Learning in Riemannian Orbifolds

Brijnesh J. Jain and Klaus Obermayer
Technische Universität Berlin
Berlin, Germany
e-mail: brijnesh.jain@gmail.com

Learning in Riemannian orbifolds is motivated by existing machine learning algorithms that directly operate on finite combinatorial structures such as point patterns, trees, and graphs. These methods, however, lack statistical justification. This contribution derives consistency results for learning problems in structured domains and thereby generalizes learning in vector spaces and manifolds.

1 Introduction

Statistical data analysis and learning in Riemannian orbifolds is motivated by applications, where the data we want to learn on are naturally represented by finite combinatorial structures such as point patterns, trees, and graphs. Examples from structural pattern recognition that learn on structured data include estimating central points of a distribution on graphs such as the mean and median [9, 16, 15, 21], central clustering of graphs [10, 12, 13, 14, 19, 15, 23], learning graph quantization [17], and multilayer perceptrons for graphs [20]. In retrospect, the structure space framework proposed by [18] theoretically justifies the above approaches in the sense that they actually minimize an empirical risk function on structures. Since minimizing an empirical risk function is usually computationally intractable, the ultimate challenge consists in constructing efficient algorithms which are capable to return optimal or at least suboptimal solutions.

From the point of view of statistical pattern recognition, however, the ultimate goal is not to determine a good solution of an empirical risk function, but rather to discover the true but unknown structure of the data with respect to its distribution. According to this perspective, we may regard the solutions of empirical risk functions as estimators of the true but unknown population parameter. One gap between statistical and structural pattern recognition is the lack of consistency results of existing estimators for the population parameters. As a consequence most methods from structural pattern recognition that directly operate in the domain of graphs still have no statistical justification.

The first contribution of this paper establishes sufficient conditions for consistency of estimators defined by empirical risk functions on attributed graphs. For this we regard graphs as points of some structure space [18]. A structure space is the quotient of a Euclidean space by some permutation group. The benefit of the structure space framework is that it provides enough mathematical structure for doing differential geometry and at the same time preserves the full relational information of the graphs. In comparison to [18], the innovations are as follows: First, we extend the more suitable concept of generalized differentiability in the sense of Norkin [22] to functions on graphs. Second, we prove the stronger result that the underlying empirical risk functions on graphs are generalized differentiable rather than locally Lipschitz. Third, equipped with these results, we apply a consistency theorem by Ermoliev and Norkin [8] for generalized differentiable loss functions. Finally, using some examples, we show that standard methods from statistical pattern recognition can be generalized to consistent learning algorithms on graphs.

The second contribution shifts the terminology from structure spaces to the more general notion of orbifold. Informally, orbifolds are topological spaces locally modeled on quotients of manifolds by finite group actions. As such, structure spaces are the simplest examples of Riemannian orbifolds. Shifting the focus to orbifolds provides a new view on the problem with the following benefits: First, the notion of orbifold more strongly emphasizes the way we exploit differential geometric tools for graphs, namely via charting and lifting as in Riemannian geometry. Second, using the notion of orbifold integrates the structure space framework into an established mathematical field providing access to useful concepts, results, and insights. Third, the notion of orbifold indicates how the theory can be generalized to structures that locally live in a quotient of a manifold by some finite group action. Fourth, since orbifolds generalize Euclidean spaces and manifolds, this framework not only establishes consistency for stochastic generalized gradient learning but also for standard stochastic gradient learning in Euclidean spaces (see [4]) under the unifying umbrella of learning on Riemannian orbifolds.

2 The Problem of Learning on Graphs

This section aims at outlining the problem of learning on structured data in order to motivate learning in Riemannian orbifolds. As an illustrative example, we consider the problem of estimating the mean of a distribution on attributed graphs.

Attributed Graphs. We begin with describing the structures we want to learn on. Let $\mathcal{A}$ be a set of *attributes* and let $\varepsilon \in \mathcal{A}$ be a distinguished element denoting the *null* or *void* element. An *attributed graph* is a tuple $X = (V, \alpha)$ consisting of a finite nonempty set V of *vertices* and an *attribute function* $\alpha : V \times V \to \mathcal{A}$. Elements of the set $E = \{(i,j) \in V \times V \,:\, i \neq j \text{ and } \alpha(i,j) \neq \varepsilon\}$ are the *edges* of X. By $\mathcal{G}_{\mathcal{A}}$ we denote the set of all attributed graphs with attributes from $\mathcal{A}$. The vertex set of an attributed graph X is often referred to as V_X and its attribute function as α_X.

Alignments. Alignments serve to compare the common structure of two given graphs. An *alignment* of a graph X is a graph X' with $V_X \subseteq V_{X'}$ and

$$\alpha_{X'}(i,j) = \begin{cases} \alpha_X(i,j) & (i,j) \in V_X \times V_X \\ \varepsilon & \text{otherwise} \end{cases} \qquad \forall\, i,j \in V_{X'}.$$

Thus, we obtain an alignment of X by adding isolated vertices with null-attribute. The set $V_{X'}^{\varepsilon} = V_{X'} \backslash V_X$ is the set of *aligned vertices.* By $\mathcal{A}(X)$ we denote the infinite set of all alignments of X. A *pairwise alignment* of graphs X and Y is a triple (ϕ, X', Y') consisting of alignments $X' \in \mathcal{A}(X)$ and $Y' \in \mathcal{A}(Y)$ together with a bijective mapping

$$\phi : V_{X'} \to V_{Y'}, \quad i \mapsto i^{\phi}.$$

A pairwise alignment (ϕ, X', Y') is *minimal* if ϕ does not map aligned vertices onto each other, that is $\phi\left(V_{X'}^{\varepsilon}\right) \subseteq V_Y$. By $\mathcal{A}(X,Y)$ we denote the set of all minimal pairwise alignments between X and Y. Note that $\mathcal{A}(X,Y)$ is finite due to the minimality condition. Sometimes we briefly write ϕ instead of (ϕ, X', Y').

Graph Edit Distance. Dissimilarity is a fundamental concept in machine learning. Here, we consider the graph edit distance, which is a common choice for measuring structural variation of two given graphs. Several distance measures reported in the structural pattern recognition literature can be derived as special cases of the graph edit distance function. Examples are geometric graph distance functions [11] and distances based on the maximum common subgraph including graph and subgraph isomorphism [5].

