# Classification Models for Historical Manuscript Recognition

S. L. Feng, R. Manmatha *
*Multimedia Indexing and Retrieval Group*
*Center for Intelligent Information Retrieval*
*University of Massachusetts*
*Amherst, MA, 01003*
[*slfeng,manmatha*]*@cs.umass.edu*

## Abstract

*This paper investigates different machine learning models to solve the historical handwritten manuscript recognition problem. In particular, we test and compare support vector machines, conditional maximum entropy models and Naive Bayes with kernel density estimates and explore their behaviors and properties when solving this problem. We focus on a whole word problem to avoid having to do character segmentation which is difficult with degraded handwritten documents. Our results on a publicly available standard dataset of 20 pages of George Washington's manuscripts show that Naive Bayes with Gaussian kernel density estimates significantly outperforms the other models and prior work using hidden Markov models on this heavily unbalanced dataset.*

## 1. Introduction

This paper investigates a number of different machine learning models for the task of historical handwritten document recognition, which include support vector machines, conditional maximum entropy models, and Naive Bayes. Our goal is to investigate the behaviors and properties of these models when trying to solve this problem.

Although handwritten document recognition is a classical vision problem and has been researched for a long time, it is far from being solved. Good results have been achieved for online handwriting recognition, which takes full advantages of the dynamic information in strokes obtained using special input devices like tablets. However, dynamic information is unavailable for huge volumes of precious handwritten documents, for example, George Washington's letters at the Library of Congress, Issac Newton's papers at Cambridge University and the collection of Joseph Grinnell in the Museum of Vertebrate Zoology at U.C.Berkeley. Efficiently accessing and reading them requires advanced offline handwriting recognition techniques. Off-line handwriting recognition is a harder problem and only successful in small-vocabulary and highly constrained domains such as mail sorting and check processing. A lot of work in handwriting recognition is done at a character-level, which requires to determine character boundaries - since character boundaries are difficult to determine this is done by jointly segmenting and recognizing the characters. In this paper, we directly recognize the entire word without character segmentation, as [4] did, and the recognition problem is formulated as a multi-class classification problem on a large-vocabulary. Classification models are investigated on how to accommodate them to the specific task.

Results from information retrieval [1] show that for print optical character recognition (OCR), the retrieval performance doesn't drop significantly even for high word error rate. By analogy although the output will not satisfy the standard for human reading, we believe it is useful for handwriting retrieval based on text queries.

### 1.1. Related Work

Although online handwriting recognition has advanced to the level of commercial application, offline handwriting recognition has only been successful in small-vocabulary and highly constrained domains. Only very recently people have started to look at offline recognition of large vocabulary handwritten documents [3]. Marti et al [5] proposed to use a Hidden Markov model (HMM) for handwritten material recognition. Each character is represented using a Hidden Markov model with 14 states. Words and lines are modelled as a concatenation of these Markov models. A statistical language model was used to compute word bigrams and

* This work was supported in part by the Center for Intelligent Information Retrieval and in part by the National Science Foundation under grant number IIS-9909073 and in part by SPAWARSYSCEN-SD grant number N66001-02-1-8903.

this improved the performance by 10%. Rath et al [4] focuses on recognizing historical handwritten manuscripts using simple HMMs one state for each word. By adding word bigrams from similar historical corpora they showed that the performance could approach an accuracy of 60%

## 2. Classification Models for Handwritten Word Recognition

We now discuss a number of different classification models for handwriting recognition - both discriminative and generative.

### 2.1. Support Vector Machines

Originally introduced as a binary linear classifier, support vector machines (SVMs) attempt to find an oriented hyper-plane which separates the linear separable space defined by the training data, while maximizing the margin. The margin is the distance of each training instance to the hyperplane.

