Improved Classification through Runoff Elections

Oleg Golubitsky
Google, Inc.
340 Hagey Blvd.
Waterloo, Ontario, Canada
oleg.golubitsky@gmail.com

Stephen M. Watt
University of Western Ontario
Dept of Computer Science
London, Ontario, Canada
watt@uwo.ca

ABSTRACT

We consider the problem of dealing with irrelevant votes when a multi-case classifier is built from an ensemble of binary classifiers. We show how run-off elections can be used to limit the effects of irrelevant votes and the occasional errors of binary classifiers, improving classification accuracy. We consider as a concrete classification problem the recognition of handwritten mathematical characters. A succinct representation of handwritten symbol curves can be obtained by computing truncated Legendre-Sobolev expansions of the coordinate functions. With this representation, symbol classes are well linearly separable in low dimension which yields fast classification algorithms based on linear support vector machines. A set of 280 different symbols was considered, which gave 1635 classes when different variants are labelled separately. With this number of classes, however, the effect of irrelevant classifiers becomes significant, often causing the correct class to be ranked lower. We introduce a general technique to correct this effect by replacing the conventional majority voting scheme with a runoff election scheme. We have found that such runoff elections further cut the top-1 mis-classification rate by about half.

1. INTRODUCTION

The problem of online handwriting recognition is becoming increasingly important, particularly because of the emergence of new hand-held mobile devices, for which pen-based input is often more convenient than typing. As every effort is made to decrease the production cost of such devices, their speed and memory capacity are usually by orders of magnitude lower than those of conventional PCs. Yet it is crucial for the recognition process not to cause delays that would be noticeable by the users. Our goal is to develop robust online handwriting recognition algorithms, which process the input data immediately as it is received from the digital pen, thereby reducing the delay after pen up to a minimum.

The process of online handwritten character recognition can be thought of as that of classification of parametric plane curves. Initially, these curves are given by sequences of points, which are sampled by the digital pen in real time. The sampling rate and resolution may vary for different devices. Other information, such as pen up/down status, pressure, angle, *etc.*, may or may not be available, depending on the device. In order to remain device-independent, we will only use the x and y coordinates of the points and rely on methods that are insensitive to the sampling rate and resolution.

Our special interest is in the recognition of handwritten mathematics. Fast and robust recognizers of handwritten mathematical formulae would significantly enhance the user interface of the already wide-spread computer algebra, scientific computing and technical document processing systems. This would allow a large community of scientists and engineers to more freely communicate mathematics to each other and open new possibilities for remote collaboration.

The following peculiarities of the problem of mathematical handwriting recognition distinguish it from the problem of recognition of handwritten text. The number of symbol classes in mathematics is about 300, more than in European, but less than in some Asian languages. Symbols tend to have only a few strokes and are distinguished by how the strokes curve. There is no fixed vocabulary for mathematical formulae, yet some subformulae do occur more often that others. Symbols in formulae are generally better segmented than in text. Formulae have a two-dimensional layout, and symbols in them may greatly vary in size. In this paper, we consider the problem of online recognition of individual mathematical symbols. Our approach is based on the representation of the character curves by a truncated expansion of their coordinate functions in an orthogonal functional basis. Earlier work [1] has shown that truncated Chebyshev series of order 10 approximate well most handwritten mathematical character curves. The same applies to the Legendre series, yet the latter have the advantage of being computable on-line, as the curve is written [2]. Legendre-Sobolev series is just as easy to compute as Legendre series, yet provides a much more accurate distance measure [3].

It has been shown [4] that, when symbols are represented by truncated Legendre-Sobolev series as points in a vector space of dimension 20 to 30, symbol classes are well linearly separable. For such classes, we can effectively use binary linear classifiers, which are fast to learn (e.g., with linear support vector machines) and very fast to apply. Moreover, classification with respect to the hyperplanes can also be done

DAS '10, June 9-11, 2010, Boston, MA, USA

on-line, as the curve is written [4]. This is especially useful when the number of classes is large, because the number of binary classifiers depends quadratically on the number of classes.

