Feature-Preserving Embeddings for Topic Transfer

Peter Krafft
Department of Computer Science
University of Massachusetts Amherst
Amherst, MA 01003
pkrafft@cs.umass.edu

Sridhar Mahadevan
Department of Computer Science
University of Massachusetts Amherst
Amherst, MA 01003
mahadeva@cs.umass.edu

Abstract

We present the theory and application of feature-preserving embeddings, a new method for transferring features of some training dataset to other datasets that have the same underlying manifold structure. Given the values of a set of features of one dataset, we create an embedding of that dataset in which the values can be predicted by a linear regression. We discuss how to apply this method to transferring topic proportions learned with a generative topic model to a parallel corpus.

1 Introduction

In natural language processing, the goal of building a generative topic model is to capture the semantics of any particular document. Since meaning should be invariant to changes in the language in which a document is written, we would expect that after learning a topic model in one language, we could transfer the topics of some document to a corresponding document in a different language.

Consider a modified machine translation problem in which we have two corpora, each composed of the same documents written in different languages, say English and German, and a topic model of the English documents that we want to transfer to the German documents. That is, we have features that should apply just as well to the German documents as to the English documents. Furthermore, suppose we do not know how the corpora match up; we may be given a small number of the correspondences, but there are some English documents that we do not know how to map to the German corpus.

More generally, given two datasets sampled from the same manifold but possibly represented in different coordinate systems, we want a method for transferring the values of features learned on one dataset to the other dataset.[1] To do this, we align the datasets using a manifold learning algorithm that maps the datasets to the same latent space according to their local geometries and any given correspondence information, and we add a novel soft constraint that the known values of the features should be able to be predicted by linear regression on the embedded points (see Figure 1).

Plain semisupervised alignment [3] in combination with nearest-neighbor inference could also be used for this problem, but the alignments are unstable when given only a small amount of correspondence information. Our goal is to avoid some of the instability of previous methods by finding a good representation of the features in the embedding space itself. The idea is that if the features of the training set vary linearly over the underlying manifold, inferences based on a regression in the embedding space should be more robust than inference based only on nearest-neighbor correspondence. That is, we should be able to infer features more easily than correspondences.

[1] Note that we only talk about aligning two domains in this paper, but as in Wang [7], the technique generalizes to any number of domains.

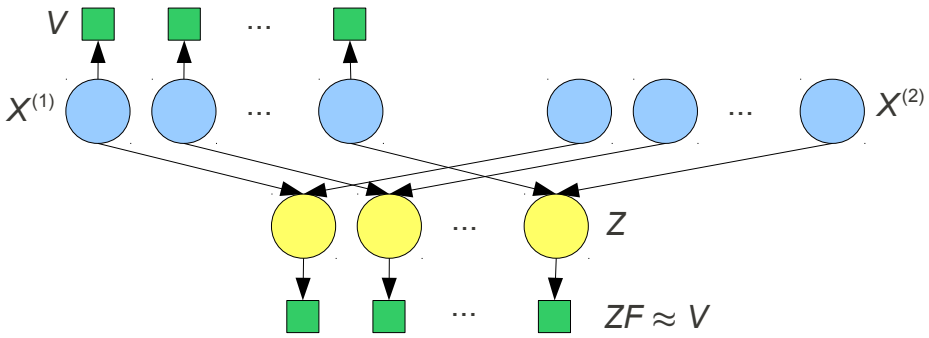

Figure 1: A diagram of our method. In the topic modeling example, $X^{(1)}$ is the English corpus, $X^{(2)}$ is the German corpus, V is the matrix of topic proportions for each English document, Z is the matrix of embedded documents, which we want to align, and ZF is the representation of the topic proportions in the embedding space.

2 Background and Related Work

We based the mechanics of our model on semisupervised alignment [3], which extends Laplacian eigenmaps [1] to the alignment case. Some previous researchers have worked on simultaneous prediction and dimensionality reduction, both for a single regression problem [5] and for multiple classification problems [6], but these methods are only designed for learning with a single dataset, so neither allows for transfer.

