Annealing and the Normalized N-Cut

Tomáš Gedeon [a,1] Albert E. Parker [a] Collette Campion [a]

[a] *Department of Mathematical Sciences, Montana State University, Bozeman, MT 59715, USA*

Zane Aldworth [b]

[b] *Center for Computational Biology, Montana State University, Bozeman, MT 59715, USA*

Abstract

We describe an annealing procedure that computes the normalized N-cut of a weighted graph G. The first phase transition computes the solution of the approximate normalized 2-cut problem, while the low temperature solution computes the normalized N-cut. The intermediate solutions provide a sequence of refinements of the 2-cut that can be used to split the data to K clusters with $2 \leq K \leq N$. This approach only requires specification of the upper limit on the number of expected clusters N, since by controlling the annealing parameter we can obtain any number of clusters K with $2 \leq K \leq N$. We test the algorithm on an image segmentation problem and apply it to a problem of clustering high dimensional data from the sensory system of a cricket.

Key words: Clustering, annealing, normalized N-cut.

1 Introduction

There is vast literature devoted to problems of clustering. Many of the clustering problems can be formulated in the language of graph theory [1–6]. Objects which one desires to cluster are represented as a set of nodes V of a graph G and the weight w associated with each edge represents the degree of similarity between

Email addresses: gedeon@math.montana.edu (Tomáš Gedeon), parker@math.montana.edu (Albert E. Parker), collettemcampion@yahoo.com (Collette Campion), zane@cns.montana.edu (Zane Aldworth).

[1] Corresponding author, tel: 1 406 994 5359, fax: 1 406 994 1789

the two adjacent nodes. After the construction of the graph, a cost function, that characterizes the putative solution, is minimized to obtain a clustering of the data.

One of the popular choices for image segmentation is the *normalized cut (Ncut)*, introduced by Shi and Malik [6],

$$Ncut(A,B) = \frac{links(A,B)}{degree(A)} + \frac{links(A,B)}{degree(B)}, \tag{1}$$

where, following Yu and Shi [7]

$$links(A,B) = \sum_{a\in A, b\in B} w(a,b) \quad \text{and} \quad degree(A) = links(A,V).$$

Here, A and B are subsets of V. While $links(A,B)$ is the total weighted connections from A to B, the degree $degree(A)$ is the total links from A to all the nodes. For a problem of an optimal partitioning of the vertex set into N clusters $A_1,\ldots,A_N$, Yu and Shi [7] define

$$Ncut(\{A_i\}_i^N) := \frac{1}{N}\sum_{i=1}^{N}\frac{links(A_i, V\setminus A_i)}{degree(A_i)}, \quad Nassoc(\{A_i\}_i^N) := \frac{1}{N}\sum_{i=1}^{N}\frac{links(A_i,A_i)}{degree(A_i)}$$

and show that $Ncut(\{A_i\}_{i=1}^N) + Nassoc(\{A_i\}_{i=1}^N) = 1$. It follows that maximizing the associations and minimizing the cuts are achieved simultaneously.

As shown by Shi and Malik [6], finding a normal cut even for a bi-partition problem is NP-complete. However, the approximate solution, the *approximate normal cut* can be found efficiently as a second eigenvector of a matrix related to the Laplacian of the graph G ([9]) after relaxing the discrete problem to a continuous problem. The advantage of this approximation is that it provides a fast near-global solution. However, to obtain a solution of the original problem another clustering problem needs to be solved, usually using heuristics such as K-means [4,6], dynamic programming [10], greedy pruning or exhaustive search [6]. A principled way to find such a solution from the approximate (relaxed) solution was recently formulated by Yu and Shi [7]. Since the approximate normal cut is closely related to the spectral clustering methods, these methods share the same limitations when applied to image segmentation of images [8].

Where the spectral graph theory based clustering algorithms provide fast near-global solutions, the annealing algorithms take a different approach. Here the emphasis is on the control of the quality of the solution via a selection of the annealing parameter. The solution "emerges" gradually as a function of this parameter, rather then being computed at once [11–13]. A special class of annealing problems involve information distortion type cost functions [15,16,12,13] which have been

applied to clustering problems in neuroscience, image processing, spectral analysis, gene expression, stock prices, and movie ratings [14,19,17,18]. A starting point for these problems is usually a joint probability distribution $p(X, Y)$ of discrete random variables X and Y, and the goal is to find the clustering of Y into a predetermined number of classes N that captures most of the mutual information between X and Y. The membership of the elements $y \in Y$ in classes $t_1, \ldots, t_N$ is described by a conditional probability $q(t_i|y)$. The annealing procedure starts at a homogeneous solution where all the elements of Y belong to all the classes with the same probability at the value of the annealing parameter $\beta = 0$. The parameter β plays the role of $1/T$, where T is the annealing temperature. The algorithm consists of incrementing the value of β, initializing a fixed point search at the solution value at the previous β and iterating until a solution at the new β is found. Various improvements of this basic algorithm are possible [20,16,13]. An alternative approach is to use agglomeration [21,22] starting at $\beta = \infty$ and lowering the value of β. This approach requires the set of classes at least as large as the cardinality of Y so that each element of Y belongs to its own class at $\beta = \infty$. As β decreases, the elements of Y aggregate to form classes. Similar ideas have been used in [23] for fast multi-scale image segmentation.

In this paper we show that the seemingly very different approaches to clustering, graph-theoretical and information-theoretical, are connected. We will show that there is an information-like cost function whose solution as $\beta \to \infty$ solves the normalized cut problem of an associated graph, and the solution at the first phase transition solves the approximate (relaxed) normalized 2-cut of the same graph. Subsequent phase transitions then provide approximate solutions to the normalized K-cut problem with $2 < K \leq N$, see Figure 1. The value of β at the first phase transition does not depend on the choice of N. This first set of results unites the discrete and the relaxed solutions of the normalized N-cut in a unified framework. Notice, that in these results we start with a joint distribution $p(X, Y)$ and derive the graph G for which the N-cut is being computed. We call this a ***forward*** problem.

Obviously, the key issue for the applications to computation of the N-cut is to start with a given graph G and then construct sets X, Y together with a probability distribution $p(X, Y)$ in such a way, that the annealing computes N-cut of G. This constitutes an ***inverse*** problem. We solve the inverse problem in section 4. Given a graph G, the set Y will correspond to the set of vertices, the set X to the set of edges of G and the distribution $p(X, Y)$ will be constructed from the edge weights of G.

This leads to the following algorithm:

(1) **Input:**
- A graph G with edge weights w_{ij};
- an integer N specifying an upper bound of the expected number of classes;
- δ- a margin of acceptance for class membership.

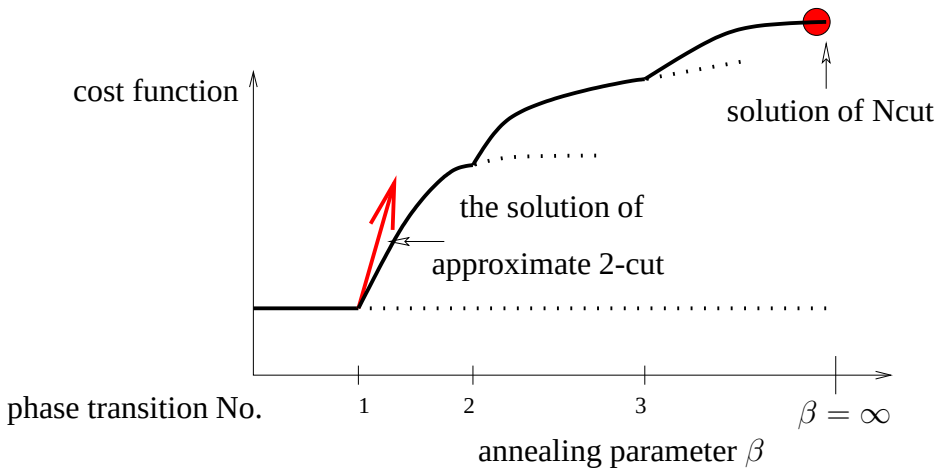

Fig. 1. The tangent vector at the first phase transition computes approximate the 2-cut, while the solution at $\beta = \infty$ computes Ncut. The subsequent phase transitions are indicated by the dotted lines along the primary branch of solutions.

- a terminal value β_{max}.

(2) **Output:** Separation of vertices of G into K classes with $2 \leq K \leq N$.

(3) **Algorithm:**

- Given the weights w_{ij} construct random variables X, Y and the probability distribution $p(X, Y)$ (see section 4);
- Starting at value $\beta_{start} = \beta^*$ and $q(\eta|y) = 1/N + \epsilon v$, where β^* and v are computed in Theorem 2, repeat:
  (a) increment $\beta = \beta + \Delta\beta$;
  (b) compute a zero of the gradient of the Lagrangian $\nabla\mathcal{L}(q, \beta + \Delta\beta)$ (see (20)).
- Stop, if one of these stopping criteria apply:
  (a) for each y there exists a class μ such that $q(\mu|y) \geq q(\nu|y) + \delta$ for all classes $\nu \neq \mu$;
  (b) annealing parameter β reached the terminal value β_{max};

Computation of the zero of the Lagrangian can be done in many different ways, ranging from a fixed point iteration to a Newton method.

The paper is organized as follows. In section 2 we review the approximation that leads to the approximate normalized cut and introduce the annealing problem. In section 3 we consider the forward problem and in section 4 we discuss the inverse problem. We finish the section by a few illustrative examples and in section 6 we apply our approach to the problem of clustering neural data and an image segmentation problem studied by Shi and Malik [6].

