Classification from a Riemannian Graph Embedding Viewpoint

Antonio Robles-Kelly
National ICT Australia (NICTA)
Locked Bag 8001
Canberra ACT 2601, Australia
Email: antonio.robles-kelly@nicta.com.au

Lin Gu
A*STAR Singapore
Bioinformatics Institute
Singapore
Email: gulin@bii.a-star.edu.sg

Ran Wei
National ICT Australia (NICTA)
Locked Bag 8001
Canberra ACT 2601, Australia
Email: ran.wei@nicta.com.au

***Abstract*—In this paper, we employ graph embeddings for classification tasks. To do this, we explore the relationship between kernel matrices, spaces of inner products and statistical inference by viewing the embedding vectors for the nodes in the graph as a field on a Riemannian manifold. This leads to a setting where the inference process may be cast as a *Maximum a Posteriori* (MAP) estimation over a Gibbs field whereby the graph Laplacian can be related to a Gram matrix of scalar products. This not only allows for a better understanding of graph spectral techniques, but also provides a means for classifying nodes in the graph without the need to compute the embedding explicitly by using a Mercer kernel. We illustrate how the developments presented here can be used for purposes of classification, where we use the graph Laplacian as a kernel matrix. We present classification results on synthetic data and four UCI datasets. We also apply our method to real-world image labelling and compare our results to those yielded by alternatives elsewhere in the literature.**

I. Introduction

The problem of embedding relational structures onto a manifold is one of a combinatorial nature which has been traditionally solved by viewing the edge-weights for the graph as distances between pairs of nodes. The embedding coordinates are then those corresponding to the isometric mapping of these pairwise distances onto an n-dimensional space. Graph embedding arises in multidimensional scaling (MDS)[1], graph drawing [2] and relational matching [3].

In order to recover an isometric mapping of the relational structure under study is common practice to pose the problem in an optimisation setting so as to minimise a measure of distortion. An option here is to perform graph interpolation by a hyperbolic surface which has the same pattern of geodesic, *i.e.* internode, distances as the graph under study [4]. Collectively, these methods are sometimes referred to as manifold learning theory. In [5], a manifold is constructed whose triangulation is the simplical complex of the graph. Other methods, such as ISOMAP [6], focus on the recovery of a lower-dimensional embedding which is quasi-isometric in nature. Related algorithms include locally linear embedding [7], which is a variant of PCA that restricts the complexity of the input data using a nearest neighbor graph. Belkin and Niyogi [8] present a Laplacian eigenmap which constructs an adjacency weight matrix for the data-points and projects the data onto the principal eigenvectors of the associated Laplacian matrix.

One of the main application of embedding methods is that of transforming a relational-matching problem into one of point-pattern matching in a high-dimensional space. The problem is, hence, to find matches between pairs of point sets when there is noise, geometric distortion and structural corruption. This problem arises in areas such as shape and motion analysis and stereo reconstruction [9].

For matching applications, one of the main challenges hinges in how to deal with differences in node and edge structure. To overcome this problem, several authors have adopted ideas from information and probability theory. For instance, in [10], the authors recover correspondences in a statistical setting using the EM algorithm. A similar approach is to associate discrete random variables with the nodes and edges to capture the graph structure [11]. Along these lines, Bagdanov and Worring [12] have used normal distributions to model the random variables in their first-order Gaussian graphs. Robles-Kelly and Hancock [13] use the relationship between the Laplace-Beltrami operator and the graph Laplacian for embedding a graph onto a Riemannian manifold. Harandi *et al.* [14] have used graph embedding to represent images in a Grassmannian manifold for image matching.

In this paper, we aim at exploiting the relationship between manifolds, Mercer kernels and probability distributions so as to view the embedding vectors of the graph vertices as a Gibbs field over a space of inner products. Thus, the developments presented here permit classification tasks to be effected over the pairwise distances across the vertex set of the graph without computing the embeddings explicitly. This treatment has a number of additional advantages. Firstly, it permits the use of *Maximum a Posteriori* (MAP) inference to estimate the hidden variables of the corresponding likelihood function making use of the embedding of the graph nodes into a metric space, *i.e.* a kernel. Secondly, it allows for unsupervised, supervised or semisupervised classification scenarios to be treated in a consistent statistical setting. Finally, it can be viewed as a natural extension of graph spectral methods to learning settings where supervised or semisupervised classification and estimation tasks with side information may be effected over the pairwise distances across the vertex set of

the graph. This provides a link between those optimisation techniques which employ invariant subspaces of matrices [15] and spectral methods by viewing the graph Laplacian as a Gram matrix [13]. This is particularly relevant to regularisation kernels on graphs [16] and pairwise clustering methods.

II. Motivation

Here, we depart from a graph theoretic setting and formulate the MAP estimation over the embedding vectors for the graph vertices in Section III. This treatment leads to an inference process that can be related to Support Vector Machines (SVMs) [17] or spectral clustering methods such as the normalised cut [18]. We provide a discussion along these lines in Section IV.

Consider the MAP estimation of a set of hidden random variables whose observables are given by a set $\mathcal{X}$ whose pairwise relationships can be represented using an undirected, weighted graph $G = (V, E, W)$, with index-set V, edge-set $E = \{(u, v)|(u, v) \in V \times V, u \neq v\}$ and edge-weight function set $W : E \rightarrow [0, 1]$. This is a common setting in spectral clustering [18], [19], relational matching [20], [21], [22] or dynamic network analysis [23] where the problem is characterised by the set of nodes V that represent the tokens under study and the set of edge weights, which account for the affinities or "distances" between them.

Here, we note that the embedding process can be effected by viewing the edge-weights in the graph as the correlation of the observables under study in a geometric sense, *i.e.* their inner products [24]. Thus, here we view the embedding vector $\mathbf{x}_i$ as the coordinates on the Riemannian manifold $\mathcal{M}$ corresponding to the node u in V. Further, let the embedding vectors $\mathbf{x}_i$ and $\mathbf{x}_j$ for nodes u and v in V be such that they span an inner product space. This is $\mathcal{K}(u, v) = \langle \mathbf{x}_i, \mathbf{x}_j \rangle$, where $\mathcal{K} : \mathcal{M} \times \mathcal{M} \mapsto \Re$ is a kernel function satisfying Mercer's condition [25].

