6-2014

Graph-based Semi-supervised Learning: Realizing Pointwise Smoothness Probabilistically

Yuan FANG
Singapore Management University, yfang@smu.edu.sg

Kevin Chen-Chuan CHANG
University of Illinois at Urbana-Champaign

Hady W. LAUW
Singapore Management University, hadywlauw@smu.edu.sg

Graph-based Semi-supervised Learning: Realizing Pointwise Smoothness Probabilistically

Yuan Fang [†‡] FANG2@ILLINOIS.EDU
Kevin Chen-Chuan Chang [†‡] KCCHANG@ILLINOIS.EDU
Hady W. Lauw [⋆] HADYWLAUW@SMU.EDU.SG

[†] University of Illinois at Urbana-Champaign, USA
[‡] Advanced Digital Sciences Center, Singapore
[⋆] Singapore Management University, Singapore

Abstract

As the central notion in semi-supervised learning, smoothness is often realized on a graph representation of the data. In this paper, we study two complementary dimensions of smoothness: its pointwise nature and probabilistic modeling. While no existing graph-based work exploits them in conjunction, we encompass both in a novel framework of Probabilistic Graph-based Pointwise Smoothness (PGP), building upon two foundational models of data closeness and label coupling. This new form of smoothness axiomatizes a set of probability constraints, which ultimately enables class prediction. Theoretically, we provide an error and robustness analysis of PGP. Empirically, we conduct extensive experiments to show the advantages of PGP.

1. Introduction

As labeled data is often scarce, semi-supervised learning (SSL) can be beneficial by exploiting unlabeled data. Consider a random tuple (X, Y), where a data point $X \in \mathcal{X} = \{x_1, \ldots, x_{|\mathcal{X}|}\}$ has a label $Y \in \mathcal{Y}$. We observe labeled data $\mathcal{L}$ comprising i.i.d. samples of (X, Y), and unlabeled data $\mathcal{U}$ comprising i.i.d. samples of X. Typically $|\mathcal{U}| \gg |\mathcal{L}|$. Potentially, we may only observe a partial $\mathcal{X}$ via $\mathcal{L}$ and $\mathcal{U}$. The task is to predict the label for every $x_i \in \mathcal{U}$.

Towards effective SSL, graph-based smoothness has attracted much research interest. In particular, the *smoothness* statement is central to SSL (Zhu, 2005; Chapelle et al., 2006): *if two points x_i, x_j are close, their respective labels y_i, y_j are likely to be the same*. The literature further suggests that it is more effective to consider smoothness on the low-dimensional manifold, where the high-dimensional data roughly live. As widely recognized, graphs are often used as a proxy for the manifold (Blum & Chawla, 2001; Zhu et al., 2003; Zhou et al., 2003; Belkin et al., 2006; Johnson & Zhang, 2008; Subramanya & Bilmes, 2011). Specifically, each point x_i is a node on a graph, and two points x_i, x_j may be connected by an edge weighted W_{ij}. The weight matrix W aims to capture the pairwise geodesic distance on the manifold. In other words, the graph reveals the structures on the manifold.

Unfortunately, although graph-based methods universally hinge on smoothness, their realizations fall short. As the thesis of this paper, we advocate that smoothness shall be pointwise in nature and probabilistic in modeling.

Nature: Pointwise smoothness. Smoothness shall inherently occur "everywhere," to relate the behavior of *each point* to that of its close points. We call this the *pointwise* nature of smoothness. As recently identified (Rigollet, 2007), precisely expressing the pointwise nature boils down to two aspects:

- How do we decide if a data point is close to another? The smoothness statement lacks a concrete definition of closeness. Thus, we need a *data closeness model* to define this. (P1)
- How is the behavior of two close points related? The smoothness statement requires that "their labels are likely to be the same", which is rather vague. Thus, we need a *label coupling model* to explicitly relate their label behavior. (P2)

Surprisingly, to date, no existing graph-based method realizes pointwise smoothness. While it has been studied in non-graph based settings (Rigollet, 2007; Lafferty & Wasserman, 2007; Singh et al., 2008), previous graph-based methods treat smoothness in an *aggregate*, rather

Proceedings of the 31[st] International Conference on Machine Learning, Beijing, China, 2014. JMLR: W&CP volume 32.

than pointwise, manner. Specifically, they optimize an energy function in a random field (Zhu et al., 2003; Zhu & Ghahramani, 2002; Getz et al., 2005) or a cost function (Zhou et al., 2003; Belkin et al., 2006; Subramanya & Bilmes, 2011) over the graph. An energy or cost function *aggregates* all pairwise differences between neighboring points across the entire graph. By minimizing the aggregated difference, some "average" smoothness is achieved. However, such aggregation is not designed for and thus does not necessarily enforce smoothness at every point—it is unclear how an aggregate function can precisely express the pointwise nature of smoothness, in terms of the two aspects (P1 & P2). After all, there exist different cost functions varying greatly in actual forms (*e.g.*, squared error, soft margin loss, or probability divergence), with limited justification to favor one over another.

Modeling: Probabilistic smoothness. Pointwise smoothness shall be modeled *probabilistically* in both aspects (P1 & P2), to ultimately infer $p(Y|X)$. First, how close is sufficiently close is difficult to be reliably captured by deterministic binary decisions (P1). Second, the smoothness statement that "their labels are likely to be the same" is meaningless (Rigollet, 2007) unless it is exploited in probabilistic terms (P2). Within a probabilistic framework, eventually each point can be classified based on $p(Y|X)$, given i.i.d. samples. Furthermore, probabilistic modeling conveys some concrete benefits, such as integrating class priors $p(Y)$ in a more principled way, naturally supporting multi-class tasks, and facilitating client applications that require probabilities as input.

We note that existing probabilistic modeling in graph-based settings (Subramanya & Bilmes, 2011; Das & Smith, 2012; He et al., 2007; Azran, 2007) only supports aggregate, but not pointwise, smoothness.

Our proposal. We propose the framework of Probabilistic Graph-based Pointwise Smoothness (PGP), hinging on two foundations that address the pointwise nature of smoothness probabilistically on a graph.

