CS 229 Final Project
OCR using an unsupervised hidden layer

Kevin Shan

2008/12/12

1 Background

Optical Character Recognition (OCR) is a supervised learning problem in which we wish to categorize letters based on similarity to a training set. Ideally, this measure of similarity should be invariant to "allowed" distortions with constraints such as continuity or monotonicity.

One particularly interesting approach that is the subject of ongoing research is elastic matching, which does exactly that. In a typical application, the sample image is warped to match a reference image, and either a cost function of the warping function or a conventional distance metric applied to residual error is used to measure the similarity between sample and reference images.

However, it has been shown[1] that elastic matching is an algorithmically hard problem with exponential time solutions.

2 Motivation

One alternative to true elastic matching is shape normalization and Gaussian blurring.

In our case, shape normalization involved finding the smallest bounding box and reshaping it to square proportions. This yields normalized images that are invariant to translation and scaling on the coordinate axes (and also in funny artifacts on sans-serif 'I's, which are stretched into squares).

Gaussian blurring is a simple and nonspecific way to increase generalizability. To make our training feasible, we also reduced the size of the normalized image to 16×16 pixels, yielding a 256-length feature vector.

Our training sets comprise 95 computer fonts in 11 different rotations (26 characters per set) and 20 sets of handwriting samples from 7 different people using various pens (61

Classifier	Base fonts	Unrotated	All fonts	Handwriting
Linear SVM trained on base fonts	0.0	13.9	42.1	53.1
LMS trained on base fonts	4.2	13.6	31.5	47.0
LMS trained on unrotated fonts	5.2	7.6	22.7	48.3
LMS trained on all fonts	11.5	11.2	13.3	52.6
LMS trained on handwriting	54.0	58.7	64.5	40.9

Table 1: Error rates (%) of various classifiers on various test sets. Note that our feature space is a 256-element vector and the test sets are 286, 2,470, 27,170, and 1,220 characters long, respectively.

characters per set). Subsets of the computer fonts include "base fonts," which is a selection of 11 very popular fonts; and "unrotated fonts," which is the 95 fonts without rotation.

We tested on many combinations of training and test data to get an idea of the generalization error of this approach. Unfortunately our ability to train SVMs was limited the availability of CPU cycles, so we were only able to train one very limited linear SVM, though ideally we would have liked to train an SVM with a Gaussian kernel.

Table 1 lists the results from our testing. While the shape normalization artifacts with the sans-serif 'I's surely account for some of the error, that effect is limited to $1/26 = 3.8\%$. Nor is it an issue of training set size, as we can see from increasing the training set.

So it seems that linear classifiers operating on shape normalized data just aren't very good at the OCR problem, motivating us to look more closely at elastic matching.

3 Approach - density estimation

Although there are numerous approximations to the elastic matching problem that utilize dynamic programming to make the problem feasible[2,3], we will approach this in a different way.

We will treat the image as a probability density function where the relative intensity of each pixel (x, y) is the relative probability of that (x, y), then fit a mixture of Gaussians to this density distribution. The parameters θ of that Gaussian fit constitute our "Density Estimate."

Figure 1 shows a summary of this process flow. In this example, we have taken an unprocessed 128 x 128 greyscale image (note that we haven't even cropped out the excess whitespace, a task that is not trivial in practical handwritten character recognition), approximated it using a mixture of 16 Gaussians, and then reconstructed from the density estimate to verify that the method works.

One minor change to the mixture-of-Gaussians distribution that we tried was forcing all Gaussians to have the same radius of minor axis. This could be interpreted as a global

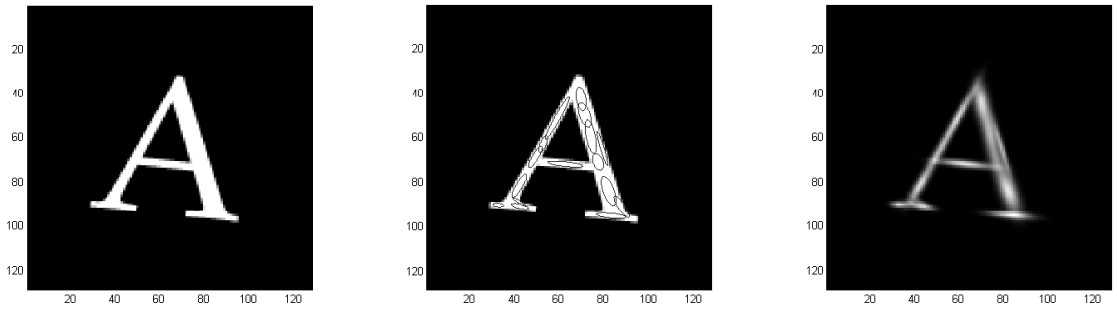

Figure 1: Density estimation process. We start with the original image (1(a)), fit a mixture of Gaussians to the pixel intensity distribution using an expectation-maximization algorithm (1(b)), and finally we can verify the result by inspecting the reconstruction of the original image using the density estimation parameters (1(c))

line width parameter. Unfortunately, this algorithm did not converge well for serif fonts, which typically use a variety of line widths per character, so we abandoned it in favor of the less-constrained mixture-of-Gaussians model.

