

Semi-Supervised Classification with Universum

Dan Zhang¹, Jingdong Wang², Fei Wang³, Changshui Zhang⁴

^{1,3,4} State Key Laboratory on Intelligent Technology and Systems,

Tsinghua National Laboratory for Information Science and Technology (TNList),

Department of Automation, Tsinghua University, Beijing, 100084, China.

{dan-zhang05, feiwang03}@mails.tsinghua.edu.cn, zcs@mail.tsinghua.edu.cn

² Internet Media Group, Microsoft Research Asia,

Microsoft Research Asia, 49 Zhichun Road, Beijing, 100080, China

i-jingdw@microsoft.com

Abstract

The Universum data, defined as a collection of "non-examples" that do not belong to any class of interest, have been shown to encode some prior knowledge by representing meaningful concepts in the same domain as the problem at hand. In this paper, we address a novel semi-supervised classification problem, called semi-supervised Universum, that can simultaneously utilize the labeled data, unlabeled data and the Universum data to improve the classification performance. We propose a graph based method to make use of the Universum data to help depict the prior information for possible classifiers. Like conventional graph based semi-supervised methods, the graph regularization is also utilized to favor the consistency between the labels. Furthermore, since the proposed method is a graph based one, it can be easily extended to the multi-class case. The empirical experiments on the USPS and MNIST datasets are presented to show that the proposed method can obtain superior performances over conventional supervised and semi-supervised methods.

1 Introduction

In pattern recognition and machine learning, a new concept, termed the Universum, has been put forward by Jeston et al. [10]. The Universum is defined as a collection of unlabeled examples known not belong to any class that is related to the classification problem at hand. It contains data that belongs to the same domain as the problem of interest and is expected to represent meaningful information related to the classification task at hand. Since it is not required to have the same distribution with the training data, the Universum can reveal some prior information for the possible classifiers. This has been justified on inductive classification problems by the Universum support vector machine (\mathcal{U} -SVM)

[10]. But \mathcal{U} -SVM is devised to deal with the supervised learning problems. In this paper, we will address a novel semi-supervised classification problem, where the Universum examples are also considered.

Semi-Supervised Learning is a very important branch in machine learning, since in many practical problems the labeled examples are always rare and the large amount of unlabeled examples are steadily available. Therefore, it has attracted significant attention [1, 5, 6, 11, 12, 13] and references therein. The motivation of semi-supervised methods is to make use of the unlabeled data to improve the performance. Among these methods, graph-based methods are very popular, where the graph nodes represent the data points, and the weights on the edges correspond to the similarities between pairwise points. The basic assumption of these methods is that all the examples are situated on a low dimensional manifold within the ambient space of the examples.

In the setup of traditional semi-supervised classification problems, the data points exactly consist of two sets: one set that has been labeled by human and the other set that is not classified but belongs to one known category. A toy example is shown in Fig.1(a). Considering the example in Fig.1(b), we are given extra data points that *do not* belong to any class, called the *Universum*, this problem turns to a typical semi-supervised Universum problem. In this paper, we will first give a general formulation for the semi-supervised Universum method. Then we will investigate how to utilize this proposed framework to integrate the Universum examples.

The rest of the paper is organized as follows: The notion of Universum will be given in Section 2. In Section 3, we will state the whole problem and give the corresponding notations. We will elaborate our proposed algorithm in Section 4. The relationship between the

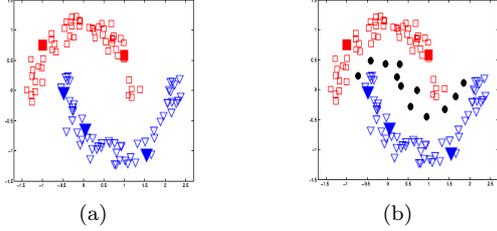


Figure 1: The comparison of SS-Universum and SSL. Here, blue \blacktriangledown denote the negatively labeled examples, red \blacksquare represent the positively labeled examples, and black \bullet refer to the Universum data. Other examples are left unlabeled. In the left figure [1], we illustrate the typical semi-supervised learning problem, while in the right figure, some Universum examples are also utilized to help determine the decision boundary.

Universum problem with some other machine learning methods will be given in Section 5. In Section 6, the experimental results are presented. In the end, conclusions and future works will be drawn in Section 7.

2 Learning with Universum

2.1 Regularization with Universum In a classification problem, the main focus is to construct a function $y = f(x)$ given a set of labeled and unlabeled examples. Let us assume that along with the labeled and unlabeled examples, we also possess a collection of examples known not belong to any classes, *i.e.*, Universum examples. Then, how can we utilize these Universum examples in the design of the function f ?

In machine learning, in order to encode the prior knowledge into an algorithm, it is quite natural to define a prior distribution on possible functions. Suppose we know a prior distribution $P(f)$ on the set of possible functions. Then, given a set of labeled examples, termed as D , if the Maximum a Posteriori (MAP) criterion is employed and D is assumed to be independent and identically distributed, the objective optimization problem can be: $\min_f -\log P(f, D) = \min_f (-\log P(D|f) - \log P(f))$. Here, in supervised learning, the first term denotes the loss on the labeled set, given a function f . In this way, it will be relatively easy to evaluate all the possible functions by taking both the loss and prior distribution together.

The problem with this approach is that the formulation of $P(f)$ is too hard. Therefore, instead of modeling this probability directly, a regularizer term can be used to encode such prior probability. One possible way is to use the norm under a Reproducing Kernel Hilbert Space (RKHS): $\|f\|_{\mathcal{H}}^2$ ($P(f) \propto e^{-\|f\|_{\mathcal{H}}^2}$ [9]), and \mathcal{H} is an RKHS where f is sampled from.). For the case when all

the training examples are linearly separable, this term can be simply $\|w\|_2^2$. If they are not, a data-dependent Mercer kernel K (For a detailed description of Mercer Kernels, please refer to [8]) can be employed and this regularizer term turns to $\alpha^T K \alpha$, where the optimal solution is $f(x) = \sum_{i=1}^l \alpha_i K(x, x_i)$ [8]. In fact, this kind of regularizer bounds the norm of the gradient of the discriminant function and hence favors the smoothness of the discriminant functions. Another possible way to formulate $P(f)$ is to use Universum examples. For an Universum example x^* in binary classification, since it doesn't belong to either of the categories, it is required that $f(x^*)$ should be near 0. By encoding such prior knowledge, another kind of regularizer term can be devised. Note that although they seem so different, these two kinds of regularization are not exclusive. In fact, in [10], it is shown that under some special cases, they are equivalent.