To extend this to classifying nonlinear separable data, SVM uses a *kernel* function $K$ to map the training data to a higher Euclidean space, in which the data may be linearly separable. The kernel function is defined as : $K(x_i, x_j) =< \phi(x_i), \phi(x_j) >$, where $\phi(x)$ is some mapping. To dealing with nonseparable data and avoid overfitting, SVM usually use a soft margin which allows some instances to be misclassified. A SVM classifier solves the optimization problem:

$$\min_{\xi,w,b} < w, w > +C \sum_{i=1}^{N} \xi_i \tag{1}$$

such that $y_i(< w, \phi(x_i) > +b) \geq 1 - \xi_i$, and $\xi_i \geq 0$,

where $y_i \in \{-1, 1\}$ is the label of instance $x_i$, slack variable is $\xi_i$ and the capability $C$ determines the cost of margin constraint violations.

### 2.2. Conditional Maximum Entropy Models

Maximum Entropy models have been recently widely applied in domains involving sequential data learning, e.g. natural languages [7, 8], biological sequence analysis [2], and very promising results have been achieved. Since maximum entropy models utilize information based on the entire history of a sequence, unlike HMM whose predications are usually based only on a short fixed length of prior emissions, we expect maximum entropy models to work well for handwritten document recognition problems since in our case each page may be taken as a long sequence of words, each of which emits a set of observations represented as word image features.

The goal of conditional maximum entropy models is to estimate the conditional distribution of label $y$ given data $x$, say $P(y|x)$. The framework is fairly straightforward. It basically specifies that the modeled distribution should be as uniform as possible, while being consistent with the constraints that are given by the features of the training data. Given a set of predicates. [1] $f_i(x, y)$, which may be real or binary values and represent some observation properties (e.g. co-occurence) of the input $x$ and output $y$, the constraints are that for each predicate its expectation value under the model $P(y|x)$ should be the same as its expectation under the empirical joint distribution $\tilde{P}(x, y)$, i.e.

$$\sum_{x,y} \tilde{P}(x)P(y|x, \lambda)f(x, y) = \sum_{x,y} \tilde{P}(x, y)f(x, y) \tag{2}$$

With these constraints, the maximum conditional entropy principle picks the model maximizing the conditional entropy:

$$H(P) = - \sum_{x \in X, y \in Y} \tilde{P}(x)P(y|x, \lambda) log P(y|x, \lambda) \tag{3}$$

It has been shown [9] that there is always a unique distribution that satisfies the constraints and maximize the conditional entropy. This distribution has the exponential form:

$$P(y|x, \lambda) = \frac{1}{Z} e^{\sum_i \lambda_i f_i(x,y)} \tag{4}$$

where $Z$ is a normalization constant such that $\sum_y P(y|x, \lambda) = 1$ and $\lambda_i$ is the weight of predicate $f_i$ in the model.

The maximum entropy model's flexibility comes from the ability to use arbitrary predicate definitions as constraints. These feature definitions represent knowledge learned from the training set. So our test of conditional maximum entropy modeling on our task focuses on the aspect of feature definitions and their effects on performance.

**2.2.1. Predicates** We do a linear vector quantization (VQ) on the original continuous features measured from the images and discretize each of them into a fixed number of bins. We define two types of binary predicate for the maximum entropy model based on the discrete features extracted from word images and the corresponding label sequence:

1. **Unigram Predicates** The frequency of a discrete feature $x$ and the current word $w$:

$$f_i^u(x, w) = \begin{cases} 1 & \text{if the feature set of } w \text{ contains } x \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

1 We use the term predicates rather than features to differentiate these from image features.

2. **Bigram Predicates** We define two sets of bigram predicates, which intuitively represent the statistical properties of a word and the features of this word's neighboring word images. For example, if in the training set the word "*force*" always follows word "*Fredericksburgh's*", then in the test set it will increase the probability of current word being recognized as "force" given that its previous word image is very long. One set of bigram predicates we defined is the frequency of word $w$ and a discrete feature $x$ which appears in the feature set of the preceding word image of word $w$:

$$f_i^{bf}(x,w) = \begin{cases} 1 & \text{if the feature set of the previous} \\ & \text{word image of } w \text{ contains } x \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

and the other set is the frequency of word $w$ and a discrete feature $x$ which appears in the features set of the following word image of word $w$:

$$f_i^{bb}(x,w) = \begin{cases} 1 & \text{if the feature set of the following} \\ & \text{word image of } w \text{ contains } x \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

### 2.3. Naive Bayes with Gaussian Kernel Density Estimate

Since our dataset is from letters which use natural language, it is unbalanced. That is, since word frequencies follow a Zipfian-like distribution their frequencies vary widely. On the other hand the dataset also provides us with reasonable prior probabilities of words in the document corpus. So instead of discriminative models like SVMs and maximum entropy, we want to use some kind of generative probability density model like Naive Bayes.

