Context-guided diffusion for label propagation on graphs

Kwang In Kim
Lancaster University

James Tompkin
Harvard Paulson SEAS

Hanspeter Pfister
Harvard Paulson SEAS

Christian Theobalt
MPI for Informatics

Abstract

Existing approaches for diffusion on graphs, e.g., for label propagation, are mainly focused on isotropic diffusion, which is induced by the commonly-used graph Laplacian regularizer. Inspired by the success of diffusivity tensors for anisotropic diffusion in image processing, we presents anisotropic diffusion on graphs and the corresponding label propagation algorithm. We develop positive definite diffusivity operators on the vector bundles of Riemannian manifolds, and discretize them to diffusivity operators on graphs. This enables us to easily define new robust diffusivity operators which significantly improve semi-supervised learning performance over existing diffusion algorithms.

1. Introduction

Physical diffusion describes how energy, mass, or substances spread over time — how their densities *smoothen out* in a medium. Simulating physical diffusion on a Euclidean space, a manifold, or their discrete approximations, e.g., grids or graphs, has application in image processing, computer vision, and machine learning. For instance, diffusion is now a standard tool for removing noise or to highlight salient structures [32]. The graph Laplacian, as a discrete approximation of the generator of the diffusion process on manifolds, i.e., the Laplace-Beltrami operator, is commonly used in spectral clustering and semi-supervised learning, which finds applications in object recognition [7, 33], image retrieval [10], and segmentation and matting [3, 25]. Similarly, stochastic diffusion process on graphs find application in multi-label classification [30] and image retrieval [12].

In these applications, typically we are given a set of objects $X = \{\mathbf{x}_1,...,\mathbf{x}_n\}$ and corresponding assignments of variables $Y^t = \{\mathbf{y}_1^t,...,\mathbf{y}_n^t\}$ at time $t = 0$. Then, (simulated) diffusion models how Y *smooths* over X. For instance, when X denotes vertices of a mesh, Y is the coordinate representations of X in an embedding space $\mathcal{X}$, leading to mesh fairing. More generally, if X denotes noisy observations of data points lying on a manifold, diffusion leads to manifold denoising. If Y represents class labels of data points in X, diffusion leads to label propagation and facilitates semi-supervised learning. In this case, Y is assumed to be a sample from an underlying classification function f on $\mathcal{X}$ (i.e., $Y = \{\mathbf{y}_1,...,\mathbf{y}_n\} = \{f(\mathbf{x}_1),...,f(\mathbf{x}_n)\}$).

Diffusion is determined by the initial condition Y^0 and the *diffusivity* defined on X or $\mathcal{X}$. Roughly, the diffusivity describes the direction and strength of f (and equivalently Y) being smoothed at each time instance t. In general, the diffusivity is inhomogeneous as it varies over X, and is anisotropic as its strength varies over different directions at each point $\mathbf{x} \in X$. For instance, in image processing, diffusivity is strong in flat regions but weaker on edges. Further, on an edge, diffusivity is stronger along the direction of edges than across it. This leads to edge-preserving image smoothing as pioneered by Weickert [32].

For graph data, diffusion can be seen as label propagation in semi-supervised learning. Thus far, label propagation has mainly focused on isotropic diffusion (i.e., the diffusivity is fixed on the entire data space and all directions at each point therein), and only recently has anisotropic diffusion been explored: Coifman and Lafon [5] apply anisotropic diffusion to the graph-based dimensionality reduction problem. They control diffusivity by normalizing the (originally isotropic) pair-wise similarity with the evaluations of diffused coordinate values. Szlam *et al*. [29] generalizes and extends this framework to semi-supervised learning by controlling diffusivity via evaluations of class labels f: If $f(\mathbf{x}_i)$ and $f(\mathbf{x}_j)$ are similar, i.e., if the class labels of $\mathbf{x}_i$ and $\mathbf{x}_j$ are likely to be the same, then diffusivity along the edge joining them is high. Otherwise, diffusivity becomes low, which prevents label propagation across class boundaries. This leads to significant performance improvement over classical isotropic diffusion. Kim *et al*. [21] proposed adapting diffusivity on Riemannian manifolds based on local curvature estimates: Diffusivity is strong in flat regions and weak along the direction of the curvature operator, which leads to an awareness of intersections between manifolds and so improves performance over isotropic equivalents. However, this requires the data X to be embedded in an ambient Euclidean space, and so does not apply to inference on general graphs.

We propose two contributions for anisotropic diffusion on graphs. First, we analyze continuous anisotropic diffusion processes on smooth manifolds, and show that anisotropic diffusion is nothing more than isotropic diffusion on a manifold with a new metric. Based on this analysis, we arrive at a new anisotropic graph Laplacian approach which is similar to the stochastic kernel smoothing approach of Szlam *et al*. [29], but with a new geometric intuition. This provides explicit criteria to define valid diffusivities on graphs and manifolds, and it facilitates non-linear diffusion on graphs. Second, we explore two possible operators which control diffusivity of each edge based on local neighborhood contexts and not just their end vertices. This *context-guided diffusion* extends to graphs the robust diffusion algorithm originally developed for image enhancement [32], and

1550-5499/15 $31.00 © 2015 IEEE
DOI 10.1109/ICCV.2015.318

we demonstrate on 11 different classification problems that this improves semi-supervised learning performance over isotropic diffusion, the stochastic anisotropic diffusion of Szlam *et al.* [29], and three existing label propagation algorithms [37, 11, 31].

To assist readers and subsequent development, we make our code available on the web.

2. Anisotropic diffusion on graphs

We develop anisotropic analogs to the existing isotropic diffusion process and to the corresponding graph Laplacian. We also introduce context-guided diffusion for semi-supervised learning. These contributions are based on the analysis of the continuous positive definite diffusivity operators on Riemannian manifolds, which we leave for Sec. 3.

Existing works [35, 17] establish the (isotropic) graph Laplacian as a discrete approximation of the Laplace-Beltrami operator on a data manifold. We build upon these works to develop isotropic and anisotropic graph Laplacians by combining local diffusivity operators defined on sub-graphs centered at each data point. As such, first, we explain existing approaches.

Discrete isotropic diffusion. A weighted graph (X,E,W) consists of sets of nodes X of size n, edges $E\subset X\times X$, and non-negative similarities $w_{ij}:=w(e_{ij})\in W$ for each edge $e_{ij}\in E$, with $w_{ij}=0$ if $e_{ij}\notin E$.

