Handwritten Zip code recognition using digit models

Shiraj Sen, Kedar Bellare
Department of Computer Science,
University of Massachusetts Amherst

May 15, 2006

1 Introduction

The use of manifold methods has become popular in the machine learning community. Such methods have been applied to various problems in unsupervised and semi-supervised learning. Recently they have also gained momentum in reinforcement learning in the context of function approximation [4].

The underlying assumption made in these approaches is that a function defined on the data points has local smoothness and the data resides on a low-dimensional manifold. Generally the steps used are similar to those in [5]:

1. Given data points $x_1, .., x_n$ in R^k generate a graph. This graph can be generated based on the local distance metric or full affinity matrix.

2. Compute the combinatorial or normalized laplacian for this weight matrix.

3. For clustering compute the embedding based on the top eigenvectors of this laplacian.

One such practical application to the above technique is handwriting recognition in which the high dimensional character data can be assumed to belong a low-dimensional underlying manifold. For example, the handwritten digit 1 can be fairly represented by a straight line, which is completely determined by the principal eigenvector. This technique assumes that the high dimensional image is a correct segmentation of the character. However, manual segmentation can be costly and a time-consuming process. In recent times, Hidden Markov Models have been a preferred way of segmenting characters by using a set of labeled training segments [6].

In this project, we have explored two different approaches to the above problem of handwriting recognition. In our first approach, we try to avoid the problem of segmentation by embedding constant slices of the image in a manifold using the graph Laplacian. This manifold is then fitted with a mixture of gaussians to learn the character models. In our second approach, we explicitly handling the problem of segmentation using semi-supervised learning, and then use these learned segments along with some labeled data for clustering on a manifold.

2 Hidden Markov Models

In this work, we have used Hidden Markov Models where the observations are modeled using mixture of gaussians. An HMM can be represented as Dynamic Bayes Network (DBN) as displayed in Figure 1

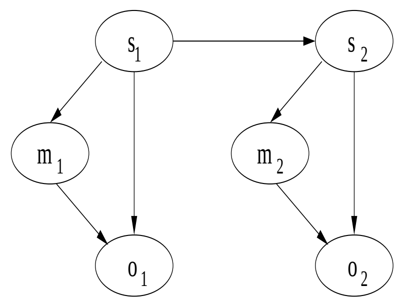

Figure 1: A Hidden Markov Model

There are three different types of nodes o_t, s_t and m_t. The edges between the nodes represent statistical dependencies. The state of the agent at time t is represented by s_t. Since, the true state of the agent is hidden, the agent at every time steps sees an observation o_t. The states are modeled using a mixture of gaussians m_t.

3 Combining Laplacian and Hidden Markov Models

3.1 First Approach

Given a set of images of length $img_{ht} \times img_{wt}$, we apply a sliding window on the image of width w and slide size s. Each of these windows W_i having $img_{ht} \times w$ pixels correspond to a point s_i in $R^{img_{ht} \times w}$. Given a set of such points $S = s_1, \ldots, s_n$ in R^l, we proceed as follows:

- Form the affinity matrix $A \in R^{n \times n}$ defined by A_{ij} being the $L2$ norm between s_i and s_j if $i \neq j$, and $A_{ii} = 0$.
- Define D to be the diagonal matrix whose (i, i)-element is the sum of A's i-th row, and construct the matrix $L = D - A$.
- Find $x_1, x_2, \ldots, x_k$, the k largest eigenvectors of L, and form the matrix $X = [x_1, x_2, \ldots, x_k] \in R^{n \times k}$ by stacking the eigenvectors in columns.
- Treating each row of X as a point in R^k, computing the embedding of the original point s_i.

Every image can now be represented by a sequence of vectors where each vector corresponds to the embedding of the image window in R^k. Assuming that the data lies in this underlying low-dimensional manifold, any image can be represented as a temporal sequence of points in this manifold. We model this manifold using a Hidden Markov Model [3] where the space is tiled using Mixture of Gaussians. Such a model is based on the assumption that the mixture of gaussians can be used to separate the labels in the manifold.

3.2 Second Approach

In this approach we use the Hidden Markov models to learn a correct segmentation of the characters. This is done by using a labeled dataset in which each partition of the image is either labeled as *start*, *middle* or *end* depending on the correct segmentation. This dataset consisting of the image window and the labels is then used to train the HMM to learn character segmentations. Once, the HMM has been trained, the test data is passed through the HMM and their corresponding segmentations are obtained. Since the obtained segments can be of varying length, they are normalised to have the same length. These segments along with some labeled segments of the characters are then used to perform semi-supervised learning on the manifold. This is done by utilizing the underlying manifold structure of the unlabeled examples to perform label propagation. The label propagation algorithm [1] proceeds as follows.