The use of the Mercer kernel as required above is important since it provides a means to using the embedding vectors for purposes of inference making use of the "kernel trick". They also allow for a general setting where any square positive semi-definite matrix $\mathcal{K} = \mathbf{J}\mathbf{J}^T$ can be viewed as a kernel whereby the i^{th} column of the matrix $\mathbf{J} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{|V|}]$ corresponds to the embedding vector $\mathbf{x}_i$.

III. Probabilistic Formulation

Consider a non-parametric Bayesian MAP estimation over the set of embedding vectors $\mathcal{X}$. We can view this MAP estimation as a maximisation problem of the form

$$\mathcal{B}^* = \mathrm{argmax}_{\mathcal{B}} \Big\{ P(\mathcal{B} \mid \mathcal{X}) \Big\} = \mathrm{argmax}_{\mathcal{B}} \Big\{ \frac{P(\mathcal{X} \mid \mathcal{B}) P(\mathcal{B})}{P(\mathcal{X})} \Big\} \tag{1}$$

where $\mathcal{B}$ is the set of real, non-negative hidden variables being inferred and, as before, the embedding vector $\mathbf{x}_i$ corresponds to the node u in V.

A. Gibbs Fields

In Section III-B we will comment further on the nature of these hidden variables. For now, we focus on the case where the conditional probability for the embedding vectors is given by the distribution over the Gibbs field defined as follows

$$P(\mathcal{X} \mid \mathcal{B}) = \frac{1}{Z_{\mathcal{B}}} \prod_{c_i \in \mathcal{C}} \mathcal{g}_{\mathcal{B}}(c_i) \tag{2}$$

where $\mathcal{C}$ is the set of cliques in the graph $\mathcal{g}_{\mathcal{B}}(c_i)$ is the potential function for the clique $c_i \in \mathcal{C}$ and $Z_{\mathcal{B}}$ is a partition function given by

$$Z_{\mathcal{B}} = \sum_{\mathcal{X}} \Big\{ \prod_{c_i \in \mathcal{C}} \mathcal{g}_{\mathcal{B}}(c_i) \Big\} \tag{3}$$

which serves as a normalising constant and, as usual, a clique is, in general, any fully connected subset of the graph.

Our choice of a Gibbs field has two intended advantages. Firstly, allows for an appropriate sampling strategy, *i.e.* the use of a Gibbs sampler, which has applications in real clustering problems in which the number of clusters is unknown. Secondly, it permits the establishment of a direct connection with a Markov Random field. To this end, we can view the clique potentials as energy functions and restrict the cliques to the first neighbours of the node under consideration. Moreover, note that, in Equation 2, the hidden variables are a parameter for the potential function. Thus, we make use of the embedding vectors and the latent variables and write

$$P(\mathcal{X} \mid \mathcal{B}) = \frac{1}{Z_{\mathcal{B}}} \exp \Big\{ -\frac{1}{T} \sum_{u \in V} \sum_{v \sim u} \beta_i \langle \mathbf{x}_i, \mathbf{x}_j \rangle \beta_j \Big\} \tag{4}$$

where we have written $v \sim u$ to imply that the node u is adjacent to v, $\mathbf{x}_i$ and $\mathbf{x}_j$ correspond to the embedding vectors of the nodes u and v in the graph, respectively, T is the "temperature" variable and $\beta_i, \beta_j \in \mathcal{B}$.

We can give an intuitive interpretation to Equation 4. Note that if two embedding vectors corresponding to nodes adjacent to one another are "close" in the embedding space, *i.e.* the value of $\varsigma = \langle \mathbf{x}_i, \mathbf{x}_j \rangle \approx \langle \mathbf{x}_j, \mathbf{x}_j \rangle$, their potential will be given by the product of the hidden variables weighted by ς. In the other hand, if the embedding vectors are far apart from each other, *i.e.* $\langle \mathbf{x}_i, \mathbf{x}_i \rangle \approx 0$, the potential value will also tend to zero.

Moreover, we can view the prior $P(\beta_i)$ as an indicator of the "relevance" of the embedding vector $\mathbf{x}_i$ in the space of inner products. This implies that the larger the probability $P(\beta_i)$ is, the more relevant the embedding vector is to the inference process. In the other hand, we would also like these "relevant" vectors to be sparse. This implies that the hidden variables should also be sparse and non-negative, *i.e.* $\beta_i \geq 0 \; \forall \; u \in V$. Thus, here we use the exponential prior

$$P(\mathcal{B}) = \kappa \prod_{u \in V} \exp \big\{ -\kappa \beta_i \big\} \tag{5}$$

where κ is the rate parameter of the distribution, which satisfies $P(\mathcal{B}) \in [0, 1]$.

Making use of Equations 4 and 5, we can rewrite Equation 1 using the log-likelihood as an optimisation over the variables β_i as follows

$$\begin{aligned}\mathcal{B}^* &= \text{argmax}_{\mathcal{B}}\Big\{ \log\big(P(\mathcal{X} \mid \mathcal{B})\big) + \log\big(P(\mathcal{B})\big)\Big\} \quad (6)\\ &= \text{argmax}_{\mathcal{B}}\Big\{ -\frac{1}{T}\sum_{u\in V}\sum_{v\sim u}\beta_i\langle \mathbf{x}_i, \mathbf{x}_j\rangle\beta_j + \\ &\qquad \eta\sum_{u\in V}\beta_i - \log\big(Z_{\mathcal{B}}\big)\Big\}\end{aligned}$$

where we have used, as a matter of convenience, the shorthand $\eta = -\kappa$. We have also removed from further consideration the terms $\log(\kappa)$ and $\log\big(P(\mathcal{X})\big)$ as they do not depend on $\mathcal{B}$ and, hence, do not affect the optimisation in hand.

B. Binary Embeddings

We now apply the MAP inference above to a binary classification setting where the embedding vectors belong to either of two distributions with conjugate parameters. This is, the prior $P(\mathcal{B})$ corresponds to the belief that the embedding vector $\mathbf{x}_i$ has a variable $1 \geq \beta_i \geq 0$ in a manifold $\mathcal{M}^+$ and $1-\beta_i$ in another one, which we denote $\mathcal{M}^-$, such that $\mathcal{M} = \mathcal{M}^+ \bigcup \mathcal{M}^-$. Here, we have used the notation $\mathcal{M}^+$ and $\mathcal{M}^-$ so as to be consistent with the common treatment in the classification literature where positive and negative labels are associated to the vectors under consideration.