For subsequent definition of diffusivity operators based on local gradients and divergences, we need spaces with defined inner products (i.e., Hilbert spaces), and so we introduce spaces $H(X)$ and $H(E)$ of functions on X and E, with inner products defined as [35, 17]:

$$\langle f,h\rangle_{H(X)}=\sum_{i=1}^{n}f(i)h(i)d_i,\forall f,g\in H(X), \tag{1}$$

$$\langle S,T\rangle_{H(E)}=\sum_{i,j=1}^{n}S(i,j)T(i,j),\forall S,T\in H(E), \tag{2}$$

where $f(i)=f(\mathbf{x}_i)$ and d_i is the degree of node $\mathbf{x}_i\in X$:

$$d_i=\sum_{j=1}^{n}w_{ij}. \tag{3}$$

For each node $\mathbf{x}_i$, a subgraph $G_i=(X_i,E_i,W_i)$ centered at $\mathbf{x}_i$ is defined as the set of nodes that are connected to $\mathbf{x}_i$ and the corresponding edges, i.e., $X_i=\{\mathbf{x}_j|e_{ij}\in E\}$, $E_i=\{e_{ij}|\mathbf{x}_j\in X_i\}$, and W_i are obtained by evaluating W at E_i. The inner-product structures on X_i and E_i are induced as restrictions of the corresponding structures on the entire graph G to the sub-graph G_i, which we denote by $H(X_i)$ and $H(E_i)$, respectively. Given these structures, we define discrete gradient and divergence operators at G_i. First, the graph gradient operator $\nabla_i:H(X_i)\to H(E_i)$ is defined as the collection of f differences along the edges:

$$[\nabla_i f](e_{ij})=\sqrt{w_{ij}}(f(j)-f(i)), \tag{4}$$

for $e_{ij}\in E_i$ and $f\in H(X_i)$. The graph divergence operator $\nabla_i^*:H(E_i)\to H(X_i)$ is defined as the formal adjoint of ∇_i: for all $f\in H(X_i),S\in H(E_i)$:

$$\langle\nabla_i f,S\rangle_{H(E_i)}=\langle f,\nabla_i^*S\rangle_{H(X_i)}. \tag{5}$$

By substituting Eq. 4 into Eq. 5, ∇_i^* is explicitly given as

$$[\nabla_i^*S](i)=\frac{1}{2d_i}\sum_{j=1}^{n}\sqrt{w_{ji}}(S(j,i)-S(i,j)). \tag{6}$$

By combining the local gradient and divergence operators, we can construct the global normalized graph Laplacian $L:H(X)\to H(X)$:

$$[Lf](i)=\nabla_i^*\nabla_i f,\ \forall f\in H(X),i=1,...,n. \tag{7}$$

Our definition of the graph Laplacian is consistent with [35, 17]. In particular, at the i-th node, it is explicitly given as:

$$[Lf](i)=f(i)-\frac{1}{d_i}\sum_{j=1}^{n}w_{ji}f(j). \tag{8}$$

If the nodes X of G are sampled from an underlying data generating manifold M, i.e., the probability distribution $P(\mathbf{x})$ is supported in M, the graph Laplacian L converges to the Laplace-Beltrami operator Δ on M as $n\to\infty$ [17, 1]. This is often regarded as the reason for using graph Laplacian as a regularizer in many applications: The semi-norm $\|f\|_\Delta$ induced by Δ is equivalent to the norm of the gradient ∇f of a function f on M (see Sec. 3). Then, Lf is obtained as a discrete approximation of the first-order regularizer on graphs. Further, Δ is the generator of isotropic diffusion process on M and accordingly, L is also a discrete approximation of the isotropic diffusion generator on G.

Anisotropic diffusion on graphs. Next, we extend isotropic graph Laplacian L to be anisotropic. Our derivation is based on Weickert's definition on positive definite (PD) diffusivity operators on $\mathbb{R}^2$ [32]. In Section 3, we introduce an extension of these operators to general Riemannian manifolds and, based on that, establish a rigorous connection between our anisotropic diffusion process on G and that of the data generating manifold M.

First, we formally introduce the local diffusivity operator $D_i:H(E_i)\to H(E_i)$:

$$D_i:=\sum_{j\neq i,\mathbf{x}_j\in X_i}q_{ij}\mathbf{b}_{ij}\otimes\mathbf{b}_{ij}$$
$$\Leftrightarrow[D_iS](e_{ij})=q_{ij}\mathbf{b}_{ij}\langle\mathbf{b}_{ij},S\rangle,\forall S\in H(E_i), \tag{9}$$

where $\otimes$ is the tensor product and the *basis function* $\mathbf{b}_{ij}$ is defined as the indicator of e_{ij}, i.e., $\mathbf{b}_{ij}=\mathbf{1}_{ij}$. Similar to the construction of diffusivity operators on $\mathbb{R}^2$ [32], our diffusivity operators are constructed based on its spectral decomposition: q_{ij} is an eigenvalue of the operator D_i corresponding to the eigenfunction $\mathbf{b}_{ij}$. This enables us to straightforwardly define a globally PD diffusivity operator on G: Our global diffusivity operator $D:H(E)\to$

$H(E)$ is obtained by identifying D_i as the restriction of D on $H(E_i)$. In this case, D is positive definite if and only if $\{q_{ij}\}$ is symmetric and positive, i.e., $q_{jk}=q_{kj}, q_{jk}>0, \forall j,k=1,...,n$. Furthermore, D is *uniformly PD* if all eigenvalues $\{q_{ij}\}$ are lower-bounded by a positive constant ν.

Now we are ready to define an anisotropic diffusion process on G. We construct an anisotropic graph Laplacian:

$$\begin{aligned}[L^D f](i) &:= [\nabla_i^* D_i \nabla_i f](i), \\ &= \left(\frac{1}{d_i}\sum_{j=1}^{n} w_{ij}q_{ij}\right) f(i) - \frac{1}{d_i}\sum_{j=1}^{n} w_{ij}q_{ij}f(j), \end{aligned} \quad (10)$$

where the equality in the second line is obtained by substituting Eqs. 4, 5, and 9 into the first line.

Except for the normalization term in $f(i)$, the construction of L^D is identical to the isotropic graph Laplacian L case: The original weights $\{w_{ij}\}$ are replaced by new weights $\{w_{ij}^D\}$:

$$w_{ij}^D = w_{ij}q_{ij}. \quad (11)$$

Given the anisotropic graph Laplacian L_D, we can define the corresponding anisotropic diffusion process on G. For instance, for label propagation applications, we propose using the explicit Euler approximation (cf. Eq. 20 for the continuous counterpart):

$$\begin{aligned}\frac{f^{t+1}-f^t}{\delta} &= -L^D f^t \\ \Leftrightarrow f^{t+1} &= f^t - \delta L^D f^t, \end{aligned} \quad (12)$$

where f^t denotes the value of f at time t and δ is the time discretization interval. The uniform positive definiteness of the diffusivity operators is crucial to the well-posedness of the corresponding diffusion process in $\mathbb{R}^2$ [32]. The same applies to the positive definiteness of our discrete diffusivity operator D: This is the only way that L_D is a conditionally PD matrix and therefore it can be a valid regularizer on G:

$$R_{L^D}(f) := \mathbf{f}^\top L^D \mathbf{f} = \sum_{i,j=1,...,n} w_{ij}^D / d_i (f(i)-f(j))^2, \quad (13)$$

where $\mathbf{f}=[f(1),...,f(n)]^\top$: For simplicity, we assume that $f(i)$ is a scalar. When $f(i)$ is a vector, e.g., for multi-class classification, $R_{L^D}(f)$ is summed over the output dimensions. If D is fixed throughout diffusion, the difference equation (12) is linear and the corresponding analytical solution f^t exists for any $\delta>0$ and $t>0$ given f^0. However, in general, D depends on f^t (e.g., Eq. 15) and so Eq. 12 becomes nonlinear, where the solution f^t can be obtained by iterating updating f^t with the right side of Eq. 12.

