KERNEL DENSITY ESTIMATION, AFFINITY-BASED CLUSTERING, AND TYPICAL CUTS

Deniz Erdoğmus *Miguel Á. Carreira-Perpiñán* *Umut Özertem*

Dept. of Computer Science & Electrical Engineering
OGI School of Science & Engineering, Oregon Health & Science University
20000 NW Walker Road, Beaverton, OR 97006, USA
Email: {deniz,miguel,ozertemu}@csee.ogi.edu

ABSTRACT

The typical cut [1, 2, 3] is a clustering method that is based on the probability p_{nm} that points $\mathbf{x}_n$ and $\mathbf{x}_m$ are in the same cluster over all possible partitions (under the Boltzmann distribution for the mincut cost function). We present two contributions regarding this algorithm. (1) We show that, given a kernel density estimate of the data, minimising the overlap between cluster densities is equivalent to the mincut criterion. This gives a principled way to determine what affinities and scales to use in the typical-cut algorithm, and more generally in clustering and dimensionality reduction algorithms based on pairwise affinities. (2) We introduce an iterated version of the typical-cut algorithm, where the estimated p_{nm} are used to refine the affinities. We show this procedure is equivalent to finding stationary points of a certain objective function over clusterings; and that at the stationary points the value of p_{nm} is 1 if n and m are in the same cluster and a small value otherwise. Thus, the iterated typical-cut algorithm sharpens the p_{nm} matrix and makes the cluster structure more obvious.

1. INTRODUCTION

Pairwise-distance algorithms for clustering use as starting point a collection of affinity values w_{nm} between pairs of points $\mathbf{x}_n$ and $\mathbf{x}_m$ (rather than the actual feature vectors). Often Gaussian affinities are used with a scale parameter σ. While a number of such algorithms exist, definition of the affinities is problematic, in particular selecting a good σ value requires running the clustering algorithm for a range of σ values.

In this paper we show a connection between kernel density estimation and the definition of affinities for clustering. We focus on a particular pairwise-distance algorithm, the typical cut, though the result is more general. We also study an extension of the typical-cut algorithm obtained by successively iterating it, which tends to polarise the affinities, so a simple inspection is enough to cluster the data.

We summarise the typical-cut algorithm in section 2. We give the relation with kernel density estimation in section 3 and experiments in section 4. We give the iterated typical-cut algorithm in section 5.

2. AFFINITY-BASED CLUSTERING, MINCUT AND TYPICAL CUT

We give a brief explanation of the typical-cut clustering algorithm of Blatt et al. [1]. Assume we have a data set $\{\mathbf{x}_n\}_{n=1}^N \subset \mathbb{R}^D$. First we define Gaussian affinities $w_{nm} = \exp\left(-\frac{1}{2}(\|\mathbf{x}_n - \mathbf{x}_m\|/\sigma)^2\right) \in (0, 1]$ where the width σ is taken as the average nearest-neighbour distance over the whole data set. A proximity graph structure can be imposed by selectively zeroing w_{nm} values: ref. [1] used a nearest-neighbour graph augmented with the minimum spanning tree (MST), but other graphs are also possible (e.g. the MST ensembles of [4]). Given a fixed integer $q > 2$, define a state vector $\mathbf{s} = (s_1, \ldots, s_N)$ where $s_n \in \{1, \ldots, q\}$ is the label for $\mathbf{x}_n$ (a Potts model with q spin values and N particles, in statistical mechanics[1]). Now we define the following cost function over states $\mathbf{s}$:

$$C(\mathbf{s}) = \sum_{n,m=1}^{N} w_{nm}(1 - \delta_{s_n,s_m}).$$

Thus, $C(\mathbf{s})$ is the sum of the affinities of all pairs of points whose states are different. Under the Potts model, $C(\mathbf{s})$ is the energy of the system state $\mathbf{s}$, where w_{nm} are the magnetic interactions between particles. An alternative view of $C(\mathbf{s})$ results if we see w_{nm} as the weight matrix of a graph: $C(\mathbf{s})$ is the value of the q-way graph cut (where each of the q subgraphs contain vertices with the same spin value). The mincut criterion [5] for graph-based clustering minimises precisely $C(\mathbf{s})$ over all clusterings. However in practice this often results in singleton clusters, which has prompted investigation of other graph-cut functions (e.g. normalised cuts [6] or the typical cut).