2.2 Universum SVM We will give an example on how the Universum examples can be used to design the regularization term. In [10], Jeston et al. integrated the penalty term constructed by Universum examples into the objective function of SVM and put forward \mathcal{U} -SVM. To formulate the objective function of \mathcal{U} -SVM, the loss function for the Universum examples is pre-defined. In their paper, the ϵ -insensitive loss is employed (in the middle of Fig.2), *i.e.*, given an Universum example x_i^* and a classifier f , its loss can be calculated as:

$$(2.1) \quad U[f(x_i^*)] = H_{-\epsilon}[f(x_i^*)] + H_{-\epsilon}[-f(x_i^*)]$$

Here, $H_{-\epsilon}(t)$ denotes the hinge loss and is shown in the left figure of Fig.2. Other loss functions, such as the quadratic loss (the right figure of Fig.2), are also possible. In this way, the prior knowledge embedded in the Universum can be reflected in the sum of the losses, *i.e.*, $\sum_{i=1}^{|\mathcal{U}|} U[f(x_i^*)]$. The smaller is this value, the higher prior possibility is this classifier f , and vice versa. Adding this term to the standard SVM objective function and supposing the classifier takes the form of $f_{w,b}(x) = w^T x + b$, we can formulate the objective function for Universum SVM algorithm (\mathcal{U} -SVM):

$$(2.2) \quad \frac{1}{2} \|w\|_2^2 + C_l^s \sum_{i=1}^l H[y_i f_{w,b}(x_i)] + C_{\mathcal{U}}^s \sum_{i=1}^{|\mathcal{U}|} U[f_{w,b}(x_i^*)],$$

where C_l^s controls the loss on the labeled examples and $C_{\mathcal{U}}^s$ controls the impact of the Universum term. It is clear that Eq.(2.2) has two regularizer terms, *i.e.* the term $\frac{1}{2} \|w\|_2^2$ that favors smoothness of the discriminant functions and the term $C_{\mathcal{U}}^s \sum_{i=1}^{|\mathcal{U}|} U[f_{w,b}(x_i^*)]$

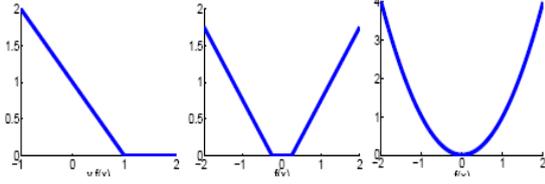


Figure 2: In the left figure, the hinge loss function, which is frequently used in SVMs is depicted. The middle and right figures are two loss functions that can be used in \mathcal{U} -SVM. The middle figure is for the ϵ -insensitive loss, while the right one is a quadratic loss function [10].

that tries to approximate $P(f_{w,b})$ through the loss on the Universum examples. Under this case, the prior for a function $f_{w,b}$ can be considered as $P(f_{w,b}) \propto e^{-\frac{1}{2}\|w\|_2^2} \times e^{-C_{\mathcal{U}} \sum_{i=1}^{|\mathcal{U}|} U[f_{w,b}(x_i^*)]}$.

In the following sections, we will address a special semi-supervised problem where the Universum examples are provided, besides the labeled and unlabeled examples.

3 Problem Statement and Notations

We are given l labeled data points: $(x_1, y_1), \dots, (x_l, y_l)$, and u ($l \ll u$) unlabeled points x_{l+1}, \dots, x_{l+u} , where $x_i \in \chi \subseteq \mathbb{R}^d$ ($1 \leq i \leq l + u = n$) is the input data, and χ is the input space. y_i is the class label and can be taken from c classes. As well as these data points, some Universum examples are also available, namely $x_1^*, \dots, x_{|\mathcal{U}|}^*$. Here, $|\mathcal{U}|$ denotes the number of the Universum examples. These Universum examples are known not belong to any classes. Our main goal is to predict the class labels of the unlabeled data points, *i.e.*, the labels for x_{l+1}, \dots, x_n by utilizing the labeled, unlabeled and the Universum examples.

4 Semi-Supervised Universum

In this section, we will first formulate the semi-supervised Universum problem. Then, we will elaborate our proposed method. In the end, the flowchart of the whole method will be given.

4.1 Formulation We aim to make use of labeled data, unlabeled data and universum data together to infer the labels. The formulation is as follows:

$$(4.3) \quad \min_{f \in \mathcal{F}} H_d(f) + H_r(f) + H_u(f),$$

where f is a classifier defined on a manifold \mathcal{M} , *i.e.*, $f : \mathcal{M} \mapsto \mathbb{R}$. $H_d(f)$ is a loss function that measures the compatibility between the estimated labels and

the given labels. $H_r(f)$ is devised to penalize the inconsistency of the labels between the data points, and $H_u(f)$ is a loss function derived from the Universum data that essentially encodes the prior distribution of f . Next, we will specialize all the terms.

4.1.1 Compatibility with the Given Labels The first term $H_d(f)$ penalizes the difference between the estimated labels and the given labels. Given the labeled and unlabeled examples, this term can take a concrete form as:

$$(4.4) \quad H_d(f) = (\hat{f} - \hat{y})^T C (\hat{f} - \hat{y})$$

Here, \hat{f} is defined as $\hat{f} = [f(x_1), \dots, f(x_n)]^T$. \hat{y} is a n -dimensional vector, and equals $[y_1, \dots, y_l, 0, \dots, 0]^T$. $C \in \mathbb{R}^{n \times n}$ is a diagonal matrix, and its i -th diagonal element c_i is computed as:

$$(4.5) \quad c_i = \begin{cases} C_l, & 1 \leq i \leq l \\ C_u, & l+1 \leq i \leq n \end{cases}$$

where C_l is a parameter that controls the loss on the labeled examples and C_u is a parameter that controls the penalty imposed on the unlabeled examples. In most cases, C_u equals zero.