Anisotropic diffusion for semi-supervised learning. With proper choices of $\{q_{ij}\}$, our diffusion equation (Eq. 12) can be used in various applications including label propagation for semi-supervised learning. Assume we are given a set of data points $X=\{\mathbf{x}_1,...,\mathbf{x}_n\}\in\mathbb{R}^d$ where only the first l-data points are provided with the ground-truth class labels $Y=\{\mathbf{y}_1,...,\mathbf{y}_l\}$. Our goal is to *propagate* these labels to the entire dataset X. We approach this problem by first building a graph $G=(X,E,W)$ with:

$$w_{jk} = \begin{cases} \exp\left(-\frac{\|\mathbf{x}_j-\mathbf{x}_k\|^2}{\sigma_{\mathbf{x}}}\right) & \text{if } \mathbf{x}_j\in N_K(\mathbf{x}_k) \\ & \text{or } \mathbf{x}_k\in N_K(\mathbf{x}_j) \\ 0 & \text{otherwise,} \end{cases} \quad (14)$$

where $N_K(\mathbf{x}_j)$ is the K-nearest neighborhood of $\mathbf{x}_j$ and $\sigma_{\mathbf{x}}>0$ is a hyper-parameter. Then, we diffuse the labels Y on G. Specifically, our label propagation algorithm adopts the approach of Zhou *et al.* [34]: For a c-class classification problem, each label $\mathbf{y}_k\in Y$ is given as a c-dimensional row vector. When the ground-truth class of $\mathbf{x}_j$ is k, the elements of $\mathbf{y}_j$ are all zero except for the k-th element that is assigned with one: $\mathbf{y}_j=[0,...,1,...,0]$. The label propagation is then performed by building the initial $f^0\in\mathbb{R}^{n\times c}$ where i-th row is $\mathbf{y}_i$ if $\mathbf{x}_i$ is labeled ($i\le l$) and 0, otherwise, and running the difference equation (explicit Euler scheme; Eq. 12) until the stopping criteria is met: As suggested by the form of regularizer $\mathcal{R}_{L^D}$, similarly to the isotropic graph Laplacian, the only null-space of anisotropic graph Laplacian is the space of constant functions. This implies that the difference equation (Eq. 12) converges to a constant function as $t\to\infty$. Accordingly, for practical applications, we stop diffusion at a finite time step T and obtain the resulting function f^T as the output. The final class label for data point $\mathbf{x}_i$ is obtained as $\operatorname{argmax} f^T(i)\in\mathbb{R}^c$ for each i.

The best choice for the eigenvalues $\{q_{ij}\}$ of the diffusivity operator D depends on the application. Intuitively, the diffusivity q_{ij} should be high when the corresponding function evaluations $f(i)$ and $f(j)$ are similar, i.e., $|\nabla_i f(e_{ij})|$ is small. One way to define such diffusivity is to use a Gaussian weight function as is common in image enhancement:

$$q_{ij} = \exp\left(-\frac{|\nabla_i f(e_{ij})|^2}{\sigma_f^2}\right), \quad (15)$$

where σ_f^2 is the scale hyper-parameter. Algorithm 1 shows pseudocode to construct the corresponding anisotropic graph Laplacian on G.

The resulting anisotropic graph Laplacian L^D can be immediately applied to any label-propagation problems. However,

Algorithm 1: Build anisotropic graph Laplacian L^D.

Input: Set of data points $X=\{\mathbf{x}_1,...,\mathbf{x}_n\}\subset\mathbb{R}^d$ with function values: $F=\{f(\mathbf{x}_1),...,f(\mathbf{x}_n)\}\subset\mathbb{R}^c$.
Output: L^D.

for $i=1,...,n$ **do**
  Find nearest neighbors $N_K(\mathbf{x}_i)$;
  Calculate isotropic weights
  w_{ij} (for $\mathbf{x}_j\in N_K(\mathbf{x}_i)$ and $\mathbf{x}_i\in N_K(\mathbf{x}_j)$; Eq. 14);
  Calculate the node degree d_i (Eq. 3);
  Calculate the diffusivity
  eigenvalues q_{ij} using one of Eqs. 15, 16, and 17;
end
Rearrange $\{w_{ij}^D\}$ (Eq. 11) to a matrix L^D based on Eq. 10.

for semi-supervised learning algorithm, naïvely applying L^D to the difference equation (12) may require many iterations before it actually starts propagating labels. The progress of diffusion can be very slow in the early stage (t is small) at the vicinity of labeled points: If a point $\mathbf{x}_i$ is labeled and $N_K(\mathbf{x}_i)\backslash\mathbf{x}_i$ are all unlabeled (this is typically the case for semi-supervised learning), the corresponding eigenvalues (Eq. 15) are all small, and accordingly, the weights $\{w_{ij}^D\}$ are also small for all $\mathbf{x}_j \in N_K(\mathbf{x}_i)$. To speed up the process, we run the isotropic diffusion (with the isotropic graph Laplacian L) and *smooth* out the initial distribution of f^0. For all experiments, the initial diffusion runs for 20 time steps while the length T of the anisotropic diffusion is regarded as a hyper-parameter.

Discussion. Our derivation of anisotropic graph Laplacian is strongly connected to the kernel-based anisotropic diffusion approach of Szlam *et al.* [29], yet the motivating ideas are different: their anisotropic kernel is based on stochastic Markov diffusion processes on graphs, while our anisotropic graph Laplacian is obtained based on a formulation of geometric diffusion on manifolds: L^D is obtained by extending Weickert's diffusivity operators in $\mathbb{R}^2$ [32] to M and then discretizing it onto a graph G (see Sec. 3).

Since the kernel smoothing corresponds to calculating analytic solution at each time step of diffusion, and our anisotropic weights $\{w_{ij}^D\}$ used in constructing L^D can be regarded as an instance of such kernels, the final diffusion algorithms of Szlam *et al.* [29] and ours are very similar when applied to linear diffusion: Kernel smoothing is given by first obtaining the continuous Gaussian smoothing as an analytical solution of the linear diffusion equation, and then discretizing it, while our explicit Euler scheme is obtained by directly discretizing both the manifold and the Laplace-Beltrami operator. In preliminary linear diffusion experiments, minor differences in weights normalization[1] led to only negligible differences in semi-supervised learning performances.

The major differences between the two diffusion algorithms are that 1) our algorithm is nonlinear, i.e. L^D depends on f^t at each time t, while the anisotropic kernel of [29] is obtained as an analytic solution of linear diffusion equation and therefore is fixed a priori to the entire diffusion process. In our experiments, we demonstrate that extending the approach of Szlam *et al.* [29] to non-linear diffusion already significantly improves semi-supervised learning performance. Furthermore, unlike Szlam, 2) our construction explicitly states sufficient conditions ($\{q_{ij}\}$ are symmetric and positive) for the well-posedness of the resulting diffusion on G as a discretization of the underlying manifold. This enables exploring various possibilities of inducing new diffusion on G.