The use of conjugate parameters has the added advantage that we can now also introduce the label set $\mathcal{Y}$ whose i^{th} entry $\mathbf{y}_i$ is associated to the embedding vector $\mathbf{x}_i$. Here, we restrict the label $\mathbf{y}_i$ to the values -1 and 1 for the positive and negative classes. As a result, the hidden variables are now dependent on the label set $\mathcal{Y}$ as follows

$$\beta_i = \begin{cases}\alpha_i & \text{if} \quad \mathbf{y}_i = 1\\ (1-\alpha_i) & \text{if} \quad \mathbf{y}_i = -1\end{cases} \quad (7)$$

where $1 \geq \alpha_i \geq 0$ is an intermediate variable introduced as a matter of convenience.

Note that the definition above is consistent with the notion that the manifold $\mathcal{M}^+$ corresponds to the embedding space for the positive labels and $\mathcal{M}^-$ accounts for the negative ones. Moreover, note that, in a supervised classification setting, the labels $\mathbf{y}_i$ are at our disposal, whereas in the unsupervised case these can be recovered making use of the sign of the hidden variables β_i. Similarly, the prior $P(\beta_i)$ is now given by

$$P(\beta_i) = \begin{cases}\kappa\exp\{-\kappa\alpha_i\} & \text{if} \quad \mathbf{y}_i = 1\\ \kappa\exp\{-\kappa(1-\alpha_i)\} & \text{if} \quad \mathbf{y}_i = -1\end{cases} \quad (8)$$

To take our analysis further, we note that the local extrema of the cost function, as given in Equation 7, should satisfy the condition

$$\frac{\partial}{\partial\beta_i}\Big\{ \log\big(P(\mathcal{X} \mid \mathcal{B})\big) + \log\big(P(\mathcal{B})\big)\Big\} = 0 \quad (9)$$

Thus, we use the shorthand $\beta_i = \mathbb{I}[\mathbf{y}_i = -1] + \mathbf{y}_i\alpha_i$, where $\mathbb{I}[\mathbf{y}_i = -1]$ is an indicator function which is unity if $\mathbf{y}_i = -1$ and zero otherwise, to obtain

$$\frac{\partial\log(Z_{\mathcal{B}})}{\partial\beta_i} = -\frac{1}{T}\sum_{u\in V}\sum_{v\sim u}(\mathbb{I}[\mathbf{y}_j = -1] + \mathbf{y}_j\alpha_j)\langle\mathbf{x}_i, \mathbf{x}_j\rangle + \sum_{u\in V}\eta \quad (10)$$

By back-substituting the condition above into the likelihood function and removing those terms that are constant from further consideration we reach the expression

$$\begin{aligned}\mathcal{A}^* = \text{argmax}_{\mathcal{A}}\Big\{ &-\frac{1}{T}\sum_{u\in V}\sum_{v\sim u}\mathbf{y}_i\alpha_i\langle\mathbf{x}_i, \mathbf{x}_j\rangle\alpha_j\mathbf{y}_j + \\ &\eta\sum_{u\in V}\mathbf{y}_i\alpha_i + Q(Z_{\mathcal{B}})\Big\}\end{aligned} \quad (11)$$

where

$$Q(Z_{\mathcal{B}}) = 2\frac{\partial\log(Z_{\mathcal{B}})}{\partial\beta_i} - \log\big(Z_{\mathcal{B}}\big) \quad (12)$$

C. Extension to Multiclass Classification

We now turn our attention to the case where, instead of dealing with a binary embedding, such as that in the section above, we aim at inferring the hidden variables of the embedding vectors drawn from C different distributions in the set Ω such that $\mathcal{M} = \bigcup_{\omega\in\Omega}\mathcal{M}^\omega$, where the manifold $\mathcal{M}$ now is the union of the submanifolds $\mathcal{M}^\omega$, each of which corresponds to the domain in which the distribution ω is defined. As a result, we now index the alpha-variables and labels with respect to distribution pairs so as to write the Gibbs field as follows

$$\begin{aligned}&P(\mathcal{X}|\mathcal{B}_{c,d}) = \\ &\quad\frac{1}{Z_{\mathcal{B}_{c,d}}}\exp\Big\{ -\frac{1}{T}\sum_{u\in V}\sum_{v\sim u}y_{i,c,d},\beta_{i,c,d}\langle\mathbf{x}_i, \mathbf{x}_j\rangle y_{j,c,d},\beta_{j,c,d}\Big\}\end{aligned}$$

where $\beta_{i,c,d}$ is defined in a manner akin to that in the previous section as

$$\beta_{i,c,d} = \begin{cases}\alpha_{i,c,d} & \text{if} \quad y_{i,c,d} = 1\\ (1-\alpha_{i,c,d}) & \text{if} \quad y_{i,c,d} = -1\\ 0 & \text{otherwise}\end{cases} \quad (13)$$

and $y_{i,c,d} = 1$ implies that the embedding vector $\mathbf{x}_i$ is drawn from the distribution indexed c. In a similar fashion, $y_{i,c,d} = -1$ indicates the embedding vector is drawn from the distribution indexed d. The third case in the definition above accounts for the possibility that the embedding vector has not been drawn from either of the two distributions under consideration.

Similarly, the prior used previously can be generalised as follows

$$P(\mathcal{B}_{c,d}) = \kappa_{c,d}\prod_{u\in V}\exp\big\{ -\kappa_{c,d}\beta_{i,c,d}\big\} \quad (14)$$

Note that, if we assume independence between the distributions in Ω, we obtain the cost function

$$\arg\max_{\mathcal{A}} \Bigg\{ -\frac{1}{T} \sum_{\substack{c\in\Omega \\ d\in\Omega \\ c\neq d}} \sum_{\substack{u\in V \\ v\sim u}} y_{i,c,d} y_{j,c,d} \alpha_{i,c,d} \alpha_{j,c,d} \langle \mathbf{x}_i, \mathbf{x}_j \rangle + \sum_{\substack{c\in\Omega \\ d\in\Omega \\ c\neq d}} \eta_{c,d} \sum_{u\in V} \mathbf{y}_{i,c,d} \alpha_{i,c,d} + Q(Z_{\mathcal{B}_{c,d}}) \Bigg\} \tag{15}$$

where $\eta_{c,d} = -\kappa_{c,d}$ is a quantity which now depends on the cluster-pair and $Q(Z_{\mathcal{B}_{c,d}})$ is akin to the shorthand shown in Equation 12.

