

Diffusion on a Tensor Product Graph for Semi-Supervised Learning and Interactive Image Segmentation

Xingwei Yang

*Department of Computer and Information Sciences
Temple University
Philadelphia, PA 19122, USA*

XINGWEI@TEMPLE.EDU

Daniel B. Szyld

*Department of Mathematics
Temple University
Philadelphia, PA 19122, USA*

SZYLD@TEMPLE.EDU

Longin Jan Latecki

*Department of Computer and Information Sciences
Temple University
Philadelphia, PA 19122, USA*

LATECKI@TEMPLE.EDU

Editor: Peter Hawkes

Abstract

We derive a novel semi-supervised learning method that propagates label information as a symmetric, anisotropic diffusion process (SADP). Since the influence of label information is strengthened at each iteration, the process is anisotropic and does not blur the label information. We show that SADP converges to a closed form solution by proving its equivalence to a diffusion process on a tensor product graph. Consequently, we obtain a semi-supervised learning framework on a tensor product graph, which does not require any setting of the iteration number as a time scale, stopping parameter. The complexity of SADP is shown to be $\mathcal{O}(n^2)$, for n data points. The theoretical properties of SADP and presented experimental results demonstrate several advantages of SADP over previous diffusion-based and other classical graph-based semi-supervised learning algorithms. SADP is less sensitive to noise, outliers, and differences in the number of label data for different classes. In particular, we clearly demonstrate that the diffusion on the tensor product graph is superior to diffusion on the original graph in the context of semi-supervised learning. We also show that the proposed approach can be utilized in interactive image segmentation, which is also called semi-supervised image segmentation.

Keywords: Tensor Product Graph, Diffusion Process, Anisotropic Diffusion Process, Semi-supervised Learning, Tensor Product Graph Diffusion, Semi-supervised Image Segmentation

1. Introduction

Traditional classifiers use only labeled data (feature / label pairs) to train. Labeled instances however are often difficult, expensive, or time consuming to obtain, as they require the efforts of experienced human annotators. Meanwhile unlabeled data may be relatively easy

to collect, but there has been few ways to use them. Semi-supervised learning addresses this problem by using large amount of unlabeled data, together with a small amount of labeled data, to build better classifiers. Because semi-supervised learning requires less human effort and gives higher accuracy than unsupervised learning, it is of great interest both in theory and in practice.

Given are the data set $X = \{x_1, x_2, \dots, x_\ell, x_{\ell+1}, \dots, x_n\}$ and the label set $\mathcal{L} = \{1, 2, \dots, c\}$. The first ℓ points x_i ($i \leq \ell$) are labeled as $y_i \in \mathcal{L}$ and the remaining points x_u ($\ell + 1 \leq u \leq n$) are unlabeled. The goal of semi-supervised learning is to predict the labels y_u of the unlabeled points x_u .

There are many types of semi-supervised learning algorithms, such as generative models, Nigam et al. (2000); Fujino et al. (2005), self-training, Rosenberg et al. (2005); Culp and Michailidis (2007), co-training, Zhou et al. (2007); Blum and Mitchell (1998), and so on. All of them have similar but slightly different assumptions. For example, co-training assumes that (1) features can be split into two sets; (2) each sub-feature set is sufficient to train a good classifier; (3) the two sets are conditionally independent given the class. In self-training, a classifier is first trained with the small amount of labeled data. The classifier is then used to classify the unlabeled data. Different from all the above methods, the proposed algorithm can be classified as graph-based semi-supervised learning method, Zhou et al. (2003); Zhu et al. (2003); Szummer and Jaakkola (2001). In these methods, the relations between data are described by a weighted graph $G = (X, E, w)$. The graph is determined by an input set of data points $X = \{x_1, \dots, x_n\}$, where $E \subseteq X \times X$ represents the set of edges between the data points and the weight w assigns a nonnegative real number to every edge $w : X \times X \rightarrow \mathbb{R}_{\geq 0}$. Usually the weight w represents the strength of the connection or some similarity relation between the elements of X . To regularize the graph by its local structure a subgraph G_K of G is obtained by restricting each data point to connect only its K nearest neighbors, i.e., $w(x_i, x_j) = 0$ if x_j does not belong to the K nearest neighbors of x_i , Szummer and Jaakkola (2001); Jebara et al. (2009). Once the graph is constructed, the label information is propagated to unlabeled data points by balancing between the fitness to labeled data and the smoothness described by the constructed graph.

It has been pointed out that these methods are closely related to the diffusion process, Zhou et al. (2003); Szlam et al. (2008), even though they contain some fundamental differences. However, due to the limitations of the diffusion process, only a few diffusion-based methods have been proposed for semi-supervised learning, Szummer and Jaakkola (2001); Szlam et al. (2008). One of the main problems is the setting of time scale parameter t for the diffusion process. When $t \rightarrow \infty$, all the points become indistinguishable given that the weighted graph is connected. On the other hand, a small value of t cannot reveal the intrinsic structure of the data manifold. Thus, t controls the resolution at which we look at the data and it is essential for the performance of the diffusion process based algorithms, Szummer and Jaakkola (2001); Szlam et al. (2008). Another problem is that isotropic diffusion, which is usually utilized, smoothes out the weights of graph edges, and consequently, removes valuable information, in particular, that is related to labeled data points. This problem is closely related to the first one. The third problem is that the diffusion process is sensitive to outliers, which may make the label information propagate incorrectly. There have been some approaches in the literature that address these problems; in particular, Pan et al. (2004) propose solutions to the problems one and two, whose closed form solution is

quite similar to Zhou et al. (2003). However, to our best knowledge, there does not exist a satisfactory solution to problem three. We propose a radically different approach to address the three problems. We introduce a diffusion process on the tensor product of graph G_K with itself as a novel semi-supervised learning method.

We define a **tensor product graph** (see, e.g., Weichsel (1962)) as

$$G_K \otimes G_K = (X \times X, E \times E, \omega), \quad \text{where } \omega(x_\alpha, x_\beta, x_i, x_j) = w(x_\alpha, x_\beta) w(x_i, x_j).$$

Thus, the weight ω between nodes of $G_K \otimes G_K$ relates four data points. Our proposed approach consists of analyzing together these group of four points. The rationale for this is that since the edge weights of G_K represents the original similarity of pairs of data points, $G_K \otimes G_K$ represents the similarities of quadruplets of data points.

Since the weighted adjacency matrix of $G_K \otimes G_K$ is a $n^2 \times n^2$ matrix, the diffusion process on $G_K \otimes G_K$ may be computationally too demanding for large data sets. However, as we will show, the diffusion on $G_K \otimes G_K$ is equivalent to a **symmetric anisotropic diffusion process (SADP)** on the original graph G_K , which we also introduce in this paper. To be precise, we will prove that the iterative computation of SADP on G_K is equivalent to a closed form solution of the diffusion process on $G_K \otimes G_K$. As a consequence of this fact we also obtain a proof that the iterative computation of SADP is guaranteed to converge. In other words, instead of an $\mathcal{O}(n^6)$ method, we will deal with an $\mathcal{O}(n^3)$ method. In fact, we show that the new method is actually only $\mathcal{O}(n^2)$.

The key idea of SADP is to iteratively add a constant value to the labeled data in the diffusion matrix. This drastically changes the behavior of the diffusion process. In particular, this is the main reason why SADP is guaranteed to converge to a closed form solution, and consequently, the diffusion result is independent from the time scale parameter. At the same time, the smoothing out of the relevant information of labeled points is also removed. Consequently, SADP addresses the first two problems with the classical diffusion stated above. To address the third problem of sensitivity to outliers, SADP propagates the local similarity information in a symmetric way on the locally constrained graph G_K . A detailed explanation is provided in Section 3. We stress that all the properties of SADP, in particular, the symmetric propagation on G_K and adding a constant value to the labeled data, are necessary for the equivalence of SADP to the diffusion process on the tensor product graph. The main contributions of this paper are threefold:

- We propose a novel semi-supervised learning method that iteratively propagates label information as a symmetric, anisotropic diffusion process.
- We prove that this diffusion process is equivalent to a diffusion process on a tensor product graph.
- As a consequence of this proof we obtain that the iterative diffusion process is guaranteed to converge.

We also provide experimental evaluation that clearly demonstrates that the proposed diffusion outperforms state-of-the-art methods on many standard semi-supervised learning, test datasets. We would like to stress that, to best of our knowledge, this is the first time that diffusion process on a tensor product graph is utilized in the context of semi-supervised learning.