2.1. Context-guided diffusion.

We have seen how defining positive eigenvalues $\{q_{ij}\}$ leads to a PD diffusivity operator D and to the corresponding anisotropic graph Laplacian L^D. This can be regarded as updating the similarity measure between data points in $X \subset \mathbb{R}^d$: The isotropic graph Laplacian matrix L is constructed from the positive weights $\{w_{ij}\}$ which are the pair-wise similarities of data points measured by the original Euclidean metric of $\mathbb{R}^d$ (see Eq. 14). By construction, the information in L is precisely the same as the pair-wise similarities and, therefore, defining a graph Laplacian L corresponds to defining a similarity measure. Now, defining the anisotropic diffusivity operator L^D, which is constructed based on the original similarity measure plus the eigenvalues $\{q_{ij}\}$, can be interpreted as introducing a new similarity measure $\{w_{ij}^D\}$ on G.[2]

In particular, we have seen how the Gaussian function (Eq. 15) measures the deviation between the two function evaluations $f(i)$ and $f(j)$ as each edge e_{ij}. This is only an example and there are various possibilities given the positivity constraint. Furthermore, q_{ij} does not have to defend only based on $f(i)$ and $f(j)$ and it can take into account the neighborhood context as well. For instance, spatially smoothing the diffusivity operator, e.g., by convolving it with a Gaussian kernel, leads to much more stable image enhancement than using the original diffusivity operators (which is commonly constructed based on gradient vectors): Theoretically, the smoothing operation guarantees the well-posedness of the resulting diffusion equation even when the corresponding original version is not. From a practical perspective, this operation offers robustness against noise in the image f since the gross effect of smoothing the diffusivity is to take the spatial averaging of the gradients of f [32].

The spatial smoothing of the diffusivity operator can be regarded as an instance of controlling the diffusivity based on *local context*. We investigate two possibilities of exploiting this local context. The first case is to adapt the idea of Gaussian smoothing on images to graphs: For a given edge e_{ij} and the corresponding local neighborhoods at each end node, $N_K(\mathbf{x}_i)$ and $N_K(\mathbf{x}_j)$, the *smooth diffusivity* w_{ij}^D is obtained based on weighted averages of the diffusivities in the mutual neighborhood $N_M(x_i,x_j) := N_K(x_i) \cap N_K(x_j)$.

$$w_{ij}^D = \sum_{x_k \in N_M(x_i,x_j)} w_{ij}(q_{ij} + q_{ik}q_{kj})/(s_i^q + s_j^q), \qquad (16)$$

where $s_i^q = \sum_{x_k \in N_K(x_i)} q_{ik}$ and $s_j^q = \sum_{x_k \in N_K(x_j)} q_{kj}$. The interpretation of our smooth diffusivity is straightforwardly transferred from the smooth diffusivity operators in the image domain: The resulting diffusion process is robust against noise in edge weights.

Another example of exploiting the context is to adopt the intuitive notion of matching between the two entities in context: If a pair of objects $\mathbf{x}_i$ and $\mathbf{x}_j$ matches, then often spatial neighbors of $\mathbf{x}_i$, $\mathbf{x}_l \in N_K(\mathbf{x}_i)$ have the corresponding matching elements in their neighborhoods $N_K(\mathbf{x}_j)$ of $\mathbf{x}_j$, i.e., the match of $(\mathbf{x}_i,\mathbf{x}_j)$ is *supported* if the neighborhoods of $N_K(\mathbf{x}_i)$ and $N_K(\mathbf{x}_j)$ find matches in each pair of elements. Our *local match diffusivity*

[1] In L^D, the normalization coefficients $\{d_i\}$ are constructed from $\{w_{ij}\}$ (see Eq. 10), while the diffusion kernel in Szlam *et al.* [29] is normalized so that it leads to a stochastic matrix.

[2] This intuition holds rigorously on the Laplace-Beltrami operator Δ on a Riemannian manifold M: 1) Indeed, Δ uniquely defines a Riemannian metric g on M [27] and 2) Section 3 shows that defining a diffusivity operator $\mathcal{D}$ on M corresponds to defining the corresponding new metric $\overline{g}$.

is defined as a smooth version of considering this match context:

$$w_{ij}^{D} = w_{ij} q_{ij} \sum_{\mathbf{x}_k \in N_K(\mathbf{x}_i)} (1+q_{ik}^{*})/(k+1), \tag{17}$$

where $q_{ik}^{*} = \max_{\mathbf{x}_l \in N_K(\mathbf{x}_j)} q_{kl}$. The max in the definition of q_{ik}^{*} implies that if there's any entity in $N_K(\mathbf{x}_j)$ that matches $\mathbf{x}_k$, the corresponding diffusivity between $\mathbf{x}_i$ and $\mathbf{x}_j$ is supported. The normalization factor $k+1$ is actually obtained as $k+1$ times the maximum possible value of q_{ij} (which corresponds to the *match* case) which is 1 (Eq. 15).

3. Connection to continuous operators

As we have seen in Sec. 2.1, our anisotropic diffusion process on $G=(X,E,W)$ is nothing more than isotropic diffusion on a new graph (X,E,W^D) (regularization-form definition of L^D in Eq. 13, and corresponding diffusion process in Eq. 12) — our (discrete) diffusivity operator D (Eq. 9) changes the notion of similarity. In this section, first, we show that this intuition applies to the continuous limit case of Laplace-Beltrami operator Δ on a data generating manifold M, i.e., anisotropic diffusion on M is isotropic diffusion with a new metric. Then, we discuss the convergence properties of our anisotropic graph Laplacian to the continuous anisotropic Laplace-Beltrami operator.

Anisotropic diffusion on Riemannian manifolds. On a Riemannian manifold (M,g) with g being a Riemannian metric on M, the isotropic diffusion of a smooth function $f \in C^{\infty}(M)$ is described as a partial differential equation:

$$\frac{\partial f}{\partial t} = \nabla^{g*} \nabla^{g} f = -\Delta^{g} f, \tag{18}$$

where $\nabla^{g} f$ is the gradient of f, ∇^{g*} is the formal adjoint of ∇^{g}, and Δ^{g} is the Laplace-Beltrami operator defined by $\Delta^{g} = -\nabla^{g*}\nabla^{g}$.