IV. Discussion

Note that, so far, we have focused on the MAP estimation of a set of hidden variables for the embedding vectors of a graph G, devoid of their connections to spectral clustering methods such as that in [18] or supervised classification techniques, such as support vector machines [17]. In this section, we examine these links more closely.

A. Link to Graph Spectral Methods

Clustering can greatly benefit from a graph-theoretic treatment in which the objects to be grouped are represented using a weighted graph whereby the nodes account for the objects under consideration and the edge-weights represent the strength of pairwise similarity relations between them. Indeed, one of the most elegant solutions to the pairwise clustering problem comes from spectral graph theory, *i.e.* the characterisation of the eigenpairs of the graph Laplacian and the adjacency matrix. The result that is key to the grouping problem is that the eigenvalue gap (the difference between the first and second eigenvalues of the Laplacian matrix) is a measure of the degree of bijectivity of the graph, *i.e.* the extent to which its nodes form two distinct clusters which can be separated by a minimum cut.

In this section, we motivate the link between spectral methods [26] and the developments presented in previous sections. To this end, we resort to the Laplacian of the graph and the Gram matrix of scalar products used previously, *i.e.* the kernel $\mathcal{K}$. Recall that the weight matrix W is related to the normalised Laplacian $\mathcal{L} = \boldsymbol{D}^{-\frac{1}{2}}(\boldsymbol{D} - W)\boldsymbol{D}^{-\frac{1}{2}} = \mathbf{I} - \boldsymbol{D}^{-\frac{1}{2}} W \boldsymbol{D}^{-\frac{1}{2}}$, where $\boldsymbol{D}$ is a diagonal matrix such that $\boldsymbol{D} = diag(deg(1), deg(2), \ldots, deg(|V|))$ and $deg(v) = \sum_{u\in V} W_{i,j}$ is the degree, *i.e.* the sum over the entries $W_{i,j}$ of the weight matrix corresponding to the node v over the whole of the graph G.

As noted by Chung [26], the graph Laplacian always admits a Young-Householder decomposition [27] of the form

$$\mathcal{L} = \mathcal{I}\mathcal{I}^T$$

where $\mathcal{I}$ is a $|\ V\ | \times |\ E\ |$ real matrix.

Making use of the matrix $\mathcal{I}$, we can rewrite the cost function in Equation 7 as follows

$$\mathcal{A}^* = \text{argmax}_{\mathcal{A}} \Bigg\{ -\frac{1}{T} \boldsymbol{\phi}[\mathcal{I}\mathcal{I}^T]\boldsymbol{\phi}^T + \eta \sum_{u\in V} \phi_i + \log(Z_{\mathcal{B}}) \Bigg\} \tag{16}$$

where the i^{th} entry of the vector $\boldsymbol{\phi} = [\phi_1, \phi_2, \ldots, \phi_{|V|}]^T$ is given by $\phi_i = \mathbf{y}_i \alpha_i$.

Further, from inspection, it is straightforward to note that the normalised cut [18] solution to segmentation, *i.e.* the eigenvector $\boldsymbol{\phi}$ corresponding to the second smallest eigenvalue τ of the graph Laplacian, yields the expression

$$\frac{\partial \log(Z_{\mathcal{B}})}{\partial \phi_i} = \tau \sum_{u\in V} \phi_i + \sum_{u\in V} \eta \tag{17}$$

when the shorthand $\boldsymbol{\phi}[\mathcal{I}\mathcal{I}^T] = \tau\boldsymbol{\phi}$ is substituted into Equation 10.

Moreover, note that by setting $\eta = \tau$ and adding the constraint

$$\tau \sum_{u\in V} \phi_i = 0 \tag{18}$$

both the derivative above and $\log(Z_{\mathcal{B}})$ become constant. This is as $\frac{\partial \log(Z_{\mathcal{B}})}{\partial \phi_i} = \sum_{u\in V} \eta$ and, as a result, $\log(Z_{\mathcal{B}}) = \tau \sum_{u\in V} \phi_i + o = o$, *i.e.* a linear function of the eigenvector $\boldsymbol{\phi}$ up to the integration constant o.

Thus, the optimisation in Equation 16 becomes

$$\mathcal{A}^* = \text{argmax}_{\mathcal{A}} \Bigg\{ -\boldsymbol{\phi}\mathcal{L}\boldsymbol{\phi}^T \Bigg\} \tag{19}$$

where we have set $T = 1$ and removed from further consideration the terms in the cost function which are constant and, hence, do not affect the maximisation in hand.

It is worth noting that the constraint in Equation 18 is always satisfied by the Fiedler vector [28], *i.e.* the solution to the normalized cut. A similar treatment can be given to those spectral methods which employ the leading eigvenvector of the weight matrix W for purposes of clustering. This is quite telling since it implies that spectral methods based upon the Laplacian can be viewed as a MAP estimate when the embedding vectors, *i.e.* the columns of the matrix $\mathcal{I}$, correspond to a Gibbs field with an exponential prior. This is also in accordance with the developments in [29].

B. Relation to SVMs

We now discuss the relationship of the MAP estimation above to Support Vector Machines (SVMs) [17]. Note that the primals of SVMs and, in particular those arising from formulations elsewhere in the literature such as relevance support vector machines [30] and Laplacian SVMs [31] are quite similar to Equation 16. Moreover, SVMs have been traditionally cast in a regularisation setting pertaining a variational problem in a reproducible kernel Hilbert space [32] and relevance SVMs can be viewed as a Gaussian process under a particular choice of covariance.

