Estimating the Readability of Handwritten Text – A Support Vector Regression Based Approach

Andreas Schlapbach and Frank Wettstein and Horst Bunke
Institute of Computer Science and Applied Mathematics
Neubrückstrasse 10, CH-3012 Bern, Switzerland
{*schlpbch, wettstei, bunke*}*@iam.unibe.ch*

Abstract

This paper presents a new approach to estimating the readability of handwritten text. The estimation task is posed as a regression problem. A novel Support Vector Regression (SVR) system is used to estimate the recognition rate of a text recognizer on a given text. The estimated recognition rates are used to classify text as either readable or unreadable. Unreadable text can then be filtered out prior to recognition, thus avoiding needless recognition attempts or a high cost caused by manual correction. The system is systematically evaluated on a data set of 1,830 text lines from 50 writers.

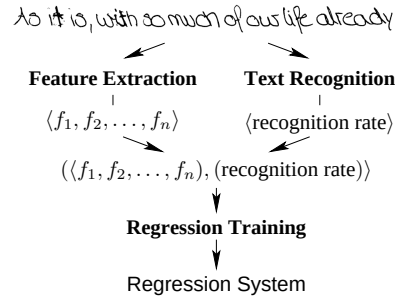

Figure 1. Training of the Regression-based Readability Estimation System

1. Introduction

Automatic reading of general handwritten text is still a challenging problem. Filtering out unreadable text prior to recognition would help us avoiding needless recognition attempts and reduce costs associated with false classification. In previous work, we have cast the readability estimation problem as a classification problem [13]. A text is classified as either *readable* or *unreadable*. In the training phase, each text is transformed into a feature vector and the text recognition rate is determined. Next, the feature vectors are labelled, i.e., assigned to one of the two classes depending on the recognition rate and used to train a classifier.

In this paper, instead of training a classifier with labelled data we directly use the recognition rate to train a regression system. A novel approach based on Support Vector Regression (SVR) has been chosen [11]. The goal of the regression system is to find a function that has a small deviation from the actually achieved recognition rates for all the training data [11]. In the recognition phase, this function is used to estimate the recognition rate obtained on a text prior to recognition. Based on the estimated value, it can be decided whether to actually submit the handwritten text to the recognition engine or to enter its transcription completely manually.

Only few works on readability estimation exist. The problem of predicting the accuracy of an Optical Character Recognition (OCR) system for machine printed text has been studied in [1, 6]. A performance prediction model for handwritten word recognizers is presented in [14]. Related to this work are writer identification [10], handwriting style classification [7], and subcategory classification analysis of handwriting [3]. To the best of the authors' knowledge, the problem of estimating the readability of handwritten text using SVR has never been addressed in the literature before.

2 System Overview

The basic units of handwritten text considered by the system are individual text lines. In the training phase, each text unit is transformed into a feature vector. Simultaneously, the text is submitted to a recognition system and the recognition rate achieved on the text is determined. Then, for each text line, the feature vector and

the recognition rate are merged to one vector and used to train an SVR system. A systematic overview of the training procedure is shown in Fig. 1. In the operational phase, the same features as the ones used for training are extracted from a text and passed on to the trained regression system to estimate the recognition rate.

2.1 Feature Extraction

As mentioned before, this paper considers individual text lines as the basic units. The features extracted from a text line image have initially been defined for writer identification and have shown very good results [5]. As these features are able to distinguish different writers, it is reasonable to apply them to the readability estimation task as well.

Before feature extraction, a text line image is normalised. The normalisation operations are designed to reduce the variability of the features so as to improve prediction accuracy [5]. The 100 features extracted from a text line can be divided into five groups [5]. Features of the first group, also called *basic features*, represent basic properties of a text line, such as the skew or the slant angle. The second group of *component features* describe the writing style of a writer with respect to its connectedness. The basic idea behind the third group of *fractal features* is to measure how the area of a handwritten text grows when a dilation operation is applied on the image [12]. From the resulting characteristic lower and upper contour the *features of the characteristic contour* are extracted. The last group are the *features of the enclosed regions*. As the extracted features can have very different numerical ranges, each feature is normalised with respect to its mean and standard deviation.