If we extend Weickert's diffusivity operator originally definied on $\mathbb{R}^2$ [32] to a manifold M, then we introduce a smooth positive definite operator $\mathcal{D} : \mathcal{T}(M) \to \mathcal{T}(M)$ with $\mathcal{T}(M)$ being the tangent bundle of M, i.e., $\mathcal{D}$ is a smooth field of symmetric positive definite operators each defined on a tangent space $T_{\mathbf{x}}(M) \in \mathcal{T}(M)$ at $\mathbf{x} \in M$. The corresponding anisotropic diffusion process is given as:

$$\frac{\partial f}{\partial t} = \nabla^{g*} \mathcal{D} \nabla^{g} f, \tag{19}$$

Defining an anisotropic Laplacian operator $\Delta_{\mathcal{D}}^{g} = \nabla^{g*}\mathcal{D}\nabla^{g}$, we restate Eq. 19 similarly to the isotropic case:

$$\frac{\partial f}{\partial t} = -\Delta_{\mathcal{D}} f. \tag{20}$$

We show that our anisotropic diffusion (Eq. 20) boils down to isotropic diffusion on M with a new metric $\overline{g}$:

Proposition 1 (The equivalence of $\Delta_{\mathcal{D}}$ and $\Delta_{\overline{g}}$)**.** *The anisotropic Laplacian operator $\Delta_{\mathcal{D}}$ on a compact Riemannian manifold (M,g) is equivalent to the Laplace-Beltrami operator $\Delta_{\overline{g}}$ on $(M,\overline{g})$ with a new metric $\overline{g}$ depending on $\mathcal{D}$. Specifically, when the diffusivity operator $\mathcal{D}$ is uniformly positive definite, $\overline{g}$ is explicitly obtained as $c(\mathbf{x})\overline{\mathbf{g}}(\mathbf{x}) = \mathbf{g}(\mathbf{x})\mathbf{D}^{-1}(\mathbf{x})$, where $\mathbf{g}(\mathbf{x})$, $\overline{\mathbf{g}}(\mathbf{x})$, and $\mathbf{D}(\mathbf{x})$ are the coordinate representations (matrices) of g, $\overline{g}$, and $\mathcal{D}$ at each point $\mathbf{x}$, and $c(\mathbf{x}) = \frac{\sqrt{\det \mathbf{g}(\mathbf{x})}}{\sqrt{\det \overline{\mathbf{g}}(\mathbf{x})}}$ which is a smooth function on M.*

Proof. The proof is obtained by applying the techniques developed for analyzing maps between general *weighted manifolds* [14]. For any function $f,h \in C^{\infty}(M)$, we have:

$$\begin{aligned} \int f \Delta_{\mathcal{D}} h dV &= \int f \nabla^{g*} \mathcal{D} \nabla^{g} h dV \\ &= -\int \langle \nabla^{g} f, \mathcal{D} \nabla^{g} h \rangle_{g} dV \\ &= -\int df(\mathcal{D}\nabla^{g} h) dV, \end{aligned} \tag{21}$$

where dV is the *natural volume element* [23] corresponding to g ($dV = \sqrt{\det g} d\mathbf{x}$) and the second equality is obtained by applying the divergence theorem on (M,g). The third equality corresponds to the definition of gradient ∇^{g} based on the differential operator d [23]. Applying Green's theorem to $(M,\overline{g})$, we obtain:

$$\begin{aligned} \int f \Delta_{\overline{g}} h d\overline{V} &= -\int \langle \nabla^{\overline{g}} f, \nabla^{\overline{g}} h \rangle_{\overline{g}} d\overline{V} \\ &= -\int \left\langle \nabla^{\overline{g}} f, \frac{\sqrt{\det \overline{g}}}{\sqrt{\det g}} \nabla^{\overline{g}} h \right\rangle_{\overline{g}} dV \\ &= -\int df \left(\frac{\sqrt{\det \overline{g}}}{\sqrt{\det g}} \nabla^{\overline{g}} h \right) dV. \end{aligned} \tag{22}$$

Now, identifying the two integrals, and using $\nabla^{g} h = g^{-1} dh$ and $\nabla^{\overline{g}} h = \overline{g}^{-1} dh$, we obtain

$$\frac{\sqrt{\det \overline{\mathbf{g}}(\mathbf{x})}}{\sqrt{\det \mathbf{g}(\mathbf{x})}} \overline{\mathbf{g}}^{-1}(\mathbf{x}) = \mathbf{D}(\mathbf{x}) \mathbf{g}^{-1}(\mathbf{x}) \tag{23}$$

$$\therefore \quad c(\mathbf{x}) \overline{\mathbf{g}}(\mathbf{x}) = \mathbf{g}(\mathbf{x}) \mathbf{D}^{-1}(\mathbf{x}). \tag{24}$$

□

It is always possible to find a coordinate representation of the Riemannian metric g at each point $\mathbf{x} \in M$ such that it becomes Euclidean (up to second order) [19]. This implies that, up to scale,[3] the metric $\overline{g}(\mathbf{x})$ in Eq. 24 boils down to well-established Mahalanobis distance, with $\mathbf{D}(\mathbf{x})$ being the corresponding covariance matrix in $T_{\mathbf{x}}(M)$. This greatly helps to understand of the anisotropic diffusion process. For any PD diffusivity operator $\mathcal{D}$, there is a corresponding isotropic Laplace-Beltrami operator $\Delta_{\overline{g}}$ on $(M,\overline{g})$. If we discretize in time the differential equation of the isotropic diffusion process (Eq. 18) on $(M,\overline{g})$ (see [18] for derivation):

$$\frac{f^{t+\delta} - f^{t}}{\delta} = -\Delta_{\overline{g}} f^{t+\delta}, \tag{25}$$

[3] Note that the ratio $\frac{\sqrt{\det \overline{g}}}{\sqrt{\det g}}$ is coordinate independent.

then the solution $f^{t+\delta}$ at time $t+\delta$, is obtained as the minimizer of the following regularization energy:[4]

$$\mathcal{E}(f) = \|f - f^t\|^2 + \delta \int \|\nabla^{\overline{g}} f\|_{\overline{g}} d\overline{V}, \tag{26}$$

which is now equivalent to:

$$\mathcal{E}(f) = \|f - f^t\|^2 + \delta \int c\left\langle \nabla^g f, \mathbf{D}^{-1} \nabla^g f \right\rangle_g dV. \tag{27}$$

Accordingly, the anisotropic diffusion process (Eq. 19) can be regarded as continuously solving a regularized regression problem where the regularizer penalizes at each point $\mathbf{x}$, the first-order deviation heavily along the direction where the *covariance matrix* $\mathbf{D}(\mathbf{x})$ is less spread, i.e. the corresponding diffusivity is weak along that direction.

This perspective provides a connection to the problem of inducing anisotropic diffusion as a special instance of metric learning on Riemannian manifolds and, as the corresponding discretization, learning a graph structure from data. See [2] for an example of data-driven graph construction which relies on the known dimensionality of the underlying manifold.