To begin with, we need a *data closeness model* to determine if a point is close to another (P1). Since the graph captures the pairwise geodesic distance on the manifold, a random walk on the graph—which moves from X to X' in each step—naturally "connects" X and X' as close points on the manifold. Hence, for a *pair* of random points (X, X') such that X is close to X', we can describe their distribution $p(X, X')$ using the *second-order* stationary distribution of the random walk. In contrast, the distribution of a *single* point $p(X)$ has been traditionally represented by the *first-order* stationary distribution.

Next, we also need a *label coupling model* to relate the label behavior of a point x_i to that of its close points (P2). We leverage the notion of *statistical indistinguishability* (Goldreich, 2010). In particular, whether X is x_i, or X is some point close to x_i, the label Y of X shall be produced in an indistinguishable manner. In other words, we cannot tell apart the distributions of Y in these two cases.

Together, these two foundations constitute our smoothness framework, which further entails a solution to SSL. While the given labels naturally constrain the labeled data, our smoothness framework axiomatizes a set of probability constraints on the unlabeled data. Solving these constraints eventually infers $p(Y|X)$ for class prediction. Note that the constraints can be either discriminative over $p(Y|X)$, or generative over $p(X|Y)$. Although the ultimate goal is $p(Y|X)$, generative models that learn $p(X|Y)$ and $p(Y)$ are often favorable in SSL (Chapelle et al., 2006). Thus, although our framework can accommodate both forms, we adopt the generative form here and leave the discriminative counterpart to future work[1].

Finally, we present a theoretical analysis of our solution. First, to see that PGP can utilize both labeled and unlabeled data, we derive a generalization error in $\mathcal{L}$ and $\mathcal{U}$. Second, to show that PGP is not sensitive to noisy input graphs, we assess the robustness of our solution.

Our contributions. We summarize the contributions in this paper as follows.

- We propose PGP, the first work to realize pointwise smoothness on a graph probabilistically.
- We conduct an error and robustness analysis of PGP.
- We demonstrate the advantages of PGP through extensive experiments.

2. Smoothness Framework

To express the pointwise nature of smoothness, we must address its two aspects. Under a probabilistic graph-based framework, we propose a data closeness model to capture how a point is close to another (Sect. 2.1), as well as a label coupling model to conceptualize how the label behavior of a point is related to that of its close points (Sect. 2.2).

2.1. Data Closeness Model (P1)

We first propose a probabilistic model for capturing data closeness on the graph.

Graph. For a set of points $\mathcal{X} = \{x_1, \ldots, x_{|\mathcal{X}|}\}$, we construct a graph G to capture the pairwise geodesic distance on the underlying manifold. Each point $x_i \in \mathcal{X}$ is a vertex of G, and each pair of points (x_i, x_j) form an edge of G with a weight W_{ij}. W_{ij} is also known as the *affin-*

[1] See a preliminary discussion in the supplementary material.

ity between x_i and x_j, an approximate description of the geodesic distance between the two points. W is often defined as follows:

$$W_{ij} = \begin{cases} \exp\left(-\|x_i - x_j\|^2/2\sigma^2\right) & i \neq j \\ 0 & i = j, \end{cases} \quad (1)$$

where $\|\cdot\|$ is a symmetric distance function, and σ is a scaling parameter. In our notation, unquantified indices such as i, j belong to $\{1, \ldots, |\mathcal{X}|\}$, unless stated otherwise.

Random walk-based closeness. As argued in Sect. 1, it is difficult to reliably capture how close is sufficiently close in a deterministic manner. In order to develop probabilistic closeness, we need to represent the event that two points, say x_i and x_j, are close.

We capture the closeness event based on a random walk on the graph. Consider a random walk on G visiting a sequence of points $\{V_t : t = 0, 1, \ldots\}$. While traditionally a *visit* at x_i ($V_t = x_i$) models a single point x_i, a *traversal* walking from a point x_i to another x_j ($V_t = x_i, V_{t+1} = x_j$) naturally "connects" x_i and x_j to imply that x_i is close to x_j on the underlying manifold.

Note that our use of random walk serves a novel purpose. It specifically models the first pointwise aspect (P1) of relating the points X through the closeness event, which, together with the second aspect (P2) of relating the labels Y in Sect. 2.2, is necessary for pointwise smoothness. On the contrary, existing use of random walk in SSL (Szummer & Jaakkola, 2001; Azran, 2007; Wu et al., 2012) models the "propagation" of label Y among X altogether, without treating the two aspects explicitly.

Formally, let $(X = x_i, X' = x_j)$ denote the event that x_i is close to x_j, which follows the distribution of observing a random walk traversal from x_i to x_j in the long run as $t \to \infty$. Hence, (X, X') is a pair of limiting random variables in the sense that a traversal (V_t, V_{t+1}) converges *in distribution* to (X, X') jointly:

$$(V_t, V_{t+1}) \xrightarrow{d} (X, X'). \quad (2)$$

In other words, (X, X') describes the closeness between two points with the joint *second-order* limit, while X describes a single point with the marginal *first-order* limit. Their convergence will be shown later.

Probability space of closeness. We describe the probability space of the random walk-based closeness model.

Sample space. An outcome that x_i is close to x_j is a pair of points (x_i, x_j), which corresponds to a random walk traversal from x_i to x_j. Hence, the sample space is $\Omega = \mathcal{X}^2$. An outcome can be denoted by a pair of random variables $(X, X') \in \Omega$, as defined in Eq. 2.

Events. As discussed, each outcome (x_i, x_j) is an event that x_i is close to x_j, *i.e.*, $\{(X, X') \in \Omega : X = x_i, X' = x_j\}$ or denoted $(X = x_i, X' = x_j)$. In order to relate the behavior of two close points, we are also interested in the events that X is x_i or X is close to x_i.

First, $\{(X, X') \in \Omega : X = x_i\}$, or denoted $X = x_i$, is the event of *observing X as a point x_i*. It corresponds to a traversal from x_i to some point.