The Naive Bayes framework is pretty simple:

$$P(w|f) = \frac{P(f|w)P(w)}{\sum_w P(f|w)P(w)} \tag{8}$$

We estimate the prior probability of word $w$ directly as its relative frequency in the training set. We calculate the probability of the visual features of a word image given a word $w$, using a non-parametric Gaussian kernel density estimate:

$$P(f|w) = \frac{1}{\|w\|} \sum_{i=1}^{\|w\|} \frac{\exp\{-(f-f_i)^T \Sigma^{-1}(f-f_i)\}}{\sqrt{2^k \pi^k |\Sigma|}} \tag{9}$$

This equation arises out of placing a Gaussian kernel over the feature vector $f_i$ of every word image labelled as word $w$. Each kernel is parametrized by the feature covariance matrix $\Sigma$. We assumed $\Sigma = \beta \cdot I$, where $I$ is the identity matrix and $\beta$ plays the role of kernel bandwidth, which determines the smoothness of $P(f|w)$ around the support points $f_i$. The value of $\beta$ is selected empirically on a validation set.

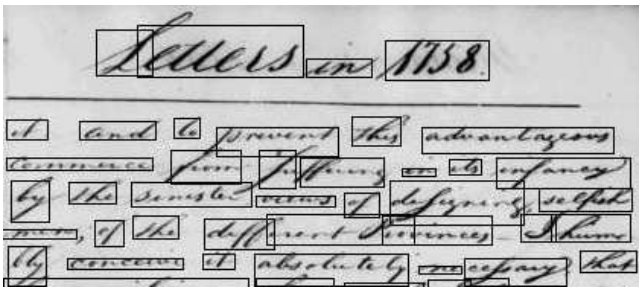

**Figure 1. A part of one segmented page in our dataset.**

## 3. Experimental Results

Our evaluation dataset consists of 20 pages from a collection of letters by George Washington. This is a publicly available standard dataset provided by [4]). Each page is accurately segmented into individual word images, each of which has been manually transcribed. We don't lowercase transcribed words, so "*region*" and "*Region*" are taken as two different words. There are 4865 words in the corpus in total and 1187 of them are unique. Figure 1 shows a part of a segmented page in our dataset.

27 features are extracted from each word image, which consists of 6 scalar features(i.e. height, width, aspect ratio, area, number of descenders in the word, and number of ascenders in the word) and 21 profile-based features which obtained through a discrete Fourier Transform(DFT) over three time series generated from the word image, which are projection profile, upper word profile and lower word profile. Please refer to [4] for the feature details.

We use word accuracy rate as our performance measure, i.e. the proportion of the words that are recovered exactly as they were in the manual transcript. 20-fold cross-validation is used to get a stable performance evaluation for each model. Each iteration leaves one page for test, and trains the model over the other 19 pages. We use the mean accuracy rate as the final evaluation measure. Since our dataset is relative small, many words in the test set don't occur in any training pages - these are called out-of-vocabulary(OOV) terms as in [4] and cause errors of the recognition, we use two types of mean accuracy rate – mean accuracy rate with OOVs and mean accuracy rate without OOVs.

Since our data are from a collection of natural language documents(letters), the frequency of words can be approximated by a Zipf distribution, in which a few words have very high frequencies, and most words occur very infre-

quently. Over our whole dataset, 681 words has only one occurrence; 1008 words have less than 5 occurrences each but 30 words have 1856 occurrences in total. The unbalance and sparsity of training data for different words make the multi-classification problem untractable for some standard classifiers such as decision trees, neural networks, as shown in [6]. Here, we investigate the three models in 2 and their behaviors when dealing with this unbalanced data problem.