4.1.2 Consistency between the Labels The second term $H_r(f)$ is a graph regularization term that penalizes the unsmoothness of f . In fact, Laplacian matrix provides such an approximation for this smoothness by m examples that are sampled from \mathcal{M} , and are denoted as: $(x'_1, x'_2, \dots, x'_m)$.

$$(4.6) \quad H_r(f) = \int_{\mathcal{M}} \|\nabla f(x)\|^2 \approx \tilde{f}^T R \tilde{f}.$$

Here, $\tilde{f} = [f(x'_1), f(x'_2), \dots, f(x'_m)]^T$. R is a regularization matrix defined on these m examples. Many existed graph-based semi-supervised algorithms are designed to devise the regularizer matrix R . Among them, the Laplacian Regularizer (Lap-Reg) [7] and Normalized Laplacian Regularizer (NLap-Reg) [6] are two very popular ones. Under these two cases, R takes the form of L and L_n , respectively. To compute Lap-Reg, we can first build a weighted k -nearest neighbor graph and use the heat kernel to determine the weights among edges. The adjacency matrix W ($W = [w_{ij}] \in \mathbb{R}^{m \times m}$), can then be determined by:

$$w_{ij} = \begin{cases} \exp\left(-\frac{1}{2\sigma^2} \|x'_i - x'_j\|^2\right), & i \in \mathcal{N}_k(x'_j) \cup j \in \mathcal{N}_k(x'_i) \\ 0, & \text{otherwise} \end{cases}$$

where, $\mathcal{N}_k(x'_j)$ denotes the k nearest neighbors of x'_j , and σ is the bandwidth for the RBF kernel. Then, the

Laplacian matrix L can be calculated as:

$$(4.7) \quad L = D - W$$

D is a diagonal matrix, with its i -th diagonal element $D_{ii} = \sum_{j=1}^n w_{ij}$. In fact, using this Laplacian matrix, the second term of Eq.(4.3) can be expressed as:

$$(4.8) \quad \tilde{f}^T L \tilde{f} = \frac{1}{2} \sum_{i=1}^m \sum_{j=1}^m w_{ij} (f(x'_i) - f(x'_j))^2$$

As for the NLap-Reg, it takes the form of:

$$(4.9) \quad L_n = I - D^{-1/2} W D^{-1/2}$$

where, I is the identity matrix. W and D are the same as in Eq.(4.7). Taking this regularizer, the second term in Eq.(4.3) can be converted to:

$$(4.10) \quad \tilde{f}^T L_n \tilde{f} = \frac{1}{2} \sum_{i=1}^m \sum_{j=1}^m w_{ij} \left(\frac{f(x'_i)}{\sqrt{D_{ii}}} - \frac{f(x'_j)}{\sqrt{D_{jj}}} \right)^2$$

The basic motivation of both Eq.(4.8) and Eq.(4.10) is that the value of \tilde{f} should not change too much between nearby points.

4.1.3 Prior from Universum The last term $H_u(f)$ is devised to penalize the loss on the Universum examples. For an Universum example x^* , its soft label $f(x^*)$ should be close to zero, which provides some prior knowledge on f . By utilizing the quadratic loss in Fig.2, this term can be devised as:

$$(4.11) \quad H_u(f) = C_{\mathcal{U}} \sum_{i=1}^{|\mathcal{U}|} f(x_i^*)^2,$$

where, $C_{\mathcal{U}}$ is a parameter that controls the impact of the Universum examples.

So far, we have analyzed the three parts of Eq.(4.3). It can be seen that, unlike traditional graph-based semi-supervised methods, our formulation contains two regularizers, i.e. the graph regularizer and the regularizer on the Universum. In fact, typical graph-based semi-supervised learning [1] [14] [6] [12] can be deemed as a special case of our formulation Eq.(4.3), with $C_{\mathcal{U}}$ equals zero, and $H_r(f)$ being defined only on the labeled and unlabeled examples.

4.2 A Simple Method We can directly use the formulation Eq.(4.3), and devise a semi-supervised Universum method as follows:

$$(4.12) \quad \min_{\tilde{f} \in \mathbb{R}^{n+|\mathcal{U}|}} (\tilde{f} - \tilde{y})^T \tilde{C} (\tilde{f} - \tilde{y}) + \tilde{f}^T \tilde{R} \tilde{f}.$$

Here, \tilde{R} is a regularizer designed on all the $n + |\mathcal{U}|$ examples, either by Eq.(4.7) or by Eq.(4.9). \tilde{f} and \tilde{y} are the corresponding $(n + |\mathcal{U}|)$ -dimensional column vector that specify the estimated and given labels, respectively. Note that for the unlabeled and Universum examples, their corresponding given labels in \tilde{y} are assumed to be 0. \tilde{C} is a diagonal matrix, and its i -th diagonal element c_i is computed as: $c_i = C_l > 0$ for $1 \leq i \leq l$, $c_i = C_u \geq 0$ for $l + 1 \leq i \leq n$, $c_i = C_{\mathcal{U}} \geq 0$ for $n + 1 \leq i \leq n + |\mathcal{U}|$. In this way, the penalty on the Universum is encoded in \tilde{C} . The solution for this formulation can be simply:

$$\tilde{f} = (\tilde{R} + \tilde{C})^{-1} \tilde{C} \tilde{y}$$

This method is quite straightforward and reasonable. But when the number of the Universum examples is very huge ¹, this method becomes very time consuming, since the calculation burden of the neighborhood relationship will become very heavy and the scale of $(\tilde{R} + \tilde{C})$ can also be so large. Therefore, we try to encode the prior information of Universum examples in a different way. The concrete methods will be elaborated in the next section.

4.3 Lap/NLap-Universum As we have mentioned in Section 4.1, graph-based semi-supervised learning is a special case of Eq.(4.3). In traditional graph-based semi-supervised learning, the out-of-sample examples are defined as examples that do not previously exist in the labeled and unlabeled set. Since the Universum examples do not exist in the labeled and unlabeled set either, we can treat the Universum as out-of-sample examples and the Universum can be removed from the graph-regularization term, i.e., R is defined only on the labeled and unlabeled sets.

Then, Eq.(4.3) can be transformed to a compact form:

$$(4.13) \quad \min_{\hat{f} \in \mathbb{R}^n} (\hat{f} - \hat{y})^T C (\hat{f} - \hat{y}) + \hat{f}^T R \hat{f} + C_{\mathcal{U}} \sum_{i=1}^{|\mathcal{U}|} f(x_i^*)^2,$$

where \hat{f} and \hat{y} have been previously defined in Eq.(4.4). The focus now turns to how to approximate $f(x^*)$. In fact, since we have treated the Universum examples as out-of-sample ones, the soft label of the Universum examples can be acquired by utilizing some developed induction methods, which are designed to get the soft labels of the out-of-sample examples, such as [3]. In this

¹This is always the case. We will show, in the experiment part, the Universum examples can be obtained in large numbers with some convenient strategies.

paper, we try to generalize the results of [3] and make it adapted to our framework.

