Heeding more than the top template

Prateek Sarkar George Nagy
sarkap@rpi.edu *nagy@ecse.rpi.edu*
Department of Electrical, Computer, and Systems Engineering
Rensselaer Polytechnic Institute, Troy NY 12180

Abstract

We present a method of classifying a pattern using information furnished by a ranked list of templates, rather than just the best matching template. We propose a parsimonious model to compute the class-conditional likelihood of a list of templates ranked on the basis of their match scores. We discuss the estimation of parameters used in the model. The results of maximum likelihood classification on isolated digit patterns consistently show a 10-20% relative gain in recognition accuracy when we use more than one top-template.

Keywords: Character recognition, template, ranked list

1. Introduction

The simplest definition of template matching may be the following: "Masks match unknown shapes to standard shapes, or templates." [4].

We identify hand-printed digits according to a list of templates ranked by match-scores. Each template is constructed using only samples from a single class, but there may be more than one template per class. According to the literature, most commonly the class of the top-ranking template is chosen as the class of the underlying pattern. The relative values of the scores of the top and subsequent templates are sometimes used in deciding whether to reject the pattern. We endeavor to use the rank-order of several templates to improve on top-template classification.

Template or mask matching is one of the oldest techniques of character recognition and has a rich literature. In 1929 Tauschek applied for a patent (granted in 1935) on a recognition system for the ten printed digits based on optical stencils [6]. The work up to 1961 is admirably analyzed in Mary Stevens' great survey [5]. Young and Calvert, in their 1974 text, stated that "There is no real evidence that multi-font character recognition machines with high quality input can profit from features more complicated than character masks, except possibly by building in tests for characters that are frequently confused." [8]. Among the pattern recognition texts that we have seen, the most thorough treatment of template matching for OCR is Ullman's [7]. Design and construction of templates, similarity measures, matching algorithms, and hardware and software implementation of template based classifiers have been subjects of extensive research for decades.

In view of the attention received by template matching over the years, it is surprising that we have not been able to find any work on classification based on the likelihood of an entire list of templates rather than only on that of the top template. While ranked lists have been used in combination of several classifiers in a post-processing stage as described in [1, 2, 3], we focus on information furnished by lower ranked templates in a single classifier. Our task begins after the templates have been constructed and applied to a set of sample patterns. Therefore the only information available to us for classification is a list of template scores for each pattern in a training set and a test set. We compare our results to that obtained by top-template classification.

Conceptually, the principal difficulty is that the number of possible rankings of templates grows exponentially with the length of the ranked list. The problem is not so much that of computing power, but of the sample size required to estimate the probability of occurrence of each list. Therefore either the underlying probabilistic model must be drastically simplified, or the number of templates considered must be curtailed.

In the next section we develop our probabilistic model for the generation of template lists. In Section 3, we describe the estimation of the model parameters from a training set of ranked lists. In Section 4 we summarize our experimental results and conclude with a discussion in Section 5.

2. Likelihood from ranked templates

Let there be N classes $w_n, \quad 1 \leq n \leq N$ and M templates $T_m, \quad 1 \leq m \leq M$. There is at least one template

for each class, so $M \geq N$. For each input pattern, a template matching procedure reports a *ranked list* of Q ($\leq M$) best matching templates $[t_1, \ldots, t_Q]$, where each t_i takes up values in the set $\{T_1, \ldots, T_M\}$. The list is sorted by goodness of match and t_1 represents the best fitting template *(top template)*.

The parameters of our model are $p_{w_n,T_m} \stackrel{\triangle}{=} P\{T_m|w_n\}$ *i.e.*, the probability that the template matching procedure picks T_m as the top template, given a pattern of class w_n. We can think of p_{w_n,T_m} as the *win probability* for the template T_m given the class w_n.

We now model the likelihood of observing a ranked template list $[t_1, \ldots, t_Q]$ given a class w_n in the following way. The probability that the top template is t_1 is given by p_{w_n,t_1}. Once t_1 is chosen as the top template, the second ranking template can be picked from the remainder of the pool. We assume that the history of the first pick does not alter the relative win probabilities of the remaining templates. Then the probability that the second template is t_2 is $\frac{p_{w_n,t_2}}{1-p_{w_n,t_1}}$. The template for the third rank is chosen from the template pool without t_1 and t_2, and so on. Arguing thus, we write:

$$P\{[t_1, \ldots, t_Q]|w_n\} = p_{w_n,t_1} \times \frac{p_{w_n,t_2}}{1-p_{w_n,t_1}} \times \frac{p_{w_n,t_3}}{1-p_{w_n,t_1}-p_{w_n,t_2}} \ldots \times \frac{p_{w_n,t_Q}}{1-p_{w_n,t_1}-p_{w_n,t_2}-\ldots-p_{w_n,t_{Q-1}}} \quad (1)$$

For any w_n, the likelihood is maximized by a list that corresponds to the top Q templates ranked by win probability.