In the typical cut, we do not minimise $C(\mathbf{s})$ directly. Instead, we define a Boltzmann distribution over states at thermal equilibrium:

$$Q(\mathbf{s}) = \frac{1}{Z}\exp\left(-C(\mathbf{s})/T\right) \qquad Z = \sum_{\text{states } \mathbf{s}} \exp\left(-C(\mathbf{s})/T\right)$$

where $T > 0$ is the temperature of the system, which acts as a scale parameter. When $T \to 0$, $Q(\mathbf{s})$ tends to a delta distribution at the global minimum of the energy, which occurs at the state where all spins are aligned ($\delta_{s_n,s_m} = 1$), i.e., there is a single cluster. When $T \to \infty$, $Q(\mathbf{s})$ tends to a uniform distribution, with all states equally likely. Intermediate T values result in intermediate clusterings. At fixed T, we define the probability p_{nm} that points $\mathbf{x}_n$ and $\mathbf{x}_m$ are in the same cluster as their average co-alignment under the Boltzmann distribution, i.e., $p_{nm} = \langle\delta_{s_n,s_m}\rangle_Q = \sum_{\text{states } \mathbf{s}} Q(\mathbf{s})\delta_{s_n,s_m}$. Blatt et al. [1] argue that, when a cluster structure exists in the data, the clusters can be found by: (1) locating an appropriate T value by inspecting the behaviour of a certain function $\chi(T)$ (whose details we omit) which displays the phase transitions of the system; and (2) building a graph where points $\mathbf{x}_n$ and $\mathbf{x}_m$ are connected if $p_{nm} > \frac{1}{2}$. The connected components of this graph give the clusters. Blatt et al. [1] show good clustering results in several data sets.

[1] The resulting number of clusters is not restricted by the choice of q [1]. It is advisable to use a large q (which gives sharper results) though this also increases the state space and so the computational complexity.

The averages over Q (e.g. $p_{nm} = \langle \delta_{s_n,s_m} \rangle_Q$) must be computed approximately because the partition function Z is a sum over q^N states for which we lack a closed-form expression. Ref. [1] uses a Markov-chain Monte Carlo approach, where an average $A = \langle A(\mathbf{s}) \rangle_Q$ is approximated as $\frac{1}{M} \sum_{k=1}^{M} A(\mathbf{s}_k)$ by means of a sample $\{\mathbf{s}_k\}_{k=1}^{M}$ from the Q distribution, obtained using Swendsen-Wang sampling (vastly more efficient than Gibbs sampling for this problem). Other methods for approximating p_{nm} have been proposed (randomized trees sampling [2], generalised belief propagation [3]).

3. RELATION WITH KERNEL DENSITY ESTIMATION

Assume we have a kernel density estimate for the data set:

$$P(\mathbf{x}) = \frac{1}{N} \sum_{n=1}^{N} K_n(\mathbf{x} - \mathbf{x}_n).$$

For example, we can take a Gaussian kernel with fixed isotropic covariances: $K_n(\mathbf{x}-\mathbf{x}_n) = (2\pi\sigma^2)^{-D/2} \exp\left(-\frac{1}{2}(\|\mathbf{x} - \mathbf{x}_n\| / \sigma)^2\right)$. Given a partition of the data set into q clusters ($\mathbf{s} = (s_1, \ldots, s_N)$ with $s_n \in \{1, \ldots, q\}$), define the cluster densities

$$P_i(\mathbf{x}; \mathbf{s}) = \frac{1}{N_i} \sum_{n=1}^{N} K_n(\mathbf{x} - \mathbf{x}_n) \delta_{i,s_n}$$