It is obvious that in, Eq.(4.13), if $C_{\mathcal{U}}$ equals zero, this formulation turns to a traditional semi-supervised classification problem:

$$(4.14) \quad \min_{\hat{f} \in \mathbb{R}^n} (\hat{f} - \hat{y})^T C (\hat{f} - \hat{y}) + \hat{f}^T R \hat{f}.$$

For an out-of-sample example x , in order to get its soft label, we can make the following assumptions: (i) As for the first term in Eq.(4.14), the constraints on the labeled and unlabeled set remains the same, but adding a new unlabeled example. (ii) the type of the smoothness constraint is the same as the second term of Eq.(4.14), but including a test example.

Based on these assumptions, for an out-of-sample example x , we then try to minimize the following criterion:

$$(4.15) \quad \begin{aligned} & C_{W,D}^*(f(x)) \\ &= \sum_{j \in U \cup L} W(x, x_j) \text{dist}(f(x), f(x_j)) + C_u (f(x) - 0)^2. \end{aligned}$$

Here, $W(x, x_j)$ is the graph weight between x and x_j , $j \in U \cup L$. $\text{dist}(f(x), f(x_j))$ is defined to be a distance function, takes value $(f(x) - f(x_j))^2$ for Lap-Reg, and $(\frac{f(x)}{\sqrt{\sum_{i \in U \cup L} W(x, x_i)}} - \frac{f(x_j)}{\sqrt{D_{jj}}})^2$ for NLap-Reg.

Taking the derivative of Eq.(4.15) with respect to $f(x)$ and equals it to zero, the minimizer of Eq.(4.15) can be obtained. When the Laplacian matrix in Eq.(4.7) is used, the above objective function can be minimized when

$$(4.16) \quad f(x) = \frac{1}{\sum_{j \in U \cup L} W(x, x_j) + C_u} \sum_{j \in U \cup L} W(x, x_j) f(x_j).$$

If we employ the normalized Laplacian matrix, *i.e.*, Eq.(4.9), the optimal f can be calculated as:

$$(4.17) \quad f(x) = \frac{1}{\sqrt{\sum_{j \in U \cup L} W(x, x_j) (1 + C_u)}} \times \sum_{j \in U \cup L} \frac{W(x, x_j)}{\sqrt{D(x_j)}} f(x_j)$$

Furthermore, we can have a compact form for the estimation of the soft labels for the Universum examples. For an Universum example x_i^* , its soft label can be estimated as: $f(x_i^*) = W_{\mathcal{U}}^i \hat{f}$. The form of $W_{\mathcal{U}}^i$ depends on

the type of regularization matrix R that we use. When the Laplacian matrix is used,

$$(4.18) \quad W_{\mathcal{U}}^i = \frac{1}{\sum_{j \in U \cup L} W(x_i^*, x_j) + C_u} \times [W(x_i^*, x_1), W(x_i^*, x_2), \dots, W(x_i^*, x_n)]$$

If we employ the normalized Laplacian matrix,

$$(4.19) \quad W_{\mathcal{U}}^i = \frac{1}{\sqrt{\sum_{j \in U \cup L} W(x_i^*, x_j) (1 + C_u)}} \times \left[\frac{W(x_i^*, x_1)}{\sqrt{D_{11}}}, \frac{W(x_i^*, x_2)}{\sqrt{D_{22}}}, \dots, \frac{W(x_i^*, x_n)}{\sqrt{D_{nn}}} \right]$$

The soft labels of the Universum examples should be near zero. Taking the quadratic cost function (as in Fig.2), the Universum term $H_u(f)$ becomes:

$$H_u(f) = \hat{f}^T W_{\mathcal{U}}^T W_{\mathcal{U}} \hat{f}$$

$W_{\mathcal{U}}$ is a matrix, with its i -th row being $W_{\mathcal{U}}^i$. Note that since we have employed the weighted k nearest neighbors graph, $W_{\mathcal{U}}$ will also be sparse. In this way, Eq.(4.13) can be transformed to:

$$(4.20) \quad \begin{aligned} \hat{f}^* &= \arg \min_{\hat{f}} (\hat{f} - \hat{y})^T C (\hat{f} - \hat{y}) + \hat{f}^T R \hat{f} \\ &+ C_{\mathcal{U}} \hat{f}^T W_{\mathcal{U}}^T W_{\mathcal{U}} \hat{f} \\ &= \arg \min_{\hat{f}} (\hat{f} - \hat{y})^T C (\hat{f} - \hat{y}) + \hat{f}^T \hat{R} \hat{f}. \end{aligned}$$

Here, $\hat{R} = R + C_{\mathcal{U}} W_{\mathcal{U}}^T W_{\mathcal{U}}$. The definition of C is the same as that in Eq.(4.3).

The final solution for Eq.(4.20) can be determined as:

$$(4.21) \quad \hat{f}^* = (\hat{R} + C)^{-1} C \hat{y}.$$

In this formulation, the information brought by the Universum has been encoded in \hat{R} , and \hat{R} can be considered as a new regularization matrix, where the possibility for function \hat{f} can be determined as: $P(\hat{f}) \propto e^{-\hat{f}^T \hat{R} \hat{f}}$. In Eq.(4.20), when R takes the form of the Laplacian matrix, we name the method "Lap-Universum", and if the normalized Laplacian matrix is used, it will be called "NLap-Universum".

Also note that since our objective function takes the form of Eq.(4.20), the Leave-One-Out classification error can be easily acquired by utilizing the lemma achieved in [11].

4.3.1 Multi-Class Classification Although \mathcal{U} -SVM performs quite well on binary classification problems, it can not be directly applied to the multi-class

case. But our proposed method can be easily extended to this case. Here, instead of using the soft label vector f , we employ the label matrix $F \in \mathcal{F}$, where \mathcal{F} is a set of $n \times c$ matrix with nonnegative entries. This amounts to assigning a row vector F_i to each data point x_i . Define a matrix $Y \in \mathcal{F}$, with $Y_{ij} = 1$ if x_i is labeled as $y_i = j$, $Y_{ij} = -1$ if x_i is labeled but $y_i \neq j$, and $Y_{ij} = 0$ if x_i is unlabeled. In this way, Eq.(4.20) can be transformed to:

$$(4.22) \quad F^* = \arg \min_{F \in \mathcal{F}} \text{tr}((F - Y)^T C (F - Y) + F^T \hat{R} F).$$

Here, $\text{tr}(\cdot)$ stands for the trace of a matrix. The optimal classification matrix can be obtained by:

$$(4.23) \quad F^* = (\hat{R} + C)^{-1} C Y$$

The labels of the unlabeled examples can be determined by $\arg \max_j (F_{ij}^*), l + 1 \leq i \leq n$.

