Appendix: Kernelized Sorting

Novi Quadrianto
RSISE, ANU & SML, NICTA
Canberra, ACT, Australia
novi.quad@gmail.com

Le Song
SCS, CMU
Pittsburgh, PA, USA
lesong@cs.cmu.edu

Alex J. Smola
Yahoo! Research
Santa Clara, CA, USA
alex@smola.org

A Data Visualization

We compare a number of different object layout algorithms. Figure 1, 2, and 4 depict the results obtained by laying out 320 image objects into a 2D grid with Kernelized Sorting (our method), Self Organizing Map `http://www.cis.hut.fi/projects/somtoolbox/`, and Generative Topographic Mapping (GTM) `http://www.ncrg.aston.ac.uk/GTM/`. Figure 3 and 5 show the corresponding cluster members of SOM and GTM.

Figure 1: Layout of 320 images into a 2D grid of size 16 by 20 using Kernelized Sorting. Gaussian RBF kernel is used for the image objects and also for the positions of the grid. One can see that images are laid out in the 2D grid according to their color grading.

Figure 2: Layout of 320 images into a 2D grid of size 16 by 20 using Self-Organizing Map (SOM). Gaussian neighborhood and inverse learning rate functions are used. For SOM, several images are mapped to the same neuron and some neurons are left with no images assigned. This creates the blank spaces in the visualization. For the neuron with multiple images assigned, we only display one of the images assigned to it.

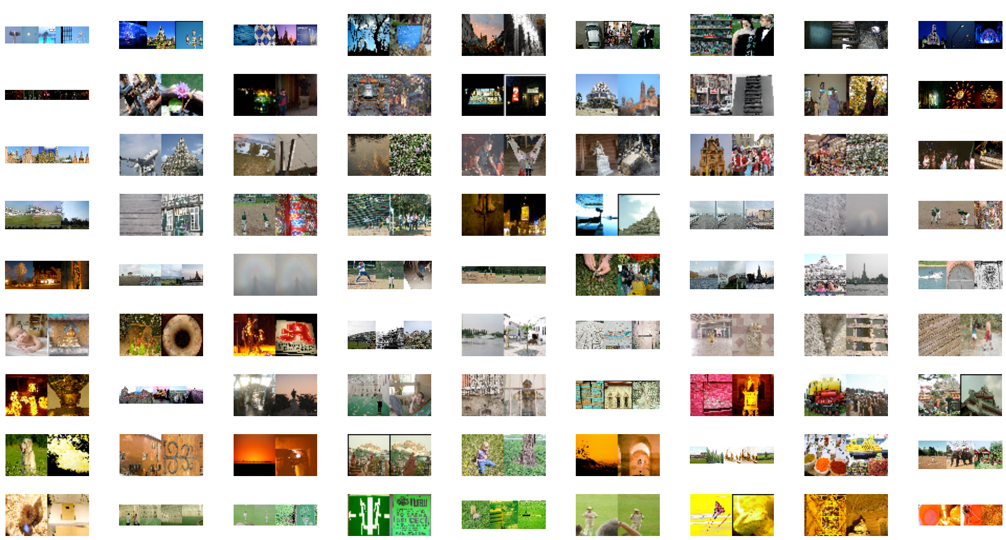

Figure 3: 81 out of the 320 neurons in SOM are assigned more than one image. This effectively cluster the images into 81 groups. This figure shows the cluster membership of each group.

Figure 4: Layout of 320 images into a 2D grid of size 16 by 20 using Generative Topographic Mapping (GTM). Principal components initialization is used and the projection of data into latent space (2D grid) is done via mode projection. Some grid points in the latent space does not have images assigned to them, and hence creating the blank spaces in the visualization. For the latent variables with multiple images assigned, we only display one of the images assigned to it.

