Comparing and combining ORCCA symbol recognizers

Scott MacLean, George Labahn

ORCCA JLM, December 2010

Abstract

Both of our labs have worked on symbol recognition problems.

It is an interesting exercise to compare the techniques we have developed.

Research question: Can our classifiers be combined into a grand unified clORCCAfier?

Contents:

Recognition

We consider here the simplified case of stroke matching.

A stroke s is a sequence of points.

$$s = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$$

We possess a model set L of labeled strokes. The labels are symbol names: "a", "γ", "$\int$", etc.

$$L = \{(s_1, \ell_1), (s_2, \ell_2), \ldots, (s_N, \ell_N)\}$$

Given an input stroke s, the recognition problem is to determine the matching label $\ell = \ell(s)$.

This can be framed as a minimization problem over L:

$$\ell(s) = \ell_i \text{ such that } i = \text{argmin}_j\{d(s, s_j)\},$$

where d is some distance function between strokes.

Elastic matching [1]

Given two strokes $s = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ and $t = \{(u_1, v_1), \ldots, (u_m, v_m)\}$.

Find a function $f : \{1, \ldots, n\} \to \{1, \ldots, m\}$ that minimizes the distance function

$$d_f(s, t) = \sum_{i=1}^{n} g(i, f(i)),$$

where $g(i, j)$ is a pointwise distance function between (x_i, y_i) and (u_j, v_j) (e.g. Euclidean distance).

f must satisfy three constraints:

1. $f(1) = 1$ (first points are matched together)
2. $f(n) = m$ (last points are matched together)
3. $f(i+1) \in \{f(i), f(i)+1, f(i)+2\}$ (no two consecutive points may be unmatched)

Greedy elastic matching [2]

A dynamic programming solution for elastic matching requires quadratic time.

Given the same strokes s and t, we greedily construct f as follows:

1. Set $f(1) = 1, f(n) = m$.
2. Choose $f(2)$ from $\{f(1), f(1)+1, f(1)+2\}$ such that $g(2, f(2))$ is minimal.
3. Similarly, choose $f(n-1)$ from $\{f(n-1), f(n-1)-1, f(n-1)-2\}$ minimizing $g(n-1, f(n-1))$.
4. Proceed to choose $f(3), f(n-2)$, and so on, selecting the best match locally for each point.

This process gives a fast approximation to the elastic matching distance.

Lagrange-Sobolev approximation [3]

Given s and t, represent the strokes by degree d polynomial approximations, parametrized by arclength:

$$s \sim \sum_{i=0}^{d} a_i P_i(\lambda), \; t \sim \sum_{i=0}^{d} b_i P_i(\lambda),$$

where the $P_i(\lambda)$ form a Lagrange-Sobolev basis.

That is, the $P_i(\lambda)$ satisfy

$$\int_0^1 P_i(\lambda) P_j(\lambda)\, d\lambda + \mu \int_0^1 P_i'(\lambda) P_j'(\lambda)\, d\lambda = \delta(i, j).$$

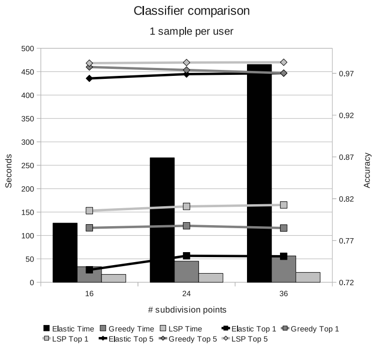
To obtain a distance between s and t, take the Euclidean distance between their (normalized) respective coefficient vectors.

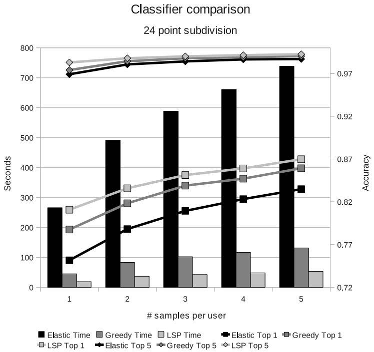
Classifier comparison

Experimental setup:

1. Using symbol data from [4]: $\sim$70 one-stroke symbol classes, $\sim$17000 samples in all.
2. Model set composed of $k = 1, 2, 3, 4, 5$ random samples from each of 20 writers.
3. Testing set was the rest of the data.
4. Strokes subdivided into $n = 16, 24, 32$ points.
5. Results are averaged over two runs for each parameter combination.
6. Top 1 means the label of the stroke minimizing the distance was correct.
7. Top 5 means the correct label appeared amongst the 5 most highly-ranked.

Comparison results





Comments

The Lagrange-Sobolev approach (LSP) was consistently slightly more accurate than the greedy elastic matching algorithm, and over twice as fast.

Both fast methods gave better results than the elastic matching algorithm on this data set.