To define the graph edit distance, we regard each minimal pairwise alignment $(\phi, X', Y') \in \mathcal{A}(X,Y)$ as an *edit path* with *edit cost*

$$d_{\phi}\left(X', Y'\right) = \sum_{i,j \in V_{X'}} d_{\mathcal{A}}\left(\alpha_{X'}(i,j), \alpha_{Y'}(i^{\phi}, j^{\phi})\right),$$

where $d_{\mathcal{A}} : \mathcal{A} \times \mathcal{A} \to \mathbb{R}_+$ is a distance function defined on the set $\mathcal{A}$ of attributes. The edit cost d_{ϕ} can be decomposed into deletion cost $d_A(a, \varepsilon)$, insertion cost $d_A(\varepsilon, a')$, and substitution cost $d_A(a, a')$ of vertices and edges, where $a, a' \in \mathcal{A} \setminus \{\varepsilon\}$ are non-null attributes. Since $d_{\mathcal{A}}$ is a distance function, we have $d_{\mathcal{A}}(\varepsilon, \varepsilon) = 0$. This can only occur for pairs of non-edges by definition of minimal pairwise alignments and therefore can safely be ignored. Observe that deletion (insertion) of vertices also deletes (inserts) all edges the respective vertices are incident to. The *graph edit distance* of X and Y is then defined as the edit path with minimal cost

$$d(X,Y) = \min\left\{d_{\phi}\left(X', Y'\right) \,:\, (\phi, X', Y') \in \mathcal{A}(X,Y)\right\}.$$

The Problem of Learning. Let $(\mathcal{G}_{\mathcal{A}}, d)$ be a graph distance space. As an illustrative example, consider the expected risk

$$R(W) = \frac{1}{2} \int_{\mathcal{G}_{\mathcal{A}}} d(X, W)^2 \, dP_{\mathcal{G}_{\mathcal{A}}}(X),$$

where $W \in \mathcal{W} \subseteq \mathcal{G}_{\mathcal{A}}$ is the optimization variable and $X \in \mathcal{G}_{\mathcal{A}}$ is a random variable with probability distribution $P_{\mathcal{G}_{\mathcal{A}}}$. Since the distribution on the set $\mathcal{G}_{\mathcal{A}}$ of graphs is usually unknown, the goal of learning is to minimize the risk $R(W)$ on the basis of empirical data.

To point out the problems of learning in the domain of graphs, we consider the counterpart of minimizing the risk $R(W)$ in a Euclidean vector space $\mathcal{X}$. The goal is to minimize the expected risk

$$R(\boldsymbol{w}) = \frac{1}{2}\int_{\mathcal{X}} \|\boldsymbol{x} - \boldsymbol{w}\|^2 \, dP_{\mathcal{X}}(\boldsymbol{x}),$$

based on independent and identically distributed random points $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N \in \mathcal{X}$, where $P_{\mathcal{X}}$ is a probability measure on $\mathcal{X}$. Since the loss function $\|\boldsymbol{x} - \boldsymbol{w}\|^2$ is continuously differentiable, the interchange of integral and gradient is valid, that is

$$\nabla R(\boldsymbol{w}) = -\int_{\mathcal{X}} (\boldsymbol{x} - \boldsymbol{w}) dP_{\mathcal{X}}(\boldsymbol{x}).$$

We can minimize the risk $R(\boldsymbol{w})$ using the following stochastic gradient method

$$\boldsymbol{w}_{t+1} = \boldsymbol{w}_t + \frac{1}{t+1}\,(\boldsymbol{x}_t - \boldsymbol{w}_t),$$

where $\boldsymbol{w}_1 = \boldsymbol{x}_1$ and $t \geq 1$. The elements $\boldsymbol{w}_t$ of the sequence $(\boldsymbol{w}_t)_{t \geq 0}$ are sample means

$$\boldsymbol{w}_t = \frac{1}{t}\sum_{i=1}^{t} \boldsymbol{x}_t.$$

It is well-known that the sample mean is a consistent estimator of the population mean $\boldsymbol{\mu}$, which in turn is the unique global minimizer of the expected risk $R(\boldsymbol{w})$.

After this short digression in vector spaces, let us return to the problem of minimizing the expected risk $R(W)$ in graph spaces. As opposed to vector spaces, the following factors complicate learning on graphs in a statistically consistent way: (i) the graph edit distance $d(X, Y)$ is in general not-differentiable; and (ii) neither a well-defined addition on graphs nor the notion of derivative for functions on graphs is known.

We therefore address the following questions: (i) How can we extend gradient-based learning problems from Euclidean spaces to $\mathcal{G}_{\mathcal{A}}$? (ii) How can we minimize the expected risk of a learning problem with structured input- and/or output-space $\mathcal{G}_{\mathcal{A}}$ in a statistically consistent way?

The ansatz to answer both questions is to identify graphs as points of a Riemannian orbifold and to extend the concept of generalized differentiability in the sense of Norkin [22] in order to apply methods from stochastic optimization for non-differentiable and non-convex loss functions.

3 Riemannian Orbifolds

This section introduces Riemannian orbifolds. To keep the treatment technically as uncluttered as possible, we assume that $\mathcal{X} = \mathbb{R}^n$ is the n-dimensional Euclidean space, and

Γ is a permutation group acting on $\mathcal{X}$. In doing so, we can refer to [18] for proofs of statements and claims made in this section. In a more general setting, however, $\mathcal{X}$ can also be a Riemannian manifold. In this case, we refer to [3] for more details.

3.1 Riemannian Orbifolds

The binary operation

$$\cdot : \Gamma \times \mathcal{X} \to \mathcal{X}, \quad (\gamma, \boldsymbol{x}) \mapsto \gamma(\boldsymbol{x})$$

is a group action of Γ on $\mathcal{X}$. For $\boldsymbol{x} \in \mathcal{X}$, the *orbit* of $\boldsymbol{x}$ is the set defined by $[\boldsymbol{x}] = \{\gamma(\boldsymbol{x}) : \gamma \in \Gamma\}$. The quotient set $\mathcal{X}_\Gamma = \mathcal{X}/\Gamma = \{[\boldsymbol{x}] : \boldsymbol{x} \in \mathcal{X}\}$ consisting of all all orbits carries the structure of a *Riemannian orbifold*. Its *orbifold chart* is the surjective continuous mapping

$$\pi : \mathcal{X} \to \mathcal{X}_\Gamma, \quad \boldsymbol{x} \mapsto [\boldsymbol{x}]$$

that projects each point $\boldsymbol{x}$ to its orbit $[\boldsymbol{x}]$. With $\Gamma = \{\mathrm{id}\}$ being the trivial permutation group, $\mathcal{X}$ is also an orbifold. Hence, orbifolds generalize the notion of Euclidean space and manifold.