2.2 Text Recognition

The text recognition system described in this section uses *Hidden Markov Models (HMMs)* to model handwritten texts [9]. Only a short description of the system is given here; for a detailed presentation we refer to [8]. Before feature extraction, the skew angle, the slant angle, and baseline positions of each text line are normalised. These normalisation steps are necessary to reduce the impact of the different writing styles. After preprocessing, a handwritten text line is transformed into a sequence of feature vectors using a sliding window approach

For each upper and lower case character an individual HMM is built. Additionally, frequent punctuation marks, such as full stop, colon and space are modelled. Other, infrequent punctuation marks are mapped to a special garbage HMM. Each character HMM consists of 16 states. The states are connected in a linear topology. The character models are concatenated to word models which in turn are concatenated to model a complete text line.

The system is trained by applying the Baum-Welch algorithm which iteratively maximises the probability for a given sequence of observations [9]. Recognition is performed by the Viterbi algorithm using dynamic programming to recursively maximise the likelihood of the state sequence [9]. The recognition is supported by a statistical bi-gram language model [15].

2.3 Support Vector Regression

Support Vector Regression (SVR) is used to estimate the recognition rate from the extracted feature set of a text line. Given training data $\{(x_1, y_1), \ldots, (x_l, y_l)\} \in \mathcal{X} \times \mathbb{R}$, where $\mathcal{X}$ denotes the space of the input patterns (e.g. $\mathcal{X} = \mathbb{R}^d$) the goal is to find a function $f(x)$ that has at most ε deviation from the given targets y_i for all the training data, and at the same time is as flat as possible [11]. In other words, we do not care about errors as long as they are less than ε, but will not accept any larger deviation. We start with a linear function f, taking the form $f(x) = \langle w, x \rangle + b \quad \text{with} \quad w \in \mathcal{X}$, $b \in \mathbb{R}$ where $\langle \cdot, \cdot \rangle$ denotes the dot product in $\mathcal{X}$. *Flatness* means that one seeks a small w. One way to ensure this is to minimize the norm, i.e., $\|w\|^2 = \langle w, w \rangle$. This problem can be stated as a convex optimization problem. As we can not assume that such a function f actually exists that approximates all pairs (x_i, y_i) with ε precision, slack variables ξ_i, ξ_i^* are introduced to cope with otherwise infeasible constraints of the optimization problem:

$$\text{minimize} \qquad \frac{1}{2}\|w\|^2 + C \sum_{i=1}^{l} (\xi_i + \xi_i^*) \tag{1}$$

$$\text{subject to} \qquad \begin{cases} y_i - \langle w, x_i \rangle - b & \leq & \varepsilon + \xi_i \\ \langle w, x_i \rangle + b - y_i & \leq & \varepsilon + \xi_i^* \\ \xi_i, \xi_i^* & \geq & 0. \end{cases}$$

The constant $C > 0$ determines the trade-off between the flatness of f and the amount up to which deviations larger than ε are tolerated.

Next, it can be shown that w can be completely described as a linear combination of the training patterns x_i [2]. The key observation is that the SVR algorithm only depends on dot products between patterns x_i. This observation is used to extend the SVR algorithm to the non-linear domain by mapping the training patterns $x_i \in \mathcal{X}$ into some feature space $\mathcal{F}$ using a function

$\Phi : \mathcal{X} \rightarrow \mathcal{F}$. Because only dot products are needed it suffices to know $\kappa(x, x') = \langle \Phi(x), \Phi(x') \rangle$ rather than Φ explicitly. Hence the SVR optimization problem is restated such that w is no longer given explicitly and corresponds to finding the *flattest* function in the *feature* space, not in the input space.

Next we need to determine functions $\kappa(x, x')$ which correspond to a dot product in some feature space. A function $\kappa(x, x)$ is a valid kernel function if it fulfils the *Mercer* condition [11]. While different kernel functions exist, the *radial basis function (RBF)* kernel function was chosen in this work because it contains only one meta-parameter γ. The parameter γ of the kernel function as well as the weighting parameter C of Eq. 1 need to be optimized on a validation set. The SVR system is implemented using the LIBSVM library [4].

3 Experimental Setup

The text lines used in the experiments are part of the IAM handwriting database [8][1]. The database currently contains over 1,500 pages of handwritten text from over 650 writers. In total, 3,308 text lines from 347 writers are used in the experiments described in this paper. The text lines are divided into two disjoint sets. The first set is used to train and validate the text recognition system while the second set is used to train, validate, and test the regression-based readability estimation system.