To evaluate the reliability of these reconstructions we ran the density estimation algorithm on our 95 fonts and 20 handwriting samples and compared the reconstructed images to the original samples. Figure 2 shows the result of this analysis on an alphabet-by-alphabet basis (rather than a character-by-character basis).

Visual inspection of the failed cases shows that the issue is not typically failure to converge but rather fluctuations in density due to the unevenness of Gaussians compared to the flat zero/one pixel intensities found in the font samples.

4 Discussion

We can tell, at least qualitatively, that the density estimation parameters capture much of the relevant information in the original image.

However, we have reduced the dimensionality of the image significantly, from a 128×128 greyscale image to $k = 16$ sets of density estimation parameters in $\mathbb{R}^6$, each describing a Gaussian distribution.

Second, we vectorized a raster image. Now scaling/rotation/translation transformations are all affine operations (these operations would otherwise require resampling with interpolation).

These two effects make subsequent operations much faster, allowing for a wider range of iterative classification algorithms, including elastic matching.

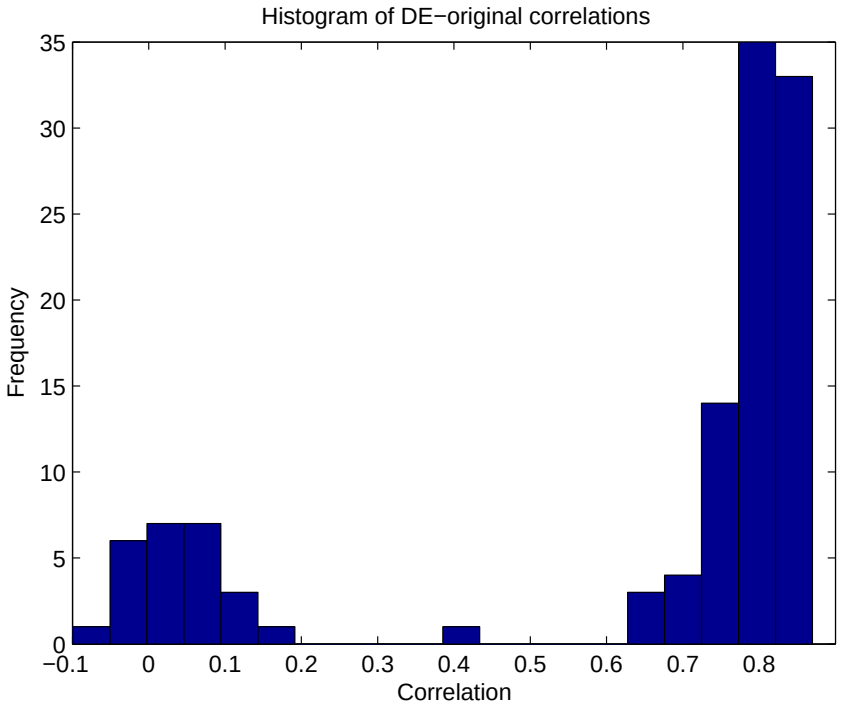

Figure 2: Histogram of correlations between original alphabet and alphabet reconstructed from density estimation

For the OCR problem, it would be ideal to use elastic matching as a determinant function and an SVM for the top layer classifier. For example, we could use the warping cost function or post-warp distance as a kernel function for an SVM kernel.

Although we incur a overhead due to the expectation maximization algorithm, it transforms our data into a form that is much easier to work with. For example, in a regularized SVM operating in a high dimensional feature space, we expect to have many nonzero α_i. For each test sample, the kernel function needs to be calculated for every nonzero α_i, so operating in the density estimation space can still offer a significant performance improvement despite the overhead of the mixture of gaussians fit.

5 Directions for further research

The next step would be to actually kernelize an elastic matching distance metric to operate in density estimation space using something a variant of something like Uchida and Sakoe's quadratic discrimination technique[4].

Finally, in instances where we are interested in warping of more complex images (rather than line art like type and handwriting), this technique of density estimation for elastic matching can be used in conjunction with conventional elastic matching at the pixel level.

For example, it can be used to guide a fast routine such as piecewise linear elastic matching, which is a relatively fast algorithm but requires careful selection of pivot points[5].

Notes

[1]D. Keysers and W. Unger. Elastic Image Matching is NP-Complete. *Pattern Recognit. Lett.* 24(1-3), 445-453 (2003)

[2]S. Uchida and H. Sakoe. A Survey of Elastic Matching Techniques for Handwritten Character Recognition. *IEICE Trans. Inf. & Syst.* E88-D, 1781 (August 2005)

[3]D. Keysers, T. Deselaers, C. Gollan, and H. Ney. Deformation models for image recognition. *IEEE Trans. on Pattern Analysis and Machine Intelligence* 29(8), 1422-1434 (August 2007)

[4]H. Mitoma, S. Uchida, and H. Sakoe. Online Character Recognition Based on Elastic Matching and Quadratic Discrimination. *Proc. ICDAR '05* (2005)

[5]S. Uchida and H. Sakoe. Piecewise linear two-dimensional warping. *Systems and Computers in Japan* 32(12), 1-9 (2001)