The intuitive reason behind good linear separability of symbol classes is the following: If we take two samples from a particular class and morph one sample into the other using linear homotopy, the intermediate characters will usually belong to the same class. As we gradually morph the first character into the second, the corresponding feature vector in the space of truncated Legendre-Sobolev series traces a straight line segment. Therefore our classes, when mapped to this space, satisfy the property that, whenever two points belong to a class, the entire segment connecting them is contained in that class as well. This means that classes are convex sets. It is a well-known fact that any pair of disjoint convex sets is linearly separable.

Earlier experimental results generally confirmed this intuition, showing that symbol classes are 96.5% linearly separable [4]. However, a significant number of samples (the remaining 3.5%) have still fallen on the wrong side of a separating hyperplane. In this paper, we show that these errors are due to the presence of different variants (also called *allomorphs*) of a symbol in the same class. If two symbols belong to the same class but are written in different ways (e.g., one clockwise and the other counterclockwise), linear homotopy from one symbol to the other may produce strange intermediate curves that do not belong to this class anymore and may even belong to other symbol classes. Therefore, for better linear separability and more accurate classification, allomorphs should be labeled, and different linear classifiers should be trained for pairs of allomorph classes, rather than pairs of symbol classes. In this paper we show for a dataset of about 50,000 handwritten mathematical symbols that subclassing into allomorphs leads to nearly 100% linearly separable classes and improves the correct retrieval rate by about 3%, to about 85%.

It turns out that the remaining classification errors are due to at least two different factors. First, each individual binary classifier, when trained on an incomplete set of samples (which is always the case in practice), produces a certain number of classification errors. Second, the majority voting scheme for N classes, which combines $N(N-1)/2$ binary classifiers into a multi-class classifier, adds a certain amount of randomness to the final decision. Indeed, suppose that a sample belongs to class A. Following [6], let the binary classifiers involving class A be called *relevant* for this sample, and the remaining classifiers *irrelevant*. If all relevant classifiers conclude in favor of class A, this class wins the majority vote. However, if some of them fail, another class may win the majority vote because of a coincidental consensus of the irrelevant classifiers. Note that the proportion of irrelevant votes grows with the number of classes, and our experiments confirm that correct retrieval rates drop accordingly. To reduce the influence of the irrelevant binary classifiers, we propose the following runoff election scheme. First, determine the top K classes using conventional majority voting. Then, run a second round of majority voting among only the top K classes. Our experiments with this scheme show an increase in correct retrieval rates by about 6%, compared to the conventional majority voting. This runoff election scheme may apply equally to other settings where binary classifiers are combined.

This paper is organized as follows. In Section 2 we outline the theory of Legendre-Sobolev expansions and describe the algorithm for computing them. In Section 3 we discuss in detail the properties of convexity and linear separability for symbol classes. We show some examples when the presence of different allomorphs in the same class leads to nonconvexity and causes linear classifiers to fail. In Section 4 we describe the dataset used in our experiments and present the results of cross-validation analysis for classes with allomorph labels. In Section 5 we discuss the runoff election scheme and present the results of comparison with the majority voting scheme. Section 6 concludes the paper.

2. LEGENDRE-SOBOLEV SERIES

As mentioned, the problem of online handwritten character recognition is that of classification of parametric plane curves. A parametric plane curve is given by a pair of its coordinate functions $x(\lambda), y(\lambda)$, where λ is an increasing parameter that ranges over a certain interval, such as time or arc length. The digital pen samples the values of these coordinate functions at a certain fixed rate and returns them in real time. Using these values, a numeric feature vector representing the curve in a finite-dimensional Euclidean space may be constructed and used to classify the curve among N symbol classes.

The choice of the numeric feature vector is the key step in this process. Earlier work [1, 2, 3] proposes to represent the curve by the coefficients of the truncated expansions of the coordinate functions with respect to a certain orthogonal functional basis. Various choices of the functional inner product and the corresponding orthogonal polynomial bases have been studied, in particular Chebyshev, Legendre, and Legendre-Sobolev inner products and bases. It has been established that, for any of these bases, truncated series of order about 10 provide accurate approximations to most character curves, which to a human eye are indistinguishable from the original curves. Moreover, the coefficients of such series are insensitive to noise and to the parameters of the ink sampling device (such as sampling rate and resolution). The Legendre and Legendre-Sobolev bases have the advantage that the corresponding truncated series can be computed on-line, as the curve is written. Therefore, the time spent on the computation of the feature vector after pen-up is reduced to a minimum.