To draw a parallel between the binary setting in Section III-B and the SVM, we commence by noting that the dual

form of the support vector machine depends on a single set of parameters, *i.e.* the alpha-weights, where the dividing hyperplane is given by $\langle \mathbf{w}, \mathbf{x}_i \rangle + b = 0$ where b is the intersect of the hyperplane and $\mathbf{w}$ is its normal vector given by

$$\mathbf{w} = \sum_{u \in V} \mathbf{y}_i \alpha_i \mathbf{x}_j \tag{20}$$

This yields the following optimisation problem

$$\min_{\mathbf{w}} \{ \langle \mathbf{w}, \mathbf{w} \rangle \} \tag{21}$$

subject to

$$\mathbf{y}_i (\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geq 0$$

whose primal Lagrangian is given by

$$\mathbb{L}(\mathbf{w}, b, \mathcal{A}) = \frac{1}{2} \langle \mathbf{w}, \mathbf{w} \rangle - \sum_{u \in V} \alpha_i [\mathbf{y}_i (\langle \mathbf{w}, \mathbf{x}_i \rangle + b)] \tag{22}$$

where we have, intentionally, used the variables $\alpha_i \in \mathcal{A}$ as the slacks for the Lagrangian.

Recall that the corresponding dual is found by differentiating the Lagrangian with respect to $\mathbf{w}$ and b. The former yields Equation 20 whereas the latter is given by

$$\frac{\partial \mathbb{L}(\mathbf{w}, b, \mathcal{A})}{\partial b} = - \sum_{u \in V} \alpha_i \mathbf{y}_i = 0 \tag{23}$$

By substituting these relations into Equation 22 we get

$$\mathbb{L}(\mathbf{w}, b, \mathcal{A}) = -\frac{1}{2} \sum_{u \in V} \sum_{v \sim u} \mathbf{y}_i \alpha_i \langle \mathbf{x}_i, \mathbf{x}_i \rangle \alpha_j \mathbf{y}_j \tag{24}$$

Note that this is essentially the same as the cost function in Equation 19 when $T = 2$ and the condition in Equation 23 is imposed explicitly. This condition can be viewed, from a statistical viewpoint, as aiming at removing the bias from the slack variables. This can be easily viewed by noting that the condition in Equation 23 is equivalent to

$$\sum_{\mathbf{x}_i \in \mathcal{X}} \mathbb{I}[y_i = -1] \alpha_i = \sum_{\mathbf{x}_i \in \mathcal{X}} \mathbb{I}[y_i = 1] \alpha_i \tag{25}$$

Moreover, note that the traditional SVM dual corresponds to that in Equation 22 with the term $\sum_{v \in V} \alpha_i$ added. This indeed corresponds to the regularised case where the constraint $\sum_{u \in V} \alpha_i = 1$ is included into Equation 24 using the Lagrange multiplier λ, which yields the maximisation

$$\mathcal{A}^* = \text{argmax}_{\mathcal{B}} \left\{ -\frac{1}{T} \sum_{u \in V} \sum_{v \sim u} \mathbf{y}_i \alpha_i \langle \mathbf{x}_i, \mathbf{x}_j \rangle \alpha_j \mathbf{y}_j + \lambda \sum_{u \in V} \alpha_i \right\} \tag{26}$$

Moreover, by viewing b as the expected zero-loss value of the distribution across the set $\mathcal{X}$, we can obtain the intersect using the average over the vertex set $V \in G$. This is

$$b = \frac{1}{|V|} \left(\sum_{\substack{u \in V \\ v \sim u}} \alpha_i \mathbf{y}_i \langle \mathbf{x}_i, \mathbf{x}_j \rangle - \sum_{u \in V} \mathbf{y}_i \right) \tag{27}$$

The importance of this interpretation resides in the fact that we can now provide a statistical interpretation of the classifier output. Thus, by making use of a testing strategy akin to that of an SVM, we can classify the novel vector $\mathbf{z}$ making use of the rule

$$\mathbf{y}_{\mathbf{z}} = \text{sign} \left\{ \sum_{u \in V} \alpha_i \mathbf{y}_i \langle \mathbf{x}_i, \mathbf{z} \rangle - b \right\} \tag{28}$$

where $\text{sign}\{\cdot\}$ is the sign function.

Once the label $\mathbf{y}_z$ corresponding to the vector $\mathbf{z}_k$ has been recovered, the probability of the embedding vector given the observables in $\mathcal{X}$ is given by

$$P(\mathbf{z} \mid \mathcal{X}, \mathcal{B}) = 1 - \exp \left\{ -\frac{1}{2} \mathbf{y}_{\mathbf{z}} \left(\sum_{u \in V} \alpha_i \mathbf{y}_i \langle \mathbf{x}_i, \mathbf{z} \rangle - b \right) \right\} \tag{29}$$

V. Experiments

We now illustrate how the developments in previous sections can be used to tackle supervised, semisupervised and unsupervised classification tasks. To this end, we commence by illustrating the behaviour of the method for purposes of binary and multiclass classification on synthetic data. We then turn our attention to three datasets from the UCI repository and real-world image labelling.

For all our experiments, we have recovered a weighted graph using their Delaunay triangulation, where the entries of the weight matrix W are given by $\exp\left(\frac{-d^2(u,v)}{t}\right)$ where $d^2(u, v)$ is the Euclidean distance between the pair of points u and v and t is a bandwidth parameter. Here, unless otherwise noted, t has been set to 2 and the optimisation of the cost functions involved in our experiments has been effected using the Biconjugate gradient optimisation method in [33].

Once the weight matrix is in hand, the graph Laplacian is computed as described in Section IV-A. The Laplacian is then used to recover the alpha parameters via the optimisation in Equation 16, where the condition $\sum_{u \in V} \alpha_i = 1$ and that in Equation 25 have been enforced to remove bias and provide a box constraint on the alpha parameters. Thus the optimisation becomes

$$\mathcal{A}^* = \text{argmax}_{\mathcal{A}} \left\{ -\frac{1}{T} \boldsymbol{\phi} \mathcal{L} \boldsymbol{\phi}^T + \lambda \sum_{u \in V} \alpha_i \right\} \tag{30}$$

subject to the constraint

$$\sum_{u \in V} \phi_i = 0$$

where $\mathbf{y}_i \alpha_i = \phi_i$ and $\alpha_i \in [0, 1] \ \forall \ u \in G$. It is worth noting in passing that the constraint $\alpha_i \in [0, 1]$ is in accordance with the developments in Section III.