2 Preliminaries

2.1 Approximate normalized cut

Given a graph $G = (V, E)$ and the weights w_{ij} associated with the edge connecting vertices i and j, we let $d(i) = \sum_j w(i, j)$ be the total connection from node i to all other nodes. Let $n = |V|$ be the number of nodes in the graph and let D be an $n \times n$ diagonal matrix with values $d(i)$ on the diagonal, and let W be an $n \times n$ symmetric matrix with $W(i, j) = w_{ij}$.

Let x be an indicator vector with $x_i = 1$ if node i is in A and $x_i = -1$ otherwise. Shi and Malik [6] show that minimizing the normalized 2-cut over all such vectors x is equivalent to the problem

$$min_y \frac{y^T(D - W)y}{y^T Dy} \tag{2}$$

over all vectors y with $y(i) \in \{1, -b\}$ for a certain constant b and satisfying the constraint

$$y^T D\mathbf{1} = 0. \tag{3}$$

If one does not require the first constraint $y(i) \in \{1, -b\}$ and allows for a real valued vector y then the problem is computationally tractable. The computation of the real valued vector y, satisfying (2) and (3), is known as the approximate normalized cut. Given such a real vector y, the bipartition of the graph G can be achieved by putting all vertices i with $y(i) > 0$ to class A and all vertices j with $y(j) \leq 0$ to class B. Other ways to associate vertices to the classes A and B based on the vector y are certainly possible.

The problem (2) with constraint (3) can be further simplified. Following again [6] consider a generalized eigenvalue problem

$$(D - W)y = \mu Dy. \tag{4}$$

Problem (2) is solved by the smallest eigenvalue of (4). However, the smallest eigenvalue of (4) is zero and corresponds to an eigenvector $y_0 = \mathbf{1}$. This vector does not satisfy the constraint (3). It can be shown ([6]), that all other eigenvectors of (4) satisfy the constraint. Therefore the problem (2) with the constraint (3) is solved by second smallest eigenvector of problem (4).

2.2 The annealing problem

Given a joint distribution $p(X,Y)$, where X and Y are finite discrete random variables, let T be a *reproduction* discrete random variable with $|T| = N$. We will denote the values of T by the Greek letters $\mu, \eta, \nu, \ldots$. Let $q(\eta|y)$ be the conditional probability $q(\eta|y) = prob(T = \eta|Y = y)$. Given $q(\eta|y)$ we define $p(x,\mu) := \sum_y q(\mu|y)p(x,y)$ and

$$Z(X,T) = \frac{1}{2\ln 2}\sum_{x,\mu} p(x,\mu)\left(\frac{p(x,\mu)}{p(x)p(\mu)} - 1\right), \tag{5}$$

where the $\ln$ denotes the natural logarithm. Consider an annealing problem

$$max_{q(\eta|y)} H(T|Y) + \beta Z(X,T), \tag{6}$$

where maximization is over the conditional probability $q(\eta|y)$ and $H(T|Y) = -\sum_{\mu,y} p(\mu,y)\log q(\mu|y)$ is the conditional entropy. Here, and for the rest of the paper, $\log$ denotes a base 2 logarithm. Intuitively, states of the random variable T represent classes, into which we try to cluster members of Y, in a way which maximizes (6). The assignment of $y \in Y$ to $\eta \in T$ is probabilistic with the value $q(\eta|y)$.

We describe the details of the annealing algorithm elsewhere [15,16]. We first fix N the upper bound for the number of clusters we seek. Since (6) is a constrained optimization problem, we form the corresponding Lagrangian. The gradient of the Lagrangian is zero at the critical points of (6). For small values of β, that is, for $\beta \leq 1$ (see Remark 4), the only solution is $q(\eta|y) = 1/N$. Incrementing β in small steps and initializing a zero finding algorithm at a solution at the previous value of β, we find the zero of the gradient of the Lagrangian at the present value of β. For many zero finding algorithms, such as the implicit solution method discussed in [15], the cost of solving (6) is proportional to $N \times |X| \times |Y| \times \mathcal{B}$, where $\mathcal{B}$ is the number of increments in β. Newton-type zero-finding algorithms need to evaluate the Hessian of (6), and so the cost in this cases is proportional to $N^2 \times |X| \times |Y|^2 \times \mathcal{B}$.

The function Z has similar properties to the mutual information function $I(X,T) = \sum_{\mu,x} p(x,\mu)\log\frac{p(x,\mu)}{p(\mu)p(x)}$ ([28]), which we view as a function of $q(\mu|y)$ in view of $p(x,\mu) = \sum_y q(\mu|y)p(x,y)$ and $p(\mu) = \sum_y q(\mu|y)p(y)$. Indeed the underlying functions $i(x) := x\log x$ and $z(x) := x(x-1)$ are both convex, satisfy $i(1) = z(1) = 0$ and eventhough $i(0)$ is not defined, the limit $\lim_{x\to 0} i(x) = 0 = z(0)$. Further, the maximum of both $-\sum_{i=1}^N p_i \log p_i$ and $-\sum_{i=1}^N p_i(p_i - 1)$ on the space of admissible probability vectors satisfying $\sum_i p_i = 1$ is attained at $p_i = 1/N$. By Lemma 7 in the Appendix the function $Z(X,T)$ is a non-negative convex function of $q(\eta|y)$. Note that $Z(X,T)$ shares these important properties with the mutual

information function $I(X,T)$. As a consequence, for a generic probability distribution $p(X,Y)$ the maximizer of

$$max_{q(\eta|y)} Z(X,T) \tag{7}$$

is deterministic, i.e. the optimal $q(\eta|y)$ satisfies $q(\eta|y) = 0$ or $q(\eta|y) = 1$ for all η and y. (see Corollary 8 in the Appendix).

3 The forward problem

In this section we show that when annealing (6), the solution of the normalized N-cut and approximate normalized 2-cut for an associated graph G are connected by a curve of solutions parameterized by β.

Given a joint distribution $p(X,Y)$ with X and Y finite discrete random variables we define the graph $G(V,E)$ in the following way. Each element $y \in Y$ corresponds to a vertex in V and the weight w_{kl} associated with the edge e_{kl} is

$$w_{kl} := \sum_x p(y_k|x)p(x,y_l). \tag{8}$$

The next Theorem is one of the key results of this paper. It connects explicitly the solution of a normalized N-cut problem to a solution of an annealing problem, see Figure 2.

Theorem 1 *Maximization of the function $Z(X,T)$ with $|T| = N$ over the variables $q(\mu|y)$ solves the maximal normalized association problem with N classes for the graph G, defined in (8).*

Proof. By definition, $p(x,\mu) = \sum_y q(\mu|y)p(x,y)$. By Corollary 8 at the maximum we have $q(\eta|y) = 0$ or $q(\eta|y) = 1$. We will write $y \in \mu$ whenever $q(\mu|y) = 1$. Then we have $p(x,\mu) = \sum_y q(\mu|y)p(x,y) = \sum_{y\in\mu} p(x,y)$. Let $\tilde{Z}(X,T) := 2\ln 2 Z(X,T)$. Then

$$\begin{aligned}
\tilde{Z}(X,T) &= -1 + \sum_\mu \sum_x \frac{1}{p(\mu)} \frac{p(x,\mu)p(x,\mu)}{p(x)} \\
&= -1 + \sum_\mu \frac{1}{p(\mu)} \sum_{y_k,y_l\in\mu} \sum_x p(y_k|x)p(x,y_l) \\
&= -1 + \sum_\mu \frac{\sum_{y_k,y_l\in\mu} w_{kl}}{p(\mu)}.
\end{aligned}$$

Before we continue our computation, we observe that a straightforward computation shows that the numbers w_{kl} have all the properties of the joint distribution on $Y \times Y$:

$$\sum_k w_{kl} = p(y_l), \sum_l w_{kl} = p(y_k), \sum_{k,l} w_{kl} = 1.$$

From this we get

$$p(\mu) = \sum_{y_k} q(\mu|y_k)p(y_k) = \sum_{y_k \in \mu} p(y_k) = \sum_{y_k \in \mu} \sum_l w_{kl}.$$

Using this expression for $p(\mu)$ we obtain

$$\begin{aligned}\tilde{Z}(X,T) &= -1 + \sum_\mu \frac{\sum_{y_k \in \mu, y_l \in \mu} w_{kl}}{\sum_{y_k \in \mu, y_l \in V} w_{kl}} \\ &= -1 + \sum_\mu \frac{links(\mu,\mu)}{links(\mu, Y)} \\ &= -1 + N \cdot Nassoc(\{\mu\}_{\mu=1}^N)\end{aligned}$$

It follows that the maxima of $\tilde{Z}(X,T)$, and therefore maxima of $Z(X,T)$, are maxima of $Nassoc(\{\mu\}_{\mu=1}^N)$. □

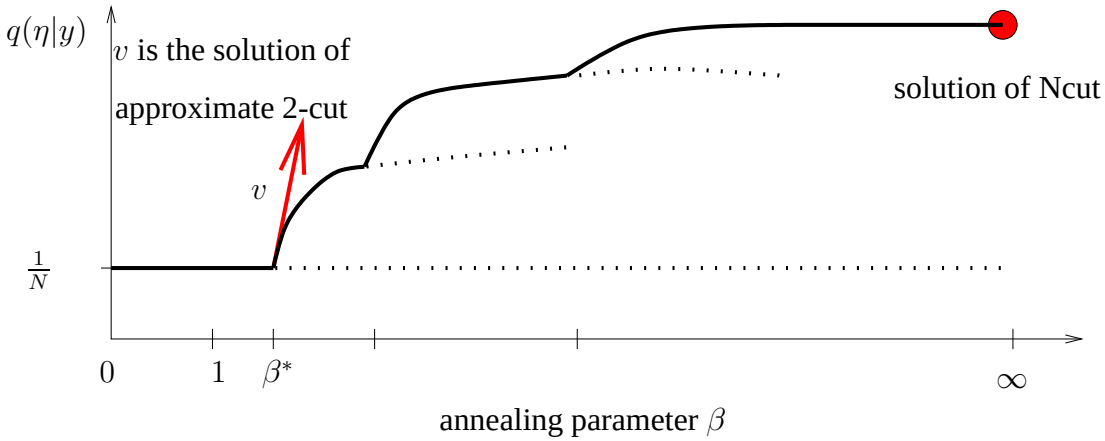

Fig. 2. The tangent vector v at the first phase transition computes approximate 2-cut, while the solution $q(\eta|y)$ at $\beta = \infty$ computes Ncut. The subsequent phase transitions are indicated by dotted lines along the primary branch of solutions.