Given k points $x_1, \ldots, x_k \in \Re^l$, with the first $s < k$ points having labels c_i, where $c_i \in -1, 1$, and the rest being unlabeled.

Step 1: [*Constructing the Adjacency Graph with n nearest neighbors*] Nodes i and j corresponding to the points x_i and x_j are connected by an edge if i is among n nearest neighbors of j or j is among n nearest neighbors of i. The distance metric can be a standard Euclidean distance or a Gaussian kernel.

Step 2: [*Eigenfunctions*] Compute p eigenvectors $e_1, \ldots, e_p$ corresponding to the p smallest eigenvectors for the eigenvector problem $Le = \lambda e$ where $L = D - W$ is the graph Laplacian for the adjacency graph. Here W is the adjacency matrix defined above and D is a diagonal matrix of the same size as W satisfying $D_{ii} = \sum_j W_{ij}$.

Step 3: [*Building the classifier*] To approximate the class we minimize the error function $Err(a) = \sum_{i=1}^{s}(c_i - \sum_{j=1}^{p} a_j e_j(i))^2$, where p is the number of eigenfunctions we wish to employ, the sum is taken over all labeled points and the minimization is considered over the space of coefficients $a = (a_1, \ldots, a_p)^T$. The solution is given by

$$a = (E_{lab}^T E_{lab})^{-1} E_{lab}^T c \tag{1}$$

where $c = (c_1, \ldots, c_s)^T$ and E_{lab} is an $s \times p$ matrix whose i, j entry is $e_j(i)$. For the case of several cases, a one-against-all classifier is build for each individual class.

Step 4: [*Classifying unlabeled points*] If x_i, $i > s$ is an unlabeled point, we put

$$c_i = \begin{cases} 1, & \text{if } \sum_{j=1}^{p} a_j e_j(i) \geq 0 \\ -1, & \text{if } \sum_{j=1}^{p} a_j e_j(i) < 0 \end{cases} \tag{2}$$

When there are several classes, the one-against-all classifiers compete using $\sum_{j=1}^{p} a_j e_j(i)$ as a confidence measure.

4 Dataset

We used the USPS handwritten digit dataset for our purposes of both training and testing our algorithms. This dataset consists of 8 bit gray-scale images of 0 through 9 with 1100 examples of each class. In order to create a sequence of digits we concatenated five digits randomly as seen in Figure 3. As can be seen from Figure 2, none of the digits are normalized with respect to slope and slant. We decided to use these images without normalizing for both our approaches. For our experiments, we used 100 zip codes for training the HMM and 900 zip codes for testing it.

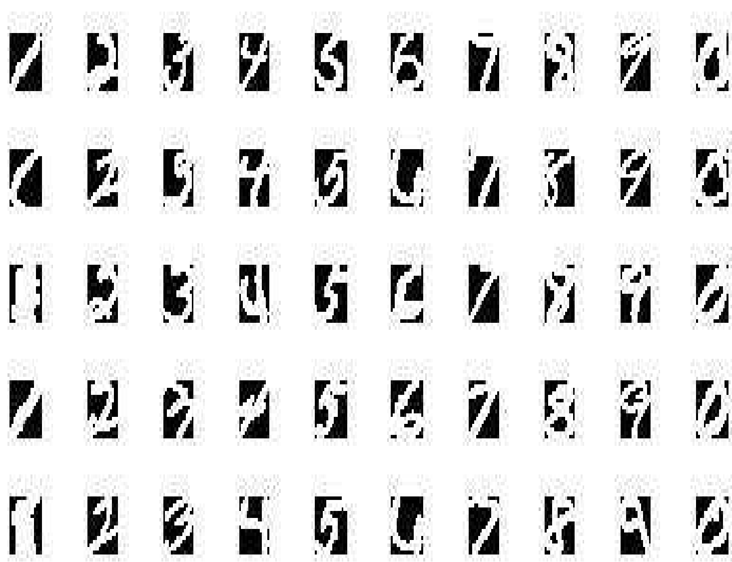

Figure 2: Some of the characters in the dataset

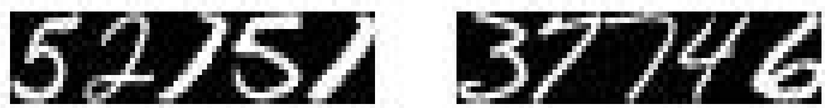

Figure 3: Some of the concatenated characters obtained from the dataset

5 Experimental Results

Our initial experiments were performed using our first approach on the two moons dataset. In this experiment, points were sampled from the two moons, and these were then used to compute the graph Laplacian and the embedding. These embeddings were then fed as a sequence to the HMM for training and classification. Figure 4 shows one such sample of the training data.We used the HMM using mixture of Gaussians observations defined in the Bayes Net Toolbox(BNT) [2].