The rest of the paper is organized as follows: Closely related works are introduced in Section 2. The classical diffusion process and the symmetric diffusion process on the locally constrained graph are described in Section 3. The proposed iterative computation of SADP for semi-supervised learning is introduced in Section 4. The proposed diffusion process on the tensor product graph is described in Section 5. The equivalence of SADP and the tensor product graph diffusion is proved in Section 6. As a consequence we obtain that SADP converges to a closed form solution. Our experimental evaluation on toy examples and several benchmark datasets is presented in Section 8. Finally in Section 9 we present a very promising application of the proposed approach to interactive image segmentation.

2. Related Work

We mainly discuss closely related graph-based semi-supervised learning methods. A detailed survey of semi-supervised learning methods is available in Zhu (2008). Graph transduction methods have achieved state-of-art results in many applications. Two widely used classic methods are the Gaussian fields and harmonic functions, Zhu et al. (2003), and the local and global consistency, Zhou et al. (2003). In these methods, the label information is propagated to unlabeled data following the intrinsic geometry of the data manifold, which is described by the smoothness over the weighted graph connecting the data samples. With the similar motivation, graph Laplacian regularization terms are combined with regularized least squares (RLS) or support vector machine (SVM). These methods are denoted as Laplacian RLS (LapRLS) and Laplacian SVM (LapSVM), Belkin et al. (2006); Sindhwani et al. (2005). The above methods can be viewed as balancing between label consistency and smoothness over the graph. Many methods utilize the same intuition. Chapelle and Zien (2005) use a density-sensitive connectivity distance between nodes to reveal the intrinsic relation between data. Blum and Chawla. (2001) treat semi-supervised learning as a graph mincut problem. One problem with mincut is that it only gives hard classification without confidence. Joachims (2003) proposes a novel algorithm called spectral graph transducer, which can be viewed as a loss function with regularizer. To solve the problem of unstable label information, Wang et al. (2008) propose to minimize a novel cost function over both a function on the graph and a binary label matrix. They provide an alternating minimization scheme that incrementally adjusts the function and the labels towards a reliable local minimum. They solve the imbalanced label problem by adding node regularizer for labeled data.

There are some other works focusing on different aspects of graph-based semi-supervised learning. A transductive algorithm on directed graph is introduced in Zhou et al. (2005). Zhou et al. (2006) propose to formulate relational objects using hypergraphs, where an edge can connect more than two vertices, and extend spectral clustering, classification and embedding to such hypergraphs. Nadler et al. (2009) discuss the limit behavior of semi-supervised learning methods based on the graph Laplacian.

The proposed method belongs to diffusion-based semi-supervised learning methods. Szummer and Jaakkola (2001) introduce a graph transduction algorithm based on the diffusion process. Recently, Szlam et al. (2008) improve the algorithm by considering the geometry of the data manifold with label distribution. However, neither of them solve the common problems with the diffusion process; see Section 1.

The most closely related work to the proposed approach on a tensor product graph is the diffusion kernel defined by Kondor and Lafferty (2002) and Vishwanathan et al. (2010). However, their construction of diffusions over the tensor product graph is completely different from the one proposed here. Moreover, Kondor and Lafferty (2002) and Vishwanathan et al. (2010) focus on defining new kernels, whereas we derive a novel semi-supervised learning framework on the tensor product graph.

3. The Diffusion Process

From graph G defined by (X, E, w) , one can construct a reversible Markov chain on X . The degree of each node and the transition probability are defined as

$$D(x_i) = \sum_{j=1}^n w(x_i, x_j) \quad \text{and} \quad P_{ij} = \frac{w(x_i, x_j)}{D(x_i)}.$$

It follows that the transition probability matrix P inherits the positivity-preserving property and that P is stochastic:

$$\sum_{j=1}^n P_{ij} = 1, \quad i = 1, \dots, n.$$

From a data analysis point of view, the reason for studying this diffusion process is that the matrix P contains geometric information about the data set X . Indeed, the transitions that it defines directly reflect the local geometry defined by the immediate neighbors of each node in the graph of the data. In other words, P_{ij} represents the probability of transition in one time step from node x_i to node x_j and it is proportional to the edge-weight $E(x_i, x_j)$. For $t \geq 0$, the probability of transition from x_i to x_j in t time steps is given by P_{ij}^t , which is the t th power P^t of P . One of the main ideas of the diffusion framework is that the chain running forward in time, or equivalently, taking larger powers of P , allows us to propagate the local geometry, and therefore, reveals relevant geometric structures of X at different scales, where t plays the role of a scale parameter. However, the performance is closely dependent on t . If t is too small, diffusion process cannot reveal the intrinsic geometric relation. On the other hand, if t is too large, it is known that diffusion process will reach a stable situation, which loses the discriminability. As we shall see, this problem is solved by the method proposed here.

In the original diffusion process setting, all paths between nodes x_i and x_j are considered to describe the probability of a walk from x_i to x_j . If there are several noisy nodes, the paths passing through these nodes will affect this probability, Yang et al. (2009). A natural way to solve this problem is to restrict the diffusion process within the K nearest neighbors (KNN) for each data. This can be done by defining a modified transition probability $P_K(x_i, x_j)$ from x_i to x_j by:

$$(P_K)_{ij} = \begin{cases} P_{ij} & x_j \in KNN(x_i) \\ 0 & \text{otherwise.} \end{cases} \quad (1)$$

Clearly, the graph of the matrix P_K is precisely G_K defined in Section 1.

In the presence of noise, which is the case considered in this paper, the rows of P are pretty full, i.e., they have very few zero elements. Therefore, every row has some elements

outside the set of KNN, for any reasonable value of $K < n$, and these are the nonzeros which are being removed. It follows then, that

$$\sum_{j=1}^n (P_K)_{ij} < 1, \quad i = 1, \dots, n. \quad (2)$$

We emphasize that we do not renormalize P_K to a stochastic matrix. In Szummer and Jaakkola (2001), the diffusion process is also restricted by the neighborhood structure. However, after P_K is obtained by (1), they re-normalize P_K into stochastic matrix by $(P_K^S)_{ij} = (P_K)_{ij} / \sum_j (P_K)_{ij}$. Though it seems like this is only a minor difference between their method and ours, we stress that it makes theoretically a fundamental difference. As we show in Section 5, when a stochastic matrix P_K^S is used, the convergence of the proposed method cannot be guaranteed.

By replacing the P by P_K the effect of noise is reduced, but the diffusion process is still not robust enough to noise.

In order to solve this problem, Yang et al. (2009) consider the paths between the KNN of x_i and the KNN of x_j simultaneously, which can be viewed as a soft and symmetric measure of their KNN compatibility. They have demonstrated the advantage of their algorithm by improving the performance of shape retrieval. Following them, we define a symmetric version of the diffusion process as

$$P_{KK}^{(t+1)} = P_K P_{KK}^{(t)} (P_K)^T,$$

where $P_{KK}^{(1)} = P$, and $(P_K)^T$ is the transpose of P_K , and call it **symmetric locally constrained diffusion process (SLCDP)**.

We now show how SLCDP solves the problem of points x_i and x_j being in the same dense cluster, but without any common KNNs. Let x_k and x_ℓ be two different neighbors of x_i and x_j respectively, i.e., $x_k \in KNN(x_i)$ and $x_\ell \in KNN(x_j)$ and $x_k \neq x_\ell$. Since x_i and x_j belong to the same dense cluster, x_k and x_ℓ are very likely to be similar to each other. Exactly this property is utilized by SLCDP to increase the similarity of x_i and x_j . To see this, let us consider a single iteration of SLCDP

$$P_{KK}^{(t+1)}(x_i, x_j) = \sum_{k \in KNN(x_i), \ell \in KNN(x_j)} P(x_i, x_k) P_{KK}^{(t)}(x_k, x_\ell) P(x_j, x_\ell)$$

Consequently, the similarity between x_i and x_j will be correctly increased by SLCDP. This property explains why SLCDP on G_K is more robust to noise and outliers than the classical diffusion process both on G_K and on G . This property is also confirmed by the experimental results in Yang et al. (2009) and Temlyakov et al. (2010) that demonstrate that SLCDP performs better than the original diffusion process for shape retrieval.