On the convergence of L^D to $\Delta^{\mathcal{D}}$. It is well known that when data points X are generated from an underlying Riemannian manifold M embedded in an ambient Euclidean space, the isotropic graph Laplacian L on $G = (X, E, W)$ converges to the Laplace-Beltrami operator Δ on M as $n \to \infty$, with the neighborhood size $K \to \infty$ controlled accordingly [1, 17]. However, despite its strong connection to the (continuous) anisotropic Laplacian $\Delta^{\mathcal{D}}$ on M, our discrete anisotropic graph Laplacian L^D is not by itself, *consistent*, i.e. it does not converge to $\Delta^{\mathcal{D}}$ as $n \to \infty$. This is because, by design, our diffusivity operator is agnostic to the dimensionality m of the manifold M. To elaborate this further, note that given fixed n-data points X and the corresponding local neighborhood size K, our local diffusivity operator D_i at $\mathbf{x}_i$ (Eq. 9) defines a (new) inner-product in $H(E_i)$:

$$\begin{aligned} &D_i : H(E_i) \to H(E_i) \\ \Rightarrow &\langle \cdot, D_i \cdot \rangle_{H(E_i)} : H(E_i) \times H(E_i) \to \mathbb{R}. \end{aligned} \tag{28}$$

The convergence of L^D to $\Delta^{\mathcal{D}}$ requires a certain form[5] of convergence of D^i to $\mathcal{D}(\mathbf{x}_i)$ at each $\mathbf{x}_i$. In particular, the continuum limit D_i^∞ (as $n \to \infty$) of D_i should induce an inner-product on $T_{\mathbf{x}_i}$. However, in general, D_i^∞ cannot induce any inner product since D_i^∞ has infinite degrees of freedom (i.e., D_i^∞ has infinitely many parameters): D_i has $K(n)$-eigenvalues and $K(n) \to \infty$ as $n \to \infty$. Actually, for a given fixed n with corresponding G_i, D_i can be defined as the restriction of D_i^∞ on E_i. On the other hand, the continuous diffusivity operator $\mathcal{D}$ on $T_{\mathbf{x}_i}$ has only up to $\frac{m(m+1)}{2}$-degrees of freedom with m being the dimensionality of M. This implies that D_i^∞ cannot be a bi-linear operator on $T_{\mathbf{x}_i}$. Actually, this is the only property that prevents D_i^∞ being an inner-product: By construction, the limit of D_i is non-negative and positive definite.

The relation between $\mathcal{D}(\mathbf{x}_i)$ and D_i is exactly the same as the relationship between the inner-product in the Euclidean space $\mathbb{R}^m$ and a nonlinear positive definite kernel $k : \mathbb{R}^m \times \mathbb{R}^m \to \mathbb{R}$ as commonly used in kernel machines: k induces a similarity measure on $\mathbb{R}^m$. However, in general, it is not bi-linear and therefore it does not corresponds to an inner-product. Instead, k induces an inner-product in a (potentially infinite-dimensional) feature space $\mathcal{K}$ which is mapped by a nonlinear function $\phi : \mathbb{R}^m \to \mathcal{K}$.

This insight leads to an algorithm to build *consistent* local graph diffusivity operators $\{D_i^{\mathcal{C}}\}$ (and the corresponding global operator $D^{\mathcal{C}}$) by reducing the degree of freedom of each D_i from $K(n)$ to $\frac{m(m+1)}{2}$. In the accompanying supplemental material, we show how $\{D_i^{\mathcal{C}}\}$ can be explicitly constructed and it converges to $\mathcal{D}$.

Discussion. While the consistent diffusivity operators might be of theoretical interest and may deserve further analysis, in this paper we focus on using the inconsistent diffusivity operator D (Eq. 9). This design choice is made based on two facts: 1) In general, estimating the dimensionality m of a manifold M and the corresponding tangent bundle $\mathcal{T}(M)$ based on a finite sample $X \subset M$ are difficult problems [20]. Therefore, existing approaches that involve estimating m make it a hyper-parameter. Optimizing *many* hyper-parameters is a difficult problem in semi-supervised learning due to the limited number of labeled points. 2) More importantly, some semi-supervised learning problems are inherently formulated as an inference on a graph G that may not have any explicit connection to a manifold M or the corresponding ambient space. For instance, if each node $\mathbf{x}_i \in X$ represents an image, and if each edge $e_{ij} \in E$ and corresponding weight $w_{ij} \in W$ represents the possibility of match and match score between $\mathbf{x}_i$ and $\mathbf{x}_j$, respectively, then there is no natural manifold or ambient space structure defined on X. Accordingly, our algorithm is obtained as a design choice that favors general applicability over theoretical consistency.

Lastly, we would like to add that it is tempting to build a consistency argument based on the fact that any graph with positive weights can be embedded into a manifold M with a sufficiently high-dimensionality m, and therefore any data X and the corresponding PD graph diffusivity operator D can be regarded as a sample from such a manifold M and the operators on $\mathcal{T}(M)$, respectively. Unfortunately, this does not lead to a useful interpretation.

4. Experiments

We evaluate our anisotropic diffusion algorithm in classification on seven standard semi-supervised learning datasets [15, 36, 4] and four object recognition datasets for which semi-supervised learning has been successful in the literature in retrieval contexts. We report performance for isotropic diffusion and the original kernel smoothing-type anisotropic diffusion approach of Szlam *et al.* [29]. We also report the performances of three existing semi-supervised learning algorithms including Zhu *et al.*'s

[4] This applies even when Eq. 25 is nonlinear, i.e. $\mathcal{D}$ depends on f.

[5] Although $X \to M$ and $L \to \Delta$, the convergence of $H(E)$ to $\mathcal{T}(M)$ cannot be uniquely defined (see [16] for details) and therefore the convergence of L^D (which depends on $D : H(E) \to H(E)$) to $\Delta^{\mathcal{D}}$ is also not uniquely defined.

Algorithm	USPS	BCI	MNIST	COIL1	COIL2	RealSim	Pcmac	MPEG7	SWDLEAF	ETH-80	C-PASCAL	Avg. %
I	8.76	41.60	10.65	7.32	4.37	23.61	11.77	3.36	2.39	11.49	54.54	148.1
A_{lin} [29]	5.55	41.80	8.47	7.36	4.11	25.02	12.58	3.01	2.54	11.30	54.47	137.0
A_{nlin}	4.48	**39.53**	7.62	6.85	2.98	23.46	11.88	2.63	2.47	**9.91**	52.22	120.8
A_{LM}	4.31	42.00	7.55	6.48	2.22	19.55	11.47	2.54	**2.17**	10.05	**51.19**	111.7
A_S	**3.93**	42.13	**7.18**	6.21	2.13	20.08	**11.34**	2.59	2.33	10.01	51.30	**110.5**
GRF [37]	6.13	42.68	10.96	**4.93**	**1.65**	28.09	11.78	2.96	2.76	12.16	61.91	127.6
FLAP [11]	5.66	44.63	10.99	6.97	2.73	20.08	14.49	**2.16**	2.84	12.59	57.97	131.0
LNP [31]	7.27	44.33	13.25	5.53	3.12	**16.02**	14.39	N/A	N/A	11.94	62.36	139.1

Table 1. Performance of different diffusion algorithms for semi-supervised learning: The three best results for each dataset are marked with boldface blue, plain green, and plain orange fonts, respectively. LNP [31] requires explicitly calculating the Euclidean distances between data points, and so it cannot be directly applied to *MPEG7* and *SWDLEAF* data sets. The final Avg. % column shows the mean percentage difference from the best result across all datasets, where 100% would indicate that particular technique was best across all datasets.

Gaussian random fields (*GRFs*)-based algorithm [37], Gong and Tao's label propagation algorithm (*FLAP*: Fick's Law Assisted Propagation, [11]) inspired by Fick's first law which describes the diffusion process at a steady state, and Wang and Zhang's [31] linear neighborhood propagation (*LNP*) algorithm which automatically determines the edge weights $\{w_{ij}\}$ by representing each input point based on a convex combination of its neighbors [31].

