Multisource transfer learning for host-pathogen protein interaction prediction in unlabeled tasks

Meghana Kshirsagar [1] **Jaime Carbonell** [1] **Judith Klein-Seetharaman** [1,2,3]
[1]Language Technologies Institute, Carnegie Mellon Univ., USA
[2]Forschungszentrum Jülich, Institute of Complex Systems (ICS-5), Germany
[3]Systems Biology Centre, University of Warwick, UK

Abstract

We consider the problem of building a predictive model for host-pathogen protein interactions, when there are no known interactions available. Our goal is to predict the protein protein interactions (PPIs) between the plant host *Arabidopsis thaliana* and the bacterial species *Salmonella typhimurium*. Our method based on transfer learning, utilizes labeled data i.e known interactions from other species (we call these the *source* tasks). The first challenge is to pick the best instances from the source tasks, such that the resultant model when applied on the target task generates high confidence predictions. Towards this, we use the instance reweighting technique Kernel Mean Matching (KMM). The reweighted instances are used to build a kernelized support vector machine (SVM) model, which is applied on the target data. This brings forth the second challenge - selecting appropriate hyperparameters while building a model for a task with no labeled data. For the purposes of evaluation, we apply our method on a task where we have some labeled data available. We find that choice of the right source examples makes a significant difference in performance on the target task.

1 Introduction

The subdiscipline of plant pathology aims at understanding the workings of the immune system in plants, how they develop resistance to various pathogens and has economic importance in food production. *Salmonella typhimurium* is one of the few bacterial species that infects not only animals, but also plants. Some protein interactions between *Salmonella* and some mammalian proteins have been determined. However there exists no plant-*Salmonella* interactions data. In this work, we build a model to predict interactions between *Arabidopsis thaliana* and *Salmonella*. Since there is no labeled data available, we use known host-pathogen protein-protein interactions (PPI) data from other organisms.

We cast the PPI prediction problem as a binary classification task, where given a pair of proteins the goal is to learn a function that predicts whether the pair would interact or not. We derive features on every pair of proteins using protein sequence data. Each host-pathogen PPI prediction problem is considered as one task - the tasks are homogenous in the sense that the class labels are the same. For simplicity we also use the same set of features across each task (protein sequence features). However the data distribution will be different across tasks due to the different organisms involved.

Our transfer learning scenario consist of the following setting: multiple 'source' tasks with small amounts of labeled data, a single 'target' task with no labeled data. This setting has been called *transductive transfer learning* in literature. Our approach is based on instance-transfer where the goal is to pick from each of the source tasks, the most relevant instances w.r.t the target task. We use a two-step process: (1) the first step does the instance weighting on the source tasks. (2) the second step uses the reweighted instance to build several SVM classifier models corresponding to various

hyper-parameter settings. We present two heuristic methods to select the best set of hyperparameters and find improvements in prediction performance.

The literature on addressing sample selection bias [9] presents various methods to perform the first step: instance reweighting. There has also been some work on combining the two steps that jointly optimizes an objective function composing of both criteria. LogReg[1] derives a kernel logistic regression classifier for separating training and test data, using which the importance of source examples can be estimated. [7] combine the KMM and SVM stages into a unified objective called KMM-LM (Large-Margin) where they minimize the classification error on source data along with the KMM objective. Transductive SVM has also been used to jointly learn labels on the target task while minimizing the error on the labeled source examples [4].

2 Approach

2.1 Step-1: Instance reweighting

The similarity between the source and target data can be expressed using the similarity in their distributions $\mathbf{P}_S(x, y)$ and $\mathbf{P}_t(x, y)$. Here $\mathbf{P}_S$ represents the joint distribution of all source tasks. Since we do not have access to the labels y on the target, we make a simplifying assumption that there is only a covariate shift between the source and target tasks - i.e the conditional distribution $P(y|x)$ is the same for both tasks. Mathematically, $\frac{\mathbf{P}_S(x,y)}{\mathbf{P}_t(x,y)} = \frac{\mathbf{P}_S(x)}{\mathbf{P}_t(x)} = r(x)$.

Many methods have been proposed for estimating the ratio r. [6] proposed an algorithm Kullback-Leibler Importance Estimation Procedure (KLIEP) to estimate r directly without estimating the densities of the two distributions. We use the nonparametric Kernel Mean Matching (KMM) [3], which was originally developed to handle the problem of covariate shift between the training and test data distributions. KMM reweighs the training data instances such that the means of the training and test data distributions are close in a reproducing kernel Hilbert space (RKHS). This approach does not require distribution estimation.