3 Feature-Preserving Embeddings

In the Section 4, we develop the technique for alignment (i.e. embedding multiple datasets simultaneously). Here, we discuss the simpler case of finding an embedding of a single dataset sampled from some manifold. Let the $n \times p$ matrix X be the dataset, which has n data points and is represented on p variables. Assume we have a set of features of X, and let the values of these f features, which for now we assume are given, be stored in the $n \times f$ matrix V. We want to find a k-dimensional embedding of X that respects the local similarity of the data points in X and affords linear representations of the features of X. In other words, we want to find an $n \times k$ embedding matrix, Z, and a $k \times f$ weight matrix, F, with the following properties:

1. Two embedded points (rows in Z) are close in terms of Euclidean distance if the corresponding points in the dataset, X, were similar according to some similarity metric, and

2. ZF takes values that are as close as possible to V.

Given L, the normalized Laplacian matrix associated with X, the first requirement is captured by the Laplacian eigenmap cost function,

$$C(Z) = \sum_i ||Z_{i,\cdot} - Z_{j,\cdot}||^2 L = tr(Z'LZ).$$

The second requirement is satisfied by adding an additional soft constraint to get

$$C(Z, F) = tr(Z'LZ) + \sum_{i,j}((ZF)_{i,j} - V_{i,j})^2,$$

or $C(Z, F) = tr(Z'LZ) + tr((ZF - V)'(ZF - V))$, which we can minimize by solving

$$\frac{\partial C}{\partial Z} = 2LZ + 2(ZF - V)F' = 0 \qquad \text{and} \qquad \frac{\partial C}{\partial F} = 2Z'(ZF - V) = 0.$$

Unfortunately, we do not have a simultaneous solution to these equations. Instead, we alternate between the partial solutions. When Z is fixed, the second equation reduces to a simple least squares,

$$F = (Z'Z)^{-1}Z'V.$$

When F is fixed we get

$$vec(Z) = (I_k \otimes L + FF' \otimes I_n)^{-1} vec(VF'),$$

where $vec(A)$ stacks the m columns of the $n \times m$ matrix A into an $nm \times 1$ vector, I_n is the $n \times n$ identity matrix, and $\otimes$ denotes the Kronecker product. In either case, if the inverse does not exist, we use the Moore-Penrose pseudoinverse instead.

4 Alignment

Just as in semisupervised alignment, aligning two datasets, an $n \times p_1$ matrix $X^{(1)}$ and an $m \times p_2$ matrix $X^{(2)}$, associated with their Laplacians, $L^{(1)}$ and $L^{(2)}$, is equivalent to embedding the concatenated datasets using the $(n+m) \times (n+m)$ joint Laplacian $L = \begin{pmatrix} L^{(1)} & L^{(12)} \\ (L^{(12)})' & L^{(2)} \end{pmatrix}$, where $L^{(12)}$ is an $n \times m$ matrix that captures the correspondence information between the two datasets and L must be renormalized as a whole, so we can write the cost function for plain alignment as

$$C(Z^{(1)}, Z^{(2)}) = \sum ||Z^{(1)}_{i,\cdot} - Z^{(1)}_{j,\cdot}||^2 L^{(1)} + \sum ||Z^{(2)}_{i,\cdot} - Z^{(2)}_{j,\cdot}||^2 L^{(2)} + \sum ||Z^{(1)}_{i,\cdot} - Z^{(2)}_{j,\cdot}||^2 L^{(12)},$$

where $Z^{(1)}$ and $Z^{(2)}$ are the $n \times k$ and $m \times k$ latent coordinate matrices of the two domains. Just as in the single dataset case, this can be written as $C(Z) = tr(Z'LZ)$, where $Z = [Z^{(1)}\ Z^{(2)}]'$ and L is the joint Laplacian. Similarly, we can initially write the second part of the cost function as

$$\sum_{i,j} ((Z^{(1)}F)_{i,j} - V^{(1)}_{i,j})^2 + \sum_{i,j} ((Z^{(2)}F)_{i,j} - V^{(2)}_{i,j})^2,$$

where $V^{(1)}$ is the $n \times f$ matrix of known values for the first domain and $V^{(2)}$ is the matrix of unknown values for the second domain (which for now we treat as also being known), but letting $V = [V^{(1)}, V^{(2)}]'$, this equation also simplifies to the single dataset notation giving us

$$C(Z, F) = tr(ZLZ) + tr((ZF - V)'(ZF - V)). \quad (1)$$

The main purpose of using feature-preserving embeddings is to be able to infer the unknown feature values of the second dataset, $X^{(2)}$. To infer values of $V^{(2)}$, we introduce the following notation. Let $V_{\{C\}}$ be the rows of $V^{(2)}$ associated with the data points in $X^{(2)}$ that we know how to map directly to $X^{(1)}$. If a data point in $X^{(2)}$ has only one known correspondence in $X^{(1)}$, we let the values of that row in $V_{\{C\}}$ equal the corresponding row in $V^{(1)}$, and if there are multiple correspondences, we take a weighted average of the corresponding rows. Now let $V_{\{M\}}$ be the remaining values of $V^{(2)}$ that we must still infer. At every step of the algorithm, we impute these missing values by taking the current entries of XF in the missing locations. This is equivalent to removing those $((Z^{(2)}F)_{i,j} - V^{(2)}_{i,j})^2$ terms from the cost function in Equation 1 and replacing them with zeros of the form $((ZF)_{i,j} - (ZF)_{i,j})^2$, so the imputed values do not affect the embedding. Finally, letting $(ZF)_{\{M\}}$ be the rows we use to impute $V_{\{M\}}$, we get the following algorithm:

1. Extract features of $X^{(1)}$. Store the values in $V^{(1)}$
2. Find the individual Laplacians $L^{(1)}$ and $L^{(2)}$, and the correspondence matrix $L^{(12)}$, and form the joint Laplacian, L, as in Ham, Lee, and Saul [3]
3. Set $V = [V^{(1)}\ \mathbf{0}]'$
4. Find the values for $V_{\{C\}}$ according to any given correspondence information
5. Initialize Z (e.g. with $Unif(-1, 1)$ entries) // random embedding/alignment
6. Set $F = (Z'Z)^{-1}Z'V$ // calculate the least squares regression weights
7. Set $V_{\{M\}} = (ZF)_{\{M\}}$ // fill in missing values according to the regression
8. **repeat**
   1. Set $vec(Z) = (I_k \otimes L + FF' \otimes I_n)^{-1} vec(VF')$ // recalculate embedding/alignment
   2. Set $F = (Z'Z)^{-1}Z'V$ // recalculate regression weights
   3. Set $V_{\{M\}} = (ZF)_{\{M\}}$ // fill in missing values according to the new regression

5 Results

We tested this method in two domains. The first domain was a toy example in which both $X^{(1)}$ and $X^{(2)}$ were 16 two-dimensional data points lying on a grid (equally spaced over the region $[.2, .5] \times [.5, .8]$). The features of $X^{(1)}$ were defined by the function $[-10, 10]'$, and in setting the correspondences, we considered $X^{(2)}$ to be a 180^{o} rotation of $X^{(1)}$. In this domain, given between one and seven correspondences, our method did much better than both semisupervised alignment and a simple control of imputing every row of $V_{\{M\}}$ as the mean column values of $V^{(1)}$. Given more than seven correspondences, semisupervised alignment did about as well as our method.

For the second task, we used the Europarl dataset [4] to test whether we could transfer a topic model from a set of English documents to a corresponding set of German documents. First, we used 10,000 documents to learn a 20-topic LDA topic model [2] for the English documents. We then took a small subset of those documents with high proportions of a topic that looked semantically meaningful. To facilitate quantitative comparison, we assumed that the topic proportions for corresponding documents would be the same. We then tried to infer the topic proportions for the German documents with missing correspondence information. Surprisingly, both k-nearest neighbor inference with semisupervised alignment and feature-preserving embeddings performed much worse than imputing the missing proportions with the mean value from the English topics. The problem with the two alignment-based methods may have been that the similarity metric we used to define the Laplacians within each dataset did not capture their semantics, their underlying joint manifold.

6 Future Directions and Conclusions

There are a few changes we would like to make to the current method. First, we want an algorithm that can account for partial alignments, which requires modification of the framework, and related to that, we would like to integrate probabilistic correspondence information. Currently, we must assume that any correspondence information we are given represents the true similarity between states in different domains, but a better method would be to assume that two states in different domains have only a certain probability of being in correspondence.

The novel contribution of this paper is to use known values of some features to find variables on which to represent those features effectively, and to use that representation for transfer. This algorithm is a new method for the transfer of models that should be invariant to changes in representation of the training dataset, such as those learned by generative topic models or other rich generative models.

References

[1] M. Belkin and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 2003.

[2] D. Blei, A. Ng, and M. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 2003.

[3] J. Ham, D. Lee, and L. Saul. Semisupervised alignment of manifolds. In *Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics*, 2005.

[4] P. Koehn. Europarl: A parallel corpus for statistical machine translation. MT Summit, 2005.

[5] J. Nilsson, F. Sha, and M. Jordan. Regression on manifolds using kernel dimension reduction. In *Proceedings of the* 24^{th} *International Conference on Machine Learning*, 2007.

[6] G. Obozinski, B. Taskar, and M. Jordan. Joint covariate selection and joint subspace selection for multiple classification problems. *Statistics and Computing*, 2010.

[7] C. Wang. *A Geometric Framework for Transfer Learning Using Manifold Alignment*. PhD thesis, University of Massachusetts Amherst, 2010.