4. Semi-supervised Locally Constrained Diffusion Process

We introduce in this section the novel symmetric anisotropic diffusion process for semi-supervised learning. We construct a diagonal $n \times n$ matrix Δ such that the diagonal entries of the labeled data points are set to one and all other entries are set to zero, i.e.,

$$\Delta(i, i) = \begin{cases} 1 & i = 1, \dots, \ell \\ 0 & \text{otherwise,} \end{cases} \quad (3)$$

or equivalently,

$$\Delta = \begin{bmatrix} I_\ell & O \\ O & O \end{bmatrix},$$

where I_ℓ is the identity of order ℓ .

The proposed **symmetric anisotropic diffusion process (SADP)** is defined as

$$Q^{(t+1)} = P_K Q^{(t)} (P_K)^T + \Delta \quad (4)$$

where $Q^{(1)} = P$. We can iterate (4) until convergence and denote the limit matrix by $Q^* = \lim_{t \rightarrow \infty} Q^{(t)}$. The proof of the convergence of (4) and a closed form expression for the unique solution Q^* are given in Theorem 2 in Section 6.

The key difference of the proposed SADP in comparison to SLCDP is the fact that at each iteration the influence of labeled data points is increased. Since SLCDP does not consider any labeled data point, its diffusion is still isotropic. The proposed addition of labeled data points makes SADP an anisotropic diffusion process. This algorithm can be intuitively understood as spreading the heat from the labeled data to the unlabeled data while at the same time adding a constant heat source at the labeled data.

We utilize Q^* to classify the unlabeled data points following a simple classification strategy as in Zhou et al. (2003). The classification of a point x_i is based on its average similarity to all labeled data points that hold the same label. Let $\mathcal{X}_\lambda \subset \{x_1, x_2, \dots, x_\ell\}$ be the set of labeled data points with the same label λ , i.e., $x_k \in \mathcal{X}_\lambda$ iff $y_k = \lambda$, for $\lambda \in \mathcal{L} = \{1, 2, \dots, c\}$. We define the average strength of label λ for x_u as

$$F_\lambda(x_u) = \frac{\sum_{x_k \in \mathcal{X}_\lambda} Q^*(x_u, x_k)}{|\mathcal{X}_\lambda|}. \quad (5)$$

This average strength $F_\lambda(x_u)$ can be interpreted as the average influence of the labeled data in class λ on datum x_u . The normalization by the number of labeled points in class λ makes the final classification robust to differences in the number of label data in different classes. Therefore, the vector $F(x_u) = (F_1(x_u), \dots, F_c(x_u))$ represents normalized label strengths for different classes. Finally, we assign to x_u the label with the greatest strength, i.e., $y_u = \operatorname{argmax}\{F_\lambda(x_u) | \lambda = 1, \dots, c\}$. We note that it is extremely unlikely that there be two labels with identical maximal strength; but if this were the case, one can randomly assign one of them.

5. Tensor Product Graph Diffusion

We introduce in this section a novel diffusion process on the tensor product graph. Its equivalence to SADP is proved in Section 6. We begin with some preliminary notation. Given an $n \times n$ matrix B , we define a $n^2 \times n^2$ **supermatrix** \mathbb{B} as the matrix with elements $\mathbb{B}_{\alpha\beta, ij} = b_{\alpha\beta} b_{ij}$ i.e., $\mathbb{B} = B \otimes B$, where \otimes denotes the Kronecker product of matrices; see, e.g., Jain (1989); Lancaster and Rodman (1995); van Loan (2000). The operator $\operatorname{vec} : \mathbb{R}^{n \times n} \rightarrow \mathbb{R}^{n^2}$ is defined as $\operatorname{vec}(B)_k = B_{ij}$, where $i = \lfloor (k-1)/n \rfloor + 1$ and $j = k \bmod n$. The inverse operator vec^{-1} which maps a vector into a matrix is often called the reshape operator¹. The following is a very useful identity:

$$\operatorname{vec}(B S B^T) = (B \otimes B) \operatorname{vec}(S) = \mathbb{B} \operatorname{vec}(S). \quad (6)$$

1. The operator vec applies as well to rectangular matrices, but this is not needed here.

If we let $A = B S B^T$, we can write this identity as

$$\text{vec}(A) = \begin{pmatrix} a_{11} \\ a_{12} \\ \cdot \\ a_{1n} \\ \cdot \\ a_{\alpha\beta} \\ \cdot \\ a_{n1} \\ a_{n2} \\ \cdot \\ a_{nn} \end{pmatrix} = \mathbb{B} \cdot \text{vec}(S) = \begin{pmatrix} b_{11}b_{11} & b_{11}b_{12} & \cdot & \cdot & b_{1n}b_{1n} \\ b_{11}b_{21} & b_{11}b_{22} & \cdot & \cdot & b_{1n}b_{2n} \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ b_{11}b_{1n} & b_{11}b_{2n} & \cdot & \cdot & b_{1n}b_{nn} \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & b_{\alpha\beta}b_{ij} & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ b_{n1}b_{11} & b_{n1}b_{12} & \cdot & \cdot & b_{nn}b_{1n} \\ b_{n1}b_{21} & b_{n1}b_{22} & \cdot & \cdot & b_{nn}b_{2n} \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ b_{n1}b_{n1} & b_{n1}b_{n2} & \cdot & \cdot & b_{nn}b_{nn} \end{pmatrix} \cdot \begin{pmatrix} s_{11} \\ s_{12} \\ \cdot \\ s_{1n} \\ \cdot \\ s_{ij} \\ \cdot \\ s_{n1} \\ s_{n2} \\ \cdot \\ s_{nn} \end{pmatrix}. \quad (7)$$

We will use the identity (6) for $B = P_K$ and $S = \Delta$, and for simplicity of notation we will use $\mathfrak{s} = \text{vec}(\Delta)$.

With this notation, we observe that \mathbb{B} is the adjacency matrix of the tensor product graph $G_K \otimes G_K$ (Weichsel (1962)), which we introduced in Section 1. We define a **tensor product diffusion process (TPDP)** on $G_K \otimes G_K$ at discrete time t as

$$\sum_{i=0}^t \mathbb{B}^i \mathfrak{s}. \quad (8)$$

Theorem 1. The tensor product diffusion process converges to a closed form solution given by

$$\lim_{t \rightarrow \infty} \sum_{i=0}^t \mathbb{B}^i \mathfrak{s} = (I - \mathbb{B})^{-1} \mathfrak{s} \quad (9)$$

Proof The identity (9) holds if and only if the maximum of the absolute values of the eigenvalues of \mathbb{B} is smaller than 1. Since \mathbb{B} has nonnegative entries, this maximum is smaller than or equal to the maximum of the row-wise sums of matrix \mathbb{B} , Varga (1962). Therefore, it is sufficient to show that the sum of each row of \mathbb{B} is smaller than 1, i.e., $\sum_{\beta, j} \mathbb{B}_{(\alpha\beta, ij)} < 1$, where β, j both range from 1 to n . Since $B = P_K$, we have

$$\sum_{\beta j} \mathbb{B}_{(\alpha\beta, ij)} = \sum_{\beta j} b_{\alpha\beta} b_{ij} = \sum_{\beta} (P_K)_{\alpha\beta} \sum_j (P_K)_{ij} < 1 \quad (10)$$

where the last inequality follows from (2), applied twice. This completes the proof.

We stress that the fact that (2) is essential for the proof of Theorem 1. Thus, the fact that we replace matrix P with P_K (in Section 3) for $K < n$ is an important step in our framework. It is not only intuitively justified, since usually only the similarities to nearest neighbors are reliable, but also essential for the convergence of TPDP. In contrast, in Szummer and Jaakkola (2001) and Szlam et al. (2008) the sum of each row of the truncated matrix P_K is renormalized to be equal to 1, so that P_K remains a stochastic matrix. If $\sum_{j=1}^n (P_K)_{ij} \geq 1$ even for one i , the sum in (9) would not converge. This fact shows that the proposed approach is fundamentally different from approaches in Szummer and Jaakkola (2001) or Szlam et al. (2008).

6. Equivalence of SADP and TPDP

The goal of this section is to prove that the symmetric anisotropic diffusion process (SADP) and the tensor product diffusion process (TPDP) are equivalent.