Formally, the Legendre-Sobolev inner product of two functions $f(\lambda)$ and $g(\lambda)$ defined on an interval $[a, b]$ is given by

$$\langle f, g \rangle = \int_a^b f(\lambda) g(\lambda)\, \mathrm{d}\lambda + \mu \int_a^b f'(\lambda) g'(\lambda)\, \mathrm{d}\lambda.$$

Here μ is a numeric parameter whose value can be chosen experimentally. For $\mu = 0$, we obtain the Legendre inner product. For a fixed interval $[a, b]$ and value of μ, the Legendre-Sobolev basis consists of the polynomials $B_0(\lambda)$, $B_1(\lambda)$, $B_2(\lambda)$, ..., one per polynomial degree $0, 1, 2, \ldots$, which are mutually orthogonal with respect to the Legendre-Sobolev inner product. We also assume that the polynomials are normalized. This completely specifies the B_i. Any function sat-

isfying certain general conditions can be represented by its infinite Legendre-Sobolev series $f(\lambda) = \sum_{i=0}^{\infty} c_i B_i(\lambda)$, where the coefficients are given by the inner products $c_i = \langle f, B_i \rangle$, and approximated by the truncated Legendre-Sobolev series of order d: $f(\lambda) \approx \sum_{i=0}^{d} c_i B_i(\lambda)$. This approximation is the projection of the function $f(\lambda)$, viewed as an element of the infinite-dimension functional vector space, onto the $(d+1)$-dimensional space of polynomials of degree at most d. In other words, it is equal to the polynomial of degree at most d that is the closest to the function $f(\lambda)$ in terms of the Legendre-Sobolev norm (induced by the Legendre-Sobolev inner product). Thus, the approximation is good if there exists a polynomial of degree d that approximates function $f(\lambda)$ well. This is the case if $f(\lambda)$ and its derivatives up to a certain order are smooth and uniformly bounded, and if $f(\lambda)$ has less than d extrema. The coordinate functions $x(\lambda)$ and $y(\lambda)$ of handwritten character curves appear to satisfy these requirements for $d \approx 10$. It turns out that the coordinate functions are smooth even if the curve has cusps and corners, because the pen slows down at such singularities. This is the intuitive reason behind the experimental results of [1, 2], which confirm that truncated series of order about 10 accurately approximate the coordinate functions of most character curves.

As we saw, the Legendre-Sobolev inner product and basis depend on the domain interval $[a, b]$. In the case of handwritten curves we can set $a = 0$, yet the value of b is unknown until the entire curve is finished, so we cannot directly compute the Legendre-Sobolev coefficients until then. However, notice that, because $B_i(\lambda)$ are polynomials, the inner products $\langle f, B_i \rangle$ can be expressed as linear combinations of the inner products $\langle f(\lambda), \lambda^k \rangle$, $k = 0, \ldots, i$, which can be easily obtained from the moments $m_k = \int_0^b f(\lambda)\lambda^k \, d\lambda$. These moment integrals can be accumulated on-line, as the curve is traced, and scaled to the domain $[0, 1]$ via the substitution $\lambda \mapsto \lambda/b$, once the pen is lifted. This technique [2] reduces the number of arithmetic operations after pen-up to a constant, which is independent of the number of points sampled from the curve.

As a result we obtain a feature vector consisting of the first $(d+1)$ Legendre-Sobolev coefficients for the coordinate functions $x(\lambda)$ and $y(\lambda)$. Note that, since $B_0(\lambda) = 1$, order-0 coefficients are equal to $\int_a^b x(\lambda)\,\mathrm{d}\lambda$ and $\int_a^b y(\lambda)\,\mathrm{d}\lambda$. By setting their value to 0, we can center the curve. Furthermore, the norm of the resulting coefficient vector is proportional to the size of the character; therefore, by normalizing this vector, we normalize the character size. These operations can be performed instantly.