Our model is a generalization of *sampling without replacement*. When there are M templates and only the Q top ranking templates are listed for each input pattern, there are $M!/(M-Q)!$ possible ranked lists per pattern. Given that $\sum_m p_{w_n,t_m} = 1$ we can show that when conditioned on any class w_n, the probabilities of these $M!/(M-Q)!$ ranked lists sum to 1. For a quick example to demonstrate this, we set the win probabilities of all templates to be equal to $1/M$. Then the likelihood for any given ranked list reduces to $(M-Q)!/M!$, and consequently the sum of the likelihoods of the $M!/(M-Q)!$ possible rank ordered lists is 1.

To classify a given ranked list of templates, the probability of the list is computed for each of the N classes according to (1), and the class corresponding to the highest probability is assigned as the label.

3. Estimation of model parameters

Let us consider the case of supervised training for the model. Then for each class w_n, we have data of the following form:

Frequency	Ranked list of Q top templates
f_1	List1=$[T_1, T_2, T_3, \ldots, T_Q]$
f_2	List2=$[T_2, T_1, T_3, \ldots, T_Q]$
$\ldots$	$\ldots$

Our goal is to estimate, from such data, the parameters p_{w_n,T_m} for the given class w_n.

3.1. Maximum likelihood (ML) estimation

From top templates only. If a template T_m ranks on top for N_{T_m,w_n} out of N_{w_n} patterns from class w_n, the ML estimate for the win probability of T_m, conditioned on w_n, is:

$$\hat{p}_{w_n,T_m} = \frac{N_{T_m,w_n}}{N_{w_n}} \quad (2)$$

This method does not make full use of the data, which in practical applications is always scarce.This motivates the estimation of parameters using entire ranked lists rather than just the top templates.

From ranked lists of templates. Assuming independence of observations, the likelihood of a set of observations is given by the product of the individual likelihoods which, in turn, can be calculated using (1). Thus the ML estimates of the class-conditional win probabilities is given by the set of values that maximizes this product, or equivalently maximizes the sum of the log-likelihoods.

3.2. Maximum a posteriori estimation

For a given class, if a template is not seen in the training sample, the maximum likelihood estimate of the corresponding conditional win probability is zero – an undesirable effect of estimation from a finite sample set. To bypass this problem, we lean on maximum *a posteriori* (MAP) estimation of the parameters. For any class, w_n we set the *a priori* density in the space of parameters p_{w_n,t_m} to

$$p_0(p_{w_n,T_1}, \ldots, p_{w_n,T_M}) = \frac{1}{\kappa} \prod_{m=1}^{M} p_{w_n,t_m} \quad (3)$$

where $0 \leq p_{w_n,t_m} \leq 1$, $\sum_{m=1}^{M} p_{w_n,t_m} = 1$, and κ is chosen such that the above integrates to unity over the region satisfying the constraints. This prior distribution, being symmetric with respect to the parameters, peaks when all the parameters are equal, *i.e.*, $p_{w_n,t_m} = 1/M$, and is zero if p_{w_n,t_m} is zero or one for any m.

The objective function for MAP estimation is obtained by multiplying the observation likelihood (as discussed in Section 3.1) by the prior.

3.3. Example

To show how the likelihood of observations is computed, we present a small example with $M = 3$ templates, of

which the top two are reported by the matching procedure ($Q = 2$).