In the following, an orbifold is a triple $\mathcal{Q} = (\mathcal{X}, \Gamma, \pi)$ consisting of a Euclidean space $\mathcal{X}$, a permutation group Γ acting on $\mathcal{X}$ and its orbifold chart π. We call the elements of $\mathcal{X}_\Gamma$ *structures*, since they represent combinatorial structures such as graphs. We use capital letters $X, Y, Z, \ldots$ to denote structures from $\mathcal{X}_\Gamma$ and write $\boldsymbol{x} \in X$ if $\pi(\boldsymbol{x}) = X$. Each vector $\boldsymbol{x} \in X$ is a *vector representation* of structure X and the set $\mathcal{X}$ of all vector representation is the *representation space* of $\mathcal{X}_\Gamma$.

3.2 The Riemannian Orbifold of Graphs

Riemannian orbifolds of attributed graphs arise by considering equivalence classes of matrices representing the same graph. To identify graphs with points from some orbifold, some technical assumptions to simplify the mathematical treatment are necessary. For this, let $(\mathcal{G}_\mathcal{A}, d)$ be a graph distance space with graph edit distance $d(\cdot|\cdot)$. Then we make the following assumptions:

A1. There is a feature map $\Phi : \mathcal{A} \to \mathcal{H}$ of the attributes into some finite dimensional Euclidean feature space $\mathcal{H}$ and a distance function $d_\mathcal{H} : \mathcal{H} \times \mathcal{H} \to \mathbb{R}_+$ such that $\Phi(\varepsilon) = \mathbf{0} \in \mathcal{H}$ and

$$d_\mathcal{A}(a, a') = d_\mathcal{H}(\Phi(a), \Phi(a')) \quad \forall\, a, a' \in \mathcal{A}.$$

A2. All graphs are finite of bounded order n, where n is a sufficiently large number. A graph X of order less than n, say $m < n$, is aligned to graph X' of order n by inserting $p = n - m$ isolated vertices with null attribute ε.

Let us consider the above assumptions in more detail. Both conditions do not effect the graph edit distance, provided an appropriate feature map for the attributes can be found. Restricting to finite dimensional Euclidean feature spaces $\mathcal{H}$ is necessary for deriving consistency results and for applying methods from stochastic optimization. Limiting the maximum size of the graphs to some arbitrarily large number n and aligning smaller graphs to graphs of oder n are purely technical assumptions to simplify mathematics.

For machine learning problems, this limitation should have no practical impact, because neither the bound n needs to be specified explicitly nor an extension of all graphs to an identical order needs to be performed. When applying the theory, all we actually require is that the order of the graphs is bounded.

With both assumptions in mind, we construct the Riemannian orbifold of attributed graphs. Let $\mathcal{X} = \mathcal{H}^{n \times n}$ be the set of all $(n \times n)$-matrices with elements from feature space $\mathcal{H}$. A graph X is completely specified by a *representation matrix* $\boldsymbol{X} = (\boldsymbol{x}_{ij})$ from $\mathcal{X}$ with elements

$$\boldsymbol{x}_{ij} = \begin{cases} \phi\left(\alpha_X(i,j)\right) & i = j \text{ or } (i,j) \in E \\ \mathbf{0} & \text{otherwise} \end{cases}$$

for all $i, j \in V_X$. The form of a representation matrix $\boldsymbol{X}$ of X is generally not unique and depends on how the vertices are arranged in the diagonal of $\boldsymbol{X}$.

Now suppose that Π^n be the set of all $(n \times n)$-permutation matrices. For each $\boldsymbol{P} \in \Pi^n$ we define a mapping

$$\gamma_{\boldsymbol{P}} : \mathcal{X} \to \mathcal{X}, \quad \boldsymbol{X} \mapsto \boldsymbol{P}^{\mathsf{T}} \boldsymbol{X} \boldsymbol{P}.$$

Then $\Gamma = \{\gamma_{\boldsymbol{P}} : \boldsymbol{P} \in \Pi^n\}$ is a permutation group acting on $\mathcal{X}$. Regarding an arbitrary matrix $\boldsymbol{X}$ as a representation of some graph X, then the orbit $[\boldsymbol{X}]$ consists of all possible matrices that can represent X. By identifying the orbits of $\mathcal{X}_\Gamma$ with attributed graphs, the set $\mathcal{G}_{\mathcal{A}}$ of attributed graphs of bounded order n is a Riemannian orbifold.

3.3 Metric Structures

Let $\mathcal{Q} = (\mathcal{X}, \Gamma, \pi)$ be an orbifold. We derive an intrinsic metric that enables us to do Riemannian geometry. Note that in the case of graph orbifolds, the intrinsic metric is a special graph edit distance based on a generalization of the concept of maximum common subgraph. This graph metric occurs in various different guises as a common choice of proximity measure [1, 6, 7, 11, 24, 25].

Any inner product $\langle \cdot, \cdot \rangle$ on $\mathcal{X}$ gives rise to a maximizer $k : \mathcal{X}_\Gamma \times \mathcal{X}_\Gamma \to \mathbb{R}$ of the form

$$k(X, Y) = \max \left\{ \langle \boldsymbol{x}, \boldsymbol{y} \rangle \,:\, \boldsymbol{x} \in X, \boldsymbol{y} \in Y \right\}.$$

We call the kernel function $k(\cdot|\cdot)$ *optimal alignment kernel*, induced by $\langle \cdot, \cdot \rangle$. Note that the maximizer of a set of positive definite kernels is an indefinite kernel in general. Since Γ is a group, we find that

$$k(X, Y) = \max \left\{ \langle \boldsymbol{x}, \boldsymbol{y} \rangle \,:\, \boldsymbol{x} \in X \right\},$$

where $\boldsymbol{y}$ is an arbitrary but fixed vector representation of Y.

Example 3.1 *Suppose that X and Y are attributed graphs where edges have attribute 1 and vertices have attribute 0. The optimal alignment kernel $k\left(X, Y\right)$ induced by the standard inner product of $\mathcal{X}$ is the number of edges of a maximum common subgraph of X and Y.*

Suppose that $X \in \mathcal{X}_\Gamma$. Since $k(X, X) = \langle \boldsymbol{x}, \boldsymbol{x} \rangle$ for all $\boldsymbol{x} \in X$, we can define the *length* of X by

$$l(X) = \sqrt{k(X, X)}.$$

Since the Cauchy-Schwarz inequality $|k(X, Y)| \leq l(X) \cdot l(Y)$ is valid, the geometric interpretation of $k(\cdot|\cdot)$ is that it computes the cosine of a well-defined angle between X and X' provided both are normalized.