Theorem 2 SADP and TPDP are equivalent, i.e.,

$$\text{vec} \left(\lim_{t \rightarrow \infty} Q^{(t+1)} \right) = \lim_{t \rightarrow \infty} \sum_{i=0}^{t-1} \mathbb{B}^i \mathfrak{s} = (I - \mathbb{B})^{-1} \mathfrak{s} \quad (11)$$

where $\mathbb{B} = P_K \otimes P_K$ and $\mathfrak{s} = \text{vec}(\Delta)$. Consequently, SADP proposed in (4) converges to Q^* defined as

$$Q^* = \text{vec}^{-1} (I - \mathbb{B})^{-1} \mathfrak{s}, \quad (12)$$

Proof: We rewrite (4) as

$$\begin{aligned} Q^{(t+1)} &= P_K Q^{(t)} P_K^T + \Delta = P_K (P_K Q^{(t-1)} P_K^T + \Delta) P_K^T + \Delta \\ &= P_K^2 Q^{(t-1)} (P_K^T)^2 + P_K \Delta P_K + \Delta = \dots \\ &= P_K^t P (P_K^T)^t + P_K^{t-1} \Delta (P_K^T)^{t-1} + \dots + \Delta \\ &= P_K^t P (P_K^T)^t + \sum_{i=0}^{t-1} P_K^i \Delta (P_K^T)^i \end{aligned} \quad (13)$$

Lemma 1 below shows that the first summand in (13) converges to zero. Hence by Lemma 1 and by (13) we obtain

$$\lim_{t \rightarrow \infty} Q^{(t+1)} = \lim_{t \rightarrow \infty} \sum_{i=0}^{t-1} P_K^i \Delta (P_K^T)^i. \quad (14)$$

It remains to consider the second summand in (13). Lemma 2 below states that $\text{vec}(P_K^i \Delta (P_K^T)^i) = \mathbb{B}^i \mathfrak{s}$. Thus, we obtain that

$$\text{vec} \left(\sum_{i=0}^{t-1} P_K^i \Delta (P_K^T)^i \right) = \sum_{i=0}^{t-1} \mathbb{B}^i \mathfrak{s}. \quad (15)$$

It follows from (14), (15), and Theorem 1 that

$$\text{vec} \left(\lim_{t \rightarrow \infty} Q^{(t+1)} \right) = \text{vec} \left(\lim_{t \rightarrow \infty} \sum_{i=0}^{t-1} P_K^i \Delta (P_K^T)^i \right) = \lim_{t \rightarrow \infty} \sum_{i=0}^{t-1} \mathbb{B}^i \mathfrak{s} = (I - \mathbb{B})^{-1} \mathfrak{s} = \text{vec}(Q^*). \quad (16)$$

This proves the theorem.

Lemma 1 $\lim_{t \rightarrow \infty} P_K^t P (P_K^T)^t = 0$

Proof: It suffices to show that P_K^t and $(P_K^T)^t$ go to 0, when $t \rightarrow \infty$. This is true if and only if every eigenvalue of P_K is less than one in absolute value. Since P_K has nonnegative

entries, this holds if its row sums are all less than one, Varga (1962). But this follows directly from (2) and the proof is complete.

Lemma 2 $vec(P_K^i \Delta (P_K^T)^i) = \mathbb{B}^i \mathfrak{s}$ for $i = 1, 2, \dots$

Proof: Our proof is by induction. Suppose $vec(P_K^k \Delta (P_K^T)^k) = \mathbb{B}^k \mathfrak{s}$ is true for $i = k$, then for $i = k + 1$ we have

$$\begin{aligned} vec\left(P_K^{k+1} \Delta (P_K^T)^{k+1}\right) &= vec\left(P_K (P_K^k \Delta (P_K^T)^k) P_K^T\right) \\ &= vec\left(P_K vec^{-1}(\mathbb{B}^k \mathfrak{s}) P_K^T\right) = \mathbb{B} \mathbb{B}^k \mathfrak{s} = \mathbb{B}^{k+1} \mathfrak{s}, \end{aligned}$$

and the proof is complete.

7. SADP Algorithm

Converting expression (14) to an iterative algorithm is simple:

```

Compute  $W = \Delta$ .
Compute  $T = P$ .
  For  $i = 1, 2, \dots$ 
    Compute  $T \leftarrow P_K T P_K$ 
    Compute  $W \leftarrow W + T$ 
  end
    
```

This algorithm requires two full $n \times n$ matrix multiplications per step and thus has $O(n^3)$ time complexity, which is of course much more efficient than using directly the TPDP (8) which uses matrices of order n^4 . Now we propose a more efficient algorithm that takes advantage of the special form of Δ matrix in (3), since usually the number of label points $\ell \ll n$.

Let L be a $n \times \ell$ matrix containing the first ℓ columns of P_K . We write $P_K = [L|R]$ and $P_K \Delta = [L|0]$. It follows then that $P_K \Delta P_K^T = LL^T$. Furthermore, if we denote $P_K^j \Delta (P_K^T)^j = L_j L_j^T$, with L_j being $n \times \ell$, it follows that

$$P_K^{j+1} \Delta (P_K^T)^{j+1} = P_K (P_K^j \Delta (P_K^T)^j) P_K^T = P_K L_j L_j^T P_K^T = (P_K L_j) (P_K L_j)^T = L_{j+1} L_{j+1}^T.$$

We are now ready to propose our more efficient algorithm:

```

Compute  $W = \Delta$ .
Compute  $L =$  first  $\ell$  columns of  $P_K$ 
Compute  $W \leftarrow W + LL^T$ .
  For  $i = 2, 3, \dots$ 
    Compute  $L \leftarrow P_K L$ 
    Compute  $W \leftarrow W + LL^T$ 
  end
    
```

As it can be appreciated, we replaced one of the two $n \times n$ matrix products with one matrix product between an $n \times n$ and an $n \times \ell$ matrix, and the other with a product of an $n \times \ell$ by an $\ell \times n$ matrix. Consequently, the complexity of the multiplication step can be

reduced from $\mathcal{O}(n^3)$ to $\mathcal{O}(\ell \times n^2)$. This is a significant speed gain in practical applications, where usually $\ell \ll n$. In fact, in practice, we perform a fixed number of iterations, and it follows that this algorithm is $\mathcal{O}(n^2)$.