Theorem 1 relates the solution of the N-cut problem to the solution of the annealing problem at $\beta = \infty$. We now show that the solution of the approximate 2-cut problem is associated to the first phase transition of the annealing procedure, see Figure 2.

The value $\beta = \beta^*$ where the phase transition occurs can be computed explicitly from Proposition 3 below. The eigenvector v that solves the relaxed bi-partition normalized cut problem is used as an initial seed of the annealing procedure at β^*. This result considerably speeds up the annealing procedure. Standard continuation algorithms [20,16] can then be used to trace this solution as β increases. Consecutive phase transitions then produce approximations of the normalized K-cut problem for all $2 < K \leq N$.

Theorem 2 *The eigenvector v associated with the phase transition at the solution ($q = 1/N$, $\beta^* = 1/\lambda^*$) induces an approximate normalized cut of the graph G. The value of β^* and v do not depend on the choice of the number of classes N.*

The key step in the proof is the following Proposition. The proof is in the Appendix.

Proposition 3 *The phase transition (bifurcation) from the solution $q(\nu|y) = \frac{1}{N}$ occurs at $\beta^* = \frac{1}{\lambda^*}$ in the direction v, where λ^* is the second eigenvalue of the matrix R with elements*

$$r_{lk} := \sum_x p(y_k|x)p(x|y_l) \tag{9}$$

and v is the corresponding eigenvector. The matrix R is independent of the choice of N.

Remark 4 Since

$$\sum_k r_{lk} = \sum_x p(x|y_l) \sum_k p(y_k|x) = 1$$

for every l and so R^T is a stochastic matrix. Thus its maximal eigenvalue is 1. As a consequence no bifurcation can occur for $\beta < 1$.

Proof of Theorem 2. We evaluate the matrices D and W in the approximate normalized 2-cut formulation (4) for a graph G. The matrix W is the matrix of weights of the graph G and therefore it consists of the elements w_{kl} defined in (8). Note that W is symmetric. The elements of the diagonal matrix D are given by

$$d(l) := \sum_k w_{kl} = \sum_k \sum_x p(y_k|x)p(x, y_l) = p(y_l)$$

for all l. The problem (4) turns into the problem of computing the second smallest eigenvalue μ_2 of

$$(D - W)y = \mu D y.$$

Multiplying the by matrix D^{-1} we arrive at the problem

$$(I - D^{-1}W)y = \mu y. \tag{10}$$

Since D^{-1} is the diagonal matrix with elements $1/p(y_l)$ it follows from (8) and (9) that

$$D^{-1}W = R.$$

A straightforward computation shows that (10) is equivalent to

$$(1-\mu)y = Ry. \tag{11}$$

Since R^T is a stochastic matrix (see Remark 4), it has the largest eigenvalue $\lambda = 1$. By (11) this corresponds to the value $\mu = 0$ in the approximate normalized 2-cut problem. The second largest eigenvalue, λ_2, of R corresponds to the second smallest eigenvalue, μ_2, of the approximate normalized 2-cut problem. Proposition 3 now finishes the proof of the Theorem.

4 The inverse problem

In the previous section we have shown that the solution of the annealing problem (6) at $\beta = \infty$ solves the normalized N-cut problem in a graph G for any predetermined number N of classes, while from the first phase transition we can recover a solution of the approximate normalized 2-cut of the same graph G. The edge weights of the graph G are determined by the joint probability distribution $p(X, Y)$ and the number of vertices of G is $|Y|$.

In this section we aim to solve the inverse problem. Given a graph G, we would like to determine X and the probability distribution $p(X, Y)$ such that the annealing problem will solve the normalized cut for the graph G. More precisely, given a graph $G = (Y, w_{ij})$ of vertices $y_1, \ldots, y_n$ and a symmetric set of edge weights w_{ij}, we seek to find a set X and a discrete probability distribution $p(X, Y)$ such that

$$w_{ij} = \sum_x p(y_j|x)p(x, y_i). \tag{12}$$

In other words we seek to split the matrix of weights as a product of a probability distribution and a conditional probability. If such a random variable X and a distribution $p(X, Y)$ exist, then we can apply the annealing procedure with the function $H + \beta Z$ (see (6)) to compute the normalized and the approximate normalized cuts of the given graph G.

We present a construction with $|X| = n^2$ and where we assume that $w_{ii} = 0$ for all i. This does not compromise the applicability of this method to real data since

usually self-similarities are not taken into account. We denote the elements of X by x_{kl}; there is one element for each edge in the graph G. Then set

$$p(x_{kl}, y_s) := 2w_{kl} \text{ for } s = k, l;$$
$$p(x_{kl}, y_s) := 0 \text{ for } s \neq k, l; \tag{13}$$

Then $p(x_{kl}) = p(x_{kl}, y_k) + p(x_{kl}, y_l) = 4w_{kl}$ for all k, l and

$$\sum_x p(y_k|x)p(x, y_l) = \sum_x \frac{p(x, y_k)p(x, y_l)}{p(x)}$$
$$= \frac{p(x_{kl}, y_k)p(x_{kl}, y_l)}{p(x_{kl})} = \frac{4w_{lk}^2}{4w_{lk}} = w_{lk}.$$

A legitimate question of performance of the annealing procedure arises when the joint probability $p(X, Y)$ has the size $n^2 \times n$. In our applications this has not been an issue, but certainly it would be desirable to find the set X with the least possible cardinality. Below we provide a condition under which construction of $p(X, Y)$ with size $n \times n$ is possible. When it is not possible to find a set X with $|X| = n$, we propose an approximation scheme.

Assume that (12) has a solution with $|X| = |Y| = n$ i.e. there exists a probability distribution p_0 on $X \times Y$ satisfying (12). Let P_0 be the $n \times n$ matrix representing the distribution p_0, such that $[P_0]_{ij} = p_0(x_i, y_j)$. Let Q_0 be a matrix of the conditional distribution $p_0(x_i|y_j)$. In the matrix form we have

$$Q_0 = P_0 D_0, \tag{14}$$

where D_0 is a diagonal matrix with the j-th diagonal element $1/p_0(y_j)$. Then (12) is equivalent to the following problem. Given $n \times n$ symmetric matrix P_1 find a matrix P_0 such that

$$P_1 = Q_0^T P_0. \tag{15}$$

As we will see in the next Lemma, this is similar to taking the square root of a matrix.

Lemma 5 *Given a $n \times n$ matrix of weights W for a graph G, let P be a scaled matrix W so that P is a $n \times n$ matrix of a probability distribution. Let Q be a conditional probability matrix corresponding to the matrix P. If Q is positive definite, then the problem (12) has (generically) a solution with $|X| = |Y|$.*

Proof. Observe that if $P_1 := P$ satisfies (15), the corresponding conditional probability Q_1 satisfies

$$Q_1 = P_1 D_1 = Q_0^T P_0 D_1,$$

where D_1, as in (14), is a diagonal matrix with the j-th diagonal element $1/p_1(y_j)$. Since P_1 satisfies (12) we have

$$p_1(y_i) = \sum_j \sum_x p_0(y_j|x) p_0(x, y_i) = p_0(y_i),$$

and thus $D_1 = D_0$. Therefore Q_1 is a square of Q_0

$$Q_1 = Q_0^T P_0 D_1 = Q_0^T P_0 D_0 = Q_0^T Q_0.$$

If $Q_1 = Q$ is positive definite as assumed, it has a square root Q_0 by Gantmacher [24] . Generically, the matrix Q_0 is non-singular. Then $P_0 = (Q_0^T)^{-1} P_1$ solves the problem (12). □

Example 6 *We show that for a general matrix P_1 it is not possible to find the matrix P_0 satisfying (15). This shows that it is not, in general, possible to find X with $|X| = |Y|$ such that (12) is satisfied. Take the matrix*

$$P_1 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \text{ and set } P_0 = \begin{bmatrix} p_{11} & p_{12} \\ p_{12} & p_{22} \end{bmatrix}.$$

Then the condition (12) for w_{12} and w_{21} reads $0 = \frac{p_{11}^2}{p_{11}+p_{12}} + \frac{p_{12}^2}{p_{12}+p_{22}}$ and $0 = \frac{p_{22}^2}{p_{11}+p_{12}} + \frac{p_{12}^2}{p_{12}+p_{22}}$, which implies that P_0 must be the zero matrix. Obviously, such a P_0 does not satisfy (12).

Finally, we present an approximate solution for the problem (15). We take

$$P_0 := P_1. \tag{16}$$

With this choice the annealing problem does not compute the normalized cut of a graph with weights P_1, but rather with weights $P_2 = Q_1^T P_1$ (see (15)) or, in other words, with weights w_{lk} given by (12) where $p(x, y)$ is given by P_1. This choice computes the normalized cut of an approximation of the given graph G. We discuss the performance of this approximation in the next section.