Likewise, $k(\cdot|\cdot)$ gives rise to a distance function defined by

$$d(X, Y) = \sqrt{l(X)^2 - 2k(X, Y) + l(Y)^2}.$$

From the definition of $k(\cdot|\cdot)$ follows that d is a metric. In addition, we have

$$d(X, Y) = \min \{ \|\boldsymbol{x} - \boldsymbol{y}\| \,:\, \boldsymbol{x} \in X, \boldsymbol{y} \in Y \}, \tag{1}$$

where $\|\cdot\|$ denotes the Euclidean norm induced by the inner product $\langle \cdot, \cdot \rangle$ of the Euclidean space $\mathcal{X}$.

Equation (1) states that $d\,(\cdot|\cdot)$ is the length of a minimizing geodesic of X and Y and therefore an intrinsic metric, because it coincides with the infimum of the length of all admissible curves from X to Y. In addition, we find that the topology of $\mathcal{X}_\Gamma$ induced by the metric d coincides with the quotient topology induced by the topology of the Euclidean space $\mathcal{X}$.

3.4 Orbifold Mappings

This section introduces mappings between orbifolds and investigates local analytical concepts of orbifold functions. We assume that $\mathcal{Q} = (\mathcal{X}, \Gamma, \pi)$ and $\mathcal{Q}' = (\mathcal{X}', \Gamma', \pi')$ are orbifolds.

Mappings. An *orbifold mapping* between $\mathcal{Q}$ and $\mathcal{Q}'$ is a mapping $f : \mathcal{X}_\Gamma \to \mathcal{X}'_{\Gamma'}$ between their underlying spaces. The *lift* of f is a mapping $\tilde{f} : \mathcal{X} \to \mathcal{X}'$ between their representation spaces such that $f \circ \pi = \pi' \circ \tilde{f}$. Since $\mathbb{R}$ is an orbifold of the form $\mathcal{Q}_\mathbb{R} = (\mathbb{R}, \{\mathrm{id}\}, \mathrm{id}_\mathbb{R})$, we can define an *orbifold function* between $\mathcal{Q}$ and $\mathcal{Q}_\mathbb{R}$ as a function $f : \mathcal{X}_\Gamma \to \mathbb{R}$. The lift of f is a function $\tilde{f} : \mathcal{X} \to \mathbb{R}$ satisfying $\tilde{f} = f \circ \pi$. The lift $\tilde{f}$ is invariant under group actions of Γ, that is $\tilde{f}(\boldsymbol{x}) = \tilde{f}\,(\gamma(\boldsymbol{x}))$ for all $\gamma \in \Gamma$.

We say, an orbifold function $f : \mathcal{X}_\Gamma \to \mathbb{R}$ is continuous (locally Lipschitz, differentiable) at $X \in \mathcal{X}_\Gamma$ if its lift $\tilde{f}$ is continuous (locally Lipschitz, differentiable) at some vector representation $\boldsymbol{x} \in X$. The definition is independent of the choice of the vector representation that projects to X.

Gradients. Suppose that $f : \mathcal{X}_\Gamma \to \mathbb{R}$ is differentiable at $X \in \mathcal{X}_\Gamma$. Then its lift $\tilde{f} : \mathcal{X} \to \mathbb{R}$ is differentiable at all vector representations that project to X. The *gradient* $\nabla f(X)$ of f at X is defined by the projection

$$\nabla f(X) = \pi(\nabla \tilde{f}(\boldsymbol{x}))$$

of the gradient $\nabla \tilde{f}(\boldsymbol{x})$ of $\tilde{f}$ at a vector representation $\boldsymbol{x} \in X$. This definition is independent of the choice of the vector representation. We have

$$\nabla \tilde{f}(\gamma(\boldsymbol{x})) = \gamma(\nabla \tilde{f}(\boldsymbol{x}))$$

for all $\gamma \in \Gamma$. This implies that the gradients of $\tilde{f}$ at $\boldsymbol{x}$ and $\gamma(\boldsymbol{x})$ are vector representations of the same structure, namely the gradient $\nabla f(X)$ of the orbifold function f at X. Thus, the gradient of f at X is a well-defined structure pointing to the direction of steepest ascent.

4 Generalized Gradients

This section extends the concept of generalized differentiability in the sense of Norkin [22] to orbifold functions. We begin with introducing generalized differentiable functions. Let $\mathcal{X} = \mathbb{R}^n$ be a finite-dimensional Euclidean space. A function $f : \mathcal{X} \to \mathbb{R}$ is *generalized differentiable* at $\boldsymbol{x} \in \mathcal{X}$ if there is a multi-valued map $\partial f : \mathcal{X} \to 2^{\mathcal{X}}$ in a neighborhood of $\boldsymbol{x}$ such that

1. $\partial f(\boldsymbol{x})$ is a convex and compact set;

2. $\partial f(\boldsymbol{x})$ is upper semicontinuous at $\boldsymbol{x}$, that is, if $\boldsymbol{y}_i \to \boldsymbol{x}$ and $\boldsymbol{g}_i \in \partial f(\boldsymbol{y}_i)$ for each $i \in \mathbb{N}$, then each accumulation point $\boldsymbol{g}$ of $(\boldsymbol{g}_i)$ is in $\partial f(\boldsymbol{x})$;

3. for each $\boldsymbol{y} \in \mathcal{X}$ and any $\boldsymbol{g} \in \partial f(\boldsymbol{y})$ holds $f(\boldsymbol{y}) = f(\boldsymbol{x}) + \langle \boldsymbol{g}, \boldsymbol{y} - \boldsymbol{x} \rangle + o(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{g})$, where the remainder $o(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{g})$ satisfies the condition

   $$\lim_{i \to \infty} \frac{|o(\boldsymbol{x}, \boldsymbol{y}_i, \boldsymbol{g}_i)|}{\|\boldsymbol{y}_i - \boldsymbol{x}\|} = 0$$

   for all sequences $\boldsymbol{y}_i \to \boldsymbol{y}$ and $\boldsymbol{g}_i \in \partial f(\boldsymbol{y}_i)$.

We call f *generalized differentiable* if it is generalized differentiable at each point $\boldsymbol{x} \in \mathcal{X}$. The set $\partial f(\boldsymbol{x})$ is the *subdifferential* of f at $\boldsymbol{x}$ and its elements are called *generalized gradients.*

Generalized differentiable functions have the following properties [22]:

1. Generalized differentiable functions are locally Lipschitz and therefore continuous and differentiable almost everywhere.

2. Continuously differentiable, convex, and concave functions are generalized differentiable.

3. Suppose that $f_1, \ldots, f_n : \mathcal{X} \to \mathbb{R}$ are generalized differentiable at $\boldsymbol{x} \in \mathcal{X}$. Then

   $$f_*(\boldsymbol{x}) = \min(f_1(\boldsymbol{x}), \ldots, f_m(\boldsymbol{x})) \quad \text{and} \quad f^*(\boldsymbol{x}) = \max(f_1(\boldsymbol{x}), \ldots, f_m(\boldsymbol{x}))$$

   are generalized differentiable at $\boldsymbol{x} \in \mathcal{X}$.