Second, $\{(X, X') \in \Omega : X' = x_i\}$, or denoted $(X,\ X' = x_i)$, is the event of *observing X as some point close to x_i* (*i.e.*, X is implicitly constrained by $X' = x_i$). It corresponds to a traversal from some point to x_i.

Without loss of generality, here we treat X as the random variable of interest, and our ultimate goal is to estimate $p(Y|X)$. However, we could also treat X' as the variable of interest, and find $p(Y'|X')$ in a symmetric manner given that W is symmetric.

Probability measure. Finally, we evaluate the probability of the events. The random walk can be formally represented by a transition matrix Q such that

$$Q_{ij} = W_{ij}/Z_i, \quad \text{where } Z_i \triangleq \textstyle\sum_j W_{ij}. \quad (3)$$

As (V_t, V_{t+1}) converges in distribution to (X, X'), the closeness event $(X = x_i, X' = x_j)$ obeys the *second-order* stationary distribution of the random walk:

$$p(X{=}x_i, X'{=}x_j) = \lim_{t\to\infty} p(V_t{=}x_i, V_{t+1}{=}x_j). \quad (4)$$

As established in Proposition 1, a unique second-order stationary distribution exists. As a further consequence, the probability of the events can also be computed[2].

PROPOSITION 1 (PROBABILITY OF EVENTS):

(a) The limit of $p(V_t, V_{t+1})$ as $t \to \infty$ exists uniquely.

(b) Given that $(V_t, V_{t+1}) \xrightarrow{d} (X, X')$,

$$\forall ij, \quad p(X = x_i, X' = x_j) \propto W_{ij}, \quad (5)$$

$$\forall i, \quad p(X = x_i) = p(X,\ X' = x_i) \propto Z_i. \quad (6)$$

■

Intuitively, Eq. 5 means that the stronger affinity W_{ij} between x_i and x_j, the more likely they are close. Second, Eq. 6 implies that observing x_i is as likely as observing a point close to x_i, which is not surprising given that two close points lie near each other on the manifold.

2.2. Label Coupling Model (P2)

Next, we propose a label coupling model to relate the label behavior of two close points. In our realization, the label Y of X distributes similarly whether X is x_i itself,

[2] All proofs are included in the supplementary material.

or X is some point close to x_i. That is, $p(Y|X = x_i)$ and $p(Y|X, X' = x_i)$ shall be alike.

Indistinguishability. We leverage the concept of *statistical indistinguishability* (Goldreich, 2010): two distributions are statistically indistinguishable if they cannot be told apart to some extent.

DEFINITION 1 (INDISTINGUISHABILITY): Two distributions D_1 and D_2 are *ϵ-statistically indistinguishable* if and only if $\frac{1}{2}\|D_1 - D_2\|_1 \leq \epsilon$. ■

In our context, $p(Y|X = x_i)$ and $p(Y|X, X' = x_i)$ shall be statistically indistinguishable. In other words, the label Y of X is produced in an indistinguishable manner regardless of X being x_i or a point close to x_i.

Label Coupling. To achieve indistinguishability, x_i's label shall distribute similarly to that of a point close to x_i. At the same time, some "distrust" of the close points shall be allowed, as small variances in their labels are still expected. These factors can be accounted for by a simple mixture:

$$p(Y|X = x_i) = (1 - \alpha)p(Y|X, X' = x_i) + \alpha D, \quad (7)$$

where $\alpha \in (0, 1)$ is a parameter, and D is the distribution to fall back on when the close points are not trusted. In the distrust case, we assign x_i to an "unknown" class $\phi \notin \mathcal{Y}$, *i.e.*, $D(y) = 0, \forall y \in \mathcal{Y}$ and $D(\phi) = 1$. Our label coupling model represented by this mixture formally satisfies statistical indistinguishability.

PROPOSITION 2 (LABEL COUPLING): Given Eq. 7, the label distribution of x_i, $p(Y|X = x_i)$, is *α-statistically indistinguishable* from the label distribution of some point close to x_i, $p(Y|X, X' = x_i)$. ■

Note that Eq. 7 couples the label distributions in a discriminative form of $p(Y|X)$. To model the generative probability $p(X|Y)$ as Sect. 1 motivated, we also derive its generative counterpart. $\forall y \in \mathcal{Y}, \forall x_i \in \mathcal{X}$,

$$\begin{aligned} & p(X = x_i|Y = y) \\ &= p(Y = y|X = x_i)\, p(X = x_i)\, /p(Y = y) \\ &= (1-\alpha)p(Y=y|X, X' = x_i)\, p(X, X' = x_i)\, /p(Y=y) \\ &= (1 - \alpha)p(X, X' = x_i|Y = y)\,. \end{aligned} \quad (8)$$

In particular, D is eliminated since $D(y) = 0, \forall y \in \mathcal{Y}$, *i.e.*, points of class $y \in \mathcal{Y}$ cannot be generated from D. The intuition is that indistinguishability slowly "fades" along a "chain" of close points due to the $1 - \alpha$ factor.

Implication. Eq. 8 implies that the *first-order* conditional distribution $p(X = x_i|Y = y)$ can be related to the sum of the *second-order* (joint) conditional distributions $p(X = x_j, X' = x_i|Y = y)$ over $x_j \in \mathcal{X}$. The association of the first-order or point distribution, to the second-order or edge distribution, is expected, as the pointwise nature of smoothness is to relate the behavior of a point x_i to that of its close points x_j, which we shall see next.

3. Probability Constraint-based Learning

Under the smoothness framework in Sect. 2, we develop a set of generative probability constraints in terms of $p(X|Y)$, and show that a unique solution satisfying the constraints exists. Next, we use an iterative algorithm to find the solution and predict classes accordingly.

3.1. Generative Probability Constraints

For each $y \in \mathcal{Y}$, we aim to learn the generative distribution

$$\pi_y \triangleq (\pi_{y1}, \ldots, \pi_{y|\mathcal{X}|}), \quad (9)$$

where $\pi_{yi} \triangleq p(X = x_i|Y = y)$. To find π_y, we develop and solve a set of constraints on π_y. On the one hand, for $x_i \in \mathcal{L}$ the constraints can be modeled using the known labels. On the other hand, while there is no known label for $x_i \notin \mathcal{L}$, the constraints can be modeled using points close to x_i, based on our smoothness framework.