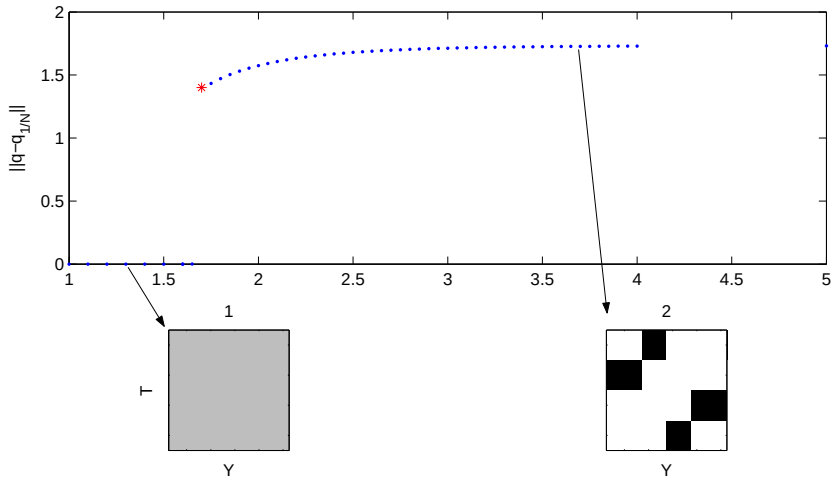

Fig. 3. Annealing on a random graph with 10 vertices and 4 subgraphs with $|X| = 100$ and $p(X, Y)$ given by (13). The horizontal axis is the annealing parameter β and the vertical axis is the norm of the difference of q and the uniform solution $q(\eta|y) = 1/4$. The bottom left panel shows the uniform solution for small β. The bottom right panel shows the deterministic solution which has clustered each subgraph successfully.

4.1 Illustrative examples

In Figure 3 and Figure 4 we present the results from annealing with two different distributions $p(X, Y)$ for a given graph G on 10 vertices.

The graph G has been chosen randomly in the following way. We divide vertices into 4 groups $V_1 = \{1, 2, 3\}, V_2 = \{4, 5\}, V_3 = \{5, 6\}, V_4 = \{7, 8, 9\}$. The weights within groups V_1 and V_4 are chosen uniformly in $[8, 12]$, within V_2 and V_3 uniformly in $[6, 10]$, between $V_1 \cup V_2$ and $V_3 \cup V_4$ uniformly in $[0, 4]$ and, finally, between V_1 and V_2 and between V_3 and V_4 uniformly in $[2, 6]$. To obtain the probability distribution P_1 these weights are re-scaled.

We denote the set of vertices by Y and we set the reproduction variable $|T| = N = 4$, which indicates that we want to split the graph into four subgraphs. In Figure 3 we present annealing results with the probability distribution $p(X, Y)$ described in (13) which is represented by a 100×10 matrix. In Figure 4 are results for the same graph G where we use the approximation (16) and thus $p(X, Y)$ is represented by a 10×10 matrix. The final clusters are indistinguishable. The time courses of the annealing procedures are different, however. The first phase transition happens around $\beta = 1.6$ in Figure 3 and around $\beta = 22$ in Figure 4. The subsequent phase transitions, that are well separated in β in Figure 4, are so close together in Figure 3, that our relatively coarse β step has not detected them. However, by results in [25] they must exist.

This behavior is not surprising. The main result of section 3 can be expressed in terms of equation (15) by saying "annealing with $p(x, y)$ given by P_0 computes

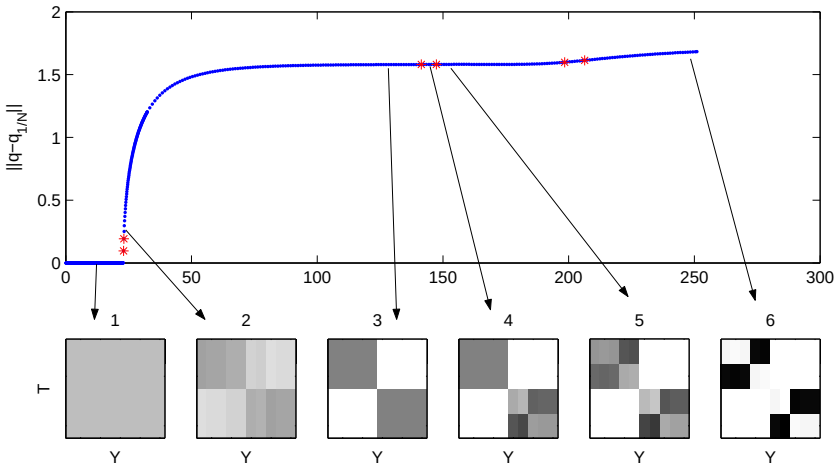

Fig. 4. Annealing on the same graph as in Figure 3 with $|X| = 10$ and $p(X, Y)$ given by (16). The bottom left panel shows the uniform solution. The bottom right panel shows the deterministic solution.

the normalized cut on a graph with weights P_1". The edge weights of a graph G on 10 vertices determine directly a 10×10 matrix of weights P_1. In Figure 3 we anneal with $p(X, Y)$ given by P_0 and thus are computing the normalized cut on the graph G, while in Figure 4 we anneal with $p(X, Y)$ given by P_1 and thus the normalized cut of a graph with weights $P_2 := Q_1^T P_1$. Intuitively, the weights P_2 are more homogeneous than those of P_1. Indeed, the condition (12) gives elements w_{ij}^2 of P_2 in terms of the elements w_{ij}^1 of P_1 as

$$w_{ij}^2 := \sum_k \frac{p(y_j, y_k)p(y_k, y_i)}{p(y_k)} = \sum_k \frac{w_{jk}^1 w_{ki}^1}{w_k^1}$$

where $w_k^1 := \sum_i w_{ik}$. Therefore the weight w_{ij}^2 of the $i \to j$ edge is proportional to a product of weights along any path with two edges joining i and j in P_1. It follows that the embedded clusters in the graph with P_2 weights will be less prominent than in the graph with P_1 weights and thus it will take longer for the annealing procedure to resolve them. This explains why the first phase transition occurs later in Figure 4.

The panels on the bottom of the Figures 3 and 4 indicate the behavior of the probabilistic assignment $q(\eta|y)$ as a function of β. The vertical axis has four values indicating the assignment to different classes, the horizontal axis has 10 values indicating vertices of the graph. Dark color indicates high probability, light color low probability. For small values of β (high temperature) all vertices belong to all four groups with probability $1/4$ (lower left panel of Figure 4). As β increases, vertices split into two groups (second panel of Figure 4); vertices in $V_1 \cup V_2$ belong with high probability to classes 1 and 4; vertices in $V_3 \cup V_4$ belong with high probability to classes 2 and 3. Following two additional phase transitions, one arrives at the last panel, where every planted group V_i has probability close to 1 to belong to a distinct class. Separation into four subgraphs has been completed.

5 Application to image segmentation

We have applied our approach to an image segmentation problem. The approximate Ncut problem is closely related to spectral clustering, since it is computed by the second eigenvector of (4) and the matrix $(D-W)$ is the graph Laplacian. Therefore it inherits all the advantages and limitations of applying spectral clustering to image segmentation [4,8]. The main advantage is ease of computation; we briefly discuss some limitations below.

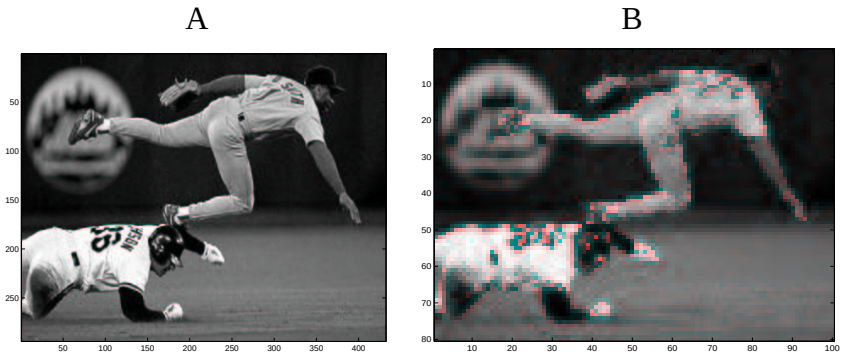

Fig. 5. The image segmentation problem. A. The original, B. subsampled 80×100 image.

To compare our approach with spectral clustering employed by Shi and Malik [6], we segmented the same image, see Figure 5.A. Since this image has 135000 pixels, we follow Shi and Malik and sub-sampled the image to get a 80×100 image in (Figure 5.B). To segment the image we construct a graph $G = (V, E)$ by taking each pixel as a node and define the edge weight w_{ij} between node i and j as a the product of a feature similarity and a spatial proximity term:

$$w_{ij} == e^{-\left(\frac{I(i)-I(j)}{\sigma_I}\right)^2} \times H(e^{-\left(\frac{X(i)-X(j)}{\sigma_X}\right)^2}),$$

where $I(j) = \frac{\iota(i)}{255}$ is the normalized intensity, while $\iota(i) \in \{0, 1, \ldots, 255\}$ is the raw intensity of pixel i. The function H is the identity when $|X(i) - X(j)| < 5$ and zero otherwise and is used to favor close spatial proximity of pixels. As in Shi and Malik's paper, we take $\sigma_I = .1$ and $\sigma_X = 4$. In agreement with (16) we view the symmetric matrix of weights $P = [w_{ij}]$, after approximate scaling, as a joint probability distribution $p(X, Y)$ with discrete random variables $|X| = |Y| = 80 \times 100 = 8000$, the number of pixels in the image. With this joint probability we performed annealing with the function (6) by incrementing the value of β. We have done this in two ways. The first way, which we call a (2×2) clustering, we choose initially to cluster into two classes (the reproduction variable $|T| = 2$), see Figure 6.