Let $x_i^S \sim P_S$ and n_S be the number of source instances from all source tasks. Let $x_i^t \sim P_t$ and n_t be the number of target instances. Let β_i represent the "importance" of the source instances. KMM uses a function based on the *maximum mean discrepancy* statistic (MMD). In the form written below, it minimizes the difference between the empirical means of the joint source and target distributions.

$$\min \mathcal{L}(\beta) = \min_\beta \left\| \frac{1}{n_S} \sum_{i=1}^{n_S} \beta_i \Phi(x_i^S) - \frac{1}{n_t} \sum_{j=1}^{n_t} \Phi(x_j^t) \right\|^2 = \frac{1}{n_S^2} \beta^T K \beta - \frac{2}{n_S^2} \kappa^T \beta + \text{constant} \text{ —- (1)}$$

subject to $\beta_i \in [0, B]$ and $\sum_i \beta_i \leq n_S$, where $K_{i,j} = k(x_i^S, x_j^S)$ is the kernel matrix over all the source examples and $\kappa_i = \frac{n_S}{n_t} \sum_{j=1}^{n_t} k(x_i^S, x_j^t)$. The function (1) is a quadratic program and can be efficiently solved using sequential minimal optimization (SMO), projected gradient based methods.

***Selecting an appropriate set of source and target instances*:**
Using all instances in the optimization problem in equation (1) is infeasible for two reasons. The optimization involves the computation of the gram matrix K of $O(n^2)$ where n is the number of instances. Typically the total number of protein-protein pairs between a host-pathogen are of the order of 100 million. Secondly, the total number of labeled source instances is quite small ($\approx$ 1500). This set is likely to get underweighted (i.e $\beta_i \approx 0$) if there are too many unlabeled source instances. To represent the source's empirical mean, in addition to the labeled instances we randomly sample four times as many unlabeled instances. For the target, we randomly sample n_S instances.

2.2 Step-2: Model learning

Once we have the optimal set of source instances, we can train a Kernel-SVM model using these. We pick a kernel-based learning algorithm since we plan to extend our work to deal with different feature spaces across the tasks. In such a scenario, the only mechanism to operate on the target data

is via similarities, i.e the kernel. The dual formulation for the weighted version of SVM optimizes: $\sum_{i=1}^{n_S} \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j K(x_i^S, x_j^S)$, subject to $\sum_i \alpha_i y_i = 0$ and $\beta_i C \geq \alpha_i \geq 0$.

2.3 Model selection

Parameter tuning and selecting the best model in the absence of labeled data is a very hard problem. The model built on the source data can not tuned using cross validation on the source data because doing so will optimize it for the source distribution. Hence we develop two heuristic approaches to select the best hyperparameters. The first one uses the expected class-skew on the target task while the second uses reweighted cross-validation.

Class-skew based parameter selection (**skew**)
We first learn several models by doing a grid-search on the classifier hyper-parameters. There are 3 parameters to tune for the Kernel-SVM: the kernel width γ, the cost parameter C, the weight parameter for the positive class w_+. The total number of parameter combinations in our grid-search were 50. We thus had 50 models trained on the reweighted source data obtained after KMM in Step-1 (Section 2.1). We apply each model on the target data and computed the predicted class-skew r_{pred} using the predicted class labels. The expected class skew based on our understanding of the PPI experimental literature is roughly 1:100 (= r_{true}). We ranked all 50 models on the statistic $|r_{pred} - r_{true}|$. The top k models are selected based on this criteria and a weighted voting ensemble was built using them. This ensemble is used to get the final class label on the target data.

Reweighted cross-validation (**rwcv**)
This method uses the assumption that the reweighted source data is distributionally similar to the target data. We do a 5 fold cross-validation on the reweighted source data. During training we use the weighted version of the classifier and during test we compute the weighted error on the source test examples. The optimal model is then applied on the target task's data.

2.4 Datasets

As source datasets we used the known PPIs between *Francisella tularensis* - human, *E.coli* - human and *Salmonella typhimurium* - human. The sizes of these three datasets are 1380, 32 and 62 respectively. The first two sets of interactions were obtained from the PHISTO [8] database and the last one from [5]. The target task is the prediction of PPIs between *Arabidopsis thaliana* and *Salmonella*. A quantitative evaluation for the target task is infeasible due to the lack of any labeled data. Hence, we used two of the labeled datasets as 'sources' for building a model and the third as the 'target' to evaluate our method on and as a proof of concept. The prediction accuracy results presented in the next section were obtained in this setting.

Negative examples: Since there is no experimental evidence about proteins that do not interact, we construct the "non-interacting" (i.e negative) class using random pairs of proteins sampled from the set of all possible bacteria-human protein pairs. The number of random pairs chosen as the negative class is decided by what we expect the interaction ratio to be. We chose a ratio of 1:100 meaning that we expect 1 in every 100 random bacteria-human protein pairs to interact with each other.