4.4 The Whole Method The whole method can be described in Table 1.

5 Relations

Since the Universum is a recently proposed concept, we would like to show some of its relations with some existed fields in machine learning.

First, the Universum problem is like a multi-class problem. The Universum examples can be deemed as belonging to a specific new category. Then, if we treat the Universum examples as an additional category, the c -class classification problem becomes a $(c + 1)$ -class classification problem. But there exist distinctions. Under the case of binary classification, the labels of Universum examples should be considered as 0, i.e. there exists some ordinal relations between the labels, i.e. $-1 < 0 < +1$. Furthermore, we don't care whether the Universum examples are categorized correctly, and their function is mainly on the regularization for the function f .

From another perspective, for the binary classification, the use of the Universum examples will make this classification more like an ordinal regression problem [2] [4]. The ordinal regression problem can be illustrated in Fig.3(b), in which the labels are ordered. In Fig.3(a), when the Universum examples are added to the training set, in some sense, the soft labels for the positive, Universum, negative examples form a order. However, the Universum problem and the ordinal regression problem are not equivalent. The ordinal regression problem needs to determine the thresholds for each ordinal category (in Fig.3(b), the thresholds are θ_1 and θ_2), while for Universum methods, the function value of Universum examples are constricted to around zero.

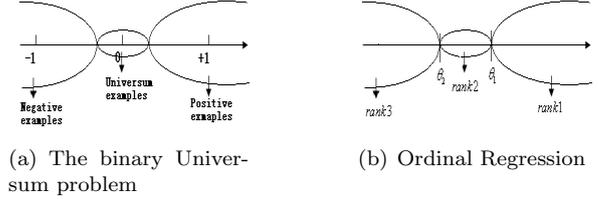


Figure 3: The relationship between a Universum problem and an ordinal regression problem. Both of them have some ordinal relations for their labels. But the Universum problem tend to place the Universum examples around zero, while the ordinal regression problem focus more on the ordinal relations between different classes

6 Experiments

In this section, we would like to show, through some experiments, the effectiveness of our proposed methods. We believe that the Universum examples are relatively easy to collect in large numbers. In this paper, we consider the following ways to generate the Universum examples:

- \mathcal{U}_{rest} : other digits that are not included in the classification tasks. For example, if the task is to classify digit 1 and 2, and the pictures of other digits (from 3 to 9) are available, these pictures can be used as Universum examples.
- \mathcal{U}_{gen} : Generate pictures by generating uniformly distributed features according to the statistic of the labeled and unlabeled pictures.
- \mathcal{U}_{mean} : Each image is generated by first selecting two images from two different categories and then combined with a specific combination coefficient. For example, for a binary classification problem, we have five positive instances and five negative ones, with one combination coefficient, twenty-five Universum data can be generated. It is true that 25 is a very small number, but when this number goes up very quickly with the increase of labeled examples and in the multi-class classification problems. Among all the experiments, the combination coefficient is set to 0.5.

For the binary classifications, we compare the performances of Lap-Universum and NLap-Universum with SVM, \mathcal{U} -SVM, Lap-Reg, NLap-Reg. Since \mathcal{U} -SVM is not designed for multi-class classification problems, its performances on these tasks can not be evaluated and are therefore not announced here. Among all the experiments, we employ RBF kernel for all the methods,

Input:

1. For binary classifications: l labeled examples: $(x_1, y_1), (x_2, y_2), \dots, (x_l, y_l), y_i \in \{-1, 1\}$.
For multi-class classifications (c classes): l labeled examples: $(x_1, Y_1), (x_2, Y_2), \dots, (x_l, Y_l)$, Here, $Y_i = [Y_{i1}, Y_{i2}, \dots, Y_{ic}]$ is a vector. $Y_{ij} = 1$ if x_i is labeled as $y_i = j$ and $Y_{ij} = -1$ otherwise.
2. u unlabeled examples: x_{l+1}, \dots, x_n .
3. a set of Universum examples: $x_1^*, \dots, x_{\mathcal{U}}^*$.
4. A set of parameters: C_l, C_u and $C_{\mathcal{U}}$.

Step 1: Construct the regularization matrix R on both the labeled and unlabeled examples:

For Lap-Universum, employ Eq.(4.7).

For NLap-Universum, employ Eq.(4.9).

Step 2: Calculate the weight matrix $W_{\mathcal{U}}$.

For Lap-Universum, employ Eq.(4.18).

For NLap-Universum, employ Eq.(4.19).

Step 3: Calculate the new regularization matrix: $\hat{R} = R + C_{\mathcal{U}} W_{\mathcal{U}}^T W_{\mathcal{U}}$.

Step 4: For binary classifications, employ Eq.(4.21) to get the final solution \hat{f}^* , while, for the multi-class classifications, utilize Eq. (4.23) to get the final solution F^* .

Ouput:

For binary classifications: the labels of the unlabeled examples can be determined by $sign(\hat{f}_i^*), l+1 \leq i \leq n$

For multi-class classifications: the labels of the unlabeled examples can be determined by $\arg \max_j (F_{ij}^*), l+1 \leq i \leq n$

Table 1: Lap-Universum and NLap-Universum

with the corresponding kernel width being tuned using 5-fold cross validation. We also tune other parameters such as C_l, C_u and $C_{\mathcal{U}}$ beforehand. The number of neighbors that is used to construct the graph is chosen from $\{7, 10, 15\}$.

6.1 USPS Dataset In this experiment, we test our method on the USPS test digits data set². For each category, the number of digits ranges from 150 to 300. Each digit is represented by a 256 dimensional vector with the range from form 0 to 1. We choose the digits "2", "3", "5", "8". We use "2" vs."3", "5" vs. "8" for the binary classification problem and "2", "3", "5", "8" as a four-class classification problem. Among all the experiments, 5 examples are randomly selected for each category as the labeled data, and the others are left as unlabeled examples. Each classification accuracy reported in Table 2, Table 3 and Table 4 is the average result of 50 independent trials. For \mathcal{U}_{rest} and \mathcal{U}_{gen} , 500 Universum examples are generated. As for \mathcal{U}_{mean} , under the binary classification case, we generate $5 \times 5 = 25$ examples by utilizing only the labeled examples, while, under the four-class classification case, 150 Universum examples are generated.