(where $i = 1, \ldots, q$ indexes the clusters and N_i is the number of points in cluster i). That is, P_i is the kernel density estimate for the points in cluster i. We also have

$$P(\mathbf{x}) = \sum_{i=1}^{q} \frac{N_i}{N} P_i(\mathbf{x}) = \sum_{i=1}^{q} \pi_i P_i(\mathbf{x}).$$

We measure the density overlap between two clusters i, j as in [7]

$$\mathcal{C}_{ij}(\mathbf{s}) = \int P_i(\mathbf{x}; \mathbf{s}) P_j(\mathbf{x}; \mathbf{s}) \, d\mathbf{x}.$$

We obtain

$$\begin{aligned} \mathcal{C}_{ij}(\mathbf{s}) &= \int P_i(\mathbf{x}; \mathbf{s}) P_j(\mathbf{x}; \mathbf{s}) \, d\mathbf{x} \\ &= \frac{1}{N_i N_j} \sum_{n,m=1}^{N} \delta_{i,s_n} \delta_{j,s_m} w_{nm} = \frac{1}{N_i N_j} \sum_{\substack{n \in i \\ m \in j}} w_{nm} \end{aligned}$$

(that is, proportional to the sum of the edges that cross between clusters i and j) if we define the affinities

$$w_{nm} = \int K_n(\mathbf{x} - \mathbf{x}_n) K_m(\mathbf{x} - \mathbf{x}_m) \, d\mathbf{x}. \tag{1}$$

For example, for the isotropic Gaussian kernel of fixed width σ we obtain $w_{nm} = (4\pi\sigma^2)^{-D/2} \exp\left(-\frac{1}{2}(\|\mathbf{x}_n - \mathbf{x}_m\| / \sqrt{2}\sigma)^2\right)$.

A plausible clustering objective function to minimise over all partitions $\mathbf{s}$ is the collective density overlap:

$$\begin{aligned} \mathcal{C}(\mathbf{s}) &= N^2 \sum_{i \neq j}^{q} \pi_i \pi_j \mathcal{C}_{ij}(\mathbf{s}) \\ &= N^2 \left(\sum_{i,j=1}^{q} \pi_i \pi_j \mathcal{C}_{ij}(\mathbf{s}) - \sum_{i=1}^{q} \pi_i^2 \mathcal{C}_{ii}(\mathbf{s}) \right) \\ &= \left(\sum_{i,j=1}^{q} \sum_{\substack{n \in i \\ m \in j}} w_{nm} - \sum_{i=1}^{q} \sum_{n,m \in i} w_{nm} \right) = 2 \sum_{\substack{n,m \in \\ \neq clusters}} w_{nm}. \end{aligned}$$

Thus we see that minimising the collective density overlap is equivalent to minimising the sum of affinities of crossing edges, which is identical to the mincut objective function, and to the energy function in the typical-cut Boltzmann distribution. Beyond those two methods, our result suggests a principled, objective way to determine the pairwise affinity between two points, namely as given by a kernel density estimate (which in turn was estimated to optimise e.g. the likelihood of the data). For example, if we want to use isotropic Gaussian affinities of fixed width σ, then the appropriate σ value is that corresponding to a Gaussian kernel density estimate with fixed isotropic covariances $2\sigma^2\mathbf{I}$. Apart from relating two different problems (clustering and density estimation), this avoids solving the clustering problem for many σ values in order to obtain a good clustering—instead, we find a good density estimate, which may be more efficient.

4. EXPERIMENTS

Here we show that deriving the affinities from a kernel density estimate using eq. (1) results in high-quality affinities that in turn improve clustering with e.g. the typical-cut algorithm. We use isotropic Gaussian affinities $w_{nm} \propto \exp\left(-\frac{1}{2}(\|\mathbf{x}_n - \mathbf{x}_m\| / \sigma_{nm})^2\right)$, i.e., we assume a Gaussian kernel density estimate. We test the following rules for selecting the widths σ_{nm}:

1. Constant $\sigma_{nm} = a$, equal to the average nearest-neighbour distance over the whole data set. This is the rule used in the original typical-cut paper [1].
2. Constant $\sigma_{nm} = \sigma$, set as in the rule in [8, pp. 85–87], which is optimal in the mean integrated square error sense.
3. Constant $\sigma_{nm} = \sigma$, set by a leave-one-out maximum likelihood estimator [9].
4. Adaptive kernel density estimator where $K_n(\mathbf{x} - \mathbf{x}_n)$ is isotropic Gaussian with width $\sigma_n = \sigma a_n$ where a_n is the average distance to the k nearest neighbours of $\mathbf{x}_n$ (k is chosen in advance and fixed for all $\mathbf{x}_n$), and σ is chosen to maximise the likelihood. This results in adaptive affinities $w_{nm} = \left(2\pi\left(\sigma_n^2 + \sigma_m^2\right)\right)^{-\frac{D}{2}} \exp\left(-\frac{1}{2}\left(\|\mathbf{x}_n - \mathbf{x}_m\| / \sqrt{\sigma_n^2 + \sigma_m^2}\right)^2\right)$.

It is also possible to use full-covariance kernels, or non-Gaussian kernels, to define the affinities w_{nm} in eq. (1).

Figure 1 shows the results for a data set containing two very different scales: one compact cluster and one loose cluster. We show the affinity matrices for each selection rule. It is apparent that, when a constant σ is used, the traditionally established rules (based on maximising likelihood or minimising the error) result in significantly better affinities than the typical-cut rule; while the adaptive kernel estimator obtains the best results.

5. ITERATED TYPICAL CUT

In the typical cut, we hope that the probabilities $p_{nm} = \langle \delta_{s_n,s_m} \rangle_Q$ are significantly high for points that should be in the same cluster; the connected components of the graph defined by thresholding p_{nm} give then the clusters. We can see the p_{nm} values as refined versions of the original affinities w_{nm}. The latter were purely local in nature, since they depend only on $\mathbf{x}_n$ and $\mathbf{x}_m$, and are thus not foolproof predictors of whether $\mathbf{x}_n$ and $\mathbf{x}_m$ should be clustered together. The p_{nm} introduce contextual information, so that pairs of points with the same affinity may have significantly different p_{nm} values.

It seems then natural to refine the affinities even further by feeding the p_{nm} values back to the typical-cut algorithm. That is, we