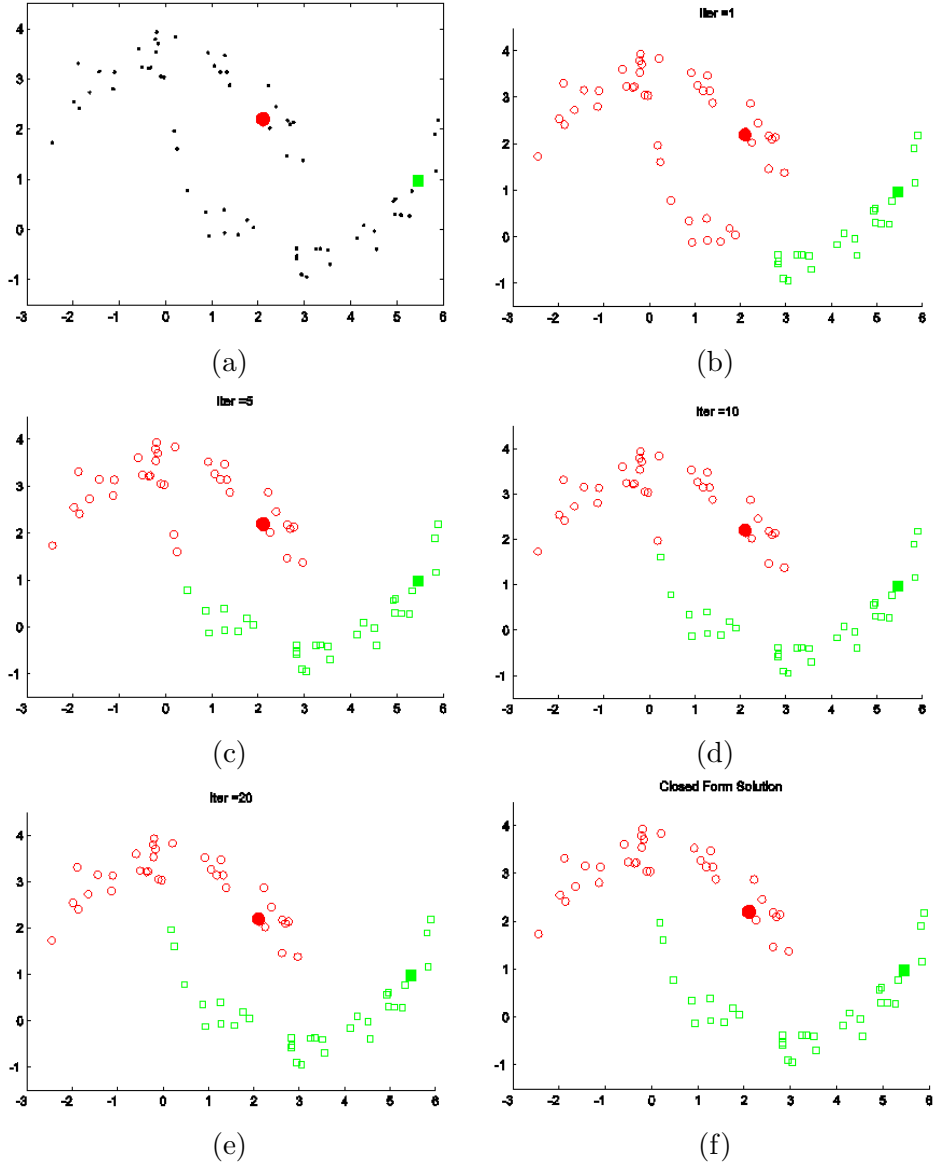


Figure 1: Classification on the pattern of two moons. The convergence process of our iterative SADP algorithm with t increasing from 1 to 20 is shown from (b) to (e). (f) shows the results of closed form solution of the supermatrix diffusion TPDP.

8. Experimental Results

In this section we evaluate the performance of the proposed SADP algorithm and compare it to other classic semi-supervised learning algorithms: Gaussian Fields and Harmonic Functions (GFHF), Zhu et al. (2003), Local and Global Consistency (LGC), Zhou et al. (2003), and classic diffusion (CD), Szummer and Jaakkola (2001), with the same parameter setting. It is known that GFHF is equal to Label Propagation, Zhu (2005).

We iterate SADP until convergence, which is guaranteed by our theoretical results (Theorem 2). We do not use the equivalent closed form solution of TPDP, since it is time consuming to calculate the inverse of $I - \mathbb{B}$ due to the size of supermatrix \mathbb{B} ($n^2 \times n^2$).

8.1 Toy Example

We first consider a widely used two half moon dataset as a toy example shown in Figure 1(a), where two labeled data points are marked as red and green dots. The affinity matrix is obtained by a Gaussian Kernel with $\sigma = 0.3$. The K for K -nearest neighbors is set to 12. The convergence process of our iteration algorithm with t increasing from 1 to 20 is shown in Figure 1(b)-1(e). It is clear that the initial label information is diffused along the moons. To demonstrate the supermatrix diffusion is able to get the same result as in (e), the result of closed form solution is shown in Figure 1(f).

8.2 Real Benchmark Datasets

We report on experiments with real benchmark data sets given by Chapelle et al. (2006). Seven datasets (BCI, Digital, g241c, g241n, USPS, COIL and TEXT) are used. All of these datasets contain two classes except COIL, which contains six classes. We use the originally suggested data splits to conduct our experiments for fair comparison, where each data set is associated with 12 different partitions of labeled and unlabeled subsets. The test errors are averaged over the trials for each data set. For all experiments, both 10 and 100 labeled samples are tested.

To fairly compare to the three well-known semi-supervised learning methods, we regularize the graph by its K nearest neighbors, which means that for each data point only the connections between it and its K nearest neighbors are kept. The Euclidean distances between data point are used for all the datasets. We use Gaussian kernel to convert distances to affinities with the kernel size $\sigma = \bar{d}_K/3$, where \bar{d}_K is the average distance between each sample and its K th nearest neighbor, Jebara et al. (2009). The parameter for these experiments is uniformly set as $K = 12$. We stress that we do not add any prior knowledge in the experiment, such as the class ratio.

The experimental results with the error rates averaged over the 12 different partitions with 10 and 100 labels are shown in Tables 1 and 2, respectively. The proposed method significantly outperforms the other three methods on 3 datasets with 10 labels and 5 datasets with 100 labels. Our error rate on COIL is 3 times smaller than LGC and over 7 times smaller than GFHF and CD in both cases. Besides, it is obvious that CD, which is based on classic diffusion, Szummer and Jaakkola (2001), has the worst performance on all datasets. Compared to it, SADP improves the performance significantly on all 7 datasets, which demonstrates a clear advantage of the proposed diffusion on the tensor product graph over

Table 1: Experimental results on the benchmark data sets (in terms of % error rate) for the variety of graph transduction algorithms with 10 labeled data. GFHF, Zhu et al. (2003), LGC, Zhou et al. (2003), CD, Szummer and Jaakkola (2001), and the proposed SADP.

	BCI	Digital	g241c	g241n	USPS	COIL	TEXT
GFHF	50.32	24.26	50.1	49.72	19.92	32.4	49.88
LGC	49.91	13.39	48.9	48.9	15.61	10.74	41.96
CD	50.3	40.91	50.06	50.21	19.98	27.75	50.02
Proposed SADP	50	13.17	48.22	48.15	19.73	3.14	49.83

Table 2: Experimental results on the benchmark data sets (in terms of % error rate) for the variety of graph transduction algorithms with 100 labeled data. GFHF, Zhu et al. (2003), LGC, Zhou et al. (2003), CD, Szummer and Jaakkola (2001), and the proposed SADP.

	BCI	Digital	g241c	g241n	USPS	COIL	TEXT
GFHF	48.03	2.17	46.25	42.52	11.5	22.51	37.79
LGC	48.86	2.56	45.61	41.14	13.57	10.7	31.83
CD	51.11	29.4	50.21	50.21	20.01	27.76	49.92
Proposed SADP	47.03	3.82	44.5	41.43	5.85	3.11	30.21

diffusion on the original graph. Furthermore, in the few cases where the new SPDP method is not the best, its error rate is very close to the best. The same cannot be said to hold for any of the other methods compared. In other words, there is no harm in using SPDP for all types of data configuration.

8.3 Imbalanced Ratios of Label Points

Since one cannot expect that the number of labeled data points is equal for all classes, an important issue in semi-supervised learning is robustness to differences in the number of labeled examples, Wang et al. (2008).

To illustrate the advantage of SADP on imbalanced label data, we first consider a noisy two half moon dataset shown in Figure 2(a), where one class contains 2 labeled data and the second class contains 50 labeled data marked with circles. Figures 2(b)-2(e) show the final classification results, which are 82.3% for GFHF, 67.7% for LGC, 60.22% for CD, and 99.27% for the proposed method. The K for K -nearest neighbors is set to 5 and the affinity matrix is obtained by the method introduced in Szlam et al. (2008).

Moreover, to perform tests on real data sets, we changed the ratio of labeled examples in two benchmark datasets from Chapelle et al. (2006), Digital and USPS. We fix the number of labeled points for one class to 4 and vary the number of labeled data points