Labeled points. We rewrite $p(X = x_i|Y = y)$ for $x_i \in \mathcal{L}$, relating it to $p(Y = y|X = x_i)$ which can be estimated from the known labels in Sect. 3.3. For a given $y \in \mathcal{Y}$,

$$\begin{aligned} & p(X = x_i|Y = y) \\ &= p(Y = y|X = x_i)\, p(X = x_i)\, /p(Y = y) \\ &\propto p(Y = y|X = x_i)\, Z_i \end{aligned} \quad (10)$$

The proportionality follows from $p(X = x_i) \propto Z_i$ (Proposition 1). We can transform this result into a constraint on π_y below, where K is the sum of π_{yi} for labeled points, and θ_{yi} is the proportion each π_{yi} gets from the sum K according to Eq. 10. Note that we write $p(y|x_i)$ as a shorthand for $p(Y = y|X = x_i)$ if there is no ambiguity.

Constraint on Labeled Data:

$$\begin{aligned} & \pi_{yi} = K \cdot \theta_{yi}, \quad \forall i : x_i \in \mathcal{L}. \\ \text{where} \quad & K = \textstyle\sum_{i:x_i \in \mathcal{L}} \pi_{yi}, \\ & \theta_{yi} = p(y|x_i)\, Z_i / \textstyle\sum_{k:x_k \in \mathcal{L}} p(y|x_k)\, Z_k. \end{aligned} \quad (11)$$

Unlabeled points. We also rewrite $p(X = x_i|Y = y)$ for unlabeled points $x_i \notin \mathcal{L}$, relating it to that of its close points. Specifically, for a given $y \in \mathcal{Y}$,

$$\begin{aligned} & p(X = x_i|Y = y) \overset{1}{=} (1 - \alpha)p(X, X' = x_i|Y = y) \\ & \overset{2}{=} (1 - \alpha) \textstyle\sum_j p(X = x_j, X' = x_i|Y = y) \\ & \overset{3}{=} (1 - \alpha) \textstyle\sum_j p(X' = x_i|X = x_j, Y = y)\, p(X = x_j|Y = y) \\ & \overset{4}{=} (1 - \alpha) \textstyle\sum_j p(X' = x_i|X = x_j)\, p(X = x_j|Y = y) \\ & \overset{5}{=} (1 - \alpha) \textstyle\sum_j W_{ji}/Z_j \cdot p(X = x_j|Y = y) \end{aligned} \quad (12)$$

Step 1 is the generative form of smoothness (Eq 8). In step 2, we relate x_i to each x_j through their second-order (joint) distribution, where each x_j has a different probability of being close to x_i. In step 3, based on our closeness model, given $X = x_j$, $X' = x_i$ only depends on W and is conditionally independent of Y. In Step 5, $p(X' = x_i | X = x_j)$ is simply the transition probability $Q_{ji} = W_{ji}/Z_j$. This result imposes another constraint on π_y.

Constraint on Unlabeled Data:

$$\pi_{yi} = (1-\alpha)\textstyle\sum_j W_{ji}/Z_j \cdot \pi_{yj}, \quad \forall i : x_i \notin \mathcal{L}. \tag{13}$$

3.2. Solving the Constraints

The goal is to solve π_y that satisfies the constraints on labeled and unlabeled data. In particular, we can show that π_y is the stationary distribution of some Markov chain with $\mathcal{X}$ as its state space. Intuitively, the unlabeled constraint (Eq. 13) already tells us how state x_j transitions to each $x_i \notin \mathcal{L}$. Thus, we only need to deduce the transition to each state $x_i \in \mathcal{L}$. Proposition 3 establishes the exact transition between the states.

PROPOSITION 3 (SOLUTION): $\forall y \in \mathcal{Y}$, if π_y satisfies the constraints in Eq. 11 and 13, then:
(a) π_y is the stationary distribution of a Markov chain $\mathcal{C}$ with states $\mathcal{X}$ and transition matrix P, where

$$P_{ji} = \begin{cases} \frac{\sum_{k:x_k \in \mathcal{L}} W_{jk} + \alpha \sum_{k:x_k \notin \mathcal{L}} W_{jk}}{Z_j} \cdot \theta_{yi} & i : x_i \in \mathcal{L} \\ \frac{(1-\alpha)W_{ji}}{Z_j} & i : x_i \notin \mathcal{L}, \end{cases} \tag{14}$$

(b) The stationary distribution of $\mathcal{C}$ exists uniquely. ■

In fact, Eq. 14 means that $\pi_y = \pi_y P$. If we rewrite it as element-wise operations, we see that a constraint is placed on every individual point.

Class prediction. Given π_y (which will be solved in Sect. 3.3), we predict the label y_i for x_i as follows:

$$\begin{aligned} y_i &= \arg\max_{y \in \mathcal{Y}} p(Y = y | X = x_i) \\ &= \arg\max_{y \in \mathcal{Y}} p(X = x_i | Y = y)\, p(Y = y). \end{aligned} \tag{15}$$

Here $p(X = x_i | Y = y)$ is simply π_{yi}, and $p(Y = y)$ is the class prior which can be estimated from $\mathcal{L}$.

3.3. Solution Computation and Estimation

Next, we discuss how π_y can be computed.

Iterative algorithm. Proposition 3 entails that π_y can be found iteratively, if the transition matrix P is *known*:

$$\pi_y^{(t+1)} = \pi_y^{(t)} P, \qquad t = 0, 1, 2, \ldots, \tag{16}$$

where $\pi_y^{(t)}$ converges uniquely as $t \to \infty$ for an arbitrary initial distribution $\pi_y^{(0)}$.