Datasets. The *MPEG7* shape dataset [22] consists of 1,400 images which show silhouettes of objects from 70 different categories. Adopting the experimental setting for data retrieval experiments [6], with 280 labels, we use shape matching [12] to infer pairwise distances from which the (isotropic) weight matrix W is constructed. In this dataset, each data point $\mathbf{x}$ in X is not explicitly presented and so the data generating manifold is not explicitly considered. Our algorithm is applicable even in this case, which justifies the use of the inconsistent diffusivity operator.[6]

The *ETH-80* dataset consists of 3,280 photographs of objects from 8 different classes [24]. The *C-PASCAL* dataset (as a subset of the PASCAL VOC challenge 2008 data, where single objects are extracted based on bounding box annotations) contains 4,450 images of 20 classes [9]. For both ETH-80 and C-PASCAL datasets, each data point is represented based on the HOG (histogram of oriented gradients) descriptors and the number of labels are set to 50 [8]. The *SWDLEAF* (Swedish leaf) datasets contains 15 different tree species with 75 leaves per species [28]. For this dataset, we use 50 labels per class, with Fourier descriptors to represent each entry [26].

Results. In Table 4, I refers to isotropic diffusion, A_{lin} is the algorithm of Szlam *et al.* [29]. A_{nlin} is an extension of [29] to nonlinear diffusion based on our diffusion approach (see Sec. 2) while A_{LM} and A_S are *local match* and *smooth* anisotropic diffusion, respectively.

Overall, all four anisotropic diffusion algorithms significantly improve classification accuracies over isotropic diffusion (I). However, for some datasets (*SWDLEAF*, *RealSim*, *Pcmac*), the performance of linear anisotropic diffusion (A_{lin}) [29] is equal to or even worse than I. In contrast, all three nonlinear diffusion algorithms outperformed both I and A_{lin}, while the *local match* (A_{LM}) and *smooth* (A_S) versions of the context-guided diffusion led to further improvement over A_{nlin} in all but the *ETH* and *BCI* datasets. These results are in accordance with the superior performance of the smooth diffusivity operators (which is an example of exploiting context) in image processing and demonstrate the effectiveness of exploiting context information in anisotropic diffusion on graphs. For the *BCI* dataset, A_{nlin} and A_S showed the best and the worst performances, while essentially all four anisotropic diffusion algorithms did not show any noticeable improvement from the isotropic case. This is because the initial labeling based on isotropic diffusion is almost random (around 40% error rate for binary classification), and so this is a poor initialization for an anisotropic diffusion and does not lead to better label propagation. Similar observation were reported in [29]. The anisotropic diffusion algorithms also demonstrated their competence in comparison with state-of-the-art label-propagation algorithms [37, 11, 31]: *GRF* is best on *COIL1* and *COIL2*, and *FLAP* and *LNP* are the best for *MPEG7* and *RealSim*. However, except for few cases, the results of A_{nlin} and A_S are included in the three best results for each dataset demonstrating the overall steady performance improvements over existing algorithms. Lastly, all three algorithms are designed for data graphs constructed based on input features rather than from function evaluations. Therefore, they can potentially benefit from our proposed anisotropic diffusion approaches.

Parameters. Isotropic diffusion has three parameters: the weight $\sigma_{\mathbf{x}}$ (Eq. 14), the size of local neighborhood N_K, and the number of diffusion steps T. We automatically determine $\sigma_{\mathbf{x}}$ based on the average Euclidean distance of $\mathbf{x}_j$ to $N_K(\mathbf{x}_j)$ [29, 18]. We determine the two other parameters with a separate validation label set which is the same size as the training label set.

For all anisotropic diffusion algorithms, an additional hyper-parameter σ_f^2 (Eq. 15) is determined in the same way. The step size δ of the explicit Euler approximation in our algorithms (Eq. 12) is fixed at 1. In general, δ can also be tuned per dataset to improve performance. *GRF*, *FLAP*, and *LNP* hyper-parameters are all determined in the same way based on the validation set.

Computational complexity. This depends upon the number n of data points, the size N_K of the local neighborhood, and the number of diffusion process iterations (Eq. 12). Each diffusion iteration requires multiplying the matrix L^D of size $n \times n$

[6]For consistent diffusivity operators, we would have to explicitly estimate the dimensionality of the data manifold; see Sec. 3.

with a vector f of size $n \times c$, where c is the number of classes. Accordingly, in theory, the complexity of each step is $O(n^2 c)$. However, typically $N_K \ll n$, which leads to a sparse matrix L^D: in practice, the computational complexity of each step is sub-quadratic. For USPS datasets with 1,500 data points, running 100 iterations of the *local match* diffusion process A_{LM} takes ≈0.3 seconds on an Intel Xeon 3.4GHz CPU in MATLAB.

5. Discussion and conclusion

We show two ways to exploit local contexts: *smooth* and *local match*. These can be extended to consider the full topological features of f evaluated at E_i and E_j. For instance, one could perform spectral analysis on W_i^D and W_j^D and measure the similarity of the corresponding Eigenspectra to define a new diffusivity operator D'. This is different from pre-calculating topological features, as is commonly used in graph matching, since features are extracted from the input X rather than from function evaluations f, and therefore the former stay constant during the diffusion process. We briefly explored this possibility in preliminary experiments, which indicate that full topological analysis is promising. However, due to the significantly increased computational complexity, we focus on *smooth* and *local match* operators and leave this extension for future work.

We adopted an explicit Euler scheme (Eq. 12) to discretize the continuous diffusion equation (Eq. 20). This scheme can be obtained as a gradient descent step of the convex regularization functional $\mathcal{E}$ (Eq. 27). An alternative implicit Euler scheme (Eq. 25) can be obtained as the analytic solution of $\mathcal{E}$. Since our diffusion equation (Eq. 20) is non-linear, both approaches eventually lead to iterative algorithms. A major advantage of an implicit Euler scheme is that it is uniformly stable with respect to δ, while our explicit Euler scheme is stable only at sufficiently small values of δ, which we regard as a hyper-parameter. On the other hand, implicit Euler approximation is computationally less favorable as it requires, at each iteration, explicitly solving a (sparse) linear system of size $n \times n$. Our explicit counterpart is computed by a matrix-vector multiplication. We choose the explicit scheme due to its fast convergence in experiments and its applicability to large-scale problems. Future work should carefully analyze the trade-off between these two approaches, especially on smaller-scale problems.

For simplicity of exposition, in Sec. 3, we assumed that the underlying probability distribution P on M is uniform. However, our interpretation applies to more general cases where P is non-uniform. If the sampling distribution P on M is non-uniform, the isotropic Laplace-Beltrami operator is locally weighted by the corresponding probability density p, rendering the *weighted Laplacian*. In particular, if p is differentiable, the weighted Laplacian is explicitly given as [17, 14]:

$$\Delta^p = \frac{1}{p} \nabla^{g*} (p \nabla^g). \tag{29}$$

The weighted Laplacian satisfies Green's theorem, and the divergence theorem holds similarly [13]. Accordingly, the corresponding weighted anisotropic Laplacian based on the diffusivity operator $\mathcal{D}$ is obtained as in Proposition 1.