A. Synthetic Data

First, we commence by using synthetic data to illustrate the behaviour of our method. Thus, we have generated sample point sets in 2-dimensional spaces for purposes of classification.

In the left-hand column of Figure 1 we show the three point clouds used for our binary classification experiments. In the panels, we have used blue stars for the positive class and red crosses for the negative one. These describe two

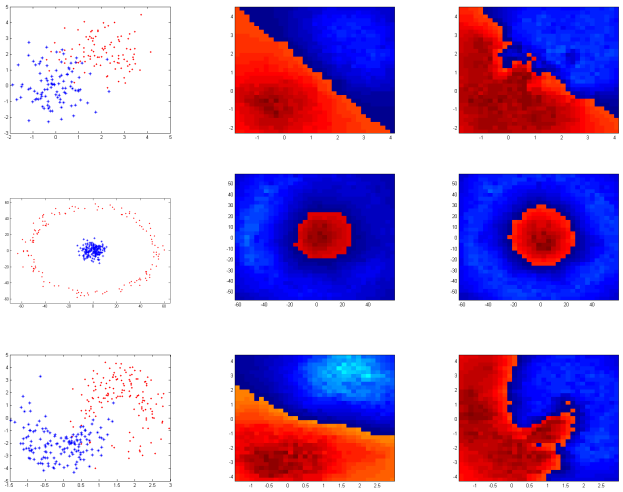

Fig. 1. Binary classification results. Left-hand column: Input point patterns; Middle column: Classification regions recovered without the use of labels for the recovery of the alpha variables; Right-hand column: Classification regions yielded by the use of label information.

Normally distributed point clouds, an annulus and two clusters with the shape of the Milky Way. Recall that, following the developments in Section IV, novel embedding vectors can be tested using the data labels and alpha parameters making use of Equations 27-28.

In the middle column of the figure we show the results obtained by testing the 2-dimensional space at regular intervals, where the colour of the regions denotes the class label and its hue the probability yielded by Equation 29. To do the testing, we set $\phi_i = \alpha_i \mathbf{y}_i$ and recover the inner product $\langle \mathbf{x}_i, \mathbf{x}_k \rangle$ by augmenting the graph, *i.e.* adding the test data point to the point cloud and computing a new Delaunay triangulation and graph Laplacian. It is worth noting that only the Laplacian rows and columns for the nearest neighbours of the testing point will be affected due to the use of the Delaunay triangulation to recover the graph G under consideration. Thus, the testing step can be done efficiently by only recomputing the Delaunay triangulation and graph Laplacian entries for the nearest neighbours of the testing point across the corresponding simplex.

Recall that, if the labels for the training data are known, the optimisation can be effected making use of Equation 26, where the inner products are given by the entries of the graph Laplacian. In the right-hand column of Figure 1 we show the classification of the 2-dimensional space when the labels are used as side information. Note that the boundaries between the two regions are in better accordance with the training data as compared to the regions shown in the middle panel.

TABLE I
STATISTICS FOR THE UCI DATASETS USED IN OUR EXPERIMENTS.

	BLD	Adult	SL	HARSP
No. of classes	2	2	6	6
No. of attributes	7	14	36	561
No. of instances	345	48842	6435	10299

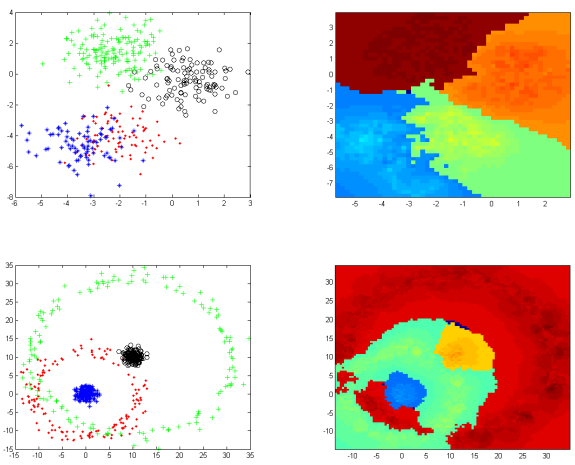

Fig. 2. Multiclass classification results. Left-hand column: Input point patterns; Right-hand column: Recovered classification regions.

TABLE II
TESTING ACCURACY OF OUR METHOD AND A NUMBER OF ALTERNATIVES ON THE UCI DATASETS USED FOR OUR EXPERIMENTS. THE BEST PERFORMANCE IS IN BOLD.

	BLD	Adult	SL	HARSP
LDA	64.9%	77.9%	82.8%	92.6%
SVM (Linear)	67.4%	83.3%	84.8%	93.7%
SVM (RBF)	69.9	84.2	88.8%	96.1
LS-SVM	75.1%	82.9%	83.1%	**96.9%**
Relevance SVM	76.8%	**84.6%**	89.7%	96.7%
KRR	75.1%	82.34%	82.9%	96.6%
Our method	**79.1%**	**84.6**%	**89.9%**	96.8%

Moreover, by using the labels and data points as training data we can perform multiclass classification in a straightforward manner using the developments in Section III-C. In Figure 2, we show two sample point sets and the corresponding classification regions obtained by testing the space sampled at regular intervals. As per Figure 1, we have used different markers for each class to plot the input point clouds in the left-hand panels. Similarly, we have used different colours for the classification regions in the right-hand panel, where the hue indicates the probability in Equation 2 for the most significant class. Here, we have effectively computed the graph Laplacian for all the class pairs and tested the space in a manner akin to that used for our binary classification experiments.

B. UCI Datasets

In order to provide a more quantitative analysis, we have used four datasets from the UCI repository [1]. These are two for binary classification, *i.e.* the Bupa Liver Disorders (BLD)

[1] Accessible at https://archive.ics.uci.edu/ml/datasets.html

and the Adult datasets, and two multiclass ones, *i.e.* the Statlog Landsat (SL) dataset and the human activity recognition using smart phones (HARSP). In Table I we show the statistics of these datasets.