Solution estimation. We can only find a solution estimator $\hat{\pi}_y$ since P is *unknown*—P is a function of W and θ_y, both of which can only be estimated, resulting in two types of error. First, *data sampling error*: W is defined on $\mathcal{X}$, but only a partial $\hat{\mathcal{X}}$ is observed through $\mathcal{L}$ and $\mathcal{U}$. Thus, we can only construct an estimator $\hat{W}$ using the incomplete $\hat{\mathcal{X}}$. Second, *label sampling error:* θ_y is defined by $p(y|x_i), \forall x_i \in \mathcal{L}$, but $p(y|x_i)$ is unknown. We can only estimate it from the given labels, $\hat{p}(y|x_i) = |\mathcal{L}_y \cap \mathcal{L}_{x_i}|/|\mathcal{L}_{x_i}|$ where $\mathcal{L}_y$ is the set of all samples with y in $\mathcal{L}$, and $\mathcal{L}_{x_i}$ is the set of all samples with x_i in $\mathcal{L}$. Subsequently, we obtain an estimator $\hat{\theta}_y$ based on $\hat{p}(y|x_i)$.

Efficiency. While efficiency is not our focus, PGP can be solved efficiently using standard iterative techniques, and its complexity is comparable to most existing SSL methods. *In terms of time*, if we use a widely accepted kNN graph, the cost is $O(k|\mathcal{X}|s)$, where s is the number of iterations till convergence (typically $k \sim 10$, $s \sim 100$). Although constructing an exact kNN graph can be quadratic, an approximate graph is often adequate (Chen et al., 2009). *In terms of space*, we can store the kNN graph sparsely, thus needing only $O(k|\mathcal{X}|)$ space.

3.4. Discussion: Comparison to Existing Methods

Our constraints on unlabeled points (Eq. 13) may appear similar to existing works, in particular GRF (Zhu et al., 2003) of the following formulation:

$$F_i = \textstyle\sum_j W_{ij}/Z_i \cdot F_j, \tag{17}$$

where $F_i \in [0, 1]$ is the label function at x_i.

Although they resemble in the surface form, their exact forms are still disparate. We stress that such resemblance—expressing x_i's label as some function of its neighbors x_j—is quite expected, since it is a common insight of graph-based SSL to relate a point and its neighbors on the graph (Zhou et al., 2003; Subramanya & Bilmes, 2011). Nonetheless, our exact function still differs in that PGP normalizes each x_j differently by Z_j and has a damping factor $1 - \alpha$, whereas GRF normalizes each x_j by the same Z_i and has no damping factor. Beneath the surface resemblance, there also exist some fundamental differences.

First, most existing cost function (Zhou et al., 2003; Belkin et al., 2006) or random walk (Szummer & Jaakkola, 2001; Azran, 2007; Wu et al., 2012) approaches, including GRF, do not correspond to an explicit formulation of pointwise smoothness. For instance, GRF boils down to the energy function of a Gaussian field, which is the aggregated sum of pairwise losses. Such aggregation is not designed for or derived from requiring smoothness at every individual point. Thus, smoothness does not necessarily occur "everywhere." Even though GRF eventually leads to a local weighted average of neighbors (Eq. 17), it is a consequence

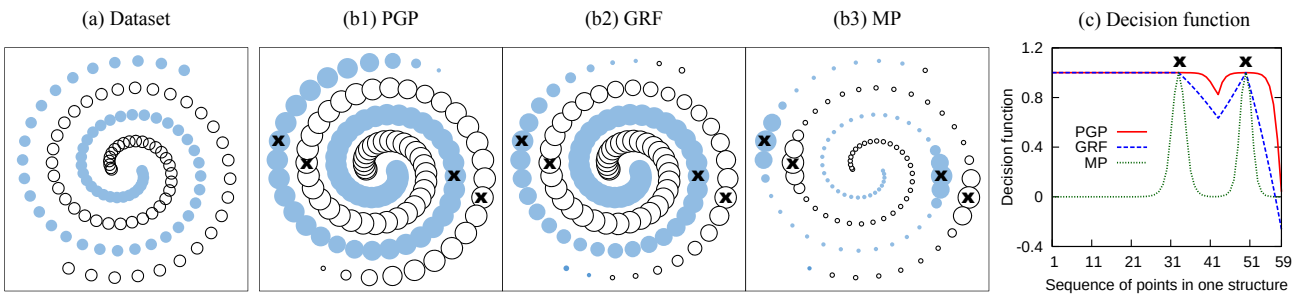

Figure 1. Toy problem: (a) Two-spiral dataset; (b1–3) Visualization of predictions and smoothness; (c) Decision function over a structure.

of minimizing an aggregate loss function, rather than originating from the pointwise nature in terms of the two aspects P1 & P2. In contrast, PGP is not derived from an aggregate function, but directly builds on the data closeness model (for P1) and label coupling model (for P2).

Second, while many approaches (Zhu et al., 2003; Wu et al., 2012) have probabilistic interpretations, they do not explicitly model $p(Y|X)$ or $p(X|Y)$. Taking GRF as an example, $\{F_i\}$ represents the most probable configuration of a Gaussian field. Equivalently, F_i is the random walk probability that a particle starting from x_i first hits a labeled point. Both interpretations do not explicitly correspond to $p(Y|X)$ or $p(X|Y)$. In contrast, in PGP, π_y directly corresponds to $p(X|Y=y)$.

Finally, we use a two-spiral dataset (Singh, 1998) to illustrate that PGP indeed results in better smoothness. As shown in Fig. 1(a), the dataset consists of two spiral structures as the two classes (*i.e.*, $\mathcal{Y}=\{1,2\}$). We compare the smoothness of PGP with the well-known GRF and the state-of-the-art MP (Subramanya & Bilmes, 2011), which are both graph-based methods albeit with different energy or cost functions.

Smoothness essentially implies that the *decision function* of a classifier changes slowly on a coherent structure (Zhou et al., 2003). A previously proposed decision function is $h(x_i) \triangleq (H_{1i}-H_{2i})/(H_{1i}+H_{2i})$, where H_{yi} is x_i's "score" for class y, as assigned by a method. Then, the decision rule is $\text{sign}(h(x_i))$, which is equivalent to the decision rule of every method compared here.