The resulting feature vector (of dimension about 20) can be used for classification. It is well-known that vector-space-based classification methods depend on the quality of the Euclidean distance measure, as a measure of dissimilarity between the characters. Experiments have shown [3] that the Legendre-Sobolev distance measure performs better than the Legendre distance. In fact, for the Legendre-Sobolev distance with $\mu = 1$, the nearest neighbor classification results are almost as accurate as for the much slower elastic matching distance. Note also that the feature vector changes if we choose a different parameterization of the curve. Among three natural parameterizations by time, Euclidean arc length, and affine arc length, the Euclidean arc length turned out to yield the most accurate classification results [3]. Therefore, we chose the vector of Legendre-Sobolev coefficients of the coordinate functions parameterized by arc length, centered and normalized as described above, as a representation of the handwritten character curves. For multi-stroke characters, we joined consecutive strokes by straight lines, in order to obtain a single connected curve, and then compute the feature vector for it.

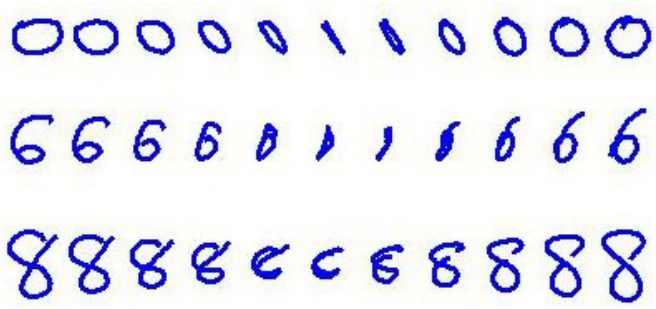


Figure 1: Homotopy between allomorphs produces intermediate curves that look different.

3. CONVEXITY, LINEAR SEPARABILITY, AND ALLOMORPH LABELING

It has been established in [4] that, when character curves are represented by the vectors of their Legendre-Sobolev coefficients, character classes turn out to be 96.5% linearly separable. As mentioned in the introduction, the reason behind the linear separability is that, in most cases, linear homotopy from one character to another character from the same class yields intermediate curves that belong to this class as well. However, this is not necessarily the case, if the two characters are of different allomorphs. Figure 1 shows that, when we morph one allomorph of "8" into another, we can obtain an intermediate curve that looks like a "c".

When a situation like the above occurs, the convex hulls of the two classes will overlap, whence the classes will not be linearly separable (even if the classes themselves do not overlap). It is natural to ask whether subclassing into allomorph classes will help to resolve this problem. Experimental results described below give a positive answer to this question.

We started the experiment with the dataset containing about 50,000 samples of single- and multi-stroke mathematical symbols from about 280 different classes. Symbols that can be attributed to more than one class have been labelled accordingly, by a label that includes the names of all classes to which the symbol belongs. Some of the resulting composite labels overlap. Furthermore, the number of strokes was included in the class label. This resulted in about 1150 subclasses, which were shown to be 96.5% linearly separable in [4]. For the present experiment, we subdivided them further into 1635 allomorph subclasses. The number of allomorphs per class varies from 1 (for about 60% of all classes) to 13 (for the class "eight").

For each pair of classes with non-overlapping composite labels and at least 10 samples, we trained a linear SVM clas-

$K \setminus$ #samples	10	20	30	40	50	60	70
1	78.0%	81.6%	84.6%	86.3%	86.6%	85.6%	86.6%
2	92.3%	94.8%	95.3%	95.8%	96.6%	96.2%	96.6%
3	95.2%	97.9%	96.9%	98.0%	98.7%	98.5%	98.7%
5	96.8%	99.0%	98.2%	98.8%	99.4%	98.9%	99.2%
10	98.9%	99.5%	99.0%	99.3%	99.6%	99.6%	99.7%

Table 1: Top-K correct retrieval rates with majority voting.

