
CS540 Machine Learning

Clustering of Typeset Mathematical Symbols Using Spectral Methods and Shape Contexts

Greg Kempe

1 Introduction

Optical character recognition (OCR) of natural languages, both typeset and handwritten, is successfully used today in a wide range of applications. OCR of mathematical expressions and mathematical symbols is not yet as advanced, however. This project demonstrates a method for recognising typeset mathematical symbols. The method involves using spectral methods to perform semi-supervised clustering of the symbol images, using shape contexts as defined in [2] to measure the similarity of symbols.

The rest of the paper is structured as follows. A background to the symbol recognition problem and its applications is given in Section 2. Section 3 describes the clustering method and its implementation. Section 4 discusses the shape context measure. Results are presented in Section 5, Section 6 briefly discusses some future areas of work, and Section 7 concludes.

2 Mathematical Symbol Recognition

There are many uses for recognition of mathematical symbols, both typeset and handwritten. They range from on-the-fly recognition of formulas entered using handheld devices to converting printed documents containing mathematics into a digital form that captures the semantics of the symbols.

Online search engines like Google are typically not very useful when it comes to searching for mathematical statements or formulas. While markup languages for displaying mathematical languages do exist (such as MathML), they are not yet widely used, and other more common languages like \TeX are not suitable for use on the web. As a result a lot of the mathematics on the Internet is displayed using images. This is especially true for the large web-based mathematical resources that it would be very useful to be able to search, such as PlanetMath¹ and MathWorld². The use of images severely reduces the searching and indexing power of Google and the like which focus primarily on words and not images. Additionally, the search engines on the sites themselves have similar constraints, indexing pages on concepts and keywords and not the actual mathematics contained within.

If OCR could be applied to the images and the resulting individual symbols parsed to form a tree that captures some of the semantics of the mathematical expression, searching for

¹PlanetMath, www.planetmath.org

²MathWorld, mathworld.wolfram.com

symbols, subexpressions or entire formulas would be far more effective. It was with this goal in mind that this project was undertaken.

Given an image of an isolated symbol, we would like to identify that symbol based the symbols we have seen before.

3 Clustering

In [1] the authors describe a clustering method based on spectral methods and the Laplacian Eigenmap. Readers are directed to the paper itself for full details. Only the basics of the approach are presented here. We then expand the approach from two to arbitrarily many clusters.

The method uses pairwise weights between the points being clustered to label them as being in one of two clusters, where at least one point in each cluster is required to start the algorithm. The explanation below is an adaption of [1] and [3].

3.1 Two Clusters

Suppose we have a number of points x_i that we wish to group into two clusters based on some measure of distance between them. Let $y_i \in \{0, 1\}$ indicate the cluster of point x_i . Define the pairwise weights between all points using the heat kernel,

$$W_{ij} = e^{-\frac{d(x_i, x_j)}{t}}$$

where t is a parameter and $d(x_i, x_j)$ is some (symmetric) measure of similarity or distance between x_i and x_j . This function gives large weights to points that are close together (or similar) and small weights to those that are far apart (or very different). The authors of [1] note that they do not have a good rule for choosing the parameter t .

We wish to assign values to y_i such that the cost function E is minimized.

$$E = \sum_{i,j} W_{ij} (y_i - y_j)^2$$

Intuitively, this requires that points with large weights are assigned the same label, thereby clustering similar points together.

Let $Y = [y_1, \dots, y_n]$ be a vector of labels, one per point. We require that each cluster contains at least one point, so denote by Y_L the labels of the pre-labelled points and Y_U the labels of the unlabelled points. We can apply similar notation to W , D and L defined below.

Let $W = [W_{ij}]$ be a matrix of pairwise costs, D be a diagonal matrix such that $D_{ii} = \sum_j W_{ij}$, and let $L = D - W$.

It can be shown that minimising E to find Y requires solving

$$\hat{Y}_U = L_{UU}^{-1} W_{UL} Y_L$$

After calculating \hat{Y}_U we set y_i for each unlabelled point to 1 if $\hat{y}_i \geq 0.5$ and 0 otherwise, and thereby have a value for Y .

However, this calculation involves inverting a potentially very large matrix which may be poorly conditioned. We therefore present the following iterative algorithm to approximate Y_U , which involves inverting only the diagonal matrix D :

1. Choose \hat{Y}_U^0 , generally by setting each entry to 0.5.
2. Set $\hat{Y}_U^1 = \hat{Y}_U^0 D^{-1} W_{UU} + D^{-1} W_{UL} Y_L$
3. Set $Z^1 = \hat{Y}_U^1 - \hat{Y}_U^0$
4. Iterate:
 - (a) Set $Z^t = D^{-1} W_{UU} Z^{t-1}$
 - (b) Set $\hat{Y}_U^t = \hat{Y}_U^{t-1} + Z^t$

Since the number of points used in the project was small, both methods were used for comparison. After approximately 1000 iterations the two solutions were very similar, if not identical.