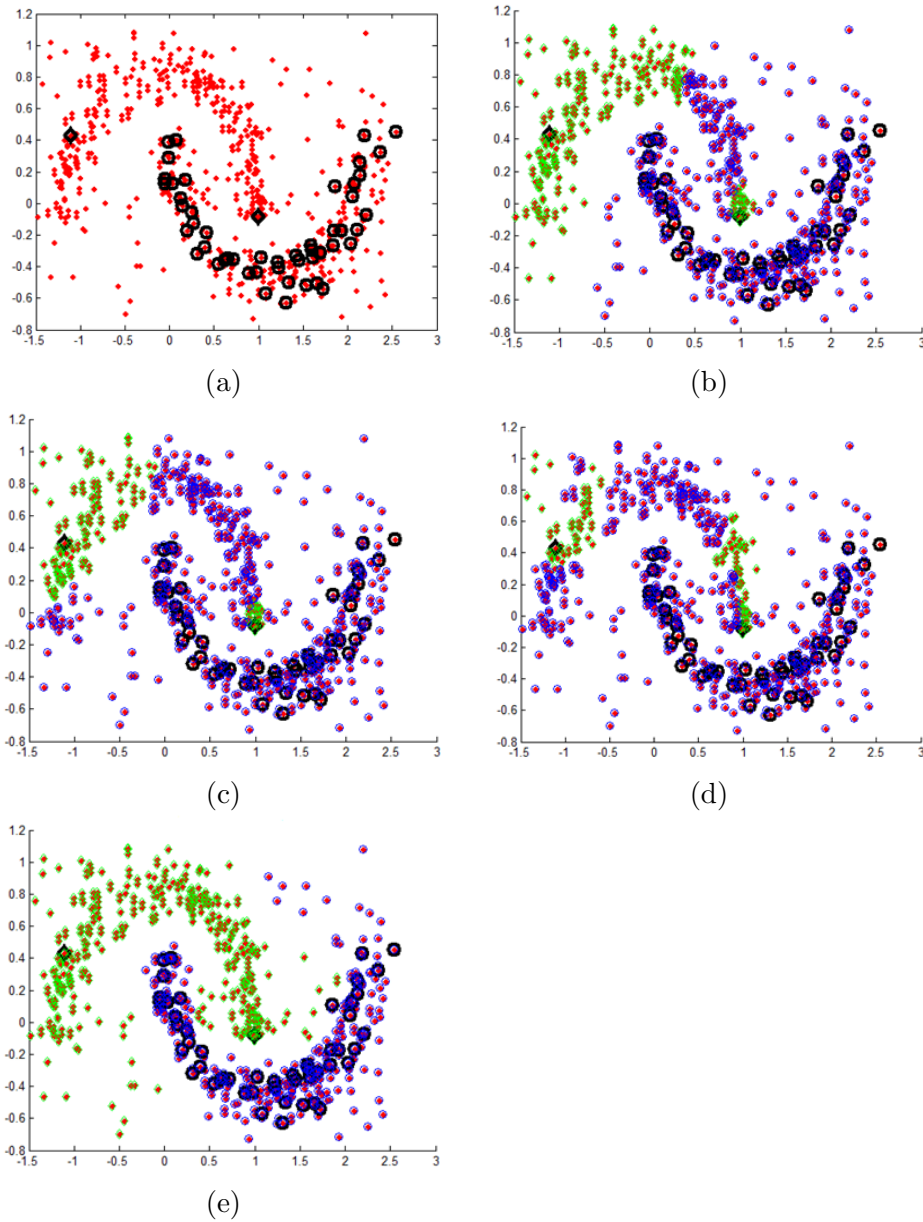
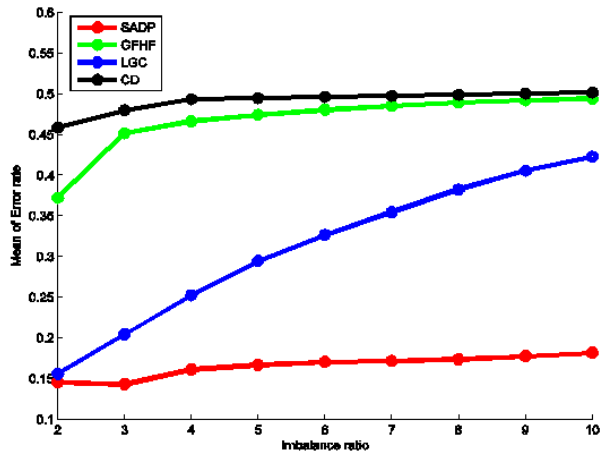
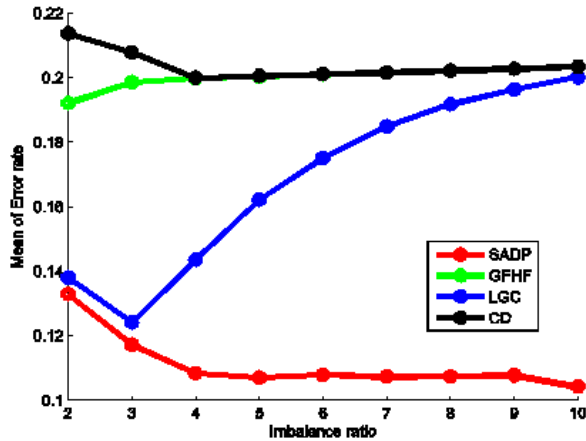


Figure 2: Classification of two noisy moons with imbalanced number of label data marked with black circles in (a); the final results are shown in (b) GFHF Zhu et al. (2003); (c) LGC Zhou et al. (2003); (d) CD Szummer and Jaakkola (2001); (e) the proposed method.

for the second class to $4 \times r$ for $r = 2, 3, \dots, 10$. Thus, r represents the imbalance ratio. We report the average classification error over 12 different partitions of these benchmark datasets in Figure 3. We set the parameters the same as the ones introduced in Section 8.2.



(a)



(b)

Figure 3: The error rate as a function of imbalance ratio r of label points shown on the horizontal axis. We compare the proposed SADP to GFHF, LGC and CD on two datasets: (a) Digital (b) USPS.

As illustrated by the red curve in Figure 3(b) for $r = 10$, the proposed SADP is not only stable but even can improve the error rate in the extremely imbalanced setting. This is due to the increasing number of total labeled data. In contrast, even though more information is provided, the error rate of LGC (blue curve) increases significantly when the imbalance ratio increases. Meanwhile, though not so obvious, the error rate of GFHF (green curve) also increases with the increasing imbalance ratio. CD is relatively robust to the imbalance ratio, but its error rates are significantly higher.

8.4 Large Number of Classes

The exceptional performance of SADP on COIL (Section 8.2), which is the only dataset with six classes, indicates that SADP is well suited for datasets with many classes. To evaluate this claim, we present results on a dataset with 70 different classes. It is the MPEG-7 CE-Shape-1 dataset that is widely used for shape retrieval. It contains 1400 binary shapes that are grouped into 70 classes; each class contains 20 shapes. Some example shapes are shown in Figure 4.



Figure 4: Typical shape images from the MPEG7 CE-Shape-1, one image from each class.

The distance between shapes is obtained by Shape Context Belongie et al. (2002), which is a well-known shape distance measure. This dataset is challenging for semi-supervised learning, because the distance distributions vary significantly among different classes. Since there are only 20 shapes in each class, we only use one label shape for each class. We randomly generate 100 sets of labeled data and the test error is averaged over the 100 trials. To fairly compare to other methods, all the parameters are the same for different methods. The error rates in classification with 70 classes are shown in Table 3. The proposed method performs better than GFHF Zhu et al. (2003) and LGC Zhou et al. (2003) and it performs much better than the classic diffusion process Szummer and Jaakkola (2001).

Table 3: The error rate in classification with 70 classes on the MPEG-7 shape dataset. Only one label datum was used for each of the 70 classes.

GFHF	LGC	CD	SADP
11.7	10.48	15.58	9.41

9. Semi-Supervised Image Segmentation

Semi-supervised Image Segmentation is also known as interactive segmentation, where some initial seeds for segmentation is provided by users. Semi-supervised segmentation methods, inspired by the user-inputs such as scribbles, which provide a partial labeling of the image, have gained popularity since these methods give the user the ability to constrain the segmentation as necessary for a particular application, Boykov and Jolly (2001); Criminisi et al. (2008); Kohli et al. (2008); Duchenne et al. (2008).

To adapt the proposed algorithm for semi-supervised image segmentation, the construction of the graph is quite critical, since it affects the affinities learned by the algorithm. Each node in the graph is an image region, called superpixel. The main benefit of superpixels as compared to image pixels is that the superpixels are more informative than pixels, Ren and

Malik (2003). Furthermore, the usage of superpixels reduces time complexity and memory requirements.

As is well known, a single image quantization is not suitable for all categories, Ladicky et al. (2009); different levels of image quantization should be considered together. Moreover, image segmentation is known to be unstable, since it is strongly affected by small image perturbations, feature choices, or parameter settings. This instability has led to advocacy for using multiple segmentations of an image, Pantofaru et al. (2008). Therefore, we utilize superpixels generated by multiple hierarchical segmentations obtained by different parameter settings. The multiple hierarchical segmentations contain information at different levels of image quantization and at different parameters of the algorithm. We use the algorithm proposed in Prasad and Swaminarayan (2008) to generate a hierarchical region structure. It is similar to Arbelaez et al. (2009) in that it not only generates different levels of segmentation, but also naturally defines the relation among each region and its parent. However, it is significantly faster.