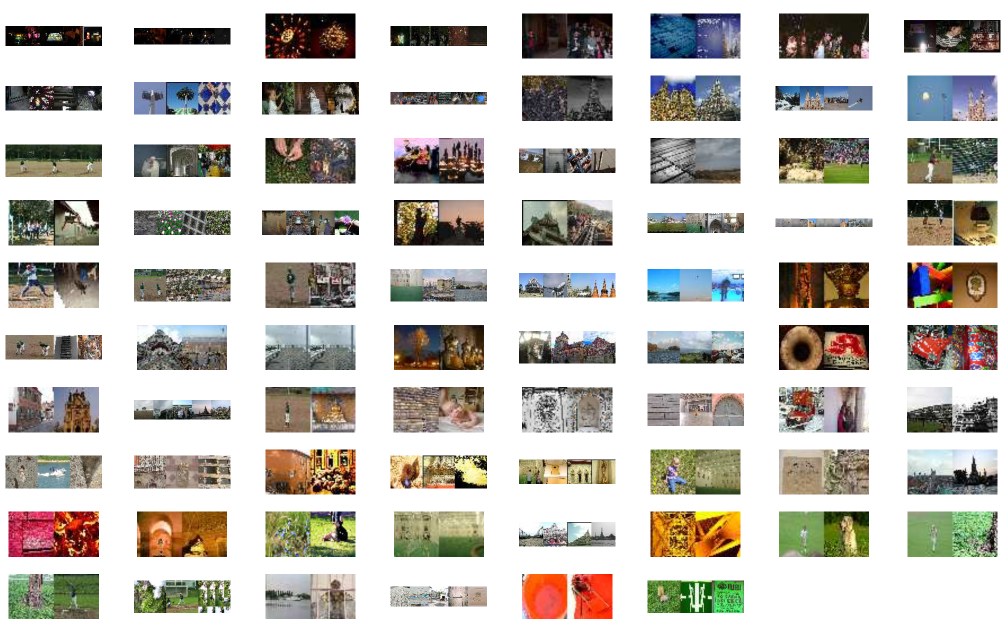

Figure 5: 78 out of the 320 latent variables in GTM are assigned more than one image. This effectively cluster the images into 78 groups. This figure shows the cluster membership of each group.

B Matching

B.1 Image matching

Figure 6: Image matching as obtained by Kernelized Sorting. The images are cut vertically into two equal halves: one half for computing entries in kernel matrix K and the other half for L. Kernelized sorting is used to pair up image halves that originate from the same images. 140 pairs out of 320 are correctly matched. This is quite respectable given that chance level would be 1 correct pair (a random permutation matrix has on expectation one nonzero diagonal entry).

B.2 Estimation

Table 1: The indexing of coordinates set members.

Type	Data set	Set A	Set B
Binary	australian	[0 1 2 3 4 5 8]	[6 7 9 10 11 12 13]
	breastcancer	[0 1 2 3 4 5]	[6 7 8 9]
	derm	[0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 24]	[17 18 19 20 21 22 23 25 26 27 28 29 30 31 32 33]
	optdigits	[0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 28]	[26 27 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51]
	wdbc	[0 1 2 3 4 5 6 7 8 9 10 11 14 15 16]	[12 13 17 18 19 20 21 22 23 24 25 26 27 28 29]
Multiclass	satimage	[0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 17 18]	[16 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35]
	segment	[0 1 3 4 5 6 7 8 11 13]	[9 10 12 14 15 16 17 18]
	vehicle	[0 1 2 3 4 5 6 7 8]	[9 10 11 12 13 14 15 16 17]
Regression	abalone	[0 1 3 4]	[2 5 6 7]
	bodyfat	[0 1 2 3 4 5 6 7]	[8 9 10 11 12 13]

B.3 Multilingual Document Matching

Table 2: Line search or automatic tuning of λ for nonconvex maximization. This table shows the number of correct matches (out of 300) for aligning documents in source languages to English documents.

Source language	Pt	Es	Fr	Sv	Da	It	Nl	De
Kernelized Sorting	241	216	193	99	83	236	211	70

B.4 Multivariate Extensions

In this experiment, we wish to sort 5 USPS digits of 0's with our multiway HSIC. Here we use the first digit as the reference set (i.e. $\pi_1 = \mathbf{1}$). The sorting performance is visualized by computing linear interpolations between sorted sets for 4 pairs of digits. Interpolation between sorted sets will result in a meaningful smooth movement and bending of the digit [1]. On each set, we use a Gaussian RBF kernel with median adjustment of the kernel width.

For comparison we also perform the same task using the method proposed by Jebara [1]. We implemented our own version as we were unable to obtain their code for reasons beyond the control of the authors of [1]. His simpler mean estimator + LAP is used for this experiment as it was observed that the experimental results were similar for the mean versus covariance estimator [1]. Here we also use a Gaussian RBF kernel with median trick as the base kernel. Although we are only interested in sorting 5 digits of 0's, we need to use more digits (200 in our experiments) to get a decent estimate of the feature space mean.

The interpolation results are shown in Figure 7 and 8. We also plot the correspondence for each digit pair in Figure 9 and 10. It can be seen from the correspondence plots that our method produced local flow consistency. As an example, in the second plot of Figure 9, all the arrows 'inside' the 0 are pointing downwards. However, in Figure 10 some arrows are pointing downwards and some are pointing upwards.