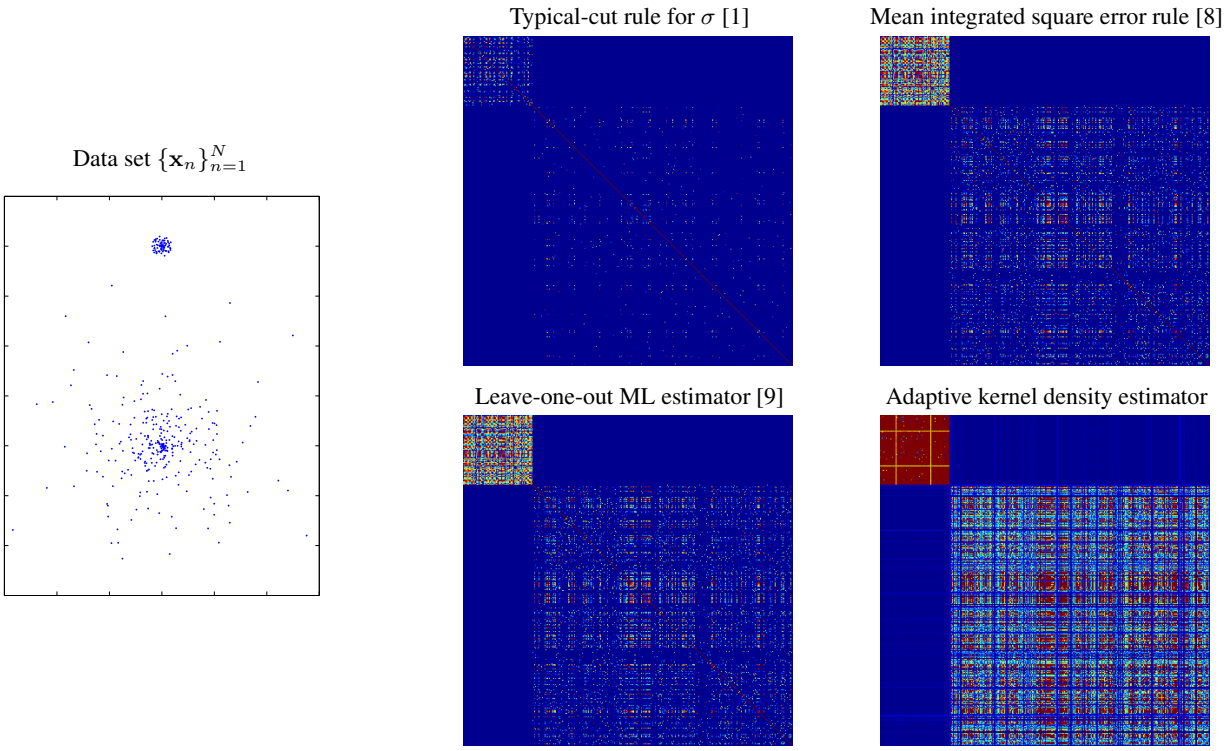

Fig. 1. Affinity matrices (w_{nm}) (colour-coded) obtained with the different rules for the affinity scale σ for the dataset shown on the left. The rules derived from the density estimate improve over the typical-cut rule. In particular, the adaptive kernel density estimator is able to use different scales in different regions of the data and so produce a much better affinity matrix, as is apparent in its block structure.

start with affinities w_{nm} and obtain p_{nm}; then we define as input to the typical-cut algorithm the refined affinities $p_{nm}w_{nm}$, and iterate. If this process converges, the $\mathbf{P} = (p_{nm})$ values must satisfy the self-consistent equation:

$$p_{nm} = \sum_{\text{states } \mathbf{s}} \frac{e^{-\frac{C(\mathbf{s},\mathbf{P})}{T}}}{Z(\mathbf{P})} \delta_{s_n,s_m} \tag{2}$$

$$Z(\mathbf{P}) = \sum_{\text{states } \mathbf{s}} e^{-\frac{C(\mathbf{s},\mathbf{P})}{T}}, \quad C(\mathbf{s},\mathbf{P}) = \sum_{n,m=1}^{N} w_{nm} p_{nm} (1 - \delta_{s_n,s_m}).$$

We can interpret this result in a Markov chain sense: successive iterations propagate information over the graph (defined by w_{nm}), sharpening the result obtained in the first iteration. Note that it makes no sense to redefine the affinities as p_{nm} directly rather than $p_{nm}w_{nm}$. In the former case, the w_{nm} would only initialise the fixed-point iteration but eq. (2) would not depend on w_{nm}, which is precisely the dataset-dependent information.

This equation can be used as the basis of a fixed-point iteration. Assuming $w_{nm} > 0 \; \forall n, m$, it is easy to show that this equation can also be derived as the stationary point of the following cost function:

$$\begin{aligned} J(\mathbf{P}) &= \sum_{\text{states } \mathbf{s}} \exp\left(-\frac{C(\mathbf{s},\mathbf{P})}{T} + \frac{1}{T}\sum_{n,m=1}^{N} w_{nm}\left(p_{nm} - \frac{p_{nm}^2}{2}\right)\right) \\ &= Z(\mathbf{P}) \exp\left(\frac{1}{T}\sum_{n,m=1}^{N} w_{nm}\left(p_{nm} - \frac{p_{nm}^2}{2}\right)\right). \end{aligned}$$

We can gain understanding of the character of the fixed points of J by examining its extreme cases:

- When $T \to 0$, the Boltzmann distribution tends to a delta centred at the global minimum of $C(\mathbf{s},\mathbf{P})$, namely the single-cluster case where all spins are aligned ($s_n = s_m \; \forall n, m$). We obtain $p_{nm} = 1 \; \forall n, m$, i.e., $\mathbf{P}$ is a matrix of ones. The Hessian of J is proportional to $-\operatorname{diag}(w_{nm})$ and so negative definite, thus the solution is a maximum of $J(\mathbf{P})$.
- When $T \to \infty$, the Boltzmann distribution tends to a uniform distribution, with all states equally likely. We obtain that the fixed point is at $p_{nm} = 1$ if $n = m$ and $\frac{1}{q}$ if $n \neq m$, and again the Hessian of J is proportional to $-\operatorname{diag}(w_{nm})$, thus negative definite, and the solution is a maximum.