Ranked List	True Label	Likelihood function	Frequency
$[T_1, T_2]$	w_1	$\frac{p_{w_1,T_1} p_{w_1,T_2}}{(1-p_{w_1,T_1})}$	6
$[T_1, T_3]$	w_1	$\frac{p_{w_1,T_1} p_{w_1,T_3}}{(1-p_{w_1,T_1})}$	1
$[T_2, T_1]$	w_1	$\frac{p_{w_1,T_2} p_{w_1,T_1}}{(1-p_{w_1,T_2})}$	3

The likelihood function for the entire training-set of observations is given by the product

$$\begin{aligned} L &= \left[p_{w_1,T_1} \frac{p_{w_1,T_2}}{1-p_{w_1,T_1}}\right]^6 \left[p_{w_1,T_1} \frac{p_{w_1,T_3}}{1-p_{w_1,T_1}}\right]^1 \\ &\quad \times \left[p_{w_1,T_2} \frac{p_{w_1,T_1}}{1-p_{w_1,T_2}}\right]^3 \\ &= \frac{p_{w_1,T_1}^{10} p_{w_1,T_2}^{9} p_{w_1,T_3}}{(1-p_{w_1,T_1})^7 (1-p_{w_1,T_2})^3} \\ &= \frac{p_{w_1,T_1}^{10} p_{w_1,T_2}^{9} (1-p_{w_1,T_1}-p_{w_1,T_2})}{(1-p_{w_1,T_1})^7 (1-p_{w_1,T_2})^3} \end{aligned}$$

The last step was obtained by enforcing the constraint $p_{w_1,T_1} + p_{w_1,T_2} + p_{w_1,T_3} = 1$. Maximizing the last expression with respect to p_{w_1,T_1} and p_{w_1,T_2} leads to the ML estimates from top 2 templates shown in Table 1.

The objective function for MAP estimation is the product of the likelihood function and the prior density:

$$\begin{aligned} &\frac{1}{\kappa} p_{w_1,T_1} p_{w_1,T_2} p_{w_1,T_3} \times L \\ &\quad = \frac{1}{\kappa} \frac{p_{w_1,T_1}^{11} p_{w_1,T_2}^{10} (1-p_{w_1,T_1}-p_{w_1,T_2})^2}{(1-p_{w_1,T_1})^7 (1-p_{w_1,T_2})^3} \end{aligned}$$

where κ, being a positive constant, can be ignored. Maximizing the above with respect to p_{w_1,T_1} and p_{w_1,T_2} yields the MAP estimates from top 2 templates shown in Table 1. For our experiments, the maximization was done numerically with MATLAB.

Estimating the parameters from just the top templates is facilitated by the simpler likelihood function, $p_{w_1,T_1}^7 p_{w_1,T_2}^3$, while the prior remains the same.

4. Experimental Results

We performed experiments on hand-printed isolated digits. For each digit sample, we were given only the true label of the sample, and the best and second best templates fitting the sample according to some template matching algorithm ($Q = 2$).

Table 1. Estimated parameters for the example.

	Top template only		Top two templates	
	ML	MAP	ML	MAP
$\hat{p}_{w_1,T_1}$	0.70	0.62	0.71	0.63
$\hat{p}_{w_1,T_2}$	0.30	0.31	0.26	0.32
$\hat{p}_{w_1,T_3}$	0	0.07	0.03	0.05

Table 2. Result summary of classification experiments - error rates.

	Test set	Training set and list length			
		Set A $Q=1$	Set A $Q=2$	Set B $Q=1$	Set B $Q=2$
	Top-template identity classifier				
1	A	19.07%			
2	B	18.92%			
	Top-template ML classifier				
3	A	19.07%	19.07%	19.07%	19.07%
4	B	18.92%	18.92%	18.92%	18.92%
	Top-two-templates ML classifier				
5	A	16.44%	15.48%	16.46%	15.01%
6	B	17.23%	16.46%	17.07%	15.62%
	Top-two-templates exhaustive classifier				
7	A		14.21%		14.94% (0.43%)
8	B		14.26% (0.48%)		13.28%

Our data, corresponding to 15720 samples spread across ten digit classes ($N = 10$), was divided into two sets, Set A and Set B, of 7860 samples each. We used two templates for each class, or a total of $M = 20$ templates.

200 parameters $p_{w_1,t_1} \ldots p_{w_{10},t_{20}}$ were MAP estimated from the training set, by two different methods, using top templates only ($Q = 1$) or top two templates ($Q = 2$). Initially, Set A was used for training. The same set of parameters was used to test on both sets, so that we can compare the error rates for consistency. The same was repeated by using Set B for training.

Table 2 shows the percentages of error for classification experiments. In our data, the true class identity of each template was known. This helped us implement classification by top-template identity, the performance of which (rows 1-2) served as a base-line for comparison.