After annealing to $\beta = 1.5$ at which point the classes are well separated, we then took each class separately and split it again to two classes, see Figure 7. This proce-

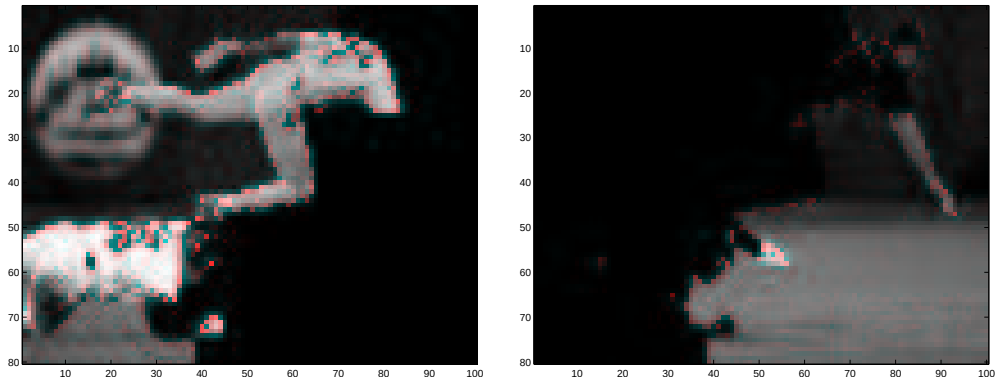

Fig. 6. The segmentation induced by the best 2-cut at $\beta = 1.5$.

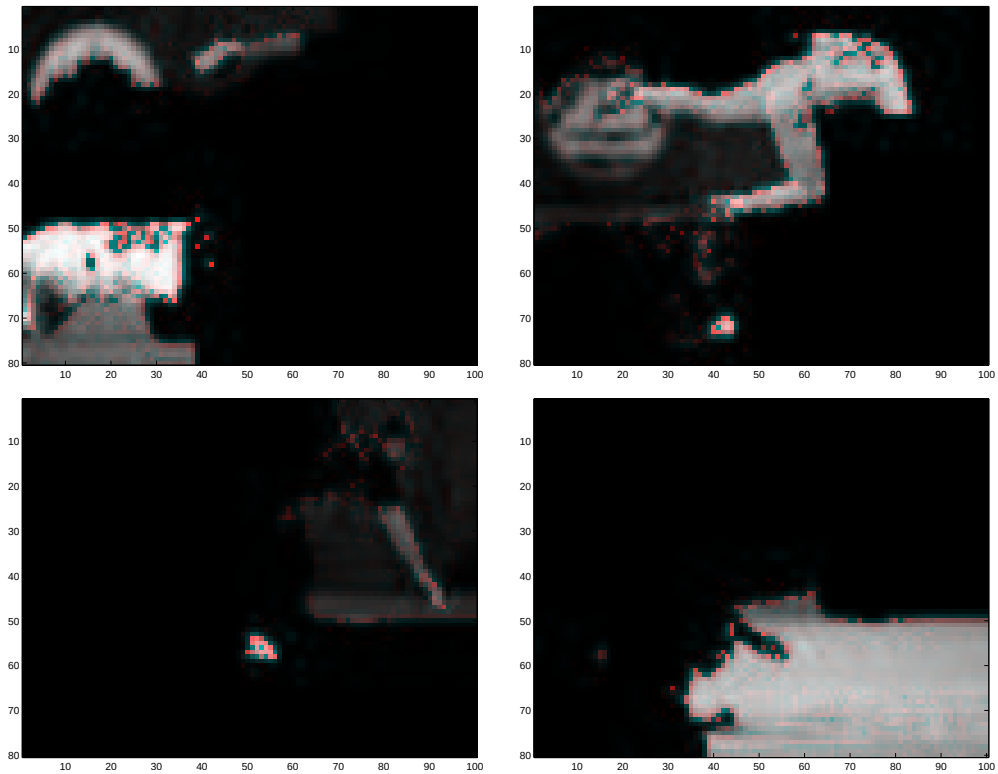

Fig. 7. The segmentation induced by the best (2×2) cut at $\beta = 1.5$.

dure most closely resembles the recursive cut of Shi and Malik [6]. They performed two additional optimization steps after computing the second smallest eigenvector.

First, they optimize the threshold which splits eigenvector values into two groups to compute maximal Ncut. Secondly, they ignore all eigenvectors that are varying continuously and thus would produce an unstable cut. Presumably, if they chose to ignore the eigenvector corresponding to the second smallest eigenvalue, they take the eigenvector corresponding to the third eigenvalue to make the cut. In their paper, this procedure is called a stability analysis.

The advantage of our method is that we can avoid recursive clustering and directly

cluster into four classes. We set the number of classes to 4 (that is, the reproduction variable $|T| = 4$) and again perform the annealing with the function (6). The results are in Figure 8.

Before we compare the results for (2×2) and 4-way clustering, we mention the inherent limitations of the spectral clustering method for the image segmentation problem. Since images are two dimensional, the graphs obtained by equating pixels with nodes and selecting weights that respect the spatial proximity, have a stereotypical structure. As a consequence, the lowest cut may be associated more with the shortest path joining edges of the image, rather than with the split between two image segments. Eigenvalues of the graph Laplacian [9] will reflect such cuts. For a review of these issues as well as a proposed solution, see [8]. Furthermore, for the images that are difficult to segment (Figure 5.B) the eigenvalues of the Laplacian will be clustered around the second smallest eigenevalue and some of the eignevectors will be "unstable" in the language of Shi and Malik, that is, smoothly varying. This problem was noticed by [4,8] and they advocate the use of a subspace corresponding to the set of eigenvectors for further processing the image. The ambiguity in selection of the correctly segmenting eigenvector is reflected in our computation as well. As we have shown in the first part of the paper, the bifurcating branches coming out of $q = 1/2$ (in (2×2) annealing) and $q = 1/4$ (in 4-way annealing) start in the direction of the eigenvector of the graph Laplacian. Since there are many of these eigenvectors closely spaced together, there are many branches bifurcating closely together. Since the continuation algorithm introduces a small error, we may land on a different branch when we repeat the annealing with the same initial value.

We have run the algorithm repeatedly and explore the different branches of solutions all the way to $\beta = 1.5$. We then selected the best segmentation based on the value of the optimized function (6). For the 4-way cut the value of the function (6) on the best branch is 3.223 and its Ncut value is 0.0079, while for the 2×2 cut the value of (6) is 2.9896 and the Ncut value is 0.0093. In fact, we have found 20 slightly different 4-cuts which correspond to different branches of solutions to 6) and the range of the values of the cost function at $\beta = 1.5$ was $[3.1283, 3.223]$ while for the 2×2 cut the range of the function values among 16 branches was $[2.9622, 2.9896]$. So even the worst 4-cut was better than the best 2×2 cut. Also notice that the Ncut value 0.0079 of the best 4-way cut is substantially lower then Ncut value 0.0093 of the best 2×2 cut. The reported Ncut value of 0.04 for the 7-way cut using recursive 2-way partitioning by Shi and Malik [6] is an order of magnitude larger. However, since the value of a 7-cut is expected to be larger then that for the 4-cut and we may have used different subsampling algorithms to obtain Figure 5.B, the direct quantitative comparison between these two results may not be appropriate.

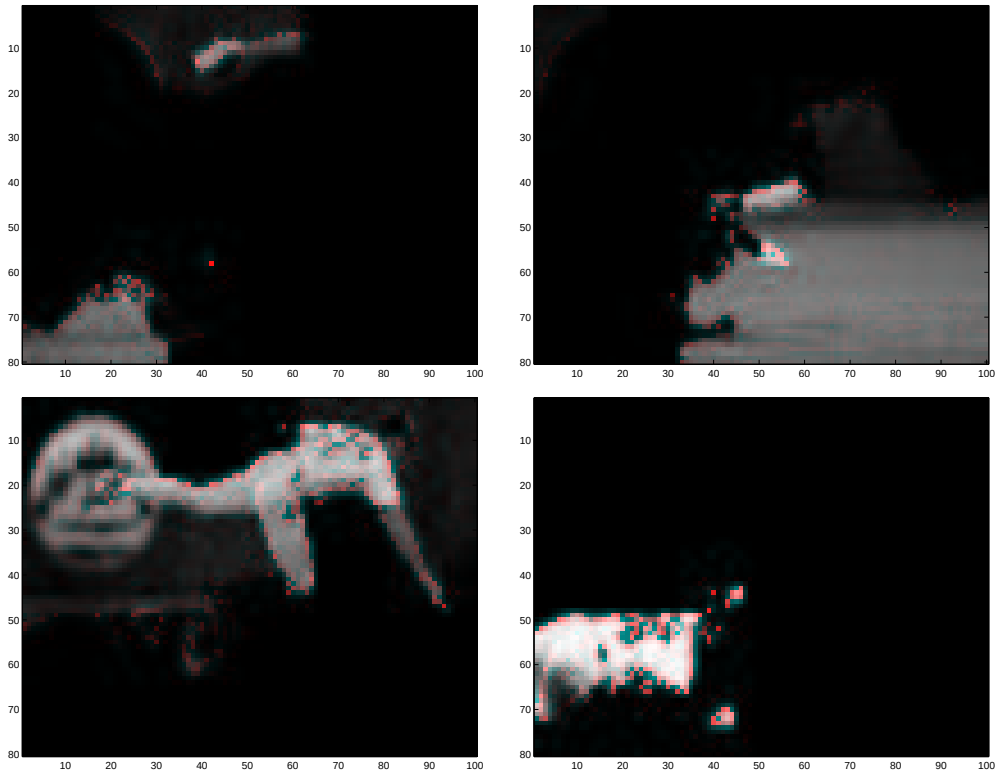

Fig. 8. The segmentation induced by the best 4-way cut solution at $\beta = 1.5$

6 Application to data from a sensory system

A fundamental problem in neurobiology is to understand how ensembles of nerve cells encode information. The inherent complexity of this problem can be significantly reduced by restricting analysis to the sensory periphery, where it is possible to directly control the input to the neural system. Complexity is further reduced by focusing on relatively simple systems such as invertebrate sensory systems, where a large proportion of the information available to the organism is transmitted by single neurons. We will try to discover such a sensory system's encoding scheme, which we define as a correspondence between the sensory input the neuron receives and the (set of) spike train (s) it generates in response to this input. We note that this correspondence is probabilistic in nature; repeated presentations of a single stimulus input will elicit variable neural outputs, and a single neural output can be associated with a range of different stimuli.