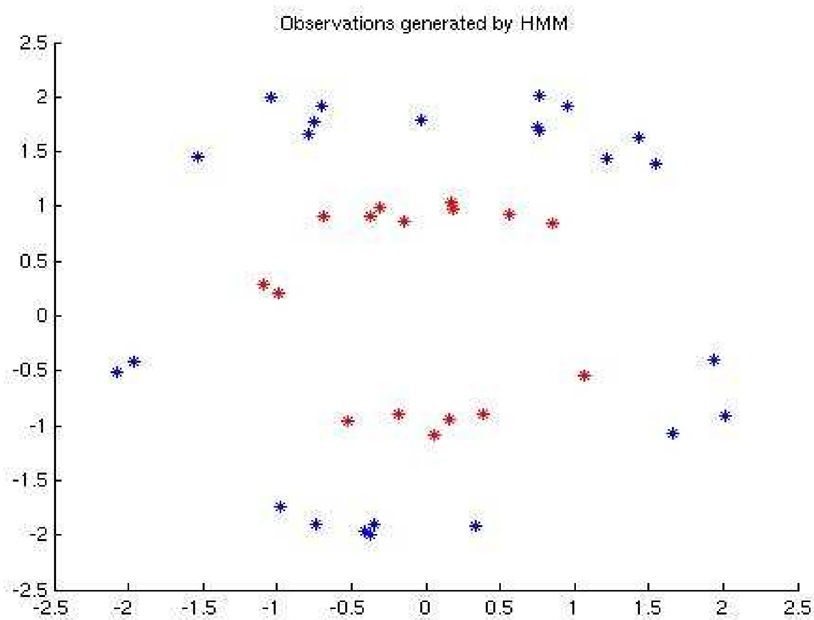

Figure 4: A sampling of the training data

We tried two different test. In the first case, we applied Viterbi to the actual data points and then used it for classification. As can be seen from Figure 5, the classifier is not able to correctly distinguish the two circles. However, when the HMM is learned on the embeddings, the two circles are easily separated out as shown in Figure 6. This is because the two moons dataset lies on a low-dimensional manifold which is exploited by the graph based algorithms.

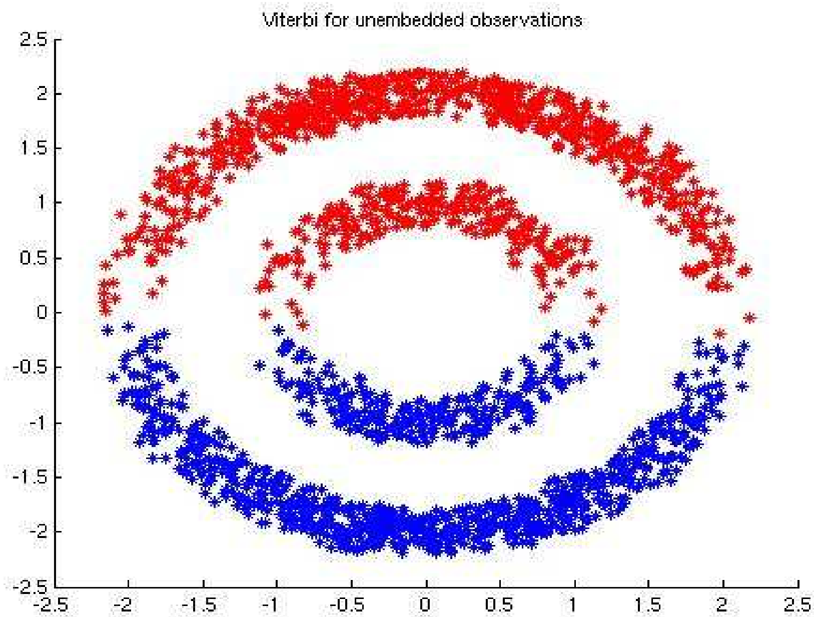

Figure 5: Classification when using the actual data points

This algorithm however did not work in the case of the digits, this is because in the manifold obtained by taking embedding of the image slices, the digits were not at all separable and seemed to be completely random. Hence, fitting a mixture of gaussians did not work. These experiments however convey the point that if the data lies in a manifold, combining Laplacian techniques with HMM can lead to good classification accuracy.