For purposes of comparison, we have performed experiments using Linear Discriminant Analysis (LDA) [34], SVMs with a linear and RBF kernel, relevance SVM [30], RBF Kernel Ridge Regressor (KRR) [35], the least-squares SVM (LS-SVM) [36] and our method. Note that the LDA can be viewed as a homoscedastic quadratic classifier (the covariances for the classes are identical) whereas the relevance SVM can be related to Gaussian processes [37]. For our LDA implementation, we have used Alglib [2] whereas for all the variants of the SVMs we have used DLib [3]. For the SVMs, when applied to the Statlog Landsat (SL) and the human activity recognition using smart phones (HARSP) datasets, we have used a one-vs-all strategy for multiclass classification.

In Table II we show the classification accuracy for our method and the alternatives. All the parameters for the SVMs were chosen by cross validation. Note that our method provides a margin of improvement over the alternatives on the BLD and SL datasets and delivers an accuracy comparable to that yielded by the relevance SVM on the Adult dataset. This is somewhat expected since the relevance SVM shares the statistical nature of our method but employs an optimisation scheme based upon the Expectation-Maximisation (EM) algorithm and, hence, can be affected by local minima. For the HARSP dataset, is the second best, after the least-squares SVM.

C. Real-world Imagery

We now turn our attention to the supervised segmentation of real-world imagery. In Figure 3, we show five real-world images which have been partially labelled using an interactive scribble tool. The pixels on the scribbles are then used to train a classifier which, in turn, is then used to classify the remaining pixels in the image. Here, the vector $\mathbf{x}_i$ is given by the three colour channel values and the row-column coordinates on the image lattice of the corresponding pixel. We have done this following the intuition that, in this manner, the classification results will take into account both, the position of the scribbles on the image relative to the testing pixels as well as their colour values.

To account for the side information provided by the set of labelled pixels Υ, we have modified Equation 30 so as to enforce the "closeness" of the recovered labels with those provided in the scribbles. As a result, we have used the cost function

$$\mathcal{A}^* = \text{argmax}_{\mathcal{A}} \left\{ -\frac{1}{T}\boldsymbol{\phi}\mathcal{L}\boldsymbol{\phi}^T + \lambda\boldsymbol{\phi}^T\boldsymbol{\phi} + \zeta \sum_{\substack{u\in V \\ u\in\Upsilon}} \mathbf{y}_i\phi_i \right\} \quad (31)$$

[2]For more information please go to http://www.alglib.net
[3]For more information please go to http://dlib.net

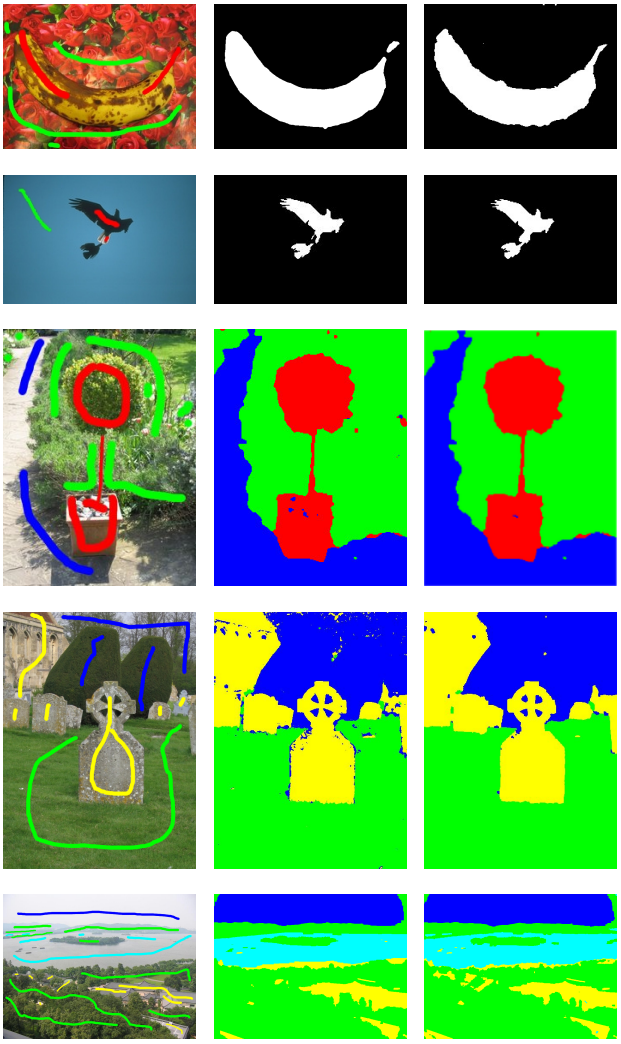

Fig. 3. Image segmentation results. Left-hand column: Input images with overlaid label scribbles; Right-hand columns: Segmented regions obtained using the Random Walker algorithm [38] and our approach, respectively.

subject to the constraint

$$\sum_{\substack{u\in V \\ u\notin\Upsilon}} \phi_i = 0$$

where, as before, $\phi_i = \mathbf{y}_i\alpha_i$ and $\zeta_i = 0.15$ is a constant that controls the influence of the last term in the right-hand side on the optimisation.

Note that the formulation in Equation 31 can still be viewed as a regularised version of Equation 24 whereby we have used the quadratic term $\boldsymbol{\phi}^T\boldsymbol{\phi}$ as an alternative to $\sum_{u\in V}\alpha_i$ for $\alpha_i \geq 0$. Further, the cost function above is convex in ϕ_i.

For comparison, we have used the random walker segmentation algorithm [38], which makes use of a simplified data term and a convex quadratic term for purposes of enforcing smoothness between neighbouring pixels in an interactive image segmentation setting. Further, the random walker can be viewed as a relaxation of the Potts model and, hence, it shares with our approach the link to an MRF.

In Figure 3 we show our real-world imagery with the label scribbles overlaid and the segmentation results yielded by both, the random walker and our approach. Again, for the sake of consistency, we use the same labelling mask for both, our method and the random walker. Note that the random walker tends to deliver more "cluttered" segmentation results. Moreover, our image segmentation results show good detail while delivering spatially compact regions. This can be seen on the tomb stone image, where the details on the green and the hollow parts of the stone are well preserved. This is also the case of the bird, where the detail on the wings, trailing falcon and tail are quite evident.