We have analyzed data from the cricket cercal system, from the interneuron designated IN 9-2a in the terminal abdominal ganglion [26]. The neuron was stimulated by a band-limited white noise stimulus, which was presented to the cricket through a custom air-flow chamber [14]. The response of the neuron was recorded intra-cellularly. From the 10 minute continuous recording a set of responses was collected by selecting all inter-spike intervals of 50 ms or less in duration. We enforced a silent prefix to the responses, such that no spike occurred in the 20 ms

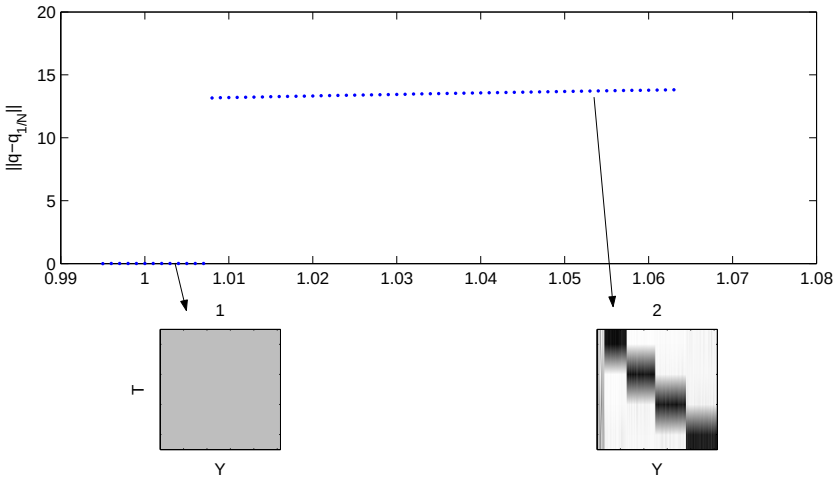

Fig. 9. Annealing on a similarity graph of patterns from the cercal sensory system of a cricket.

preceding the start of a response. The corresponding set of stimuli were formed by taking an 80 ms long interval starting 40 ms before the first spike in the response event. Therefore the data form a collection of pairs of intervals, one from the stimulus set and one from the response set. Since the sampling frequency of the input is 10 points per millisecond (10 kHz) and there are two spatial dimensions for the stimulus [14], the stimulus set lies in 1600 dimensional space. The response set lies in 800 dimensional space, and is parameterized by a single parameter which is the inter-spike interval length. We view each data point as a pair of stimulus and its corresponding response. We want to cluster these points to discover the "codewords"; that is, consistent classes in the stimulus-response space.

We employed a method of random projections, described in [27], to compute a similarity matrix between the data points. This yields a weighted graph G where each vertex represents a data point (x, y) with x a stimulus and y the corresponding response. The weight associated to the edge connecting vertices (x_1, y_1) and (x_2, y_2) is computed as the frequency of these two data points being projected to the same cluster under a series of random projections. A rescaled collection of these weights forms a matrix P_1 that has been described in section 4. We chose the approximation (16) to solve the inverse problem and set $X := Y$ and $p(X, Y) := P_1$.

We then applied our annealing algorithm to this joint probability. The results are shown in Figure 9 and Figure 10. As in Figure 3 the phase transitions from one to two, and two to three classes are so close together in the parameter β that we could not find them numerically. Since by Remark 4 the first phase transition cannot occur for $\beta < 1$, in practical applications one may have to anneal only for a very small range of β, see Figure 9. In Figure 10 we graph the projection of clusters to the response set Y. The first spike in the pattern always happens at $t = 0$. The second spike is colored according to the cluster it belongs to. This clustering has clear biological interpretation: inter-spike interval length is the important feature of the

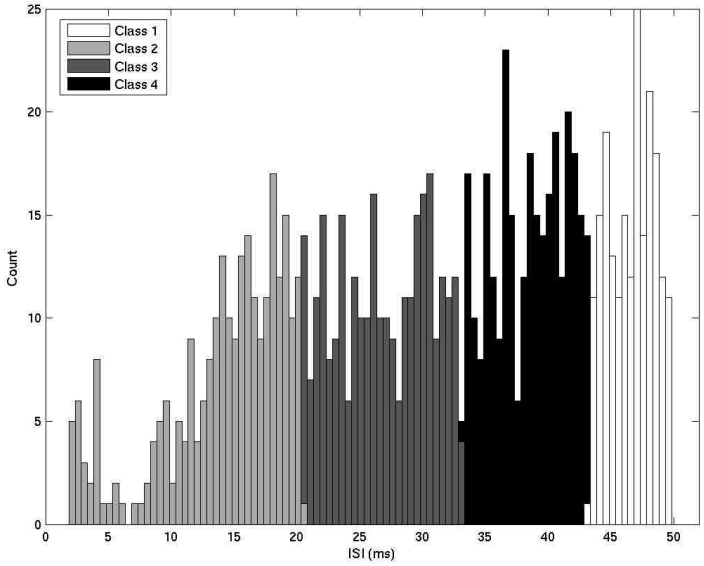

Fig. 10. Projection of the clusters to the response space and graphed according to the inter-spike intervals. The first spike of the response codeword always happens at time 0, and the second spike of the response is colored according to the class the inter-spike interval belongs to. The vertical axis is the frequency and the horizontal axis is the inter-spike interval length.

output set.

7 Conclusions

In this paper we show the seemingly very different approaches to clustering, graph-theoretical and information-theoretical, are connected. We have shown that there is an information-like cost function whose solution as $\beta \to \infty$ solves normalized cut problem of an associated graph, and the solution at the first phase transition solves the approximate (relaxed) normalized 2-cut of the same graph. Subsequent phase transitions then separate approximate solutions to the normalized K-cut problem with $2 < K \leq N$. The first phase transition does not depend on the choice of N.

Based on these results we propose an algorithm that, starting with an arbitrary graph G with n vertices, controls the quality and the number of computed clusters K in G for $2 \leq K \leq N$. The first step in this process is the construction of the random variables X, Y and the joint probability $p(X, Y)$. We provide a general algorithm which computes $p(X, Y)$ with $|X| = n^2$ and $|Y| = n$ from the edge weights of the graph G. Although we show that the appropriate $p(X, Y)$ with $|X| = n$ and $|Y| = n$ does not always exist, we provide a sufficient condition for its existence.

We also use an approximation with $|X| = n$ and discuss its performance on several examples. This approximation computes a normalized N-cut for a closely related graph G' whose weights are "smoothed out" versions of the weights of G.

We tested our algorithm on an image segmentation problem and obtained results that compare favorably with those in Shi and Malik [6]. We also applied the algorithm to a problem of clustering high dimensional data from the sensory system of a cricket.

8 Appendix

Lemma 7 1. $Z(X,T) \geq 0$.

2. *The function* $Z(X,T)$ *is a convex function of* $q(\eta|y)$.

Proof: To prove the first statement, we observe that $x - 1 \geq \log x$. Therefore from (5) and the definition of mutual information ([28]) we get

$$Z(X,Y) \geq I(X,Y) \geq 0.$$

We will indicate the main steps in the proof of convexity of $Z(X,T)$, since the proof follows closely the convexity argument for mutual information $I(X,T)$ in [29]. Since the function $f(t) = t(t-1)$ is strictly convex, one can show using Jensen's inequality [28] that

$$\sum_{i=1}^{n} a_i(\frac{a_i}{b_i} - 1) \geq (\sum_{i=1}^{n} a_i)(\frac{\sum_{i=1}^{n} a_i}{\sum_{i=1}^{n} b_i} - 1) \tag{17}$$

non-negative numbers, $a_1, a_2, ..., a_n$ and $b_1, b_2, ..., b_n$; with equality if and only if $\frac{a_i}{b_i}$ =constant. In analogy to Kullback-Leibler distance [28] we set

$$Z(p||q) = \sum_x p(x)(\frac{p(x)}{q(x)} - 1).$$

Then our function $Z(X,T)$ can be written as $Z(X,T) = \frac{1}{2\ln 2} Z(p(x,\mu)||p(x)p(\mu))$. Applying (17) one can show that the function $Z(p||q)$ is convex in the pair (p,q), i.e., if (p_1,q_1) and (p_2,q_2) are two pairs of probability mass functions, then for some real λ,

$$\begin{aligned} Z(\lambda p_1 + (1-\lambda)p_2 || \lambda q_1 + (1-\lambda)q_2) &\leq \lambda Z(p_1||q_1) \\ &+ (1-\lambda) Z(p_2||q_2). \end{aligned} \tag{18}$$

The last step is to show that the $Z(X,T)$ is a convex function of $q(\eta|y)$. Fix $p(x)$ and consider two different conditional distributions $q_1(\mu|x)$ and $q_2(\mu|x)$. The corresponding joint distributions are $p_1(x,\mu) = p(x)q_1(\mu|x)$ and $p_2(x,\mu) = p(x)q_2(\mu|x)$, and their respective marginals are $p(x), p_1(\mu)$ and $p(x), p_2(\mu)$. Consider a conditional distribution

$$q_\lambda(\mu|x) = \lambda q_1(\mu|x) + (1-\lambda)q_2(\mu|x)$$

that is a mixture of $q_1(\mu|x)$ and $q_2(\mu|x)$. Then the corresponding joint distribution $p_\lambda(x,\mu) = \lambda p_1(x,\mu) + (1-\lambda)p_2(x,\mu)$, and the marginal distribution of Y, $p_\lambda = \lambda p_1(\mu)+(1-\lambda)p_2(\mu)$ are both corresponding mixtures. Finally, if we let $q_\lambda(x,\mu) = p(x)p_\lambda(\mu)$ we have $q_\lambda(x,\mu) = \lambda q_1(x,\mu)+(1-\lambda)q_2(x,\mu)$ as well. Since $Z(X;T) = \frac{1}{2\log 2}Z(p_\lambda(x,\mu)||q_\lambda(x,\mu))$ and the function $Z(p_\lambda||q_\lambda)$ is convex (see 18), it follows that the function $Z(X;T)$ is convex function of $q(\eta|y)$ for a fixed $p(x)$. □

Corollary 8 *For a generic probability distribution $p(X,Y)$ the maximizer of*

$$max_{q(\eta|y)} Z(X,T)$$

is deterministic, i.e. the optimal $q(\eta|y)$ satisfies $q(\eta|y) = 0$ or $q(\eta|y) = 1$ for all η and y.