Both fast methods are also limited:

- The greedy approach does not scale well to long strokes.
- The LSP approach does not scale well to complicated stroke shapes.

Although LSP outperformed the greedy algorithm, it consistently recognized about 25 symbol classes less accurately.

- Some (e.g. "ξ") make sense based on the above limitation.
- Others (e.g. "(", "U") do not.

Can both methods be combined to boost recognition rates?

Combining distance-based classifiers

Consider combining the greedy and Lagrange-Sobolev classifiers.

Given s, t, the classifiers yield two distances $d_1(s,t), d_2(s,t)$.
These distances are *incomparable*.

Instead of working with distances directly, consider the probability

$$p(\alpha_1, \alpha_2) = \Pr\left(\ell(s) = \ell(t) \,\middle|\, d_1(s,t) = \alpha_1, d_2(s,t) = \alpha_2\right).$$

The conditioning event has probability zero, so define

$$p(\alpha_1, \alpha_2) = \lim_{\delta \longrightarrow 0} \Pr(\ell(s) = \ell(t) \,\big|\, d_1(s,t) \in [\alpha_1, \alpha_1 + \delta], d_2(s,t) \in [\alpha_2, \alpha_2 + \delta]).$$

Combining distance-based classifiers (2)

Let

$$f(x, y) = \Pr(d_1(s, t) \in [x, x + \delta], d_2(s, t) \in [y, y + \delta] \\ \mid \ell(s) = \ell(t))$$

and

$$g(x, y) = \Pr(d_1(s, t) \in [x, y + \delta], d_2(s, t) \in [x, y + \delta]).$$

Then by Bayes' theorem and some algebra,

$$p(\alpha_1, \alpha_2) \propto \frac{D_{(1,1)} f(\alpha_1, \alpha_2)}{D_{(1,1)} g(\alpha_1, \alpha_2)} \propto \frac{\frac{\partial f}{\partial x}(\alpha_1, \alpha_2) + \frac{\partial f}{\partial y}(\alpha_1, \alpha_2)}{\frac{\partial g}{\partial x}(\alpha_1, \alpha_2) + \frac{\partial g}{\partial y}(\alpha_1, \alpha_2)}$$

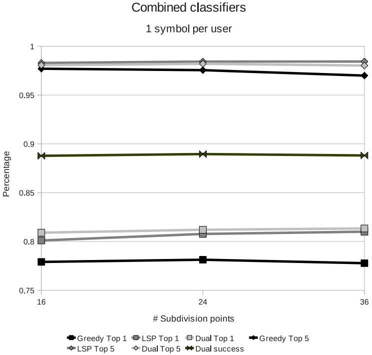
The joint distribution p is proportional to a ratio of sums of marginals (the partial derivatives).

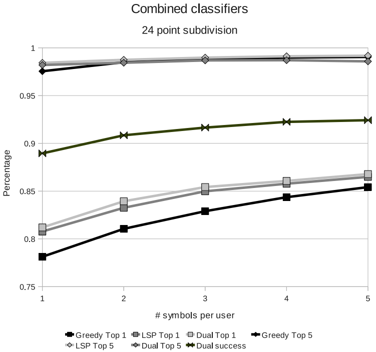
Classifier combination

Experimental setup:

1. Model set composed of $k = 1, 2, 3, 4, 5$ random samples from each of 20 writers.
2. Strokes subdivided into $n = 16, 24, 32$ points.
3. Precompute empirical distributions f and g on the model set, and discretize.
4. Run each classifier and the combined classifier for each parameter combination. Estimate derivatives by finite differences.
5. "Success" means the correct label was ranked at least as highly by the combined classifier as by both individual classifiers.

Combination results





Comments

Despite a $\sim 90\%$ success rate, the accuracy results are not yet convincing: $\sim 1\%$ increase in Top 1 accuracy.

The discretization strategy affects the results significantly. More work is needed here.

Why is the slight increase in top 1 accuracy accompanied by a decrease in top 5 accuracy?

Combination does not help (much) to disambiguate similar symbol classes, e.g. 1/(/)/l/I/etc.

Results should improve dramatically if we condition on the label of the model set symbols as well.

- Initial experiments show error rate reductions of 50-90%.
- We lack sufficient data to expand this to most symbol classes.

References

1. C. C. Tappert, *Cursive Script Recognition by Elastic Matching*, IBM J. of R&D 26 (6), 1982, pp.765-771.
2. S. MacLean and G. Labahn, *Elastic matching in linear time and constance space*, Proc. DAS (short paper), 2010, pp.551-554.
3. O. Golubitsky, S. Watt, *Distance-Based Classification of Handwritten Symbols*, IJDAR 13 (2), 2010, pp.133-146.
4. S. MacLean et al, *Grammar-based techniques for creating ground-truthed sketch corpora*, IJDAR, 2010, (forthcoming).