9.1 Hierarchical Graph Construction

We generate L hierarchical over-segmentation results by by varying the segmentation parameters, which are the input parameters of Canny edge detector. V_k^h represents the regions at level h of the k th over-segmentation. In particular, V_k^1 denotes the regions of over-segmentation k at the finest level. Since we use the hierarchical segmentation, the constructed graph $G = (V, A)$ has the total of n nodes representing all segmentation regions in the set

$$V = V_1^1 \cup V_2^1 \cup \dots \cup V_L^1 \cup V_1^2 \cup V_2^2 \cup \dots \cup V_L^H.$$

The edges are connected by different criteria according to the node types. Edge weight $a_{ij} \in A$ is non-zero if two regions i and j satisfy one of the following conditions

- regions i and j are at the finest level of some segmentation k , i.e., $i, j \in V_k^1$, and they share a common boundary, which we denote as $i \in Neighbor_k(j)$,
- regions i and j are at the finest level in two different segmentations k_1, k_2 , i.e., $i \in V_{k_1}^1$, $j \in V_{k_2}^1$, and they overlap, which we denote as $Overlap(i, j)$
- in the same k segmentation and region j is the parent of region i , i.e., $i \in V_k^h$ and $j \in V_k^{h+1}$, which we will denote $j = Parent_k(i)$.

In this cases, edge weight $a_{ij} \in A$ is defined as

$$a_{ij} = \begin{cases} \exp\left(-\frac{\|\bar{c}_i - \bar{c}_j\|}{\delta}\right), & i \in Neighbor_k(j) \\ \exp\left(-\frac{\|\bar{c}_i - \bar{c}_j\|}{\delta}\right), & Overlap(i, j) \\ \lambda, & j = Parent_k(i) \end{cases} \quad (17)$$

where \bar{c}_i represents the mean color of region i and λ controls the relation between regions in different layers. The larger λ is, the more information can be propagated following the hierarchical structure of regions. The smaller δ is, the more sensitive is the edge weight to the color differences. In all our experiments, we set $\lambda = 0.0001$ and $\delta = 60$.

10. Interactive Segmentation

Once we construct the graph, we utilize the proposed affinity learning algorithm to refine the similarities among nodes in the graph. We select the segmentation k_m that contains the largest number of superpixels at finest level. The segmentation result is then based on the regions in the $V_{k_m}^1$. Therefore, only the affinities $Q_{k_m}^*$ between regions in $V_{k_m}^1$ are extracted from Q^* to obtain the final segmentation result.

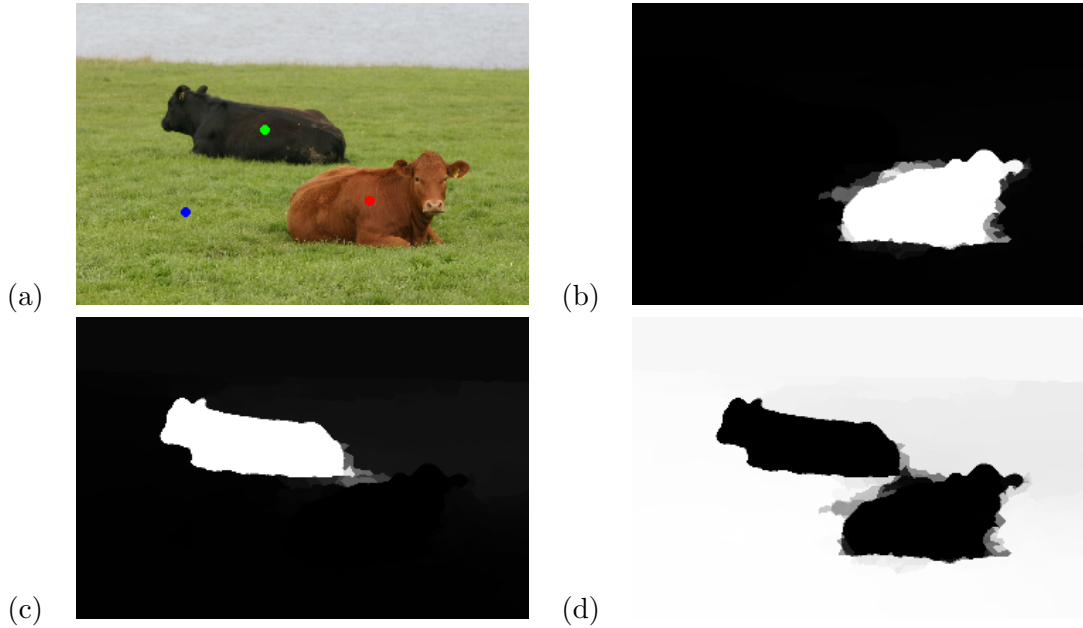


Figure 5: A user labeled 3 object classes with 3 dots in (a). The other images show the probability maps of the red label point in (b), the green label point in (c), and the blue label point in (d). The larger of the probability, the whiter the color.

Suppose that a user has labeled selected pixels in the image as belonging to C object classes. We consider the regions in $V_{k_m}^1$ that contain the labeled pixels. They are grouped into sets \mathcal{R}_c for $c = 1, \dots, C$, so that all regions in a set \mathcal{R}_c have label c . For each unlabeled region R_u , we define its average similarity to the labeled regions in class c as

$$sim(R_u, c) = \frac{1}{|\mathcal{R}_c|} \sum_{R \in \mathcal{R}_c} Q_{l_m}^*(R_u, R), \quad (18)$$

where $|\mathcal{R}_c|$ is the number of regions labeled as class c . Hence sim is a $N_u \times C$ matrix, where N_u is the number of unlabeled regions. When sim is first row wise normalized and then column wise normalized, we can interpret each column c of sim as the probability map of each image region having label c . An example is provided in Figure 5. Since the algorithm is based on superpixels, the boundaries of objects may be zigzag, but the general segmentation is quite accurate.

An unlabeled region R_u is assigned the label of regions with the largest average affinity to it, i.e., the label of R_u is given by

$$\text{label}(R_u) = \operatorname{argmax}\{\text{sim}(R_u, c) \mid c = 1, \dots, C\}. \quad (19)$$

As Figure 6 illustrates, the learned affinities lead to excellent interactive segmentation results.

11. Conclusion

We propose a novel symmetric anisotropic diffusion process for semi-supervised learning, which is equivalent to semi-supervised learning on a tensor product graph. As our experimental results demonstrate SADP compares very favorably to state-of-the-art semi-supervised learning methods. In particular, it can perform very well even if the number of classes is large, and it performs significantly better when the ratio of labeled points for different classes varies significantly. This is due to the fact that SADP averages the influence of the labeled data to unlabeled data, which performs like a regularizer to balance the influence of labels from different classes and makes SADP insensitive to differences in the number of labeled examples in different classes. Moreover, the qualitative results of interactive image segmentation demonstrate the ability to apply our algorithm to image segmentation problem.

Acknowledgement

The work of the second author was supported in part by the U.S. Department of Energy under grant DE-FG02-05ER25672, and by the U.S. National Science Foundation under grant DMS-1115520. The work was also supported by the NSF under Grants IIS-0812118, BCS-0924164, OIA-1027897, and by the AFOSR Grant FA9550-09-1-0207.

References

- Pablo Arbelaez, Michael Maire, Charless Fowlkes, and Jitendra Malik. From contours to regions: An empirical evaluation. In *CVPR*, 2009.
- Mikhail Belkin, Partha Niyogi, and Vikas Sindhwani. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *Journal of Machine Learning Research*, 7:2399–2434, 2006.
- S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Trans. PAMI*, 24:705–522, 2002.
- A. Blum and S. Chawla. Learning from labeled and unlabeled data using graph mincuts. In *ICML*, 2001.
- Avrim Blum and Tom Mitchell. Combining labeled and unlabeled data with co-training. In *COLT: Proceedings of the Workshop on Computational Learning Theory*, 1998.
- Y. Boykov and M.-P. Jolly. Interactive graph cuts for optimal boundary and region segmentation of objects in n-d images. In *ICCV*, 2001.