The first set of 1,478 text lines from 297 writers is split into a training and a validation set. The text recognition system is trained using 1,333 text lines from 268 writers. The meta-parameters are optimized using the remaining 145 text lines from 29 writers. The text lines of each writer appear only in one data set, thus the experimental setup is writer independent.

The second set of 1,830 text line from 50 writers is split into three sets: a training, a validation and a test set. This data set is the same set as the one that was used for writer identification in [5]. A total of 780 text lines from 20 writers are used as training set, 335 text lines from another 10 writers are used as validation set, and the remaining 715 text lines from 20 writers form the test set. Again, this is a writer independent setup.

During training of the handwriting recognition system, the number of Gaussian mixture components of the HMM-based recognizer is increased from 6 to 18 components in steps of 3. Optimal performance on the validation set is achieved using 15 Gaussian mixture components resulting in a word accuracy rate of 69.78%. The distribution of the recognition rates for the text lines of the training set is shown in Fig. 2.

[1]The IAM handwriting database is publicly available at: www.iam.unibe.ch/fki/iamDB

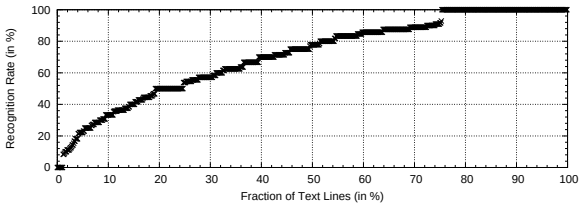

Figure 2. Distribution of the Recognition Rates on the Training Set

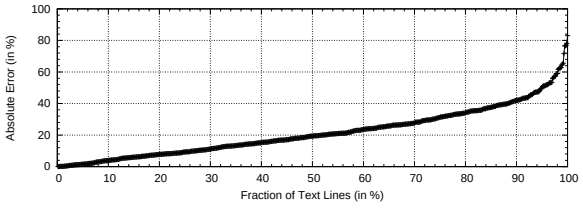

Figure 3. Absolute Error Rates on the Test Set

4 Regression Results

For a given text line t, the SVR system outputs an estimated recognition rate $r_{\text{est}}(t)$. To assess the performance of the SVR system the sum of the mean absolute error and the median absolute error are calculated. The meta-parameters of the regression system, $C \in \{2^{-5}, 2^{-3}, \ldots, 2^{15}\}$ and $\gamma \in \{2^{-15}, 2^{-13}, \ldots, 2^{3}\}$, are optimized on the validation set. The lowest mean absolute error of 20.29% is achieved by $C = 2^{11}$ and $\gamma = 2^{-13}$. This result means that in average, the recognition rate achieved by the text recognition system and the estimated recognition rate returned by the SVR system vary by approximately 20%. The median absolute error on the validation set is 17.83% and thus lower than the mean absolute error. This finding can be explained by the fact that only 32 out of 335 text lines with $|r(t_i) - r_{\text{est}}(t_i)| > 40\%$ exist.

In Fig. 3, the absolute error $e_{\text{abs}}(t) = |r(t) - r_{\text{est}}(t)|$ for each text line t of the test set is plotted in ascending order. The mean absolute error rate is 21.61% and the median absolute error rate is 19.32%. The absolute error rate is below 10% in over 27% and below 15% in over 39% of all cases.

To study the performance of the SVR system the difference of the actual recognition rate and the predicted recognition rate is plotted in Fig. 4. For each text line of the test set, the difference is calculated and then plotted at the point of the actual recognition rate. For ease

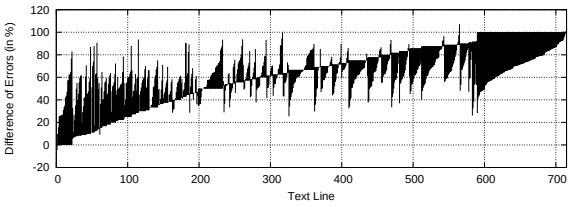

Figure 4. Difference Between Actual and Predicted Error Rates on the Test Set

of presentation, the results are sorted firstly in respect to the actual recognition rate and, secondly, in respect to the predicted recognition rate. As can been seen in Fig. 4, the SVR system has difficulties predicting the recognition results at the extremes, i.e., if the actual recognition rate is close to 0% or 100%. Especially if the actual recognition rate is 100%, the estimation system tends to predict lower recognition rates. Furthermore, Fig. 4 shows that, in a few rare cases, the SVR system can predict impossible recognition rates, i.e., recognition rates below 0% or above 100%,.