2.5 RBF Spectrum kernel

We use a variant of the spectrum kernel, based on the features used by [2] for HIV-human PPI prediction. The kernel uses the n-mers of a given input sequence and is defined as: $k_{sp}^n(x, x') = \exp\{-\frac{\|\phi_{sp}^n(x) - \phi_{sp}^n(x')\|^2}{\sigma^2}\}$, where x, x' are two sequences over an alphabet Σ. Instead of using the 20 amino acids as the alphabet Σ, we use a classification of the amino-acids. There are seven classes based on the electrostatic and hydrophobic properties of proteins, i.e $|\Sigma|$=7. Here ϕ_{sp}^n transforms a sequence s into a $|\Sigma|^n$-dimensional feature-space. One dimension of ϕ_{sp}^n corresponds to the normalized frequency of one of the 7^n possible strings in s. We use n=2,3,4,5.

Source tasks	Target task	Method	P†	R†	F1†	MAP	ROC
Francisella-human, *E.coli*-human	*Salmonella*-human	Baseline	9.5	20.9	13	0.076	0.74
		Baseline *skew*	17.8	12.9	14.9	0.11	**0.78**
		KMM-SVM *skew*	25.7	**16.1**	**19.9**	0.106	0.74
		KMM-SVM *rwcv*	**30.4**	11.3	16.5	**0.126**	0.76
		T-SVM	15	14.5	14.7	0.077	0.71
Francisella-human, *Salmonella*-human	*E.coli*-human	Baseline	5.2	15.6	7.8	0.033	0.73
		Baseline *skew*	12.9	12.5	12.7	0.076	0.83
		KMM-SVM *skew*	**15.9**	**21.9**	**18.4**	0.075	0.84
		KMM-SVM *rwcv*	14.3	3.1	5.1	**0.122**	**0.87**
		T-SVM	10.4	15.6	12.5	0.1	0.77

†- computed using the default classifier threshold: 0.5, *rwcv*, *skew*: parameter tuning methods

2.6 Evaluation

We present results on two different task configurations: (a) using the *Salm*-human task as the target in one and others as source tasks, (b) the *Ecoli*-human task as target with others as sources. None of the methods sees any labeled data from the target task. Our evaluation criteria does not use accuracy which measures performance on both the classes. Since our datasets are highly imbalanced with a large number of negative samples, we instead use precision (P), recall (R) and F1 computed on the interacting pairs (positive class). We also report the mean average precision (MAP) and area under the ROC curve (ROC). We compare the KMM-based method with the following baselines:
Inductive Kernel-SVM (Baseline) : This model assumes that the source and target distributions are identical. All available labeled examples are pooled and used to build a model using 5-fold cross validation. This model is then applied to the target task.
Transductive SVM (TSVM) : Here we apply transductive SVM in the transfer learning setting. The target data is treated as the 'test' data. Note that we do not use the labeled examples from the target task during training. Parameter tuning is done using cross validation on the source data. For the kernel we used the RBF-spectrum kernel.

Table 2.6 shows the results. The two parameter tuning methods from Section 2.3 are listed as 'skew' and 'rwcv'. We see that in general, KMM-SVM with the *skew* parameter tuning heuristic has significantly better P, R and F1. The F1 on the *Salm.* task is 3.4 points better than all others, while that on the *Ecoli* task is 5.7 points better. The *rwcv* based KMM-SVM has poor P, R, F1 on one of the tasks but a much higher MAP and ROC than other methods. However, leveraging this performance will require picking the optimal classifier threshold which is hard without access to the target labels. The performance of T-SVM is similar to that of the baselines.

Evaluation on *A. thaliana-Salmonella*: We do a qualitative analysis of these results using Gene Ontology term enrichment analysis and find many interesting terms.

References

[1] Steffen Bickel, Michael Brckner, and Tobias Scheffer. Discriminative learning for differing training and test distributions. *ICML*, 2007.

[2] M.D. Dyer, T.M. Murali, and B.W. Sobral. Computational prediction of host-pathogen protein-protein interactions. *Bioinf.*, 23(13):i159–66, 2007.

[3] J. Huang, A. Smola, A. Gretton, K.M. Borgwardt, and B. Scholkopf. Correcting sample selection bias by unlabeled data. *NIPS*, 2007.

[4] M. Kshirsagar, J. G. Carbonell, and J. Klein-Seetharaman. Transfer learning methods for the discovery of host-pathogen protein interactions. *Intelligent Systems for Molecular Biology (ISMB) poster*, 2012.

[5] S. Schleker, J. Sun, et al. The current salmonella-host interactome. *Proteomics Clin Appl.*, 2012.

[6] M. Sugiyama, S. Nakajima, H. Kashima, P.V. Buenau, and M. Kawanabe. Direct importance estimation with model selection and its application to covariate shift adaptation. *NIPS*, 2008.

[7] Qi Tan, Huifang Deng, and Pei Yang. Kernel mean matching with a large margin. *LNCS*, 2012.

[8] S.D. Tekir, Ali S., Tunahan C., and Kutlu O.U. Infection strategies of bacterial and viral pathogens through pathogen-host protein interactions. *Frontiers in Microb. Immun.*, 2012.

[9] B. Zadrozny. Learning and evaluating classifiers under sample selection bias. *ICML*, 2004.