4. Suppose that $f_1, \ldots, f_m : \mathcal{X} \to \mathbb{R}$ are generalized differentiable functions at $\boldsymbol{x} \in \mathcal{X}$ and $f_0 : \mathbb{R}^m \to \mathbb{R}$ is generalized differentiable at $\boldsymbol{y} = (f_1(\boldsymbol{x}), \ldots, f_m(\boldsymbol{x})) \in \mathbb{R}^m$. Then $f(\boldsymbol{x}) = f_0(f_1(\boldsymbol{x}), \ldots, f_m(\boldsymbol{x}))$ is generalized differentiable at $\boldsymbol{x} \in \mathcal{X}$. The subdifferential of f at $\boldsymbol{x}$ is of the form

$$\partial f(\boldsymbol{x}) = \text{con}\left\{\boldsymbol{g} \in \mathcal{X} : \boldsymbol{g} = [\boldsymbol{g}_1 \boldsymbol{g}_2 \ldots \boldsymbol{g}_m]\boldsymbol{g}_0, \boldsymbol{g}_0 \in \partial f_0(\boldsymbol{y}), \boldsymbol{g}_i \in \partial f_i(\boldsymbol{x}), 1 \leq i \leq m\right\}.$$

where $[\boldsymbol{g}_1 \boldsymbol{g}_2 \ldots \boldsymbol{g}_m]$ is a $(N \times m)$-matrix.

5. Suppose that $F(\boldsymbol{x}) = \mathbb{E}_{\boldsymbol{z}}\left[f(\boldsymbol{x}, \boldsymbol{z})\right]$, where $f(\cdot, \boldsymbol{z})$ is generalized differentiable. Then F is generalized differentiable and its subdifferential at $\boldsymbol{x} \in \mathcal{X}$ is of the form $\partial F(\boldsymbol{x}) = \mathbb{E}_{\boldsymbol{z}}\left[\partial f(\boldsymbol{x}, \boldsymbol{z})\right]$.

Now suppose that $f : \mathcal{X}_\Gamma \to \mathbb{R}$ is an orbifold function. We say f is generalized differentiable at $X \in \mathcal{X}_\Gamma$, if its lift $\tilde{f} : \mathcal{X} \to \mathbb{R}$ is generalized differentiable at all vector representations that project to X. The *subdifferential* $\partial f(X)$ of f at X is defined by the projection

$$\partial f(X) = \pi(\partial \tilde{f}(\boldsymbol{x}))$$

of the subdifferential $\partial \tilde{f}(\boldsymbol{x})$ of $\tilde{f}$ at a vector representation $\boldsymbol{x} \in X$. This definition is independent of the choice of the vector representation. We have

$$\partial \tilde{f}(\gamma(\boldsymbol{x})) = \gamma(\partial \tilde{f}(\boldsymbol{x}))$$

for all $\gamma \in \Gamma$. This implies that the subdifferentials $\partial \tilde{f}(\boldsymbol{x}) \subseteq \mathcal{X}$ and $\partial \tilde{f}(\gamma(\boldsymbol{x})) \subseteq \mathcal{X}$ are subsets that project to the same subset of $\mathcal{X}_\Gamma$, namely the subdifferential $\partial f(X)$. Proposition 4.1 summarizes and proves the statements.

Proposition 4.1 *Let $f : \mathcal{X}_\Gamma \to \mathbb{R}$ be an orbifold function. Suppose that its lift $\tilde{f} : \mathcal{X} \to \mathbb{R}$ is generalized differentiable at a vector representation $\boldsymbol{x}$ that projects to $X \in \mathcal{X}_\Gamma$. Then $\tilde{f}$ is generalized differentiable at $\gamma(\boldsymbol{x})$ for all $\gamma \in \Gamma$ and*

$$\partial \tilde{f}(\gamma(\boldsymbol{x})) = \gamma\left(\partial \tilde{f}(\boldsymbol{x})\right).$$

is a subdifferential of $\tilde{f}$ at $\gamma(\boldsymbol{x})$ for all $\gamma \in \Gamma$.

Proof: Since $\tilde{f}$ is generalized differentiable at $\boldsymbol{x}$, there is a multi-valued mapping $\partial \tilde{f} : \mathcal{U}_\delta(\boldsymbol{x}) \to 2^{\mathcal{X}}$ defined on some neighborhood $\mathcal{U}_\delta(\boldsymbol{x})$. Let $\gamma \in \Gamma$ be an arbitrary permutation and $\boldsymbol{x}' = \gamma(\boldsymbol{x})$. Then

$$\partial \tilde{f} : \mathcal{U}_\delta(\boldsymbol{x}') \to 2^{\mathcal{X}}, \quad \boldsymbol{y}' = \gamma(\boldsymbol{y}) \mapsto \gamma\left(\partial \tilde{f}(\boldsymbol{y})\right)$$

is a multi-valued mapping in a neighborhood of $\boldsymbol{x}'$. Since γ is a homeomorphic linear map, we find that $\gamma(\partial \tilde{f}(\boldsymbol{x})) = \partial \tilde{f}(\boldsymbol{x}')$ is a convex and compact set. Next we show that $\tilde{f}$ is upper semicontinuous at $\boldsymbol{x}'$. Suppose that $\boldsymbol{y}'_i \to \boldsymbol{x}'$, $\boldsymbol{g}'_i \in \tilde{f}_c(\boldsymbol{y}'_i)$ for each $i \in \mathbb{N}$, and $\boldsymbol{g}'$

is an accumulation point of $(\boldsymbol{g}_i')_{i\in\mathbb{N}}$. Then there is a $i_0 \in \mathbb{N}$ such that $\boldsymbol{y}_i' \in \mathcal{U}_\delta(\boldsymbol{x}')$ for all $i \geq i_0$. From

$$\mathcal{U}_\delta(\boldsymbol{x}') = \mathcal{U}_\delta(\gamma(\boldsymbol{x})) = \gamma\left(\mathcal{U}_\delta(\boldsymbol{x})\right)$$

follows that there are vector representations $\boldsymbol{y}_i \in \mathcal{U}_\delta(\boldsymbol{x})$ with $\gamma(\boldsymbol{y}_i) = \boldsymbol{y}_i'$ for each $i \geq i_0$. From continuity of γ^{-1} follows that $\boldsymbol{y}_i \to \boldsymbol{x}$. By construction of $\partial\tilde{f}$ follows that

$$\boldsymbol{g}_i' \in \partial\tilde{f}\left(\boldsymbol{y}_i'\right) = \partial\tilde{f}\left(\gamma\left(\boldsymbol{y}_i\right)\right) = \gamma\left(\partial\tilde{f}\left(\boldsymbol{y}_i\right)\right)$$

for each $i \geq i_0$. Hence, there are vector representations $\boldsymbol{g}_i \in \partial\tilde{f}(\boldsymbol{y}_i)$ with $\gamma(\boldsymbol{g}_i) = \boldsymbol{g}_i'$ for each $i \geq i_0$. Since $\tilde{f}$ is upper semicontinuous at $\boldsymbol{x}$, we find that $\boldsymbol{g} \in \partial\tilde{f}(\boldsymbol{x})$. Again by construction of $\partial\tilde{f}$ follows that

$$\boldsymbol{g}' = \gamma(\boldsymbol{g}) \in \gamma\left(\partial\tilde{f}(\boldsymbol{x})\right) = \partial\tilde{f}\left(\gamma(\boldsymbol{x})\right) = \partial\tilde{f}(\boldsymbol{x}').$$