6.2 MNIST Dataset We present experimental results on the MNIST handwritten digit test set³. This data set contains 10,000 28×28 pixels images, with 1000 for each category and 10 categories in total. Like the experiments on the USPS dataset, we use "2" vs."3", "5" vs. "8" for the binary classification problem and "2", "3", "5", "8" as a four-class classification problem. Among all the experiments, for each category, 5 examples are randomly selected as the labeled examples, and 500 examples are randomly selected as unlabeled ones. Each classification accuracy reported in Table 5, Table 6 and Table 7 is averaged over 50 independent trials. We use exactly the same way as the experiments on USPS dataset to generate the Universum examples.

6.3 Results and Discussions The average classification accuracies and the standard deviations for the USPS data set are shown in Table 2, 3 and 4, while the results on the MNIST dataset are shown from Table 5 to 7. The different kinds of examples that can be used in different methods are also concluded in Table 8.

SVM is a supervised large margin algorithm. Although it tries to maximize the margins on the labeled examples, it can not utilize the unlabeled and Universum examples to help improve its performance. Unlike SVM, \mathcal{U} -SVM can make use of the Universum examples as well. Therefore, in most cases, \mathcal{U} -SVM performs

²<http://www.kernel-machines.org/>

³<http://yann.lecun.com/exdb/mnist/>

Universum source	SVM	\mathcal{U} -SVM	Lap-Reg	Lap-Universum	NLap-Reg	NLap-Universum
\mathcal{U}_{rest}	86.76 \pm 3.72	89.76 \pm 3.63	88.85 \pm 4.30	95.76 \pm 0.85	93.20 \pm 2.15	94.90 \pm 1.14
\mathcal{U}_{mean}	86.76 \pm 3.72	88.76 \pm 4.57	88.85 \pm 4.30	90.30 \pm 6.31	93.20 \pm 2.15	94.98 \pm 1.46
\mathcal{U}_{gen}	86.76 \pm 3.72	88.90 \pm 3.53	88.85 \pm 4.30	89.56 \pm 3.59	93.20 \pm 2.15	94.87 \pm 1.74

Table 2: Classification results for digit 2 and 3 on the USPS dataset. For each category, 5 examples are selected as the labeled data, and the rest are left as unlabeled set.

Universum source	SVM	\mathcal{U} -SVM	Lap-Reg	Lap-Universum	NLap-Reg	NLap-Universum
\mathcal{U}_{rest}	84.27 \pm 10.14	88.42 \pm 3.21	88.56 \pm 4.32	94.15 \pm 1.74	93.55 \pm 3.41	94.66 \pm 1.30
\mathcal{U}_{mean}	84.27 \pm 10.14	87.75 \pm 3.52	88.56 \pm 4.32	90.67 \pm 3.23	93.55 \pm 3.41	94.98 \pm 1.46
\mathcal{U}_{gen}	84.27 \pm 10.14	87.87 \pm 4.65	88.56 \pm 4.32	90.16 \pm 4.61	93.55 \pm 3.41	93.89 \pm 2.57

Table 3: Classification results for digit 5 and 8 on the USPS dataset. For each category, 5 examples are selected as the labeled data, and the rest are left as unlabeled set.

Universum source	SVM	Lap-Reg	Lap-Universum	NLap-Reg	NLap-Universum
\mathcal{U}_{rest}	71.72 \pm 9.05	78.33 \pm 3.96	82.86 \pm 4.38	83.80 \pm 2.82	85.58 \pm 3.30
\mathcal{U}_{mean}	71.72 \pm 9.05	78.33 \pm 3.96	83.08 \pm 4.27	83.80 \pm 2.82	84.63 \pm 2.67
\mathcal{U}_{gen}	71.72 \pm 9.05	78.33 \pm 3.96	82.80 \pm 3.96	83.80 \pm 2.82	83.90 \pm 2.86

Table 4: Classification results for digit 2, 3, 5 and 8 on the USPS dataset. For each category, 5 examples are selected as the labeled data, and the rest are left as unlabeled set.

Universum source	SVM	\mathcal{U} -SVM	Lap-Reg	Lap-Universum	NLap-Reg	NLap-Universum
\mathcal{U}_{rest}	86.99 \pm 4.19	88.49 \pm 3.91	90.03 \pm 8.25	94.73 \pm 1.15	95.26 \pm 4.19	96.33 \pm 1.36
\mathcal{U}_{mean}	86.99 \pm 4.19	87.19 \pm 5.68	90.03 \pm 8.25	95.71 \pm 2.80	95.26 \pm 4.19	95.43 \pm 2.75
\mathcal{U}_{gen}	86.99 \pm 4.19	87.66 \pm 3.81	90.03 \pm 8.25	96.37 \pm 1.83	95.26 \pm 4.19	95.35 \pm 1.91

Table 5: Classification results for digit 2 and 3 on the MNIST dataset. For each category, 5 examples are selected as the labeled data, and 500 examples are randomly selected as unlabeled ones.

Universum source	SVM	\mathcal{U} -SVM	Lap-Reg	Lap-Universum	NLap-Reg	NLap-Universum
\mathcal{U}_{rest}	80.61 \pm 4.18	82.16 \pm 3.86	88.72 \pm 9.93	88.95 \pm 8.64	89.21 \pm 8.50	91.50 \pm 2.94
\mathcal{U}_{mean}	80.61 \pm 4.18	82.90 \pm 9.06	88.72 \pm 9.93	90.44 \pm 5.62	89.21 \pm 8.50	90.53 \pm 5.78
\mathcal{U}_{gen}	80.61 \pm 4.18	81.51 \pm 6.09	88.72 \pm 9.93	91.55 \pm 4.24	89.21 \pm 8.50	91.30 \pm 4.40

Table 6: Classification results for digit 5 and 8 on the MNIST dataset. For each category, 5 examples are selected as the labeled data, and 500 examples are randomly selected as unlabeled ones.