In Fig. 1(b1–3), we visualize the predictions made by the three methods, respectively. All methods use the same four points marked by $\times$ as labeled, whereas the rest are unlabeled. Their respective optimal parameters are adopted. The *color* of a point x_i represents the predicted class y_i, and the *size* of a point x_i represents the magnitude of the decision function at x_i, $|h(x_i)|$. Thus, a smoother decision function shall result in a sequence of points in more uniform sizes over each structure. Clearly, among the three methods, PGP generates points of the most uniform sizes, and is smooth nearly everywhere. Alternatively, we plot the decision function over the sequence of points in one of the structures in Fig. 1(c), which shows that PGP has a smoother decision function.

With better smoothness, PGP achieves a perfect result against the ideal classification in Fig. 1(a) (where the size of each point has no significance). In contrast, GRF and MP misclassify 4 and 2 points, respectively.

4. Theoretical Analysis

Error in $\hat{\pi}_y$. It is crucial that we can bound the error in the solution estimator $\hat{\pi}_y$, which is estimated from the samples $\mathcal{L}$ and $\mathcal{U}$.

We show that the expected error, $\mathbb{E}\left[\|\hat{\pi}_y-\pi_y\|_1\right]$, can be bounded by two terms, corresponding to the two types of error discussed in Sect. 3.3. Formally, as our solution is the stationary distribution of a Markov chain, the proof can be established based on the perturbation theory of Markov chains (Cho & Meyer, 2001; Seneta, 1993).

PROPOSITION 4 (ERROR): Given the two constraints (Eq. 11 and 13), for any constant $\epsilon \in (0,1)$,

$$\mathbb{E}\left[\|\hat{\pi}_y-\pi_y\|_1\right] \le O\left((1-\lambda_1)^{|\mathcal{U}|}\right) + O\left(\exp\left(-2\epsilon^2\lambda_2 \min_{x_i\in\mathcal{L}, p(y|x_i)>0} |\mathcal{L}_{x_i}|\right)\right), \quad (18)$$

where $\lambda_1 = \min_{x_i\in\mathcal{X}, p(x_i)>0} p(x_i)$, and $\lambda_2 = \min_{x_i\in\mathcal{L}, p(y|x_i)>0} p(y|x_i)^2$ are constants in $(0,1]$. ■

This result presents two major implications. First, both labeled and unlabeled data can help, as the bound improves when $\mathcal{L}$ or $\mathcal{U}$ grows. Second, the bound is fundamentally limited by $\mathcal{L}$. Given a fixed set of $\mathcal{L}$, even as $|\mathcal{U}| \to \infty$, we can achieve no better than the second error term. In other words, unlabeled data can only help so much. While our analysis is tailored to PGP, the result is consistent with previous analysis (Rigollet, 2007).

Robustness of π_y. Until now, we have assumed that the

graph construction function (Eq. 1) is perfect. If the graph were constructed differently (*i.e.*, perturbed), can we assess the robustness of our solution? In other words, do small perturbations only cause a small change in the solution?

In our perturbation model, every pairwise affinity W_{ij} can be perturbed by some scale factor $s > 1$. The goal is to show that the solution derived from the perturbed affinity matrix $\tilde{W}$ does not change much if s is small.

PROPOSITION 5 (ROBUSTNESS): Suppose a matrix $\tilde{W}$ is perturbed from W, such that for some $s > 1$, $W_{ij}/s \leq \tilde{W}_{ij} \leq W_{ij} \cdot s, \forall ij$. Let $\tilde{\pi}_y$ be the the solution vector based on $\tilde{W}$. It holds that $\|\tilde{\pi}_y - \pi_y\|_1 \leq O(s^2 - 1)$. ■

Here s is the *degree of perturbation* on W. The result implies that our solution is robust, for changes in the solution can be bounded by the degree of perturbation.

5. Experimental Evaluation

We empirically compare PGP with various SSL algorithms, and validate the claims in this paper.

Datasets. We use six public datasets shown in Fig. 2. Three of them, Digit1, Text and USPS, come from a benchmark (Chapelle et al., 2006). As the benchmark datasets are mostly balanced, we also use three datasets from UCI repository (Frank & Asuncion, 2010), namely, Yeast, ISOLET and Cancer[3]. Only a subset of Yeast (classes *cyt*, *me1*, *me2*, *me3*) and of ISOLET (classes *a*, *b*, *c*, *d*) are used. The benchmark datasets are taken without further processing. For the UCI datasets, feature scaling is performed so that all features have zero mean and unit variance.

Name	Task	Points	Features	Classes	Balanced
Digit1	synthetic digits	1500	241	2	yes
Text	newsgroups	1500	11960	2	yes
ISOLET	spoken letters	1200	617	4	yes
Cancer	breast cancer	569	30	2	no
USPS	written digits	1500	241	2	no
Yeast	protein sites	721	8	4	no

Figure 2. Summary of the datasets.

Graph. We construct a kNN graph (Chapelle et al., 2006), where k is a parameter to be selected. To instantiate Eq. 1, we use Euclidean distance for all datasets except Text, and Cosine distance for Text. σ is set to the average distance of all neighboring pairs on the graph.

Labeling. For a given $|\mathcal{L}|$, we sample 200 runs, where in each run $|\mathcal{L}|$ points are randomly chosen as labeled, and the rest are treated as unlabeled. The sampling ensures at least one labeled point for each class. 5% of the runs are reserved for model selection, and the remaining are for testing.

Evaluation. We evaluate the mean performance over the testing runs on each dataset. As classification accuracy is not a truthful measure of the predictive power on imbalanced datasets, we adopt *macro F-measure* (Forman, 2003) as the performance metric.

5.1. Comparison to Baseline Algorithms

We compare PGP to five state-of-the-art SSL algorithms, which have been shown (Zhu et al., 2003; Belkin et al., 2006; Subramanya & Bilmes, 2011) to significantly outperform earlier ones such as TSVM (Joachims, 1999) and SGT (Joachims, 2003).