This proves upper semicontinuity of $\partial\tilde{f}$ at all vector representations projecting to $X = \pi(\boldsymbol{x})$. Finally, we prove that $\tilde{f}$ satisfies the subderivative property at $\boldsymbol{x}'$. Suppose that $\boldsymbol{y}', \boldsymbol{y} \in \mathcal{X}$ with $\boldsymbol{y}' = \gamma(\boldsymbol{y})$. By Γ-invariance of $\tilde{f}$, we have $\tilde{f}(\boldsymbol{y}') = \tilde{f}(\boldsymbol{y})$. Since $\tilde{f}$ is generalized differentiable at $\boldsymbol{x}$, we find a $\boldsymbol{g} \in \partial\tilde{f}(\boldsymbol{y})$ such that

$$\tilde{f}(\boldsymbol{y}') = \tilde{f}(\boldsymbol{y}) = \tilde{f}(\boldsymbol{x}) + \langle \boldsymbol{g}, \boldsymbol{y} - \boldsymbol{x}\rangle + o(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{g})$$

with $o(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{g})$ tending faster to zero than $\|\boldsymbol{y} - \boldsymbol{x}\|$. Let $\boldsymbol{g}' = \gamma(\boldsymbol{g})$. Exploiting Γ-invariance of $\tilde{f}$ as well as isometry and linearity of γ yields

$$\begin{aligned}\tilde{f}(\boldsymbol{y}') &= \tilde{f}(\gamma(\boldsymbol{x})) + \langle \gamma(\boldsymbol{g}), \gamma(\boldsymbol{y} - \boldsymbol{x})\rangle + o(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{g}) \\ &= \tilde{f}(\boldsymbol{x}') + \left\langle \boldsymbol{g}', \boldsymbol{y}' - \boldsymbol{x}'\right\rangle + o(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{g}).\end{aligned}$$

We define $o'(\boldsymbol{x}', \boldsymbol{y}', \boldsymbol{g}') = o \circ \gamma^{-1}(\boldsymbol{x}', \boldsymbol{y}', \boldsymbol{g}') = o(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{g})$ showing that o' tends faster to zero than $\|\boldsymbol{y}' - \boldsymbol{x}\|$. This proves the subderivative property of $\tilde{f}$ at all vector representations projecting to $X = \pi(\boldsymbol{x})$. Putting all results together yields that $\tilde{f}$ is generalized differentiable at $\gamma(\boldsymbol{x})$ for all $\gamma \in \Gamma$. ■

Example 4.1 *Let $(\mathcal{G}_\mathcal{A}, d)$ be a graph space, where d is a graph edit distance. We can identify $\mathcal{G}_\mathcal{A}$ with a Riemannian orbifold $\mathcal{Q} = (\mathcal{X}, \Gamma, \pi)$ and the graph edit distance $d\left(\cdot|\cdot\right)$ with a distance function defined on $\mathcal{X}_\Gamma$. Suppose that the edit costs $d_\phi\left(\cdot|\cdot\right)$ of all edit paths are generalized differentiable. Then the distance $d\left(\cdot|\cdot\right)$ is generalized differentiable.*

Example 4.2 *Let $\mathcal{Q}$ be a graph orbifold. Then the optimal assignment kernel $k\left(\cdot|\cdot\right)$, the intrinsic metric $d\left(\cdot|\cdot\right)$, and the squared metric $d\left(\cdot|\cdot\right)^2$ are generalized differentiable.*

5 Stochastic Optimization

We assume that $\mathcal{Q}_\mathcal{W} = (\mathcal{W}, \mathsf{H}, \rho)$ and $\mathcal{Q}_\mathcal{Z} = (\mathcal{Z}, \Gamma, \pi)$ are Riemannian orbifolds and $\Omega \subseteq \mathcal{W}_\mathsf{H}$ is some (sufficiently large) bounded convex constraint set. Learning is formulated as a stochastic optimization problem of the form

$$\min R(W) = \mathbb{E}\left[L(Z, W)\right] = \int_{\mathcal{Z}_\Gamma} L(Z, W) dP_\Gamma(Z) \tag{2}$$

$$\text{s.t. } W \in \Omega, \tag{3}$$

where $R(W)$ is the *expected risk function*, $W \in \Omega$ is the optimization variable, and $Z \in \mathcal{Z}_\Gamma$ is a random variable with probability measure P_Γ. The *loss function* $L : \mathcal{Z}_\Gamma \times \Omega \to \mathbb{R}$ measures the performance of the learning system with parameter W given an observable event Z. We assume that the loss $L(Z, W)$ is generalized differentiable in W and integrable in Z. The expectation $\mathbb{E}$ is taken with respect to some probability space $(\mathcal{Z}_\Gamma, \Sigma_\Gamma, P_\Gamma)$.

Since the distribution $P_\mathcal{Z}$ of the observable events $Z \in \mathcal{Z}$ is usually unknown, the expected risk function $R(W)$ can neither be computed nor be minimized directly. In addition, the loss function $L(W, Z)$ is neither convex nor differentiable. The field of stochastic approximation provides methods to minimize $R(W)$ that are consistent under very general conditions.