Proof. By Theorem 4 of [15], this is a consequence of the convexity of Z. □

Proof of Proposition 3. Let

$$F(q,\beta) := H(T|Y) + \beta Z(X,T).$$

Recall that the vector of conditional probabilities $q = q(t|y)$ satisfies

$$\sum_\eta q(\eta|y) = 1 \quad \text{for all } y. \tag{19}$$

These equations form an equality constraint on the maximization problem (6) giving the Lagrangian

$$\mathcal{L}(q,\xi,\beta) = F(q,\beta) + \sum_{k=1}^{|Y|} \xi_k \left(\sum_{\mu=1}^{N} q(\mu|y_k) - 1 \right), \tag{20}$$

which incorporates the vector of Lagrange multipliers ξ, imposed by the equality constraints (19).

Maxima of (6) are critical points of the Lagrangian i.e. points q where the gradient of (20) is zero. We now switch our search from maxima to critical points of the

Lagrangian. We reformulate the optimization problem (6) as a system of differential equations under a gradient flow,

$$\begin{pmatrix} \dot{q} \\ \dot{\xi} \end{pmatrix} = \nabla_{q,\xi} \mathcal{L}(q, \xi, \beta). \tag{21}$$

The critical points of the Lagrangian are the equilibria of (21) since those are places where the gradient of $\mathcal{L}$ is equal to zero. The maxima of (6) correspond to those equilibria for which the Hessian ΔF, is negative definite on the kernel of the Jacobian of the constraints [30,25].

As β increases from 0, the solution $q(\eta|y)$ is initially a maximum of (6). We are interested in the smallest value of β, say $\beta = \beta^*$, where $q(\eta|y)$ ceases to be a maximum. This corresponds to a change in the number of critical points in the neighborhood of $q(\eta|y)$ as β passes through $\beta = \beta^*$. The necessary condition for such a phase transition (bifurcation) is that some eigenvalue of the linearization of the flow at an equilibrium crosses the imaginary axis [31]. Therefore we need to consider eigenvalues of the $(N|Y|+|Y|) \times (N|Y|+|Y|)$ Hessian $\Delta\mathcal{L}$. Since $\Delta\mathcal{L}$ is a symmetric matrix, a bifurcation can only be caused by a real eigenvalue crossing the imaginary axis, and therefore we must find the values of (q, β) at which $\Delta\mathcal{L}$ is singular.

The form of $\Delta\mathcal{L}$ is simple:

$$\Delta\mathcal{L} = \begin{bmatrix} B_1 & 0 & \dots & I \\ 0 & B_2 & \dots & I \\ \vdots & \vdots & \vdots & \vdots \\ 0 & \dots & B_N & I \\ I & I & \dots & 0 \end{bmatrix},$$

where I is the identity matrix and B_i is

$$B_i := \frac{\partial^2 \mathcal{L}}{\partial q(\mu_i|y_k) \partial q(\mu_i|y_l)} = \frac{\partial^2 F}{\partial q(\mu_i|y_k) \partial q(\mu_i|y_l)}.$$

The block diagonal matrix consisting of all matrices B_i represents the second derivative matrix (Hessian) of F.

It is shown in [25] that, generically, there are two types of bifurcations: the saddle-node in which two equilibria appear simultaneously, and the pitchfork-like bifurcations, where new equilibria emanate from an existing equilibrium. Furthermore,

the first kind of bifurcation corresponds to a value of β and q where $\Delta\mathcal{L}$ is singular, but ΔF is non-singular; the second kind of bifurcation happens at β and q where both $\Delta\mathcal{L}$ and ΔF are singular. Since at the bifurcation off of $q(\eta|y) = 1/N$ a new branch emanates from an existing branch, we need only investigate when the eigenvalues of the smaller Hessian ΔF are zero. We solve the system

$$\Delta F\mathbf{w} = (\Delta H(T|Y) + \beta\Delta Z(X,T))\mathbf{w} = 0 \tag{22}$$

for any nontrivial vector $\mathbf{w}$. We rewrite (22) as an eigenvalue problem

$$(-\Delta H(T|Y))^{-1}\Delta Z(X,T)\mathbf{w} = \frac{1}{\beta}\mathbf{w}. \tag{23}$$

Since this matrix ΔF is block diagonal with blocks B_i, $i = 1, \ldots, N$ and by symmetry [25] at $q(\eta|y)$ all the blocks B_i are identical, we will from now on only compute with one diagonal block $B := B_i$.

Lemma 9 *Let $Z(X,T)$ be defined as in (5) with $|T| = N$. Then the one diagonal block of ΔZ, evaluated at $q(\eta|y) = 1/N$, is*

$$\Delta Z(X,T) = \frac{N}{\ln 2}\sum_x \frac{p(x,y_k)p(x,y_l)}{p(x)} - p(y_k)p(y_l)$$

The Hessian of $H(T|Y)$ at $q(\eta|y)$ is

$$\Delta H(T|Y) = -\frac{Np(y_k)}{\ln 2}.$$

Proof. Straightforward calculation shows that the (ν,k) element $(\nabla Z)_{\nu k}$ of the gradient of Z is

$$\frac{1}{2\ln 2}\sum_x (\frac{2p(x,y_k)p(x,\nu)}{p(x)p(\nu)} - \frac{p(x,\nu)^2 p(y_k)}{p(\nu)^2 p(x)} - p(x,y_k)).$$

Differentiating such an element with respect to $q(\mu|y_l)$ yields a $(\mu,l),(\nu,k)$ element of the second derivative matrix $\frac{\partial^2 Z}{\partial q(\mu|y_l)q(\nu|y_k)}$

$$\frac{1}{2\ln 2} \quad \sum_x 2\delta_{\nu\eta}(\frac{p(x,y_k)p(x,y_l)}{p(x)p(\nu)} - \frac{2p(x,\nu)(p(x,y_k)p(y_l)}{p(x)p(\nu)^2} - \frac{p(x,\nu)p(x,y_l)p(y_k)}{p(x)p(\nu)^2} + \frac{2p(x,\nu)^2 p(y_l)p(y_k)}{p(\nu)^3 p(x)})$$

We evaluate expressions $p(x,\nu)$ and $p(\nu)$ at $q(\nu|y) = \frac{1}{N}$ to get $p(x,\nu) = \frac{1}{N}p(x)$ and $p(\nu) = \frac{1}{N}$. Therefore at $q(\mu|y) = \frac{1}{N}$ we have

$$\Delta Z(X,T) = \frac{N}{\ln 2}\delta_{\nu\eta}(\sum_x \frac{p(x,y_k)p(x,y_l)}{p(x)}) - p(y_k)p(y_l), \tag{24}$$

where $\delta_{\nu\eta} = 1$ if and only if $\nu = \eta$. For the computation of $\Delta H(T|Y)$ see ([32]).

□

Since $\Delta H(T|Y)$ is diagonal, we can explicitly compute its inverse as well as the diagonal block U of the matrix

$$(-\Delta H(T|Y))^{-1}\Delta Z(X,T).$$

Using Lemma 9 we get that the $(l,k)^{th}$ element of U at $q(\eta|y) = 1/N$ is

$$\begin{aligned} u_{lk} &:= \sum_i \frac{p(x_i,y_k)p(x_i,y_l)}{p(x_i)p(y_l)} - p(y_k) \\ &= \sum_i p(y_k|x_i)p(x_i|y_l) - p(y_k). \end{aligned}$$

We observe that the matrix B can be written as $B = R - A$, where the $(l,k)^{th}$ element of R is

$$r_{lk} := \sum_i p(y_k|x_i)p(x_i|y_l), \tag{25}$$

and $a_{lk} := p(y_k)$. Therefore the problem (23) becomes

$$(R - A)\mathbf{w} = \lambda\mathbf{w}. \tag{26}$$

Let $\mathbf{1}$ be a vector of ones in $\mathbf{R}^N$. We observe that

$$A\mathbf{1} = \mathbf{1}$$

and the l-th component of $R\mathbf{1}$

$$\begin{aligned} [R\mathbf{1}]_l &= \sum_k \sum_i p(y_k|x_i)p(x_i|y_l) \\ &= \sum_i p(x_i|y_l) \sum_k p(y_k|x_i) \\ &= \sum_i p(x_i|y_l) \\ &= 1. \end{aligned}$$

Therefore we obtain one particular eigenvalue-eigenvector pair $(0, \mathbf{1})$ of the eigenvalue problem (26)

$$(R - A)\mathbf{1} = 0$$

Since the eigenvalue λ corresponds to $1/\beta$, this solution indicates bifurcation at $\beta = \infty$. We are interested in finite values of β.