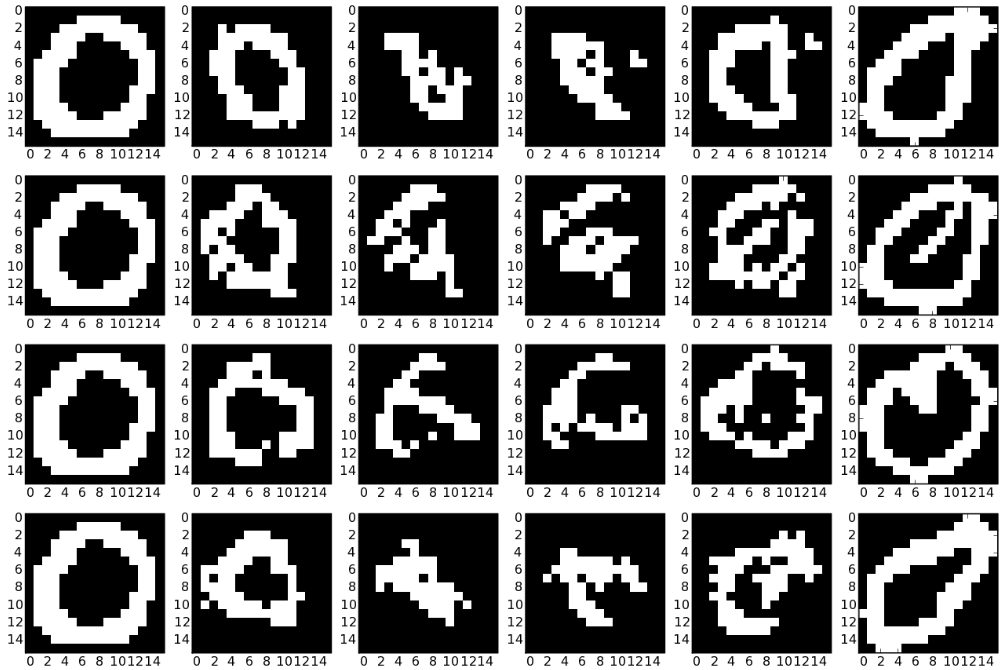

Figure 7: Linear interpolation of 4 pairs (shown per row) of 0's digits from left to right after sorting using multiway HSIC.

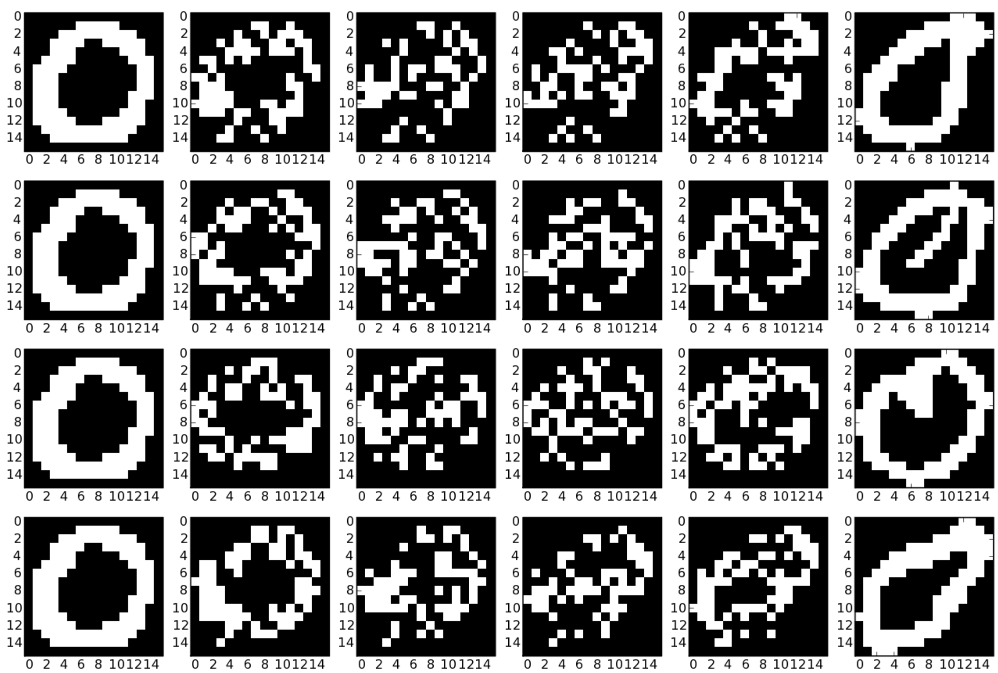

Figure 8: Linear interpolation of 4 pairs (shown per row) of 0's digits from left to right after sorting using [1].