Since the interchange of integral and generalized gradient is valid, that is $\partial_W R(W) = \mathbb{E}\left[\partial_W L(Z, W)\right]$ under mild assumptions [8, 22], we can minimize the expected risk $R(W)$ according to the following *stochastic generalized gradient* (SGG) method:

$$W_{t+1} = \Pi_\Omega \left(W_t - \eta_t S_t\right), \qquad t \geq 0,$$

where $W_0 \in \Omega$ and Π_Ω is a projection operator on Ω. The random structures S_t are *stochastic generalized gradients*, i.e. random variables defined on the probability space $(\mathcal{Z}_\Gamma, \Sigma_\Gamma, P_\Gamma)^\infty$ such that

$$\mathbb{E}\left[S_t \,|\, W_0, \dots, W_t\right] \in \partial_W R\left(W\right). \tag{4}$$

We can take $S_t = g(Z_t, W_t)$ with iid $(Z_t)_{t \geq 0}$ and some single valued selection $g(Z, W) \in \partial_W L(Z, W)$, measurable in (Z, W). We consider the following conditions for almost sure convergence of the SSG method:

A1 The sequence $(\eta_t)_{t\geq 0}$ of step sizes satisfies

$$\eta_t > 0, \; \lim_{t \to \infty} \eta_t = 0, \; \sum_{t=1}^{\infty} \eta_t = \infty, \; \sum_{t=1}^{\infty} \eta_t^2 < \infty.$$

A2 The sequence $(S_t)_{t\geq 0}$ satisfies (4).

A3 We have $\mathbb{E}\left[\|S_t\|^2\right] < +\infty$.

Then by Ermoliev and Norkin's Theorem [8], the SGG method is consistent in the sense that the sequence $(W_t)_{t\geq 0}$ converges almost surely to points satisfying necessary extremum conditions

$$\Omega^* = \{W \in \Omega \, : \, 0 \in \partial_W R(W) + \mathcal{N}_\Omega(W)\},$$

where $\mathcal{N}_\Omega(W)$ is a normal cone to the constraint set Ω at $W \in \Omega$. Besides the sequence $(R(W_t))_{t \geq 0}$ converges almost surely and $\lim_t R(W_t) \in R(\Omega^*)$.

Since orbifolds generalize Euclidean spaces and manifolds the consistency theorem is also valid for standard machine learning algorithms in Euclidean spaces with differentiable cost function (e.g multi-layer perceptron) and non-differentiable cost function (e.g. online k-means) [4].

6 Examples

This section extends some typical examples of statistical data analysis and learning problems from vector spaces to structured domains. We assume that $\mathcal{Q} = (\mathcal{X}, \Gamma, \pi)$ is a Riemannian orbifold with optimal alignment kernel $k(\cdot|\cdot)$.

Orbifold-Adaline. Orbifold adaline generalizes the *adaline* proposed by [26].

Let $\mathcal{W} = \mathcal{X}_\Gamma \times R$ be the parameter space and let $\mathcal{Z} = \mathcal{X}_\Gamma \times \{\pm 1\}$ be the space of observable data. The parameter space $\mathcal{W}$ consists of augmented parameter structures $W' = (W, b)$, where $W \in \mathcal{X}_\Gamma$ is the weight structure and $b \in \mathbb{R}$ is the bias. The observable data $Z = (X, y)$ from $\mathcal{Z}$ consists of input structures $X \in \mathcal{X}_\Gamma$ together with their labels $y \in \{\pm 1\}$.

The loss function of the orbifold-Adaline is of the form

$$L_{ada}(Z, W') = \big(y - (k(X, W) + b)\big)^2.$$

Since $k(\cdot|\cdot)$ is generalized differentiable, so is $L_{ada}(Z, W)$. Lifting the loss L_{ada} to the Euclidean space gives

$$\hat{L}_{ada}\left(\boldsymbol{z}, \boldsymbol{w}'\right) = \big(y - \max\big\{\langle \boldsymbol{x}', \boldsymbol{w}\rangle \,:\, \boldsymbol{x}' \in X\big\} - b\big)^2,$$

where $\boldsymbol{z} = (\boldsymbol{x}, y) \in \mathcal{Z}$ and $\boldsymbol{w}' = (\boldsymbol{w}, b) \in \mathcal{W}$ with vector representations $\boldsymbol{x}$ and $\boldsymbol{y}$ that project to structures $X \in \mathcal{X}_\Gamma$ and $W \in \mathcal{X}_\Gamma$, respectively. The update rule is given by

$$\begin{aligned} \boldsymbol{w}_{t+1} &= \boldsymbol{w}_t - \eta_t\,(y_t - \langle \boldsymbol{x}_t^*, \boldsymbol{w}_t\rangle\, \boldsymbol{x}_t^*) \\ b_{t+1} &= b_t - \eta_t\,(y_t - b_t), \end{aligned}$$

where $(\boldsymbol{x}_t^*, \boldsymbol{w}_t)$ is an optimal alignment.

Learning Orbifold Maps. This example presents a generic formulation of learning functional relationships between orbifolds in a supervised manner. Since orbifolds generalize Euclidean spaces, this setting covers various types of functional relationships that can be learned. Non-standard examples include multi-layer perceptrons for adaptive processing of graphs [20] and learning to predict structured data [2].

Let $\mathcal{Q}_\mathcal{W} = (\mathcal{W}, \Omega, \psi)$, $\mathcal{Q}_\mathcal{X} = (\mathcal{X}, \Gamma, \pi)$, and $\mathcal{Q}_\mathcal{Y} = (\mathcal{Y}, \Lambda, \phi)$ be Riemannian orbifolds. The parameter space is represented by orbifold $\mathcal{Q}_\mathcal{W}$ and the space of observable data by the orbifold $\mathcal{Q}_{\mathcal{Z}} = \mathcal{Q}_{\mathcal{X}} \times \mathcal{Q}_{\mathcal{Y}}$. Suppose that $\mathcal{F}$ is a class of generalized differentiable orbifold mappings of the form

$$f : \mathcal{X}_\Gamma \times \mathcal{W}_\Omega \to \mathcal{Y}_\Lambda.$$

The mean-squared-error loss function is defined by

$$L_{mse}(Z, W) = \frac{1}{2}\left(Y - f(X, W)\right)^2.$$

Lifting this loss function yields

$$\hat{L}_{mse}(\boldsymbol{z}, \boldsymbol{w}) = \frac{1}{2}\left(\boldsymbol{y} - \hat{f}(\boldsymbol{x}, \boldsymbol{w})\right)^2,$$

where $\boldsymbol{z} = (\boldsymbol{x}, \boldsymbol{y})$ projects to structure $Z = (X, Y)$ and $\boldsymbol{w}$ projects to W. The update rule is then of the form

$$\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \eta_t \left(\boldsymbol{y}_t - \hat{f}(\boldsymbol{x}_t, \boldsymbol{w}_t)\right)^{\mathsf{T}} g(\boldsymbol{x}_t, \boldsymbol{w}_t),$$

where $g(\boldsymbol{x}_t, \boldsymbol{w}_t) \in \partial \hat{L}_{mse}(\boldsymbol{z}_t, \boldsymbol{w}_t)$ is a stochastic generalized gradient of the lifted loss at $\boldsymbol{w}_t$.