This is also confirmed by analysing exactly the simple case of $N = 2$ data points, for which eq. (2) becomes $p = 1/(1+(q-1)e^{-2wp/T}) \in (\frac{1}{q}, 1)$. For small T, $p \approx 1$; for large T, $p \approx \frac{1}{q}$; and for a range of intermediate T, there are 3 fixed points, of which only $p \approx 1$ and $p \approx \frac{1}{q}$ are stable.

This suggests that for the general case, for stable clusterings at intermediate values of T (as detected e.g. using the $\chi(T)$ function in [1]) we should obtain $p_{nm} \approx 1$ if n and m are in the same cluster and $p_{nm} \approx \frac{1}{q}$ otherwise. Therefore, the (possibly permuted) matrix $\mathbf{P}$ is made up of blocks of ones along the diagonal (for the intra-cluster p_{nm}) and values $\approx \frac{1}{q}$ elsewhere. Usually q is quite larger than 2, so $\frac{1}{q} \ll 1$ and just by inspection we can tell which points are in the same cluster without the need to compute connected components. This is confirmed in the experiment of fig. 2 (using Swendsen-Wang

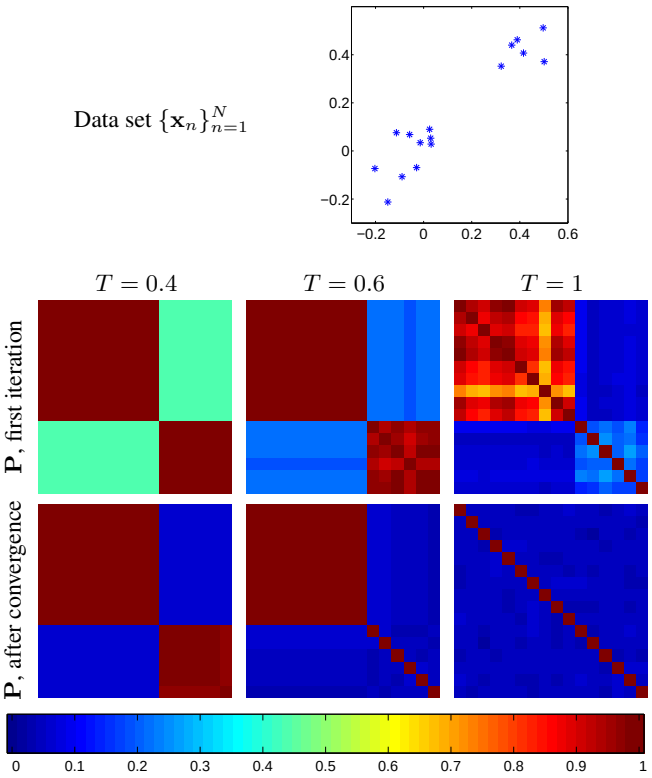

Fig. 2. Matrix $\mathbf{P} = (p_{nm})$ of contextual affinities produced after convergence of the iterated typical-cut algorithm for the dataset shown on the top, for 3 regimes of T, using Gaussian affinities with $\sigma = 0.2$, $q = 20$ spin values and $M = 1\,000$ Swendsen-Wang samples (the dataset contains only $N = 16$ points so the individual p_{nm} values can be seen). We show the $\mathbf{P}$ matrix that results from the first iteration (i.e., the original typical-cut algorithm) and after convergence of the iterated typical-cut algorithm (which occurred in a few iterations). In the first iteration, the values of p_{nm} can be farther from 1 (dark red) and $\frac{1}{q} = 0.05$ (dark blue); notice the colorbar. Thus, thresholding $\mathbf{P}$ and finding its connected components is necessary. In the iterated version, all p_{nm} approximately converge to either 1 (diagonal blocks, i.e., clusters) or $\frac{1}{q}$ (off-diagonal blocks), thus sharpening the affinities. Note the number of clusters for a given T need not be the same for both versions.

sampling to estimate the p_{nm} probabilities [1]), which shows how the initial $\mathbf{P}$ contains a wide range of values in $(0, 1)$ while the final $\mathbf{P}$ (after a few iterations) binarises into the two poles $\{\frac{1}{q}, 1\}$.