3.2 Multiple Clusters

We now extend this method up to an arbitrary number of clusters, say K . Redefine each y_i to be a soft-assignment vector for point i . Then the solution $Y = [y_i]$ is a matrix whose K columns are the assignments for each of the clusters. The cost function we wish to minimise becomes

$$E = \sum_{i,j} W_{ij} \|y_i - y_j\|^2$$

We can simplify this and find that

$$E = \sum_{i,j} W_{ij} \sum_{k=1}^K (y_{ik} - y_{jk})^2$$

And finally

$$E = \sum_{k=1}^K \sum_{i,j} W_{ij} (y_{ik} - y_{jk})^2$$

In other words, we perform K independent clusterings and then sum their costs to find the total cost which must be minimised. This implies that to cluster points into K clusters we simply perform K independent ‘‘all-against-one’’ clusterings, each time determining which points are in the current cluster and which are not (as opposed to determining points for *all* clusters in one go). The y_i vectors are still the soft-assignment vectors, as required.

For each clustering operation, different labels for Y_L are used. Specifically, only the points in the current cluster are labelled with a 1. All other pre-labelled points not in the cluster are 0, and the remaining points with unknown clusters are initialised to $\frac{1}{K}$.

An example of clustering points into three clusters for 2-dimensional data using the Euclidean metric is given in Figure 1. Both the direct and approximation methods (with 8000 iterations) were used, both producing the same result. The heat kernel parameter was $t = 0.25$.

This is exactly the method used in the project to cluster the mathematical symbols. However, we must still define $d(x_i, x_j)$ between two symbols x_i and x_j for use in determining the weights.

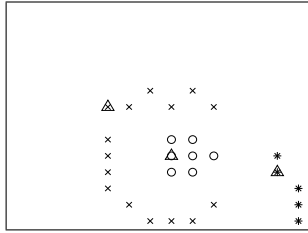


Figure 1: A clustering example using 2D points and the Euclidean metric. Those points marked with a \triangle were the initially labelled points for each cluster. This sort of clustering would not be possible with linear methods like K-means.

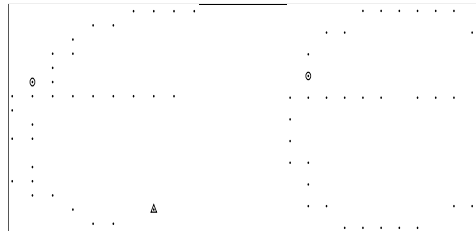


Figure 2: Examples of the points used for matching two epsilon symbols. The points marked with a circle and triangle have histograms shown in Figure 3.

4 The Shape Context Measure

In [2] the authors present a method called shape contexts for measuring the similarity between shapes in two images. The project uses this measure to calculate the “distance” between symbols for use when assigning weights.

The method is described briefly here with an emphasis on the project implementation. Readers are referred to the paper itself for a more thorough description and analysis.

4.1 Shape Context

We sample a set of n points from each of the two images that capture the overall shape of the image. Because the symbols used were simple and had thin lines, and in the interests of simplicity, a very basic method was used to choose these points. Those points that had grey values darker than some threshold were picked. The threshold was adjusted for each symbol to get a good approximation of the symbol. Generally, a more advanced method such as choosing points outputted by an edge detection algorithm would be more suitable. Figure 2 contains examples of the points used to compare two epsilon symbols.

For each point p_i on a shape we measure the relative distribution of the other $n - 1$ points. We do this by calculating the position of every other point using p_i as the origin, converting those positions to polar coordinates, and dividing the resulting r and θ values into bins to form a $\log(r)$ -by- θ histogram. As recommended in [2] we use 12 bins for θ and five for $\log(r)$. Note that using $\log(r)$ gives a histogram that is more sensitive to smaller r values than larger r values, resulting in a measure that is more locally-based than globally-based.

More formally, for each p_i we calculate a 60-bin normalised histogram h_i which is termed the *shape context* of p_i .

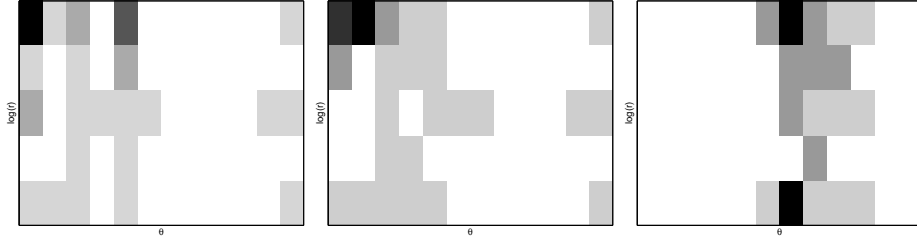


Figure 3: Histograms for the marked points in Figure 2, where a darker colour indicates a higher value. The leftmost one corresponds to the circled point on the first epsilon. The second one corresponds to the circled point on the second epsilon. The third histogram corresponds to the point in the first epsilon that is marked with a triangle. Notice the similarity between the first two histograms.

$$h_i(k) = \#\{q \neq p_i : (q - p_i) \in \text{bin}(k)\}$$

Example histograms for the three marked points in Figure 2 are given in Figure 3.