Structure Quantization. Structure quantization generalizes vector quantization to the quantization of structures. For graphs, a number of structure quantizer design techniques for the purpose of central clustering have already been proposed. Examples include competitive learning [12, 13, 17] and k-means as well as k-medoids algorithms [10, 15, 23].

Let $\mathcal{W} = \mathcal{X}_\Gamma^k$ be the parameter space and let $\mathcal{Z} = \mathcal{X}_\Gamma$ be the space of observable data. The parameter space $\mathcal{W}$ consists of k-tuples $W = (W_1, \ldots, W_k)$, called *codebook*.

The general loss function of structure quantization is defined by the distortion

$$L_{sq}(X, W) = \min_{1 \leq i \leq k} d(X, W_i).$$

For generalized differentiable distance function $d(\cdot|\cdot)$, the update rule is defined by

$$\boldsymbol{w}_{t+1}^* = \boldsymbol{w}_t^* - \eta g(\boldsymbol{x}_t, \boldsymbol{w}_t^*),$$

where $(\boldsymbol{x}_t, \boldsymbol{w}_t^*)$ is an optimal alignment of input structure X_t and its closest codebook structure W_t^*. If $d(\cdot|\cdot)$ is the squared intrinsic metric, we have $g(\boldsymbol{x}, \boldsymbol{w}_t^*) = \boldsymbol{x}_t - \boldsymbol{w}_t^*$.

Observe that structure quantization also generalizes the problem of estimating a mean graph of Section 2.4 by fixing the number k of centroids to 1.

7 Conclusion

This contribution proves consistency of learning in structured domains by reducing it to stochastic generalized gradient learning on Riemannian orbifolds. The proposed framework is applicable to learning on combinatorial structures such as point patterns, trees, and graphs. In retrospect, the proposed results provide a theoretical foundation and statistical justification of a number of existing learning methods that directly operate in the domain of graphs. In addition, the orbifold framework provides a generic technique to generalize gradient-based learning methods to structured domains. Future work aims at generalizing the theory to more general Riemannian orbifolds and to discontinuous graph edit distance functions.

Acknowledgments.

The authors are very grateful to Vladimir Norkin for his kind support and valuable comments.

References

[1] H.A. Almohamad and S.O. Duffuaa. A linear programming approach for the weighted graph matching problem. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 15(5):522–525, 1993.

[2] G. Bakir, T. Hofmann, B. Schölkopf, A.J. Smola, and B. Taskar, editors. *Predicting structured data*. The MIT Press, 2007.

[3] J.E. Borzellino. *Riemannian geometry of orbifolds*. PhD thesis, University of California, Los Angelos, 1992.

[4] L. Bottou. Stochastic learning. *Advanced lectures on machine learning*, pages 146–168, 2003.

[5] H. Bunke. On a relation between graph edit distance and maximum common subgraph. *Pattern Recognition Letters*, 18(8):689 – 694, 1997.

[6] T.S. Caetano, L. Cheng, Q.V. Le, , and A.J. Smola. Learning graph matching. In *International Conference on Computer Vision, ICCV 2007*, pages 1–8, 2007.

[7] T. Cour, P. Srinivasan, and J. Shi. Balanced graph matching. In *Advances in Neural Information Processing Systems, NIPS 2007*, volume 19, 2007.

[8] Y. M. Ermoliev and V.I. Norkin. Stochastic generalized gradient method for nonconvex nonsmooth stochastic optimization. *Cybernetics and Systems Analysis*, 34(2):196–215, 1998.

[9] M. Ferrer. *Theory and algorithms on the median graph. Application to graph-based classification and clustering*. PhD thesis, Universitat Autònoma de Barcelona, 2007.

[10] M. Ferrer, E. Valveny, F. Serratosa, I. Bardají, and H. Bunke. Graph-based k-means clustering: A comparison of the set median versus the generalized median graph. In *Computer Analysis of Images and Patterns, CAIP 2009*, pages 342–350, 2009.

[11] S. Gold and A. Rangarajan. A graduated assignment algorithm for graph matching. *Ieee Transactions On Pattern Analysis and Machine Intelligence*, 18(4):377–388, 1996.

[12] S Gold, A Rangarajan, and E Mjolsness. Learning with preknowledge: Clustering with point and graph matching distance measures. *Neural Computation*, 8(4):787–804, 1996.

[13] S. Günter and H. Bunke. Self-organizing map for clustering in the graph domain. *Pattern Recognition Letters*, 23(4):405–417, 2002.

[14] A. Hlaoui and S. Wang. Median graph computation for graph clustering. *Soft Computing-A Fusion of Foundations, Methodologies and Applications*, 10(1):47–53, 2006.

[15] B. Jain and K. Obermayer. On the sample mean of graphs. In *International Joint Conference on Neural Networks, IJCNN 2008*, pages 993–1000, 2008.

[16] B. Jain and K. Obermayer. Algorithms for the sample mean of graphs. In *Computer Analysis of Images and Patterns, CAIP 2009*, pages 351–359, 2009.

[17] B. Jain and K. Obermayer. Graph quantization. arXiv:1001.0921v1 [cs.AI], 2009.

[18] B. Jain and K. Obermayer. Structure spaces. *Journal of Machine Learning Research*, 10:2667–2714, 2009.

[19] B. Jain and F. Wysotzki. Central clustering of attributed graphs. *Machine Learning*, 56(1-3):169–207, 2004.

[20] B. Jain and F. Wysotzki. Structural perceptrons for attributed graphs. In *Structural, Syntactic, and Statistical Pattern Recognition, SSPR/SPR 2004*, pages 85–94, 2004.

[21] X. Jiang, A. Munger, and H. Bunke. An median graphs: properties, algorithms, and applications. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 23(10):1144–1151, 2001.

[22] V.I. Norkin. Stochastic generalized-differentiable functions in the problem of nonconvex nonsmooth stochastic optimization. *Cybernetics*, 22(6):804–809, 1986.

[23] A. Schenker, H. Bunke, M. Last, and A. Kandel. Clustering of web documents using graph representations. In *Applied Graph Theory in Computer Vision and Pattern Recognition*, volume 52 of *Studies in Computational Intelligence*, pages 247–265. Springer, 2007.

[24] S. Umeyama. An eigendecomposition approach to weighted graph matching problems. *IEEE Trans. Pattern Anal. Mach. Intell.*, 10(5):695–703, 1988.

[25] M.A. van Wyk, T.S. Durrani, and B.J. van Wyk. A rkhs interpolator-based graph matching algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24:988–995, 2002.

[26] B. Widrow and M.E. Hoff. Adaptive switching circuits. In *IRE WESCON Convention Record*, volume 4, pages 96–104, 1960.