- O. Chapelle and A. Zien. Semi-supervised classification by low density separation. In *AISTAT*, 2005.
- O. Chapelle, B. Scholkopf, and A. Zien. *Semi-supervised learning*. MIT Press, 2006.
- A. Criminisi, T. Sharp, and A. Blake. Geos: Geodesic image segmentation. In *ECCV*, 2008.
- Mark Culp and George Michailidis. An iterative algorithm for extending learners to a semisupervised setting. In *The 2007 Joint Statistical Meetings*, 2007.
- O. Duchenne, J.-Y. Audibert, R. Keriven, J. Ponce, and F. Segonne. Segmentation by transduction. In *CVPR*, 2008.
- Akinori Fujino, Naonori Ueda, and Kazumi Saito. A hybrid generative/discriminative approach to semi-supervised classifier design. In *AAAI*, 2005.
- Anil K. Jain. *Fundamentals of Digital Image Processing*. Prentice Hall, 1989.
- Tony Jebara, Jun Wang, and Shih-Fu Chang. Graph construction and b-matching for semi-supervised learning. In *ICML*, 2009.
- T. Joachims. Transductive learning via spectral graph partitioning. In *ICML*, 2003.
- P. Kohli, L. Ladicky, and P. Torr. Robust higher order potentials for enforcing label consistency. In *CVPR*, 2008.
- Risi Imre Kondor and John Lafferty. Diffusion kernels on graphs and other discrete structures. In *ICML*, 2002.
- L’ubor Ladicky, Chris Russell, and Pushmeet Kohli. Associative hierarchical crfs for object class image segmentation. In *ICCV*, 2009.
- Peter Lancaster and Leiba Rodman. *Algebraic Riccati Equations*. Clarendon Press, Oxford, 1995.
- Boaz Nadler, Nathan Srebro, and Xueyuan Zhou. Semi-supervised learning with the graph Laplacian: The limit of infinite unlabelled data. In *NIPS*, 2009.
- Kamal Nigam, Andrew McCallum, Sebastian Thrun, and Tom Mitchell. Text classification from labeled and unlabeled documents using EM. *Machine Learning*, 39:103–134, 2000.
- J.-Y. Pan, H.-J. Yang, C. Faloutsos, and P. Duygulu. Automatic multimedia cross-modal correlation discovery. In *KDD*, pages 653–658, 2004.
- Caroline Pantofaru, Cordelia Schmid, and Martial Hebert. Object recognition by integrating multiple image segmentations. In *ECCV*, 2008.
- L. Prasad and Sriram Swaminarayan. Hierarchical image segmentation by polygon grouping. In *CVPR 2008 Workshop on Perceptual Organization in Computer Vision*, 2008.
- Xiaofeng Ren and J. Malik. Learning a classification model for segmentation. In *ICCV*, 2003.

- Chuck Rosenberg, Martial Hebert, and Henry Schneiderman. Semi-supervised selftraining of object detection models. In *Seventh IEEE Workshop on Applications of Computer Vision*, 2005.
- Vikas Sindhwani, Partha Niyogi, and Mikhail Belkin. Beyond the point cloud: from transductive to semi supervised learning. In *ICML*, 2005.
- Arthur D. Szlam, Mauro Maggioni, and Ronald R. Coifman. Regularization on graphs with function-adapted diffusion processes. *Journal of Machine Learning Research*, 2008.
- M. Szummer and T. Jaakkola. Partially labeled classification with Markov random walks. In *NIPS*, 2001.
- Andrew Temlyakov, Brent C. Muncell, Jarell W. Waggoner, and Song Wang. Two perceptually motivated strategies for shape classification. In *CVPR*, 2010.
- Charles van Loan. The ubiquitous Kronecker product. *Journal of Computational and Applied Mathematics*, 123:85–100, 2000.
- Richard S. Varga. *Matrix Iterative Analysis*. Prentice-Hall, Englewood Cliffs, New Jersey, 1962. Second Edition, revised and expanded, Springer, Berlin, 2000.
- S.V. N. Vishwanathan, N. N. Schraudolph, R. Kondor, and K. M. Borgwardt. Graph kernels. *Journal of Machine Learning Research*, 11:1201–1242, 2010.
- Jun Wang, Tony Jebara, and Shih-Fu Chang. Graph transduction via alternating minimization. In *ICML*, 2008.
- Paul M. Weichsel. The Kronecker product of graphs. *Proceedings of the American Mathematical Society*, 13 (1):47–52, 1962.
- X. Yang, S. Köknar-Tezel, and L. J. Latecki. Locally constrained diffusion process on locally densified distance spaces with applications to shape retrieval. In *CVPR*, 2009.
- D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Scholkopf. Learning with local and global consistency. In *NIPS*, 2003.
- Dengyong Zhou, Jiayuan Huang, and Bernhard Scholkopf. Learning from labeled and unlabeled data on a directed graph. In *ICML*, 2005.
- Dengyong Zhou, Jiayuan Huang, and Bernhard Scholkopf. Learning with hypergraphs: Clustering, classification, and embedding. In *NIPS*, 2006.
- Zhi-Hua Zhou, De-Chuan Zhan, and Qiang Yang. Semi-supervised learning with very few labeled training examples. In *AAAI*, 2007.
- X. Zhu. *Semi-supervised learning with graphs*. PhD thesis, Carnegie Mellon University, 2005. Also Technical Report CMU-LTI-05-192.
- X. Zhu. Semi-supervised learning literature survey. Technical Report 1530, Carnegie Mellon University, 2008.

X. Zhu, Z. Ghahramani, and J. Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In *ICML*, 2003.

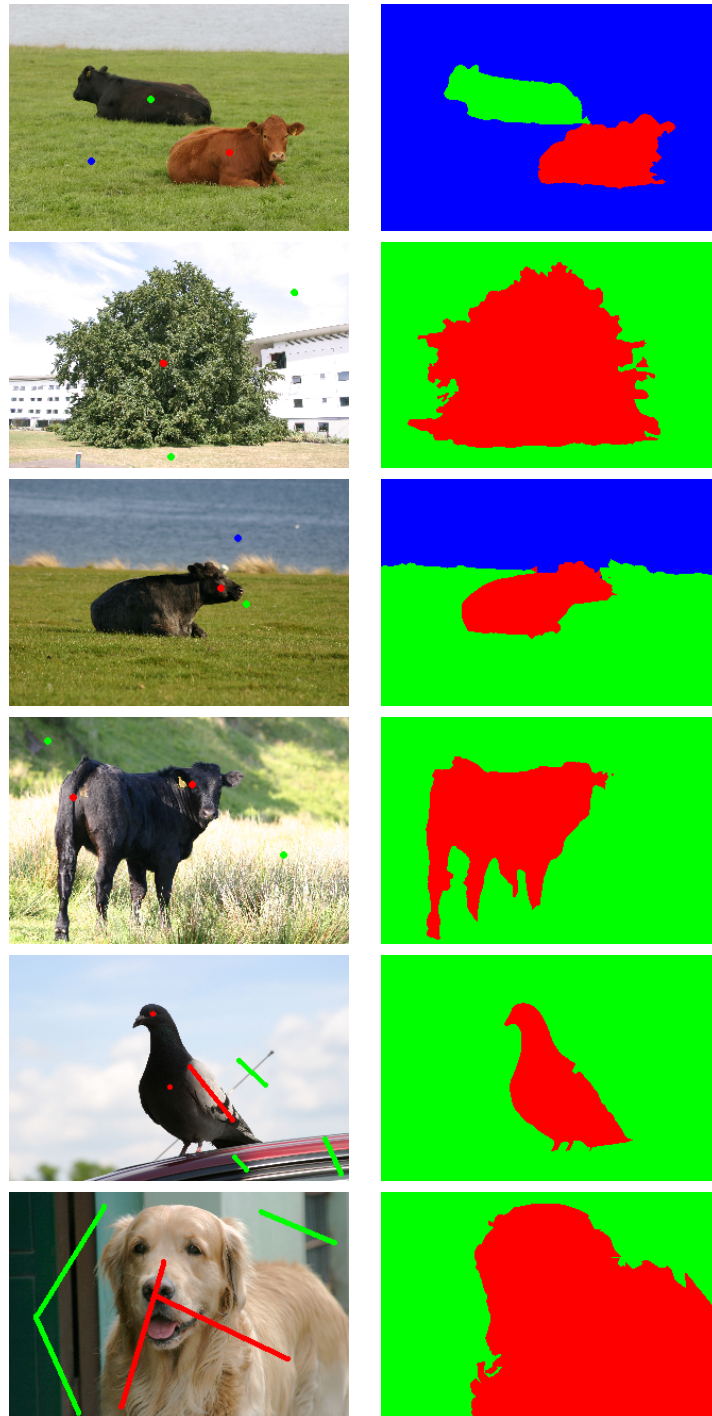


Figure 6: Interactive segmentation results. The labeled regions in the original images are enlarged to make them visible.