Lemma 10 *Let $1 = \lambda_1 \geq \lambda_2 \geq \lambda_3 \ldots \lambda_{|Y|}$ be eigenvalues of a block of the matrix R. Then the solution $q(\eta|y)$ ceases to be a maximum at $\beta = \frac{1}{\lambda_2}$. The corresponding eigenvector to λ_2 (and all λ_k for $k \geq 2$) is perpendicular to the vector $p := (p(y_1), p(y_2), \ldots, p(y_n))^T$.*

Proof. We note first that the range of the matrix A is the linear space consisting of all multiples of the vector $\mathbf{1}$ and the kernel is the linear space

$$W := \{\mathbf{w} \in \mathbf{R}^N \mid \langle \mathbf{p}, \mathbf{w} \rangle = 0\},$$

where $\mathbf{p} = (p(y_1), \ldots, p(y_n))$ and $\langle \cdot, \cdot \rangle$ denotes the dot product.

We now check that the space W is invariant under the matrix R which means that $RW \subset W$. It will then follow that all eigenvectors of $R - A$, apart from $\mathbf{1}$, belong to W and are actually eigenvectors of R. So assume $\mathbf{w} = (w_1, \ldots, w_N) \in W$, which means

$$\sum_k w_k p(y_k) = 0.$$

We compute the l-th element $[R\mathbf{w}]_l$ of the vector $R\mathbf{w}$

$$[R\mathbf{w}]_l = \sum_k \sum_i p(y_k|x_i) p(x_i|y_l) w_k.$$

The vector $R\mathbf{w}$ belongs to the space W if, and only if, its dot product with $\mathbf{p}$ is zero. We compute the dot product

$$\begin{aligned} R\mathbf{w} \cdot \mathbf{p} &= \sum_{l,i,k} p(y_k|x_i) p(x_i|y_l) w_k p(y_l) \\ &= \sum_{i,k} p(y_k|x_i) w_k \sum_l p(x_i|y_l) p(y_l) \\ &= \sum_k w_k \sum_i p(y_k|x_i) p(x_i) \\ &= \sum_k w_k p(y_k) \end{aligned}$$

and the last expression is zero, since $w \in W$.

This shows that all other eigenvectors of $R - A$, apart from $\mathbf{1}$, belong to W and are eigenvectors of R. Since bifurcation values of β are reciprocals of the eigenvalues λ_i, the result follows. □

Acknowledgements

The work of T. G. was partially supported by NSF-BITS grant 0129895, NIH-NCRR grant PR16445, NSF/NIH grant W0467 and NSF-CRCNS grant W0577. The work of C.C. was partially supported by the Summer Undergraduate Research Program sponsored by IGERT grant NSF-DGE 9972824 and the Undergraduate Scholars Program at MSU-Bozeman. We would like to thank Aditi Baker for providing us with the similarity matrix used in section 5.1 and John P. Miller for his support of this project.

References

[1] B. Everitt, *Cluster Analysis*, Oxford University Press 1993.

[2] B. Mirkin, *Mathematical Classification and Clustering*, Kluwer Academic Publishers, 1996.

[3] Z. Wu and R. Leahy, An optimal graph theoretic approach to data clustering: Theory and its applications to image segmentation, *IEEE Trans. on Pattern Analysis and Machine Intelligence* 15(11), (1993) 1101-1113.

[4] A. Y. Ng, M. Jordan and Y. Weiss, On spectral clustering: Analysis and an algorithm, *Advances in Neural Information Processing Systems*, MIT Press, vol. 14, 2002.

[5] Y. Weiss, Segmentation using eigenvectors: a unifying view, *International Conference on Computer Vision: 975-982* 1999.

[6] J. Shi and J. Malik, Normalized cuts and image segmentation, *IEEE Trans. Pattern Analysis and Machine Intel.*, 22(8), (2000) 888-905.

[7] S. X. Yu and J. Shi, Multiclass spectral clustering, *International conference on Computer Vision* 2003, 11- 17.

[8] D. Tolliver and G. L Miller, Graph partitioning by spectral rounding: applications to image segmentation and clustering, pp. 1053-1060, 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 1 (CVPR'06), 2006.

[9] F. R. K. Chung, *Spectral Graph Theory*, Providence, RI: Amer. Math. Soc., 1997.

[10] C. J. Alpert and A. B. Kahng, Multiway partitioning via geometric embeddings, orderings and dynamic programming, *IEEE Transactions on Computer-aided Design of Integrated Circuits and Systems*, 14(11), (1995)1342-58.

[11] R. Durbin, R. Szeliski and A. Yuille, An analysis of the elastic net approach to the travelling salesman problem, *Neural Computation* 1 (3),(1989), 348-358.

[12] K. Rose, Deterministic Annealing for clustering, compression, classification, regression, and related optimization problems, *Proc. IEEE* 86(11), (1998) 2210-2239.

[13] N. Tishby, F. Pereira and W. Bialek, The Information Bottleneck Method, *Proceedings of The 37th annual Allerton conference on communication, control and computing*,University of Illinios, 1999.

[14] A. Dimitrov, J. Miller, T. Gedeon, Z. Aldworth and A. Parker, Analysis of neural coding using quantization with information-based distortion function, *Network* 14, (2003) 369-383.

[15] T. Gedeon, A. Parker and A. Dimitrov, Information distortion and neural coding, *Canadian Applied Math. Q.* 10-1, (2003),33-69.

[16] A. Parker, T. Gedeon and A. Dimitrov, Annealing and the rate distortion problem, *Advances in Neural Information Processing Systems*, MIT Press, vol. 15, 2003.

[17] N. Slonim, G. S. Atwal, G. Tkačik and W. Bialek, Information based clustering, *PNAS* in press.

[18] H. Greenspan, J. Goldberger and S. Gordon, Unsupervised clustering using the Information Bottleneck method. *DAGM*, 2002.

[19] E. Schneidman, N. Slonim, N. Tishby, R. de Ruyter van Steveninck and W. Bialek, Analyzing neural codes using the information bottleneck method, *Advances in Neural Information Processing Systems*, vol. 15, 2003.

[20] W. J. Beyn, A. Champneys, E. Doedel, W. Govaerts, Y. A. Kuznetsov and B. Sandstede. Numerical continuation and computation of normal forms. In Handbook of Dynamical Systems III, 1999.

[21] N. Slonim and N. Tishby, Agglomerative Information Bottleneck, *Advances in Neural Information Processing Systems*, vol. 12 1999.

[22] N. Slonim, N. Friedman, and N. Tishby, Agglomerative multivariate information bottleneck, (2001). *In advances in Neural Information Processing Systems*, vol. 14 2001.

[23] E. Sharon, A. Brandt and R. Basri, Fast multiscale image segmentation, *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), South Carolina*, Vol. I,(2000) 70-77.

[24] F. G. Gantmacher, *The Theory of Matrices*, Chelsea Pub. Co., 2nd edition 1990.

[25] A. Parker and T. Gedeon, Bifurcation structure of a class of S_N-invariant constrained optimization problems, *J. Dynamics and Diff. Eq.*, 16(3), (2004) 629-678.

[26] G. A. Jacobs and R. K. Murphey, Segmental origins of the cricket giant interneuron system, *J. Comp. Neurol.* 265,(1987)145-157.

[27] B. Mumey, A. Sarkar, T. Gedeon, A. Dimitrov and J. Miller, Finding neural codes using random projections, *Neurocomputing* 58-60,(2004)19-25.

[28] T. Cover and J. Thomas, *Elements of Information Theory*, Wiley Series in Communication, New York, 1991.

[29] R. Grey, *Entropy and Information Theory*, New York: Springer-Verlag, 1990.

[30] J. Nocedal and S.J. Wright, *Numerical Optimization*, Springer-Verlag, New York, 2000.

[31] M. Golubitsky and D.G.Schaeffer, *Singularities and Groups in Bifurcation Theory I*, Springer-Verlag, New York, 1985.

[32] A. G. Dimitrov and J. P. Miller, Neural coding and decoding: communication channels and quantization, *Network: Computation in Neural Systems*, vol.12, N.4, (2001), 441-472.

Tomáš Gedeon received his B.A. and M.Sc. in Mathematics in 1989 at Comenius University in Bratislava, Slovak Republic (Czechoslovakia). After receiving Ph.D. in Mathematics from Georgia Institute of Technology in 1994, he spent a one-year post-doc at Northwestern University. In 1995 he joined the Department of Mathematical Sciences at Montana State University, where he is currently an Associate Professor of Mathematics. His research interests include information based methods of clustering, dynamics of complex systems in neuroscience and gene regulation, as well as dynamics of evolutionary algorithms.

He is an Associate Editor of the Rocky Mountain Journal of Mathematics.

Collette Campion received her B.S. in Applied Mathematics in May 2005 from Montana State University in Bozeman. She hopes to enter a Ph.D. program in Neuroscience in the near future.

Albert E. Parker received his B.S. in Mathematics in 1994 from Bridgewater State College in Bridgewater, MA, an M.S. in Mathematics in 1997 from the University of Vermont, a Ph.D. in Mathematics in 2003 and an M.S. in Statistics in 2004 from Montana State University in Bozeman. In 2004-2005, he worked as a post-doc for NSF's Center for Adaptive Optics while at Montana State University. His research interests include dynamical systems with symmetry, optimization, clustering, and image registration.

Zane Aldworth received a B.S. in physics from University of Puget Sound in Tacoma, WA in 1998, and received a B.S. in Biology from Montana State Univer-

sity in Bozeman, MT in 1999. He is currently working on his Ph.D. in Neuroscience in the Center for Computational Biology at Montana State University. His research interests include neural coding and structure-function relations in neural systems.