Each set of parameter estimates was used for ML classification of sets A and B, once using the top templates only (rows 3-4), and then using the (short) ranked lists of

top two templates (rows 5-6) for each pattern. As we had surmised, using more than just the top template helps improve recognition accuracy. Similarly, using all available training-data for parameter estimation (top two templates in this case), rather than just the top template, helps in classification. Thus the error rates in the $Q = 2$ columns are consistently lower than or equal to those in $Q = 1$ columns.

Results of ML classification with top templates are the same, regardless of the training set or method (rows 3-4). Further, for a given test set, the results also match up with those of the top-template-identity classifier. This is not mandated by our training or testing process in any way. Rather, as a consequence of the template design process, each template has a higher win probability given its true class than that given any other class.

To set another benchmark for how far we can improve on the isolated digits data, we performed the following experiment. To each of the $20 \times 19 = 380$ possible permutations of top two templates, we assign the class that most frequently induces this permutation in the training data. Disregarding minor irregularities owing to finite sample size, this is the best that we can do with top-template-pair observations. Rows 7 and 8 of Table 2 show the results of this scheme. In this example the scheme requires nearly twice the number of parameter estimates compared to our ranked-list model. In general the number of parameters grows exponentially with list length. Some permutations in the test data were never seen in the training data, which highlights the need for a model with fewer parameters. We classify such cases by the top-template identity. The relative frequency of "unseen observations" is shown within parentheses as percentages, and the error rates reported include these cases.

5. Discussion

We have presented a model for the rank ordering of templates as a generalization of *sampling without replacement*, and applied it to classification of hand-written digit patterns given a ranked list of matching templates. We have shown that heeding more than the top template can improve recognition accuracy. The main advantage of the proposed method is that it can be extended to using more than the top two templates without increasing the number of parameters to be estimated. We shall conclude with a few thoughts for future work.

Though we focus on isolated digit recognition in this paper, heeding more than the top template may be particularly helpful in reducing errors when contextual knowledge is available, and neighboring characters affect and guide classification. Such context may be either linguistic context (e.g., character n-gram statistics) or style context (where nearby characters are expected to be from the same font, or written by the same writer). Context information may invalidate classification according to top template, and thus require information from lower ranking templates.

In the application from which we drew our data, nearly eight hundred templates are necessary to recognize hand-printed digits with acceptable accuracy. We would like to demonstrate that using the top two templates is better even when there are many more templates per class and therefore the top template is more often the one with the correct identity. The number of parameters to be estimated for 800 templates and 10 classes is 8000, which would require a considerably larger sample size than we had available. We may also gain by (*i*) pruning the parameter list during training by considering only a few useful templates for each class and (*ii*) by utilizing the information in the match scores.

With our model, we may be able to correctly classify new classes or character shape variants without adding new templates because we do not need the class identities of templates. Further, the model can also be applied to ranked lists not derived from template matching, e.g., to post-processing of ranked-list outputs of other classifiers.

Acknowledgment. We gratefully acknowledge Hitachi Central Laboratories for sponsoring this project. In particular, we thank Dr. H. Fujisawa and his group for providing the template-based classifier outputs on which the experiments were run.

References

[1] A. Dengel, R. Hoch, F. Hönes, T. Jäger, M. Malburg, and A. Weigel. Techniques for improving OCR results. In H. Bunke and P. S. P. Wang, editors, *Character Recognition and Document Image Analysis*, pages 227–258. World Scientific Publishing Company, 1997.

[2] T. K. Ho, J. J. Hull, and S. N. Srihari. Decision combination in multiple classifiers. *IEEE Transactions on Pattrn Analysis and Machine Intelligence*, 16(1):66–75, January 1994.

[3] L. Lam, Y.-S. Huang, and C. Y. Suen. Combination of multiple classifier decisions for optical character recognition. In H. Bunke and P. S. P. Wang, editors, *Character Recognition and Document Image Analysis*, pages 79–101. World Scientific Publishing Company, 1997.

[4] M. Nadler and E. P. Smith. *Pattern Recognition Engineering*, page 158. John Wiley and Sons, 1993.

[5] M. E. Stevens. Automatic character recognition, a state of the art report. Technical Note 112, National Bureau of Standards, Washinton, D.C., May 1961.

[6] G. Tauschek. Reading machine. U. S. Patent 2 026 329, Dec 1935. Appl. May 1929.

[7] J. R. Ullman. *Pattern Recognition Techniques*. Crane, Russak & Company, New York, 1973.

[8] T. Y. Young and T. W. Calvert. *Classification, Estimation, and Pattern Recognition*, page 334. Elsevier, 1974.