$K \setminus$ #samples	10	20	30	40	50	60	70
1 (runoff at top 4)	87.0%	91.0%	89.9%	92.2%	91.4%	92.5%	92.9%
2 (runoff at top 8)	94.2%	96.7%	96.6%	97.2%	97.6%	97.7%	97.9%
3 (runoff at top 15)	96.6%	97.6%	97.3%	98.4%	98.0%	98.8%	98.4%

Table 2: Top-K correct retrieval rates with runoff elections.

sifier [5], using all available samples as training data. Only 623 out of 52,396 samples (about 1.2%) ended up on the wrong side of at least one hyperplane. Moreover, only 125 of these actually had correct labels (the labeling process is manual, and errors are inevitable). After label adjustment, the number of errors dropped to 70 (about 0.14%). Compared to the results of the linear separability test without subclassing into allomorphs, the error rate has dropped by a factor of about 2500. Since 0.14% of samples can be removed from the dataset without affecting our ability to learn the classes, we conclude that the resulting allomorph classes are fully linearly separable.

4. CROSS-VALIDATION ANALYSIS

We use repeated random sub-sampling to test the ability of the linear SVM classifiers to generalize incomplete training sets. For classes with at least 70 samples (which cover 110 original classes of mathematical symbols), we sub-sampled random subsets of size 10, 20, 30, 40, 50, 60, and 70. For each of these subsets, we randomly selected 75% of the elements into training sets and the remaining 25% into test sets. Then, linear SVMs were trained for each pair of allomorph classes with non-overlapping labels (see previous section for a description of class labels).

Each test sample was classified with respect to all separating hyperplanes, and the classes were ranked according to the number of votes they received, with ties broken at random. We say that a test sample is retrieved correctly in top K classes, if the sample's label overlaps with at least one class label that was ranked in top K. A summary of the top-K correct retrieval rates for $\mu = 1$, 111 classes, and $K = 1, 2, 3, 5, 10$ is shown in Table 1. Our experiments with other values of μ, ranging from 2^{-4} to 2^4, produced very similar results; the dependence on the number of classes is discussed in the next section.

While the top-K rates for large K look satisfactory, the gap between the top-1 and top-2 rates suggests that for small K the results could be improved, especially considering that our classes are fully linearly separable. In the following section, we propose to use a different voting scheme, which yields a significant improvement of the top-K rates for small K.

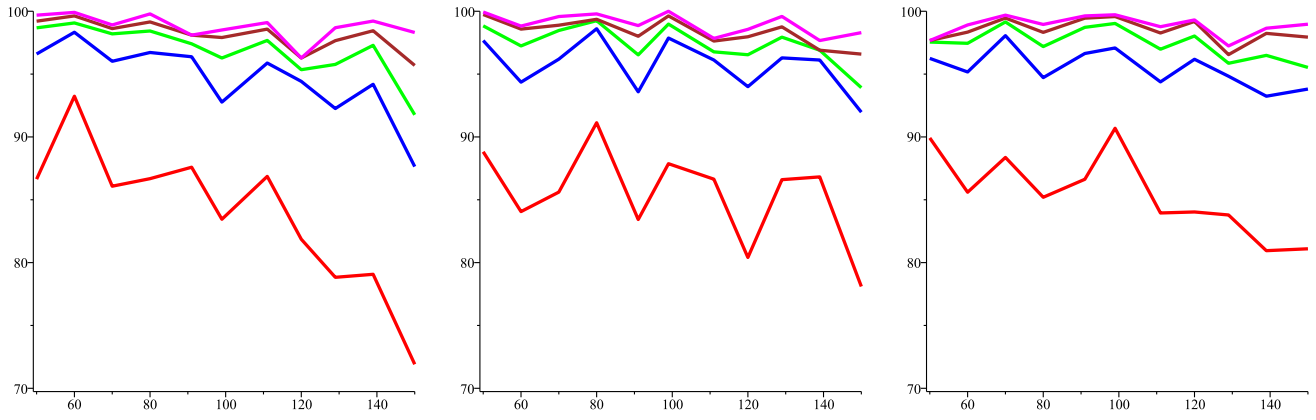
5. THE RUNOFF ELECTION SCHEME

The majority voting scheme suffers from the following drawback: for each particular test sample, most of the hyperplanes that participate in the voting process have nothing to do with the class to which the sample should be attributed. Using the terminology of [6], we call a hyperplane separating two classes *relevant* to the test samples from these classes, and *irrelevant* to all other samples.