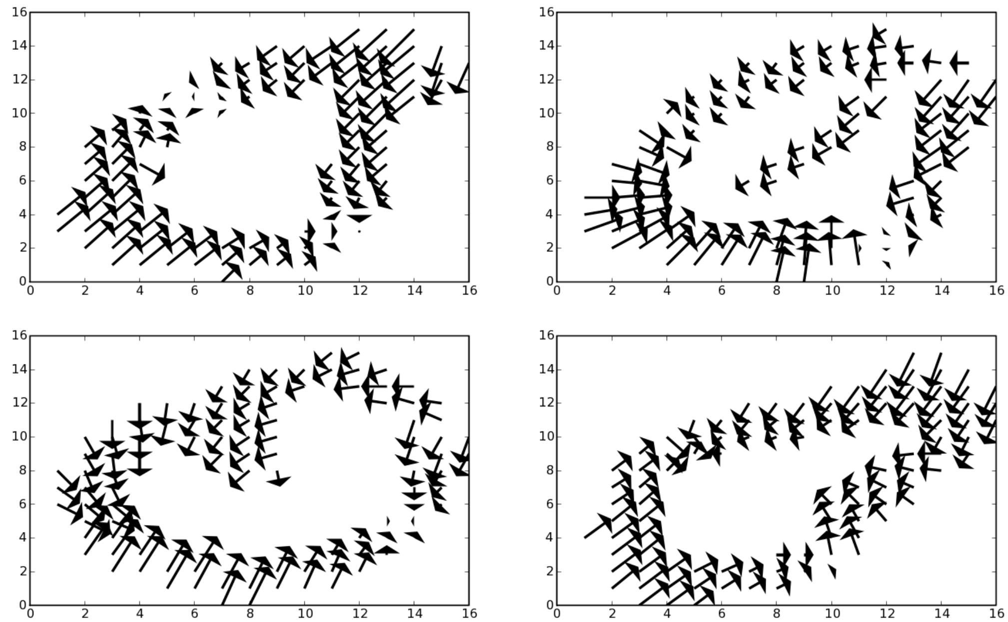

Figure 9: Arrows showing the matching of strokes of digit pairs sorted using multiway HSIC. Each sub-figure of this plot corresponds to a row in Figure 7.

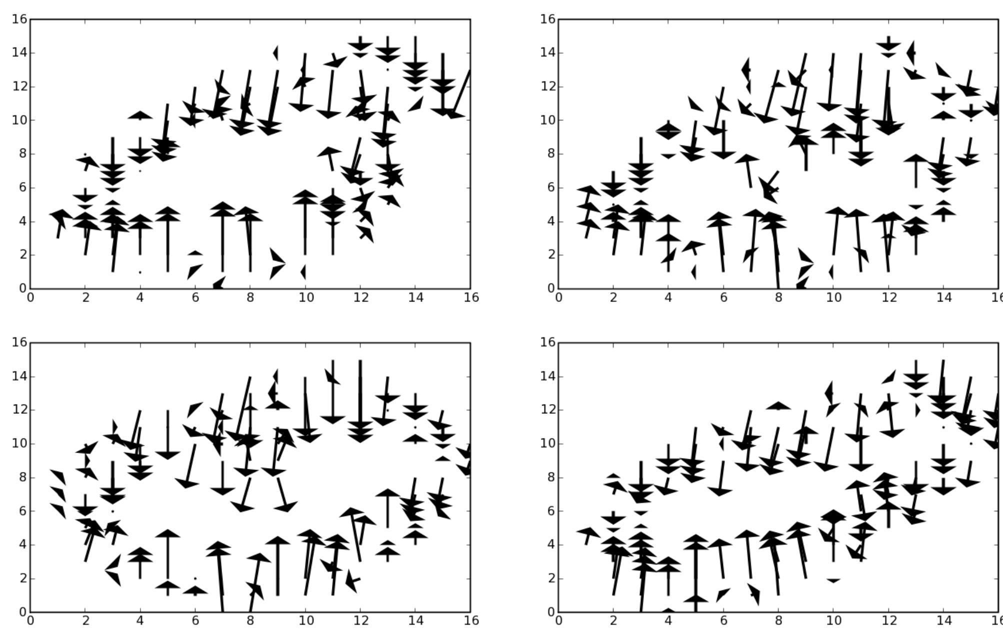

Figure 10: Arrows showing the matching of strokes of digit pairs sorted using [1]. Each sub-figure of this plot corresponds to a row in Figure 8.

References

[1] T. Jebara. Kernelizing sorting, permutation, and alignment for minimum volume PCA. In *Conference on Computational Learning Theory (COLT)*, volume 3120 of *LNAI*, pages 609–623. Springer, 2004.