4.2 Distance Measure

Given the points for two shapes we wish to establish a one-to-one mapping between the points on each. Consider a point p_i on one shape and q_j on the other, the cost of matching the two points is given in [2] as

$$C(p_i, q_j) = \frac{1}{2} \sum_{k=1}^K \frac{(h_i(k) - h_j(k))^2}{h_i(k) + h_j(k)}$$

Given $C(p_i, q_j)$ for all p_i and q_j , we need to find the mapping π that minimises the total cost

$$H(\pi) = \sum_i C(p_i, q_{\pi(i)})$$

This is simply a square assignment problem that can be solved using the Hungarian method or other, possibly faster, methods. Third-party code³ for finding a solution using the Hungarian method was used in the project, which we felt was justified since it was not part of the project's primary focus.

Once the optimal matching π is found its cost $H(\pi)$ is the distance measure (written as $d(x_i, x_j)$ in Section 3) used in the spectral methods clustering algorithm. Figure 4 shows the optimal point matchings for the two epsilons in Figure 2.

5 Results

The two algorithms described above were implemented and used to cluster the six typeset mathematical symbols shown in Figure 5. Each of the three clusters is a type of symbol (either sigma, delta or epsilon) and each cluster contains two symbols, one labelled and one

³Courtesy of Niclas Borlin, Dept. of Computing Science, Umea University, Sweden. Available at www.cs.umu.se/~niclas

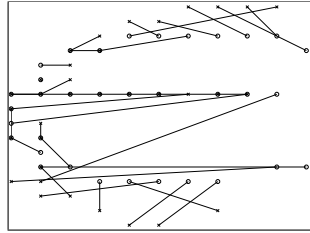


Figure 4: The optimal matching for the two sets of points in Figure 2. Because of the unequal densities of points in some areas, some points are forced to match with points far away on the other shape. This could be avoided if the points traced the true edges of the shapes, since differing densities would be avoided. The results remain useful.

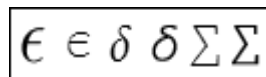


Figure 5: The symbols clustered (recognised) by the program. The leftmost of each symbol was considered known (ie. labelled) while the rightmost was the unknown symbol that had to be identified (labelled).

unlabelled. So 3 unknown symbols were shown to the system and it correctly identified each of them.

Once points describing each of the symbols had been extracted from the images (with an average of 43 points per symbol), the clustering was done using both the approximation algorithm and the direct calculation method as discussed in Section 3.1. With sufficient iterations—in this case 1000—the two methods produced the same results. The value $t = 2$ for the heat kernel parameter was found by trial and error. Values too far on either side of 2 tended to cluster all the symbols together.

Table 1 shows the distance values for the six symbols. Table 2 contains the soft cluster assignments for each symbol. The clusters are remarkably clear, given that the shape context measure produced similar distance values for delta 1 vs epsilon 1 and epsilon 1 vs epsilon 2. In all cases, the unknown symbols were correctly recognised and with a reasonably high degree of confidence.

	δ 1	Σ 1	ϵ 1	δ 2	Σ 2	ϵ 2
delta 1	0	12.2173	11.4545	8.2100	13.3168	12.5839
sigma 1	12.2173	0	12.8870	14.1745	8.0128	14.2519
epsilon 1	11.4545	12.8870	0	12.5719	14.2443	11.0181
delta 2	8.2100	14.1745	12.5719	0	13.6796	13.8002
sigma 2	13.3168	8.0128	14.2443	13.6796	0	14.9521
epsilon 2	12.5839	14.2519	11.0181	13.8002	14.9521	0

Table 1: The distances between symbols as produced by the shape context measure. Notice that it considers delta 1 and epsilon 1 to be almost as similar as the two epsilons. It also found the two epsilons harder to match than the pairs of the other two symbols. Despite this, the resulting clusters are very distinct.

	δ	Σ	ϵ
delta 1	1	0	0
sigma 1	0	1	0
epsilon 1	0	0	1
delta 2	0.7963	0.0893	0.1144
sigma 2	0.1059	0.8386	0.0555
epsilon 2	0.3277	0.1654	0.5069

Table 2: The cluster assignment probabilities for each symbol with the resulting clusters for the unknown symbols shown in bold.

6 Future Work

There are still a number of areas of future work that it would be interesting to examine. Using a large set of symbols and choosing points more intelligently (such as from an edge detection algorithm) would provide a better test of the power of the methods involved. Examining the impact of the choice of the t parameter for the heat kernel may provide a better method for choosing its value. Testing the limits of the direct calculation versus the approximation method, and the impact of the number of iterations, would also be interesting.

7 Conclusion

The project demonstrates the use of spectral clustering to identify typeset mathematical symbols. It also demonstrates use of the shape context measure for determining the similarity between images. The system was shown three unknown mathematical symbols which it correctly grouped with known images of the same symbol. The clustering of each symbol into one of the three groups was done by performing three independent “all-against-one” clusterings to produce a soft assignment vector for each symbol.

References

- [1] M. Belkin and P. Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. In T. G. Diettrich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems 14*, pages 585–591, Cambridge, 2002. MIT Press.
- [2] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(24):509–522, April 2002.
- [3] N. de Freitas. CS540 Machine Learning course notes. Dept. of Computer Science, UBC. Fall 2003.