Let N be the total number of classes, and suppose for the moment that their labels do not overlap (unlike in our dataset). If all relevant classifiers yield correct results, the correct class receives $N-1$ votes and wins the majority. However, if some of the relevant classifiers fail (due to the perpetual incompleteness of the training data), the final outcome can be determined by the irrelevant classifiers. It is significant that the number of relevant classifiers grows only linearly with N while the number of irrelevant classifiers grows quadratically. Since the *proportion* of irrelevant classifiers therefore grows linearly with N, we should expect a decline in correct majority voting retrieval rates, as N increases, even if the individual binary classifiers perform just as well for a larger set of classes. Our experimental results, shown in Figure 2, confirm these pessimistic expectations.

We propose to use the following general runoff election strategy in order to minimize this effect. Suppose that testing shows the correct class is almost always in the top K classes, for some fixed K, in majority voting. Then modify the classification to first do majority voting using all binary classifiers, and then do a round of run-off voting using only those binary classifiers involving the initial top K classes. In the second round there is a much smaller proportion of irrelevant classifiers. In our case, given that the top 10 correct retrieval rates are high, we can assume that the correct class is among the top 10. In the second round the proportion of irrelevant classifiers is 36 out of 45 compared to $(N-1)(N-2)/2$ out of $N(N-1)/2$ (*e.g.* 109/111) so we should expect the top-1 correct retrieval rates to grow.

Experimental results confirm this. They show that for the best top 1 retrieval rates, 4 classes should be used in the second round. More than two rounds of voting did not yield



Classes with ≥ 10 samples | Classes with ≥ 40 samples | Classes with ≥ 70 samples

Figure 2: Top-K recognition rates as function of number of classes. $K = 1$ (*bottom*), 2, 3, 5, 10 (*top*)**.**

any improvement. The results of runoff elections, with a second round of majority voting for the top 4 classes, are summarized in Table 2. The retrieval rates have increased by about 6%, compared to the pure majority voting scheme. Similarly, the top-2 correct retrieval rates increased by about 1% with a second round of majority voting for the top 8 classes, and the top-3 rates got just slightly higher. The likely reason why the effect of runoff elections diminishes as K increases is that, for large K, the top-K rates are not so sensitive to the number of classes.

6. CONCLUSION

For a classification problem with a large number of highly linearly separable classes, ensembles of linear support vector machines are a natural, robust, and scalable approach. By replacing the conventional majority voting scheme with a runoff election scheme, we can reduce the influence of irrelevant votes on the final outcome and increase stability of the ensemble classifier.

This has been experimentally verified in the case of on-line classification of handwritten mathematical symbols, a problem that involves hundreds of classes and uneven distribution of samples across different classes. In this setting, we have shown that runoff elections reduce the classification error by about 6%, or roughly in half, compared to previous results. Runoff elections do not cause any noticeable computational overhead, especially if used in conjunction with the on-line binary SVM classification algorithm, which can eliminate most of the irrelevant hyperplanes while the character curve is traced.

7. REFERENCES

[1] Char, B., Watt, S.M.: Representing and Characterizing Handwritten Mathematical Symbols through Succinct Functional Approximation. Proc. Intl. Conf. on Docum. Anal. and Rec. (ICDAR) (2007) 1198–1202.

[2] Golubitsky, O., Watt, S.M.: Online Stroke Modeling for Handwriting Recognition. Proc. 18th Intl. Conf. on Comp. Sci. and Soft. Eng. (CASCON) (2008) 72–80.

[3] Golubitsky, O., Watt, S.M.: Online Computation of Similarity between Handwritten Characters. Proc. Docum. Rec and Retrieval (DRR XVI) (2009) C1–C10.

[4] Golubitsky, O., Watt, S.M.: Online Recognition of Multi-Stroke Symbols with Orthogonal Series. Proc. 10th International Conference on Document Analysis and Recognition (ICDAR 2009), Barcelona, Spain, IEEE Computer Society, 1265–1269.

[5] Joachims, T.: Making Large-Scale SVM Learning Practical. Advances in Kernel Methods - Support Vector Learning. B. Schölkopf and C. Burges and A. Smola (ed.), MIT-Press (1999).

[6] Savicky, P., Fürnkranz, J.: Combining Pairwise Classifiers with Stacking. Lect. Notes in Comp. Sci. **2810** (2003) 219–229.