### 3.1. Results on Different Models

**3.1.1. SVMs** We use the *MATLAB Support Vector Machine Toolbox* developed by Gavin Cawley to build the SVM model on the data. By using the 'max wins' algorithm, we tried linear kernels and polynomial kernels of degree 2 on the data.

| Accuracy | with OOV | w/o OOV |
|---|---|---|
| Linear Kernel | 0.3827 | 0.4642 |
| Polynomial d-2 | 0.4463 | 0.5281 |

**Table 1. Experimental results using SVMs**

Table 1 shows the experimental results using support vector machines, from which we see that the polynomial kernel performances much better than the linear kernel. This is unsurprising since the kernel function plays a crucial role in SVMs. The kernel determines the mapping of instances to a high dimensional space, whether the space is separable or not. However, generally it is not easy to locate the proper kernel. In other word, deciding to which space the original data should be projected requires a deeper understanding of the data, usually background knowledge needed. In our case, both the linear kernel and the polynomial kernel of degree 2 don't work very well on the data. Other kernels that project the data into higher dimension spaces might help in this case but there is no simple way to determine these short of trying all of them.

**3.1.2. Conditional Maximum Entropy Models** We use the maximum entropy toolkit from *http://homepages.inf.ed.ac.uk/s0450736/maxent-toolkit.html*, which was developed in C++ based on the java version *maxent.sf.net*. To extract unigram and bigram discrete predicates in section 2.2, we linearly quantize each of the 27 continuous features into 19 bins. To test the influence of different numbers of bins into which the raw features is quantized, we also gradually change the number of bins and re-run the maximum entropy model. The performance only varies slightly with the change of the number of bins except at 100 bins the performance drops sharply.

| Accuracy | with OOV | w/o OOV |
|---|---|---|
| Unigram | 0.4164 | 0.4939 |
| Unigram + Bigram | 0.4432 | 0.5234 |

**Table 2. Performance Comparisons for maximum entropy models and features**

Table 2 shows the results of Maximum Entropy models using discrete predicates. These number shows using both unigram and bigram information outperforms only using unigram information by a small margin. Further experiment is needed to determine whether higher-order dependency information(e.g. trigram) is helpful. Note the concept of bigram here is defined between label states and features unlike that in HMM which depicts the dependency between label spaces. Since our dataset is relative small and the vocabulary is huge, it is more difficult to capture useful bigram information for maximum entropy.

**3.1.3. Naive Bayes with Gaussian Density Estimate** Our best results are achieved using the Naive Bayes model with Gaussian kernel density estimates, a mean accuracy of 0.542 with OOVs and 0.640 without OOVs. It is not surprising that Naive Bayes achieves good results on our task for at least two reasons. One is that the model provides prior probabilities of the words - that is the frequency of the words. This corresponds to unigram language model information used in [4] where it was shown to improve performance. Another is that the Gaussian density emphasizes the local information provided by each instance, which has been shown to be very useful in multimedia data analysis.

### 3.2. Results Summary

| Accuracy Rate | with OOV | w/o OOV |
|---|---|---|
| SVM | 0.446 | 0.528 |
| ME | 0.443 | 0.523 |
| Naive Bayes with GD | 0.542 | 0.640* |
| HMM (in [4]) | 0.497 | 0.586 |

**Table 3. Results of comparing all the models**

Table 3 shows the performance comparisons of all the models we test. The numbers shown in this table are the best accuracy rate we achieved for each kind of model. HMM results are from the recent paper [4]. To make the comparison fair the results for the HMM model include word bigrams obtained from the training set but not from the external corpora (the Naive Bayes model as well as the other models here do not use any bigrams). Using an external Thomas

Jefferson electronic text corpus boosts the HMM performance to 0.52 and 0.61 respectively [4]. From this table, we see that Naive Bayes model with a Gaussian density estimate achieved the best performance on our task. The t-test shows it outperform HMM significantly by a P-value of 0.01.

The unbalance of our dataset is a disadvantageous condition for both SVM and maximum entropy models. For many words the models starve for training data, while for some other words abundant training samples are available. When SVM dealing unbalanced data, even margins for negative instances and positive instances may be inappropriate. Uneven margins [10] for SVM may alleviate the effects of unbalance of the dataset.