Universum source	SVM	Lap-Reg	Lap-Universum	NLap-Reg	NLap-Universum
\mathcal{U}_{rest}	67.52 \pm 6.95	79.22 \pm 5.82	79.90 \pm 3.31	83.19 \pm 6.91	84.00 \pm 2.82
\mathcal{U}_{mean}	67.52 \pm 6.95	79.22 \pm 5.82	81.99 \pm 7.09	83.19 \pm 6.91	84.92 \pm 3.35
\mathcal{U}_{gen}	67.52 \pm 6.95	79.22 \pm 5.82	82.85 \pm 5.31	83.19 \pm 6.91	84.60 \pm 3.46

Table 7: Classification results for digit 2, 3, 5 and 8 on the MNIST dataset. For each category, 5 examples are selected as the labeled data, and 500 examples are randomly selected as unlabeled ones.

Labeled/Unlabeled	100	300	500	700
10/1000	89.00 \pm 6.55	90.46 \pm 5.22	91.55 \pm 4.24	91.23 \pm 2.88

Table 9: Lap-Universum is used in the '5' vs '8' classification task on MNIST dataset. Its performance for different numbers of Universum examples (\mathcal{U}_{gen} is employed to generate the Universum examples) are shown in this table.

	Labeled	Unlabeled	Universum
SVM	Y	N	N
\mathcal{U} -SVM	Y	N	Y
Lap/NLap-Reg	Y	Y	N
Lap/NLap-Uni.	Y	Y	Y

Table 8: The comparison of several different methods on the type of examples that can be used

better than SVM. However, \mathcal{U} -SVM is still a supervised method and can not take the unlabeled examples into account.

Lap-Reg and NLap-Reg are both typical graph-based semi-supervised methods. We show, in our experimental results, their performance can still be improved by Lap-Universum and NLap-Universum.

In fact, the performances of the different methods reflect the impact of different prior knowledge. SVM use the norm of functions in a RKHS defined by all the labeled examples. Lap-Reg and NLap-Reg dwell more on the smoothness of the manifold defined on both the labeled and unlabeled examples. Lap-Reg and NLap-Reg always outperform SVM when the labeled examples are rare and the number of unlabeled example is large. That is because, with more unlabeled examples, the prior knowledge encoded by Lap-Reg and NLap-Reg will become more accurate. But, in SVM, since the labeled examples are so rare, the norm of the functions defined on RKHS may not be so precise.

It is hard to distinguish the performance between \mathcal{U} -SVM and Lap/NLap-Reg. They use different kinds of regularizers. We can not tell which one is better.

Lap/NLap-Universum takes both the Lap/NLap-Reg and the Universum regularizer into account. The empirical experiments tell us this proposed method can achieve a better performance than using just one regularizer alone, such as \mathcal{U} -SVM and Lap/NLap-Reg.

The effect of the Universum is in fact controlled by the number of Universum examples and $C_{\mathcal{U}}$, *i.e.*, if $C_{\mathcal{U}}$ equals zero, the Universum examples will have no impact on the final classification results. To see the effect of the Universum term on the classification results, we have also conducted another group of experiments. For the Lap-Universum on the the "5" vs. "8" MNIST classification task, we select 5 examples and 500 examples as the labeled and unlabeled examples for each category, and the the number of \mathcal{U}_{rest} , \mathcal{U}_{gen} , and \mathcal{U}_{mean} are fixed to 500, 500, and 25, respectively. We first fix the parameter $C_{\mathcal{U}}$ to 0.01 and tune other parameters by cross validation for \mathcal{U}_{rest} , \mathcal{U}_{mean} and \mathcal{U}_{gen} . Then, we fix the other parameters, and vary $C_{\mathcal{U}}$, chart the accuracies with the varying $C_{\mathcal{U}}$. The final result is averaged over 50 independent runs and is charted in Fig.4. Note

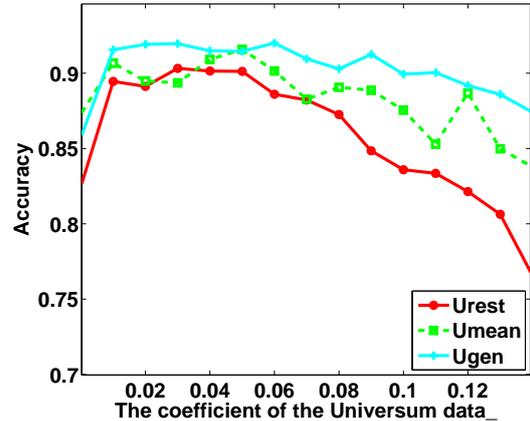
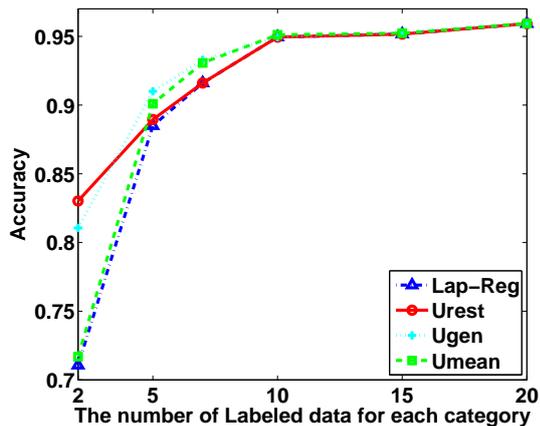


Figure 4: The effect of $C_{\mathcal{U}}$. On MNIST dataset, "5" vs "8", for each category, 5 examples are selected as labeled examples and 500 as unlabeled ones. The accuracy changes with different values of $C_{\mathcal{U}}$. The changes for different type of Universum have been plotted in this figure.

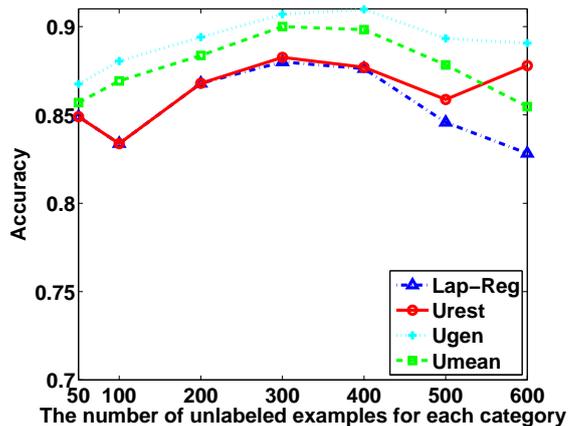
that our intuition tells us that for all the three kinds of Universum methods, when $C_{\mathcal{U}}$ equals zero, their performance should be the same. But since the parameters are tuned to their best performance for each kind of Universum when $C_{\mathcal{U}}$ is fixed, the optimal parameter values are different for \mathcal{U}_{rest} , \mathcal{U}_{mean} and \mathcal{U}_{gen} , and they are fixed when $C_{\mathcal{U}}$ varies. Therefore, when $C_{\mathcal{U}}$ equals zero, the performances of these different kinds of Universum are different. As shown in this figure, for all these three kinds of Universum examples, the accuracy goes up when $C_{\mathcal{U}}$ is appropriate. But when $C_{\mathcal{U}}$ becomes too large, the performance will deteriorate. This is understandable, since the Universum term can only be deemed as an regularization term and too large a $C_{\mathcal{U}}$ makes this term dominate the optimization problem Eq.(4.20).