5 Conclusion and Future Work

In this paper, a new approach to estimating the readability of a handwritten text is presented. It allows one to filter out unreadable data prior to recognition and thus helps avoiding needless recognition attempts. The estimation problem is posed as a regression problem where a Support Vector Regression system is used to predict the recognition rate of a text recognizer. The regression-based readability estimation system is tested on a test set of 715 text lines from 20 writers. In over half of all cases, the absolute error between the actual and the estimated recognition rates is below 22%.

The approach proposed in this paper consists of two stages, the first one involving the regression system and the second one the recognizer. The second stage is only entered if there is evidence that the recognizer will yield a result of sufficiently high quality. Thus using a regression system to predict the accuracy of the recognizer seems particularly meaningful if the computational cost of the regression system is low compared to the recognizer such as is the case in our experiments.

The readability estimation task studied in this paper can be regarded as a special case of the more general problem of estimating whether or not some presented input data is recognisable by a given system. A natural extension of this work would thus be to study the problem of *recognisability classification* in the context of other recognition tasks.

Acknowledgements

This research is supported by the Swiss National Science Foundation NCCR program “Interactive Multimodal Information Management (IM2)” in the Individual Project “Visual/Video Processing”.

References

[1] L. R. Blando, J. Kanai, and T. A. Nartker. Prediction of OCR accuracy using simple image features. In *Proc. 3rd Int. Conf. on Document Analysis and Recognition*, volume 1, pages 319–322, 1995.

[2] B. E. Boser, I. M. Guyon, and V. N. Vapnik. A training algorithm for optimal margin classifiers. *Proc. of the Annual Conf. on Computational Learning Theory*, pages 144–152, 1992.

[3] S.-H. Cha and S. N. Srihari. Apriori algorithm for sub-category classification analysis of handwriting. In *Proc. 6th Int. Conf. on Document Analysis and Recognition*, pages 1022–1025, 2001.

[4] C.-C. Chang and C.-J. Lin. *LIBSVM: a library for support vector machines*, 2001.

[5] C. Hertel and H. Bunke. A set of novel features for writer identification. *Audio- and Video-Based Biometric Person Authentication*, pages 679–687, 2003.

[6] J. Kanai, T. A. Nartker, S. V. Rice, and G. Nagy. Performance metrics for document understanding systems. In *Proc. 2nd Int. Conf. on Document Analysis and Recognition*, pages 424–427, 1993.

[7] E. D. Mandana, N. Sherkat, and T. Allen. Handwriting style classification. *Int. Journal on Document Analysis and Recognition*, 6(1):55–74, 2003.

[8] U.-V. Marti and H. Bunke. The IAM–database: An English sentence database for off-line handwriting recognition. *Int. Journal of Document Analysis and Recognition*, 5:39–46, 2002.

[9] L. R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proc. of the IEEE*, 77(2):257–285, 1989.

[10] A. Schlapbach and H. Bunke. Off-line writer identification and verification using Gaussian mixture models. *Machine Learning in Document Analysis and Recognition*, 11:409–428, 2008.

[11] A. J. Smola and B. Schölkopf. A tutorial on support vector regression. *Statistics and Computing*, 14(3):199–222, 2004.

[12] P. Soille. *Morphological Image Analysis*. Springer, 1999.

[13] F. Wettstein. Automatic readability of handwritten text lines. Master’s thesis, University of Bern, 2007.

[14] H. Xue and V. Goindaraju. Performance prediction for handwritten word recognizers and its application to classifier combination. In *Proc. 16th Int. Conf. on Pattern Recognition*, pages 241–244, 2002.

[15] M. Zimmermann and H. Bunke. N-gram language models for offine handwritten text recognition. In *Proc. 9th Int. Workshop on Frontiers in Handwriting Recognition*, pages 203–208, 2004.