Kernel selection is the key to SVM once the feature sets have been fixed. There still aren't very good theoretical methods for automatic selection of kernel functions for SVM. Although the upper bound on VC dimension is potentially useful for comparing kernels, it is necessary to estimate the radius of a hypersphere enclosing the data in the non-linear feature space, which is a very difficult task. So cross-validation is still a preferred method for selecting kernels. Kernels should accommodate to a specific task, and the specific data. In our case, each manuscript page could be considered as a sequence instance of word images. So some kernels for structure learning, e.g. Fisher kernels, may be more suitable for our task since these kernels can learning the dependency among the state space.

The performance of maximum entropy depends directly on the predicates defined for this model, which determine what information will be captured from the training data as knowledge constraints for the model. Since it is easy for maximum entropy to use information based upon of the whole sequence through predicates definition, maximum entropy is expected to perform well on sequence data. But the models should have enough training data to capture accurately high-order information. When only sparse training data available, high-order(n-gram) predicates may cause very biased estimation.

The improved performance of naive Bayes over other models in our experiments shows that, the prior probabilities (unigram information) is important for analysis on natural language document corpus (especially heavily unbalanced datasets). In contrast, prior distribution information is difficult to utilize in other discriminative models such as maximum entropy and SVM. Gaussian density estimates also show that localized models and local information are preferable for handwriting recognition. Such local information is suitable for many multimedia data problem in which each category could be a mixture of different patterns.

## 4. Conclusions and Future Work

We investigate and compare a set of machine learning models for the historical manuscripts recognition problem, including support vector machines, conditional maximum entropy models and Naive Bayes with Gaussian kernel density estimates. In the future, we will try other graphical models, for example conditional random field models. These models are more suitable for sequence data, in which the transition information between labels are important. As shown in [4], after they used bigram information, the performance improved substantially.

## 5. Acknowledgements

We thank Toni Rath and Victor Lavrenko for providing us with all the dataset, features and his previous results.

## References

[1] S. M. Harding, W. B. Croft and C. Weir Probabilistic Retrieval of OCR Degraded Text Using N-Grams, In *Proc. of the 1st European Conference on Research and Advanced Technology for Digital Libraries.* Pisa, Italy, September 1-3, 1997, pp. 345-359.

[2] Eugen C. Buehler, Lyle H.Ungar Maximum Entropy Methods for Biological Sequence Modeling, In *Workshop on Data Mining in Bioinformatics of KDD01*, 2001.

[3] A. Vinciarelli, S. Bengio and H. Bunke Offline Recognition of Large Vocabulary Cursive Handwritten Text. in *Prof. of the 7th Intl Conf. on Document Analysis and Recognition*, vol. 1. Edinburgh, Scotland, August 3-6, 2003, pp. 1101-1105..

[4] Lavrenko, V., Rath, T. and Manmatha, R., Holistic Word Recognition for Handwritten Historical Documents in *the Proceedings of Document Image Analysis for Libraries (DIAL)*, 2004,pp. 278-287.

[5] U.-V. Marti and H. Bunke Using a Statistical Language Model to Improve the Performance of an HMM-Based Cursive Handwriting Recognition System. in *ntl Journal of Pattern Recognition and Artifi cal Intelligence*, I 15:1 (2001) 65-90.

[6] N. Japkowicz and S. Stephen The class imbalance problem: A systematic study in em Intelligent Data Analysis 2002,6.5

[7] A. Berger, S. D. Pietra, and V. D. Pietra A maximum entropy approach to natural language processing in *Computational Linguistics*, (22-1), March 1996

[8] R. Rosenfeld A maximum entropy approach to adaptive statistical language modelling in *Computer, Speech and Language* , 10:187–228, 1996

[9] A. Ratnaparkhi A Simple Introduction to Maximum Entropy Models for Natural Language Processing Technical Report 97-08, Institute for Research in Cognitive Science, University of Pennsylvania, 1997

[10] Y. Li, H. Zaragoza, R. Herbrich, J. Shawe-Taylor, J. Kandola The Perceptron Algorithm with Uneven Margins in *the 2002 Proceedings of the International Conference of Machine Learning* 379–386