In our second approach, where the segmentations obtained from a trained HMM was used along with labeled digits to perform label propagation. As can be seen from table 5, we get reasonable accuracy with very few number of labeled points per digit. Moreover, the accuracy increases with the number of labeled points. Since, the unlabeled segments used for learning the manifold contained

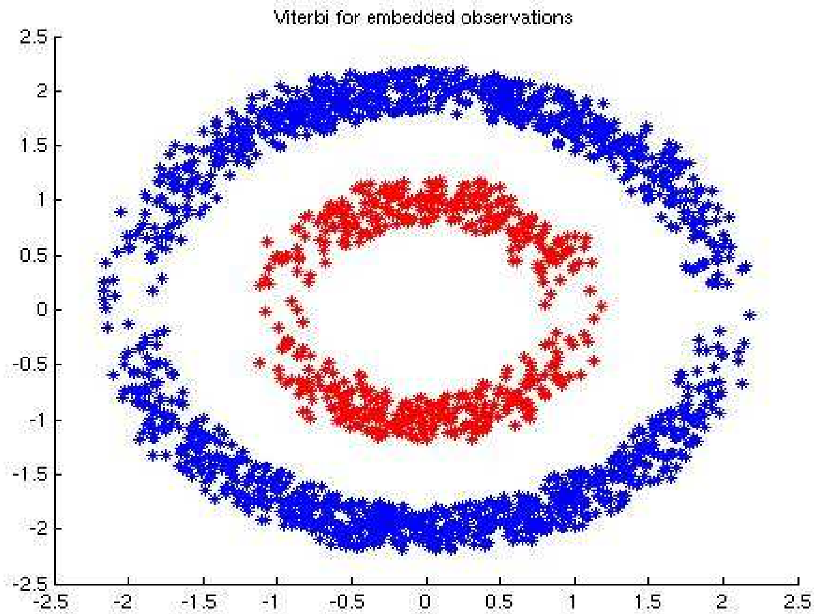

Figure 6: Classification using the embedded data points

Labeled Points	1	5	10	100
Overall Accuracy	43.3%	62.2%	69%	76.1%
Accuracy on Training segments	50.8%	75.8%	82.4%	92.8%

Table 1: Performance of second approach

the test and training segments obtained from the HMM, the accuracy seems to be lower. However, when label propagation was performed on the correct segments used for training the HMM, we got a classification accuracy of 92.8% using only 100 labeled points per digit.

Figure 7 shows some of the correct images obtained after label propagation. Figure 8 shows some of the incorrect segmentations found by the HMM.

6 Conclusions

This report presents two ways of combining graph based algorithms with Hidden Markov Models for handwritten zip code recognition. We have shown how graph based algorithms exploit the underlying

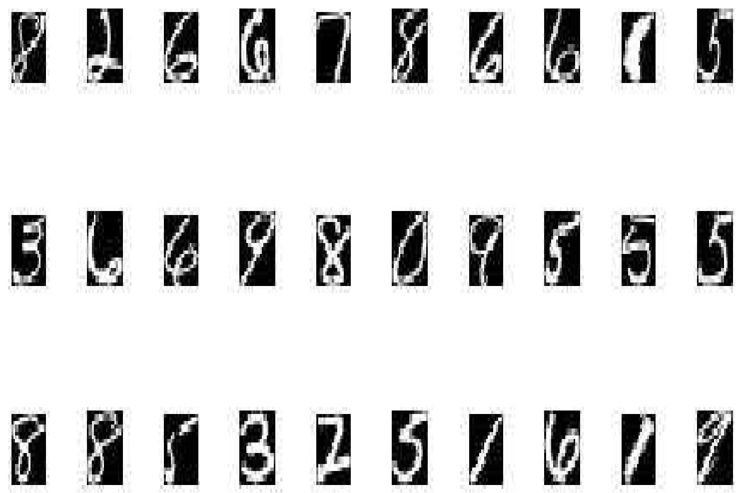

Figure 7: Some of the correct labels obtained after Label Propagation

manifold structure of the data for learning and classification. This when combined with an HMM can be a powerful learning and classification technique.

7 Future Work

Both the ideas in this report revolve around using a HMM for segmenting the dataset. As a future work, it might be worthwile considering using graph based algorithms for segmentation in a semi-supervised framework. One can modify the label propagation algorithm by having labels indicate whether its the start, middle or end of a segment, and using this to then populate the graph. It will be interesting to note how the algorithm fairs in such a situation.

References

[1] M. Belkin, P. Niyogi, "Using Manifold Structure for Partially Labelled Classification", *Neural Information Processing Systems 15 (NIPS'2002)*, pp. 929-936, 2003.

[2] K. Murphy, "The Bayes Net Toolbox for Matlab", *Computing Science and Statistics*, vol. 33, 2001.

[3] L. R. Rabiner, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition", *Proc. IEEE*, Vol. 77, No. 2, pp. 257-286: February 1989.

Figure 8: One of the incorrect segmentation found by HMM

[4] S. Mahadevan, "Proto-Value Functions: Developmental Reinforcement Learning", ICML 2005

[5] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: analysis and an algorithm", *NIPS 14, 2001.*

[6] J. Edwards, D. Forsyth, "Searching for Character Models", *NIPS 2005.*