Conclusion. We have presented an approach for anisotropic diffusion on graphs, by first extending well-established geometric diffusion on images to Riemannian manifolds and then discretizing it onto graphs. The resulting positive definite diffusivity operators on graphs leads to new diffusion possibilities that take local neighborhood structures into account, and thereby lead to robust diffusion. Applied to semi-supervised learning, our algorithms demonstrate improved accuracy over existing isotropic diffusion- and anisotropic diffusion-based algorithms.

Acknowledgements

The authors thank the reviewers for their constructive feedback. Kwang In Kim thanks EPSRC EP/M00533X/1. James Tompkin and Hanspeter Pfister thank NSF CGV-1110955, the Air Force Research Laboratory, and the DARPA Memex program. Christian Theobalt thanks the Intel Visual Computing Institute.

References

[1] M. Belkin and P. Niyogi. Towards a theoretical foundation for Laplacian-based manifold methods. *Journal of Computer and System Sciences*, 74(8):1289–1308, 2005. 2, 6

[2] C. Carey and S. Mahadevan. Manifold spanning graphs. In *Proc. AAAI*, pages 1708–1714, 2014. 6

[3] W. Casaca, L. G. Nonato, and G. Taubin. Laplacian coordinates for seeded image segmentation. In *Proc. IEEE CVPR*, pages 384–391, 2014. 1

[4] O. Chapelle, B. Schölkopf, and A. Zien. *Semi-Supervised Learning*. MIT Press, Cambridge, MA, 2010. 6

[5] R. R. Coifman and S. Lafon. Diffusion maps. *Applied and Computational Harmonic Analysis*, 21(1):5–30, 2006. 1

[6] M. Donoser and H. Bischof. Diffusion processes for retrieval revisited. In *Proc. IEEE CVPR*, pages 1320–1327, 2013. 7

[7] S. Ebert, M. Fritz, and B. Schiele. Active metric learning for object recognition. In *Proc. DAGM-OAGM*, pages 327–336, 2012. 1

[8] S. Ebert, M. Fritz, and B. Schiele. RALF: a reinforced active learning formulation for object class recognition. In *Proc. IEEE CVPR*, pages 3626–3633, 2012. 7

[9] S. Ebert, D. Larlus, and B. Schiele. Extracting structures in image collections for object recognition. In *Proc. ECCV*, pages 720–733, 2010. 7

[10] E. Elboer, M. Werman, and Y. Hel-Or. The generalized Laplacian distance and its applications for visual matching. In *Proc. IEEE CVPR*, pages 2315–2322, 2013. 1

[11] C. Gong and D. Tao. Fick's law assisted propagation for semisupervised learning. *IEEE T-NNLS*, PP(99):1,1, 2014 (Early Access). 2, 7

[12] R. Gopalan, P. Turaga, and R. Chellappa. Diffusion processes for retrieval revisited. In *Proc. ECCV*, pages 286–299, 2010. 1, 7

[13] A. Grigor´yan. Heat kernels on weighted manifolds and applications. In J. Jorgenson and L. Walling, editors, *The Ubiquitous Heat Kernel*, Contemporary Mathematics. American Mathematical Society, 2006. 8

[14] A. Grigor´yan. *Heat Kernel and Analysis on Manifolds (AMS/IP Studies in Advanced Mathematics)*. American Mathematical Society, 2013. 5, 8

[15] Y. Guo and X. N. H. Zhang. An extensive empirical study on semi-supervised learning. In *Proc. ICDM*, pages 186–195, 2010. 6

[16] M. Hein. *Geometrical Aspects of Statistical Learning Theory*. PhD thesis, Fachbereich Informatik, Technische Universität Darmstadt, Germany, 2005. 6

[17] M. Hein, J.-Y. Audibert, and U. von Luxburg. From graphs to manifolds - weak and strong pointwise consistency of graph Laplacians. In *Proc. COLT*, pages 470–485, 2005. 2, 6, 8

[18] M. Hein and M. Maier. Manifold denoising. In *NIPS*, pages 561–568, 2007. 5, 7

[19] J. Jost. *Riemannian Geometry and Geometric Analysis*. Springer, New York, 6th edition, 2011. 5

[20] B. Kégl. Intrinsic dimension estimation using packing numbers. In *NIPS*, pages 681–688, 2002. 6

[21] K. I. Kim, J. Tompkin, and C. Theobalt. Curvature-aware regularization on Riemannian submanifolds. In *Proc. IEEE ICCV*, pages 881–888, 2013. 1

[22] L. J. Latecki, R. Lakämper, and U. Eckhardt. Shape descriptors for non-rigid shapes with a single closed contour. In *Proc. IEEE CVPR*, pages 424–429, 2000. 7

[23] J. M. Lee. *Riemannian Manifolds- An Introduction to Curvature*. Springer, New York, 1997. 5

[24] B. Leibe and B. Schiele. Analyzing appearance and contour based methods for object categorization. In *Proc. IEEE CVPR*, 2003. 7

[25] D. Li, Q. Chen, and C.-K. Tang. Motion-aware knn Laplacian for video matting. In *Proc. IEEE ICCV*, pages 3599–3606, 2013. 1

[26] H. Ling and D. W. Jacobs. Shape classification using the inner-distance. *IEEE T-PAMI*, 29(2):286–299, 2007. 7

[27] S. Rosenberg. *The Laplacian on a Riemannian Manifold*. Cambridge University Press, 1997. 4

[28] O. Söderkvist. *Computer Vision Classification of Leaves from Swedish Trees*. PhD thesis, Master thesis, Linköping University, Sweden, 2001. 7

[29] A. D. Szlam, M. Maggioni, and R. R. Coifman. Regularization on graphs with function-adapted diffusion processes. *JMLR*, 9:1711–1739, 2008. 1, 2, 4, 6, 7

[30] B. Wang, Z. Tu, and J. K. Tsotsos. Dynamic label propagation for semi-supervised multi-class multi-label classification. In *Proc. IEEE ICCV*, pages 425–432, 2013. 1

[31] F. Wang and C. Zhang. Label propagation through linear neighborhoods. In *Proc. ICML*, pages 985–992, 2006. 2, 7

[32] J. Weickert. *Anisotropic Diffusion in Image Processing*. ECMI Series, Teubner-Verlag, Stuttgart, 1998. 1, 2, 3, 4, 5

[33] R. Wu, Y. Yu, and W. Wang. SCaLE: supervised and cascaded Laplacian eigenmaps for visual object recognition based on nearest neighbors. In *Proc. IEEE CVPR*, pages 867–874, 2013. 1

[34] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Schölkopf. Learning with local and global consistency. In *NIPS*, pages 321–328, 2003. 3

[35] D. Zhou and B. Schölkopf. Discrete regularization. In *Semi-supervised Learning*, pages 237–250. MIT Press, Cambridge, MA, USA, 2006. 2

[36] X. Zhou and M. Belkin. Semi-supervised learning by higher order regularization. *JMLR W&CP (Proc. AISTATS)*, pages 892–900, 2011. 6

[37] X. Zhu, Z. Ghahramani, and J. Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In *Proc. ICML*, pages 912–919, 2003. 2, 7