- Gaussian Random Fields (GRF) (Zhu et al., 2003): a pioneering method based on Gaussian fields, equivalent to optimizing the squared loss.
- LapSVM (LSVM) (Belkin et al., 2006): an effective graph-based extension of SVM.
- Graph-based Generative SSL (GGS) (He et al., 2007): a probabilistic generative approach.
- Measure Propagation (MP) (Subramanya & Bilmes, 2011): a divergence-based optimization formulation over probability distributions.
- Partially Absorbing Random Walk (PARW) (Wu et al., 2012): a random walk method on graphs.

An existing implementation (Melacci & Belkin, 2011) is used for LSVM, whereas our own implementations are used for the others. Each algorithm integrates class priors as suggested in their respective work, if any.

Model selection is performed on the reserved runs. For each algorithm, we search $k \in \{5, 10, 15, 20, 25\}$ to construct the kNN graph. GRF and GGS has no other parameters. For LSVM, we search $\gamma_A \in \{$1e–6, 1e–4, .01, 1, 100$\}$, $r \in \{0,$ 1e–4, .01, 1, 100, 1e4, 1e6$\}$. For MP, we search $\alpha \in \{.5, 1, 5, 20, 100\}$, $u \in \{$1e–8, 1e–6, 1e–4, .01, .1, 1, 10$\}$, $v \in \{$1e–8, 1e–6, 1e–4, .01, .1$\}$. For PARW, we search $\alpha \in \{$1e–8, 1e–6, 1e–4, .01, 1, 100$\}$. For PGP, we search $\alpha \in \{.01, .02, .05, .1, .2, .5\}$.

The mean macro F-measures on the testing runs are reported in Fig. 3, leading to the following findings.

First, PGP performs the *best* or *not significantly different* from the best in 15 out of the 18 cases (*i.e.*, columns), whereas GRF, LSVM, GGS, MP and PARW perform as such in only 3, 6, 4, 7, 5 cases, respectively.

Second, while PGP has relatively stable performance across all the cases, the baselines can be volatile. In particular, when PGP is not the best, there is no consistent best method, which varies between LSVM, GGS and MP.

[3] It is known as "Breast Cancer Wisconsin (Diagnostic)" in the UCI repository.

	$\|\mathcal{L}\| = 10$						$\|\mathcal{L}\| = 20$						$\|\mathcal{L}\| = 150$					
	Digit1	Text	ISOLET	Cancer	USPS	Yeast	Digit1	Text	ISOLET	Cancer	USPS	Yeast	Digit1	Text	ISOLET	Cancer	USPS	Yeast
GRF	.894	.451	.627	.871	.638	.510	**.932**	.467	.686	.913	.682	.569	**.979**	.744	.849	**.958**	.902	.718
LSVM	.833	.428	.719	.886	**.698**	.562	**.935**	.472	**.780**	.914	.780	.614	**.979**	**.771**	.901	.956	.908	**.729**
GGS	.855	.567	.677	.867	.666	.540	.886	**.648**	.771	.918	.758	.578	.965	**.771**	**.905**	.948	.906	**.727**
MP	**.901**	.558	.692	.898	**.713**	.574	**.940**	.611	.735	**.924**	**.794**	.617	**.979**	.746	.854	**.957**	.913	.718
PARW	.881	**.587**	.721	.893	**.706**	.575	.923	**.640**	**.782**	.920	**.791**	.613	.975	.729	.897	.955	.916	.715
PGP	**.910**	**.592**	**.734**	**.910**	**.704**	**.593**	**.939**	.634	**.786**	**.927**	**.796**	**.633**	**.978**	.732	**.902**	**.958**	**.931**	.721

Figure 3. Performance comparison. In each column, the *best* result and those *not significantly different* ($p > .05$ in t-test) are bolded.

Third, PGP is especially advantageous with limited labeled data (*e.g.*, $|\mathcal{L}| = 10$), which is the very motivation of SSL. In contrast, when abundant data are labeled (*e.g.*, $|\mathcal{L}| = 150$), all algorithms perform better, and thus not surprisingly, the margin between them becomes smaller.

5.2. Integrating Class Priors

A concrete benefit of probabilistic modeling is to enable better integration of class priors, which is also probabilistic in nature. We demonstrate that principled integration of class priors is more effective than heuristics, and integrating more accurate priors helps.

Integration of priors. We compare two different methods of integrating class priors:

- BAYES: integrating in PGP in a Bayesian way (Eq. 15).
- CMN: integrating in GRF using the popular heuristic Class Mass Normalization (Zhu et al., 2003).

Note that BAYES and CMN respectively apply to a different algorithm as they are originally intended for. The priors are approximated in the same way for both methods, using the labeled points with add-one smoothing.

We study the corrective power of each method: integrating priors can be seen as "corrections" to the *base model* that does not incorporate priors. Directly assessing the improvement over the base model is unfair, since the base performances of PGP and GRF differ. Instead, we compute the F-score from the precision and recall of the corrections:

$$\text{precision} = \#\text{true corrections}/\#\text{corrections made} \quad (19)$$

$$\text{recall} = \#\text{true corrections}/\#\text{corrections needed} \quad (20)$$

The results are presented in Fig. 4(a) on the imbalanced datasets, which are more interesting given their non-uniform class priors. In all but one case, BAYES possesses much better corrective power than CMN.

More accurate priors. If class priors are integrated appropriately, using more accurate priors is expected to improve the performance. Suppose we know the exact priors by considering the labels of all points. We then apply the approximate and exact priors to PGP. We directly measure the performance with or without priors, given the same base model. The results are presented in Fig. 4(b), which illustrate that, while the estimated priors are effective in most cases, the supposedly more accurate exact priors can further improve the performance.