Computationally, note that we cannot reuse the same Swendsen-Wang sample for different iterations since the Boltzmann distribution depends on $\mathbf{P}$, which changes over iterations. Experimentally we observe that the "1" values converge very fast (a couple of iterations) while the "$\frac{1}{q}$" values oscillate mildly; this is presumably due to the sampling noise.

In summary, by starting with local affinities w_{nm} and iterating the construction of contextual affinities p_{nm} defined by the typical cut, we obtain a sharply binarised affinity matrix $\mathbf{P}$ where the clustering structure at the given scale T is obvious. This is an alternative to thresholding the initial $\mathbf{P}$ and finding its connected components.

6. DISCUSSION

We have shown (1) that one can define pairwise affinities between data points given a kernel density estimate, and (2) that the iterated typical cut can produce sharp contextual affinities. Although we have focused on the typical cut, our first result applies more generally to pairwise-distance algorithms for clustering (e.g. normalised cuts, spectral clustering [6]) and dimensionality reduction (e.g. Laplacian eigenmaps [10]). Such algorithms require a search (or clever choice) over the affinity scale parameter σ to succeed. Our result transforms the problem of finding good affinities and good scales σ into finding good density estimates—a well-researched problem for which objective rules exist. (Conversely, given a ground-truth clustering of the data, we could invert this process to find σ for which this clustering occurs and use it for kernel density estimation.) This point of view brings together density estimation, proximity graphs and affinities in a principled way. For example, using a finite-support kernel (such as the Epanechnikov kernel, i.e., quadratic up to distance σ, 0 beyond) automatically results in a sparse graph. By using an adaptive kernel size we obtain both a density estimate and a proximity graph (given by the affinities).

7. REFERENCES

[1] M. Blatt, S. Wiseman, and E. Domany, "Data clustering using a model granular magnet," *Neural Computation*, vol. 9, no. 8, pp. 1805–1842, Nov. 1997.

[2] Y. Gdalyahu, D. Weinshall, and M. Werman, "Self organization in vision: Stochastic clustering for image segmentation, perceptual grouping, and image database organization," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 23, no. 10, pp. 1053–1074, Oct. 2001.

[3] N. Shental, A. Zomet, T. Hertz, and Y. Weiss, "Learning and inferring image segmentations using the GBP typical cut algorithm," in *Proc. 9th Int. Conf. Computer Vision (ICCV'03)*, Nice, France, Oct. 14–17 2003, pp. 1243–1250.

[4] M. Á. Carreira-Perpiñán and R. S. Zemel, "Proximity graphs for clustering and manifold learning," In Saul et al. [11], pp. 225–232.

[5] Z. Wu and R. Leahy, "An optimal graph theoretic approach to data clustering: Theory and its application to image segmentation," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 15, no. 11, pp. 1101–1113, Nov. 1993.

[6] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888–905, Aug. 2000.

[7] R. Jenssen, D. Erdogmus, J. C. Principe, and T. Eltoft, "The Laplacian PDF distance: A cost function for clustering in a kernel feature space," In Saul et al. [11], pp. 625–632.

[8] B. W. Silverman, *Density Estimation for Statistics and Data Analysis*, Chapman & Hall, 1986.

[9] R. P. W. Duin, "On the choice of the smoothing parameters for Parzen estimators of probability density functions," *IEEE Trans. Computers*, vol. 25, no. 11, pp. 1175–1179, Nov. 1976.

[10] M. Belkin and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," *Neural Computation*, vol. 15, no. 6, pp. 1373–1396, June 2003.

[11] L. K. Saul, Y. Weiss, and L. Bottou, Eds., *Advances in Neural Information Processing Systems*, vol. 17, 2005.