimensional embeddings of unlabeled, high-dimensional data [1]. Because of this, we name it locally linear feature regularization (LLFR).

Many authors have explored using graphs constructed from local distances between unlabeled instances to regularize semi-supervised classification models [2, 3]. A key difference between our method and these techniques is that we construct our regularizers based on similarities between *features*, rather than instances. Thus our method is more similar in spirit to the work of Toutanova et al. [4], who explored random walks on features in the context of a generative model (although the random walks were combined discriminatively). By contrast, we use the feature graph to regularize a discriminative linear model. The feature graph induces a prior covariance matrix on the model parameters, and we use information obtained from unlabeled data to learn the structure of this graph. Unlike Raina et al. [5] who also learn a covariance matrix from unlabeled data, LLFR side-steps the technical challenges associated with constructing a positive definite matrix directly and instead induces it from a graph. Consequently, it can be used to learn larger models.

Locally linear feature regularization takes as input a weighted graph $G = (V, E)$ whose vertices correspond to the features of our model and whose edges represent similarities between features with respect to how similarly they are correlated with the labels. The edges are weighted by a Markov transition matrix, P, whose entries are non-negative and whose rows sum to one. Larger weights correspond to higher similarity, while a weight of zero means that two features are unrelated. Since we want the feature coefficients to be close at neighboring nodes, we penalize each feature's coefficient by the squared amount it differs from the average of its neighbors' coefficients. For a weight vector $\boldsymbol{\beta} \in \mathbb{R}^d$, the LLFR term is $\sum_{j=1}^{d} \left(\beta_j - \sum_k P_{jk}\beta_k\right)^2$.

Given a training set $T = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ of points $\boldsymbol{x}_i \in \mathbb{R}^d$ with associated labels, y_i, and denoting the loss on a single instance as $l(\boldsymbol{x}, y; \boldsymbol{\beta})$, we seek in training to minimize the regularized loss

$$\text{loss}(\boldsymbol{\beta}, T) = \frac{1}{n}\sum_{i=1}^{n} l(\boldsymbol{x}_i, y_i; \boldsymbol{\beta}) + \alpha \sum_{j=1}^{d}\left(\beta_j - \sum_{k=1}^{d} P_{jk}\beta_k\right)^2 + \gamma\|\boldsymbol{\beta}\|_2^2\,, \tag{1}$$

where we have added a ridge penalty to ensure that the loss function is strictly convex. The regularization portion of equation (1) can be rewritten as $\boldsymbol{\beta}^\top\left(\alpha(I-P)^\top(I-P) + \gamma I\right)\boldsymbol{\beta}$, showing that LLFR penalizes $\boldsymbol{\beta}$ according to a squared Mahalanobis norm. Enforcing smoothness allows the learning of coefficients for features not seen in the training set. Additionally it can provide more robust estimates for features seen only a few times.

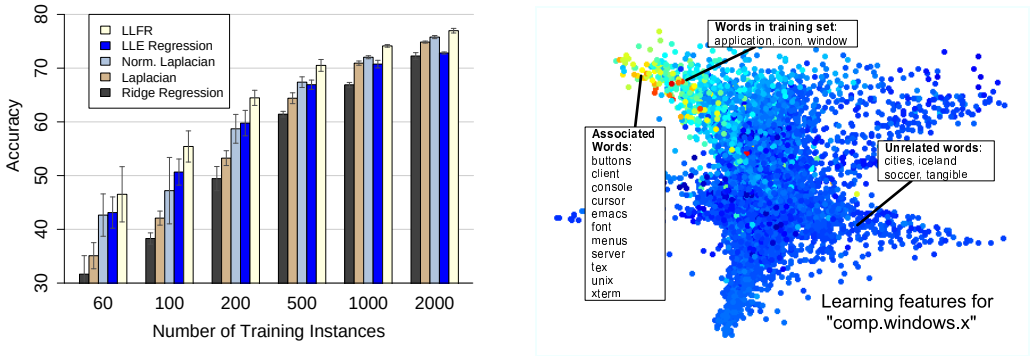

Figure 1: **Left:** Accuracy of LLFR and four baselines on "20 newsgroups" data. LLFR attains the highest accuracy. **Right:** Words were projected onto the fourth and 36th smallest eigenvectors of the regularization matrix, $(I-P)^\top(I-P)$, chosen for having highest correlation with the class `comp.windows.x`. A model was trained on only one document per class. The points are shaded according to the coefficients in the discriminant vector for the class `comp.windows.x`. "Red/light" denotes strongly positive coefficients while "blue/dark" denotes negative coefficients. The shading shows non-trivial weights are learned for unseen words.

We tested LLFR on the 20 newsgroups data set using all twenty classes. LLFR was compared against four baselines: (1) logistic regression with L_2 regularization, (2) logistic regression with a regularization penalty derived from the graph Laplacian: $\sum_{i,j} W_{ij}(\beta_i - \beta_j)^2$, (3) logistic regression with a similar penalty derived from the normalized graph Laplacian, and (4) LLE logistic regression.

The feature set was restricted to the 11,376 words which occurred in at least 20 documents. We constructed the feature graph by assigning a node to each word in the vocabulary. To create the graph edges, each word was represented by a binary vector denoting presence or absence in each of the 20,000 documents of the data set. Cosines were computed between word vectors and each word was linked to its 100 most similar neighbors. This generated 573,334 edges for the graph, each weighted by cosine score. The matrix P was constructed by L_1 normalizing each vertex's out-going edges. In LLE logistic regression, each document was projected onto the 200 smallest eigenvectors of the matrix $(I-P)^\top(I-P)$ and a classifier was trained on this representation.

Accuracies for 20 newsgroups are given in figure 1. Results are averaged over five trials with training sets sampled to contain an equal number of documents per class. LLFR with a graph constructed from unlabeled data out performs all baseline regularizations and increases accuracy by 4%-17% over L_2, a reduction in error of 17%-30%. Additionally, LLFR consistently outperforms the related LLE regression. We conjecture this is because it implicitly incorporates information from all eigenvectors and eigenvalues as opposed to just the smallest 200. Here we have demonstrated improvements with just one approach to constructing the feature graph but LLFR is flexible in that it allows priors to be derived from arbitrary associations among features.

References

[1] Sam T. Roweis and Lawrence K. Saul. Nonlinear Dimensionality Reduction by Locally Linear Embedding. *Science*, 290(5500):2323–2326, 2000.

[2] Dengyong Zhou, Olivier Bousquet, Thomas Navin Lal, Jason Weston, and Bernhard Schölkopf. Learning with local and global consistency. In *Advances in Neural Information Processing Systems (NIPS)*, 2004.

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

[4] Kristina Toutanova, Christopher D. Manning, and Andrew Y. Ng. Learning random walk models for inducing word dependency distributions. In *International Conference on Machine Learning (ICML)*, 2004.

[5] R. Raina, A.Y. Ng, and D. Koller. Constructing informative priors using transfer learning. In *International conference on Machine learning (ICML)*, 2006.