(a) Corrective power of different integration methods

	$\|\mathcal{L}\| = 10$			$\|\mathcal{L}\| = 20$			$\|\mathcal{L}\| = 150$		
	Cancer	USPS	Yeast	Cancer	USPS	Yeast	Cancer	USPS	Yeast
CMN	**.449**	.264	.310	.255	.275	.307	.087	.250	.084
BAYES	.333	**.564**	**.388**	**.373**	**.692**	**.504**	**.475**	**.781**	**.607**

(b) Using different priors on PGP

	$\|\mathcal{L}\| = 10$			$\|\mathcal{L}\| = 20$			$\|\mathcal{L}\| = 150$		
	Cancer	USPS	Yeast	Cancer	USPS	Yeast	Cancer	USPS	Yeast
None	.923	.636	.568	.927	.722	.600	.950	.873	.678
Approx	.910	.704	.593	.927	.796	.633	.958	.931	.721
Exact	**.929**	**.733**	**.622**	**.937**	**.805**	**.647**	**.960**	**.932**	**.730**

Figure 4. Effect of incorporating class priors in prediction.

6. Conclusion

We proposed a novel framework of Probabilistic Graph-based Pointwise Smoothness (PGP), hinging on the foundational data closeness and label coupling models. We further transformed such smoothness into a set of probability constraints, which can be solved uniquely to infer $p(Y|X)$. We also studied the theoretical properties of PGP in terms of its generalization error and robustness. Finally, we empirically demonstrated that PGP is superior to existing state-of-the-art baselines.

Acknowledgement

This material is based upon work partially supported by NSF Grant IIS 1018723, the Advanced Digital Science Center and the Multimodal Information Access and Synthesis Center of University of Illinois at Urbana-Champaign, and Agency for Science, Technology and Research of Singapore. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the author(s) and do not necessarily reflect the views of the funding agencies.

References

Azran, Arik. The rendezvous algorithm: Multiclass semi-supervised learning with markov random walks. In *ICML*, pp. 49–56, 2007.

Belkin, M., Niyogi, P., and Sindhwani, V. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *J. of Machine Learning Research*, 7:2399–2434, 2006.

Blum, A. and Chawla, S. Learning from Labeled and Unlabeled Data using Graph Mincuts. In *ICML*, pp. 19–26, 2001.

Chapelle, O., Schölkopf, B., and Zien, A. (eds.). *Semi-supervised learning*. MIT Press, Cambridge, MA, 2006.

Chen, Jie, Fang, Haw-ren, and Saad, Yousef. Fast approximate kNN graph construction for high dimensional data via recursive lanczos bisection. *J. of Machine Learning Research*, 10:1989–2012, 2009.

Cho, Grace E and Meyer, Carl D. Comparison of perturbation bounds for the stationary distribution of a Markov chain. *Linear Algebra and its Applications*, 335(1):137–150, 2001.

Das, D. and Smith, N.A. Graph-based lexicon expansion with sparsity-inducing penalties. In *NAACL-HLT*, 2012.

Forman, George. An extensive empirical study of feature selection metrics for text classification. *J. of Machine Learning Research*, 3:1289–1305, 2003.

Frank, A. and Asuncion, A. UCI machine learning repository. http://archive.ics.uci.edu/ml, 2010. University of California, Irvine, School of Information and Computer Sciences.

Getz, Gad, Shental, Noam, and Domany, Eytan. Semi-supervised learning–a statistical physics approach. In *ICML Workshop on Learning with Partially Classified Training Data*, 2005.

Goldreich, Oded. *A primer on pseudorandom generators*, volume 55. American Mathematical Society, 2010.

He, J., Carbonell, J., and Liu, Y. Graph-based semi-supervised learning as a generative model. In *IJCAI*, 2007.

Joachims, T. Transductive inference for text classification using support vector machines. In *ICML*, pp. 200–209, 1999.

Joachims, Thorsten. Transductive learning via spectral graph partitioning. In *ICML*, pp. 290–297, 2003.

Johnson, Rie and Zhang, Tong. Graph-based semi-supervised learning and spectral kernel design. *IEEE Transactions on Information Theory*, 54(1):275–288, 2008.

Lafferty, John and Wasserman, Larry. Statistical analysis of semi-supervised regression. In *NIPS*, 2007.

Melacci, Stefano and Belkin, Mikhail. Laplacian Support Vector Machines Trained in the Primal. *J. of Machine Learning Research*, 12:1149–1184, March 2011.

Rigollet, Philippe. Generalization error bounds in semi-supervised classification under the cluster assumption. *J. of Machine Learning Research*, 8:1369–1392, 2007.

Seneta, E. Sensitivity of finite markov chains under perturbation. *Statistics & probability letters*, 17(2):163–168, 1993.

Singh, Aarti, Nowak, Robert, and Zhu, Xiaojin. Unlabeled data: Now it helps, now it doesn't. In *NIPS*, pp. 1513–1520, 2008.

Singh, Sameer. 2D spiral pattern recognition with possibilistic measures. *Pattern Recognition Letters*, 19(2): 141–147, 1998.

Subramanya, A. and Bilmes, J. Semi-supervised learning with measure propagation. *J. of Machine Learning Research*, 12:3311–3370, 2011.

Szummer, Martin and Jaakkola, Tommi. Partially labeled classification with Markov random walks. In *NIPS*, pp. 945–952, 2001.

Wu, Xiao-Ming, Li, Zhenguo, So, Anthony M, Wright, John, and Chang, Shih-Fu. Learning with partially absorbing random walks. In *NIPS*, pp. 3086–3094, 2012.

Zhou, Dengyong, Bousquet, Olivier, Lal, Thomas Navin, Weston, Jason, and Schölkopf, Bernhard. Learning with local and global consistency. In *NIPS*, pp. 321–328, 2003.

Zhu, Xiaojin. Semi-supervised learning literature survey. Technical Report 1530, University of Wisconsin-Madison, 2005.

Zhu, Xiaojin and Ghahramani, Zoubin. Towards semi-supervised classification with markov random fields. Technical Report CMU-CALD-02-106, School of Computer Science, Carnegie Mellon University, 2002.

Zhu, Xiaojin, Ghahramani, Zoubin, and Lafferty, John. Semi-supervised learning using Gaussian fields and harmonic functions. In *ICML*, pp. 912–919, 2003.