Table 9 shows the accuracy changes with different number of Universum examples on MNIST dataset. For each category, 5 examples are randomly selected as labeled ones, and 500 examples are randomly chosen as unlabeled ones. The number of Universum examples ranges from 100 to 700. As can be seen, the performance can be improved with more Universum examples. But it is also true that this improvement may not go up endlessly with the increasing number of Universum examples.

Next, we try to analyze the accuracy changes with different number of labeled and unlabeled examples, and conduct on MNIST dataset the "5" vs "8" classifica-



(a)



(b)

Figure 5: The accuracy changes with the varying number labeled and unlabeled examples for the "5" vs. "8" classification task on MNIST data set. For Fig.5(a), the number of unlabeled examples for each category are fixed to 500. The classification accuracy changes with the number of labeled examples are plotted. For Fig.5(b), the number of labeled examples for each category are fixed to 5. The classification accuracy changes with the number of unlabeled examples are plotted.

tion experiments with varying number of labeled and unlabeled examples. The $C_{\mathcal{U}}$ is searched through $\{0, 0.01, 0.1\}$ and all the other parameters are tuned by 5-fold cross validation. In Fig.5(a), we fix the number of unlabeled examples to 500 for each category while in Fig.5(b), the number of labeled examples are set to 5 for each category. The results reported here are averaged over 50 independent runs. We use the same strategy as the experiments shown by Table 6 to generate the three different kind of Universum examples.

As can be seen from Fig.5(a), the advantage of the proposed methods over Lap-Reg is evident, when the number of labeled examples is small. We speculate that the impact of the Universum examples are more vital when the labeled examples are rare. The results shown in Fig.5(b) somewhat contradict our common sense that with more unlabeled examples, performance can be better for semi-supervised methods. It is likely that under our settings the number of unlabeled examples is already enough for their best performance. But it is evident that the proposed method performs better than the Lap-Reg.

7 Conclusions and Future Works

Universum is a recently proposed concept. With the help of the Universum examples, the description of the prior probabilities for all the possible functions can be more accurate. So far as we know, \mathcal{U} -SVM is the only method that can utilize the Universum examples. But \mathcal{U} -SVM is a supervised method. In

fact, many real-world problems have to be processed under the semi-supervised framework. Therefore, in this paper, we consider solving a semi-supervised Universum problem. Furthermore, We have also analyzed the relationship between the Universum problem with some other machine learning problems, such as multi-class classification and ordinal regression problem. In the future, we will try to find that whether the Universum problem can be solved from other perspectives, say, a modified version of ordinal regression. We will also consider if there are any other ways to generate Universum examples.

8 Acknowledgement

This work is supported by National 863 project (No. 2006AA01Z121) and NSFC (Grant No. 60721003). We would like to thank Junfeng He (Columbia University), Yangqiu Song, Feiping Nie, Shouchun Chen for their help with this work. We would also thank the anonymous reviewers for their valuable comments.

References

- [1] M. Belkin, I. Matveeva, and P. Niyogi. Regularization and semi-supervised learning on large graphs. In *COLT*, volume 3120, pages 624–638. Springer Berlin / Heidelberg, January 2004.
- [2] W. Chu and Z. Ghahramani. Gaussian processes for ordinal regression. In *Journal of Machine Learning Research*, 2005.

- [3] O. Delalleau, Y. Bengio, and N. Le Roux. Efficient non-parametric function induction in semi-supervised learning. In R. G. Cowell and Z. Ghahramani, editors, *Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics, Jan 6-8, 2005, Savannah Hotel, Barbados*, pages 96–103. Society for Artificial Intelligence and Statistics, 2005.
- [4] R. Herbrich, T. Graepel, and K. Obermayer. *Large margin rank boundaries for ordinal regression*. MIT Press, Cambridge, MA, 2000.
- [5] T. Joachims. Transductive inference for text classification using support vector machines. In I. Bratko and S. Dzeroski, editors, *Proceedings of ICML-99, 16th International Conference on Machine Learning*, pages 200–209, Bled, SL, 1999. Morgan Kaufmann Publishers, San Francisco, US.
- [6] T. Joachims. Transductive learning via spectral graph partitioning. In *ICML*, pages 290–297, 2003.
- [7] B. Schölkopf, J. Platt, and T. Hoffman. On transductive regression. In *Advances in Neural Information Processing Systems 19*, 2006.
- [8] B. Schölkopf and A. Smola. *Learning with Kernels. Support Vector Machines, Regularization, Optimization and Beyond*. MIT Press, 2002.
- [9] M. Seeger. Gaussian processes for machine learning. *International Journal of Neural Systems*, 14(2):1–38, 2004.
- [10] J. Weston, R. Collobert, F. Sinz, L. Bottou, and V. Vapnik. Inference with the universum. In *ICML '06: Proceedings of the 23rd international conference on Machine learning*, pages 1009–1016, New York, NY, USA, 2006. ACM Press.
- [11] M. Wu and Schölkopf. Transductive classification via local learning regularization. In *Proceedings of the 11th International Conference on Artificial Intelligence and Statistics*, pages 624–631, 03 2007.
- [12] D. Zhou, O. Bousquet, T. Lal, J. Weston, and B. Schölkopf. Learning with local and global consistency, 2003. In 18th Annual Conf. on Neural Information Processing Systems.
- [13] X. Zhu, Z. Ghahramani, and J. Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In *The Twentieth International Conference on Machine Learning, August 21-24, 2003, Washington, DC USA*, pages 912–919, 2003.
- [14] X. Zhu, Z. Ghahramani, and J. D. Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In *ICML*, pages 912–919, 2003.