VI. Conlusions

In this paper, we have drawn on statistical inference and machine learning concepts so as to provide a means for classification via the Riemannian embedding of a graph. By viewing the embedding vectors for the nodes in the graph as a field on a Riemannian manifold, we cast the inference process as a MAP estimation on a Gibbs field with an exponential prior. This allows for a better understanding of graph spectral techniques. This is as the use of the graph Laplacian and its relation with a Gram matrix of inner products as explored here delivers, in effect, a supervised spectral classifier. We have illustrated the use of the developments presented here for purposes of classification on synthetic and real-world data and compared the results yielded by our method with those delivered by alternatives elsewhere in the literature.

References

[1] I. Borg and P. Groenen, *Modern Multidimensional Scaling, Theory and Applications*, ser. Springer Series in Statistics. Springer, 1997.

[2] G. Di Battista, P. Eades, R. Tamassia, and I. Tollis, *Graph Drawing: Algorithms for the Visualization of Graphs*. Prentice Hall, 1998.

[3] M. F. Demirci, A. Shokoufandeh, S. Dickinson, Y. Keselman, and L. Bretzner, "Many-to-many feature matching using spherical coding of directed graphs," in *European Conference on Computer Vision*, 2004, pp. I: 322–335.

[4] H. Busemann, *The geometry of geodesics*. Academic Press, 1955.

[5] A. Ranicki, *Algebraic l-theory and topological manifolds*. Cambridge University Press, 1955.

[6] J. B. Tenenbaum, V. de Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, no. 5500, pp. 2319–2323, 2000.

[7] S. T. Roweis and L. K. Saul., "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, pp. 2323–2326, 2000.

[8] M. Belkin and P. Niyogi, "Laplacian eigenmaps and spectral techniques for embedding and clustering," in *Neural Information Processing Systems*, no. 14, 2002, pp. 634–640.

[9] P. Foggia, G. Percannella, and M. Vento, "Graph matching and learning in pattern recognition in the last 10 years," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 28, no. 1, 2014.

[10] B. Luo and E. R. Hancock, "Structural graph matching using the EM algorithm and singular value decomposition," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 23, no. 10, pp. 1120–1136, 2001.

[11] A. K. C. Wong, J. Constant, and M. You, "Random graphs," in *Syntactic and Structural Pattern Recognition*, 1990, pp. 197–234.

[12] A. D. Bagdanov and M. Worring, "First order gaussian graphs for efficient structure classification," *Pattern Recognition*, vol. 36, no. 6, pp. 1311–1324, 2003.

[13] A. Robles-Kelly and E. R. Hancock, "A riemannian approach to graph embedding," *Pattern Recognition*, vol. 40, no. 3, pp. 1042–1056, 2007.

[14] M. T. Harandi, C. Sanderson, S. A. Shirazi, and B. C. Lovell, "Graph embedding discriminant analysis on grassmannian manifolds for improved image set matching," in *Computer Vision and Pattern Recognition*, 2011, pp. 2705–2712.

[15] P. Absil, R. Mahony, and R. Sepulchre, *Optimization Algorithms on Matrix Manifolds*. Princeton University Press, 2008.

[16] A. J. Smola and R. Kondor, "Kernels and regularization on graphs," in *COLT*, 2003, pp. 144–158.

[17] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines*. Cambridge University Press, 2000.

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

[19] S. Sarkar and K. L. Boyer, "Quantitative measures of change based on feature organization: Eigenvalues and eigenvectors," *Computer Vision and Image Understanding*, vol. 71, no. 1, pp. 110–136, 1998.

[20] S. Sclaroff and A. Pentland, "Modal matching for correspondence and recognition," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 17, no. 6, pp. 545–561, 1995.

[21] S. Umeyama, "An eigen decomposition approach to weighted graph matching problems," *PAMI*, vol. 10, no. 5, pp. 695–703, September 1988.

[22] T. Caetano, T. Caelli, D. Schuurmans, and D. A. Barone, "Graphical models and point pattern matching," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 28, no. 10, pp. 1646–1663, 2006.

[23] P. Kohli and P. Torr, "Efficiently solving dynamic markov random fields using graph cuts," in *International Conference on Computer Vision*, 2005.

[24] F. Tang and H. Tao, "Probabilistic object tracking with dynamic attributed relational feature graph," *IEEE Trans. Circuits Syst. Video Techn.*, vol. 18, no. 8, pp. 1064–1074, 2008.

[25] J. Mercer, "Functions of positive and negative type and their connection with the theory of integral equations," *Philos. Trans. Royal Soc. (A)*, vol. 83, no. 559, pp. 69–70, 1909.

[26] F. Chung, *Spectral Graph Theory*. American Mathematical Society, 1997.

[27] G. Young and A. S. Householder, "Discussion of a set of points in terms of their mutual distances," *Psychometrika*, vol. 3, pp. 19–22, 1938.

[28] M. Fiedler, "A property of eigenvectors of nonnegative symmetric matrices and its application to graph theory," *Czech Math. Journal*, no. 25, pp. 619–633, 1975.

[29] A. Robles-Kelly and E. R. Hancock, "A probabilistic spectral framework for spectral clustering and grouping," *Pattern Recognition*, vol. 37, no. 7, pp. 1387–1405, 2004.

[30] M. E. Tipping, "Sparse bayesian learning and the relevance vector machine," *Journal of Machine Learning Research*, pp. 211–244, 2001.

[31] S. Melacci and M. Belkin, "Laplacian support vector machines trained in the primal," *Journal of Machine Learning Research*, vol. 12, pp. 1149–1184, 2011.

[32] B. Scholkopf and A. J. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, 2001.

[33] R. Barrett, M. Berry, T. F. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine, and H. V. der Vorst, *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods*. SIAM, 1994.

[34] G. McLachlan, *Discriminant Analysis and Statistical Pattern Recognition*. Wiley-Interscience, 2004.

[35] K. P. Murphy, *Machine Learning: A Probabilistic Perspective*. MIT Press, 2012, pp. 492–493.

[36] T. Van Gestel, J. A. K. Suykens, J. De Brabanter, A. Lambrechts, B. De Moor, and J. Vandewalle, "Bayesian framework for least-squares support vector machine classifiers, gaussian processes, and kernel fisher discriminant analysis," *Neural Computation*, vol. 14, no. 5, pp. 1115–1147, 2002.

[37] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.

[38] L. Grady, "Random walks for image segmentation," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 28, no. 11, pp. 1768–1783, 2006.