Online Mean Field Approximation for Automated Experimentation

Shaona Ghosh
University of Southampton, UK

Adam Prügel-Bennett
University of Southampton, UK

Abstract

In this paper, we propose a semi-supervised online graph labelling method that affords early learning capability. We use mean field approximation for predicting the unknown labels of the vertices of the graph with high accuracy on the standard benchmark datasets. The minimum cut is the energy function of our probabilistic model that encodes the uncertainty about the labels of the vertices. Our method shows that it can learn early given any choice of experiments that may take place in the automated experimentation systems used for scientific discovery.

1 Introduction

Often in scientific discovery, there are physically implemented interaction systems/laboratory equipment conducting vast number of automated scientific experimentation. An example of such a system from functional genomics is the Robot-Scientist framework [5]. Typically, the experiments conducted by such machines are managed through human choice of experiments or by simple selection protocols. However, some such machines that interact with biological systems, are met with a number of challenges given the restrictive nature of those systems. The available initial experimental results are far too limited to guide the model/selection of experiments. Further, the number of experiments required to completely characterize the behavioural response of the biological system is large compared to the number of unknown parameters. With the rich underlying structured graphical domain knowledge represented by the biochemical reactions taking place in the system, it is imperative for the interaction system to employ a sophisticated machine learning algorithm that learns early from the given structure regardless of the sequence of experiments, from known and unknown response variables in semi supervised nature. Our work operates in this area by designing an online graph labelling algorithm based on mean field approximation that promises such capability. Although, the traditional goal in automated experimentation systems is in experiment planning; our experiments are chosen randomly, with the goal of learning early given any sequence of experiments. We validate the prediction quality of our method on standard benchmark datasets. As future work, we plan to apply our method to the automated experimentation platform discussed in the papers [7, 5].

Semi-supervised learning is a well established approach of learning from a few examples. On a graph that is built from the given labelled and unlabelled data points, where each datum is represented as a vertex, two vertices share the same label if they are connected by an edge. The standard graph labelling semi-supervised algorithms in the literature use the graph Laplacian in order to optimize the labelling consistent with the labels seen so far [12, 11, 4, 3]. Graph Laplacian based methods suffer from the limitations in the light of many unlabelled data [8]. Our work uses a similar regularity measure for smoothness of the labelling that instead induces an Ising model distribution over the vertices of the graph. Our work is different from the approximation method in label propagation [12, 11] as we do not drop the higher order terms in the approximation. Also, our energy equation 2, is the complement of theirs. Treeopt algorithm [10, 2] is optimal over a tree without graph connectivity utilization. [1]

2 Mean Field Approximation

We consider a unit weighted graph $G = (\mathcal{V}, \mathcal{E}, w)$ where $w(e)$ is the weight of edge $e \in \mathcal{E}$ (we assume that $w(e) = 0$ if $e \notin \mathcal{E}$). For convenience, we de-

Appearing in Proceedings of the 4^{th} Workshop on Machine Learning for Interactive Systems (MLIS) 2015, Lille, France. JMLR: W&CP volume 43. Copyright 2015 by the authors.

[1] In-spite of being a decade old, to the best of our knowledge the algorithms [12, 11, 10, 2] are still the state-of-the-art in online graph/tree labelling semi-supervised setting.

note the weight of edge (i,j) by $w((i,j)) = w_{ij}$. We consider the scenario when we are given labels for a subset of the vertices, $\mathcal{L} \subset \mathcal{V}$. We denote the label of vertex i by S_i, which can take a value from the class set $\mathcal{C}$. Our task is to assign labels to the unobserved vertices $\mathcal{U} = \mathcal{V} \setminus \mathcal{L}$. The weights are taken to be a measure of similarity so that vertices that share an edge are more likely to have the same label than other vertices. Because of this, our labelling not only depends on the labelled vertices, but also on the structure of the graph (heavily connected components of the graph are likely to have the same label). In this sense this can be viewed as a semi-supervised learning problem (we have a few labelled examples, but we are also learning from unlabelled data). We use a probabilistic model to encode our uncertainty about the labels of the vertices in $\mathcal{U}$, where we assume

$$\mathbb{P}\left(\boldsymbol{S}\right) = \frac{\mathrm{e}^{-\beta\, E(\boldsymbol{S},\boldsymbol{S}^o)}}{Z}, \qquad Z = \sum_{\boldsymbol{S}} \mathrm{e}^{-\beta\, E(\boldsymbol{S},\boldsymbol{S}^o)} \tag{1}$$

where $\boldsymbol{S}$ denotes the labels at the unobserved vertices, $\boldsymbol{S}^o$ the labels at the observed vertices, β is a parameter (the inverse temperature) encoding the degree of uncertainty, the sum is over the labels of the unobserved vertices

$$\sum_{\boldsymbol{S}} \cdots = \left(\prod_{i \in \mathcal{U}} \sum_{S_i \in \mathcal{C}} \right) \cdots$$

and $E(\boldsymbol{S}, \boldsymbol{S}^o)$ is a minimum cut energy function given by [2]

$$E(\boldsymbol{S}, \boldsymbol{S}^o) = \frac{1}{2} \sum_{(i,j)\in\mathcal{E}} w_{ij} \, [\![S_i \neq S_j]\!] \tag{2}$$

As in any online learning algorithm, a sequential game is played between the learner and the adversary or environment. The learner's goal is to minimize its mistaken predictions (wrong class prediction) while the adversary's goal is the opposite. Nature selects a data-point/vertex at every trial/experiment, learner predicts the class, nature then returns the true label to verify if the learner made a mistake. The only restriction is the adversary cannot return a label that increases the minimum-cut. Note, the probability function in 1 favours labellings that minimise the number of edges whose vertices have different labels. Unfortunately, for large vertex sets, computation of this is intractable. We therefore resort to using a variational approximation where we minimise the variation free energy given by

$$\Phi(\boldsymbol{\theta}) = \sum_{\boldsymbol{S}} \mathbb{Q}\left(\boldsymbol{S}|\boldsymbol{\theta}\right) \log\left(\frac{\mathbb{Q}\left(\boldsymbol{S}|\boldsymbol{\theta}\right)}{\exp(-\beta\, E(\boldsymbol{S},\boldsymbol{S}^o))} \right)$$

where $\mathbb{Q}\left(\boldsymbol{S}|\boldsymbol{\theta}\right)$ is taken as a separable probability distribution

$$\mathbb{Q}\left(\boldsymbol{S}|\boldsymbol{\theta}\right) = \prod_{i\in\mathcal{U}} \sum_{\mu\in\mathcal{C}} \theta_i^\mu \; [\![S_i = \mu]\!]$$

with $\theta_i^\mu \geq 0$ and for all vertices $\sum_{\mu\in\mathcal{C}} \theta_i^\mu = 1$. The parameters θ_i^μ can be interpreted as the marginal probability of the label for vertex i to be in class μ. As is well known, we can rewrite the variational free energy as

$$\Phi(\boldsymbol{\theta}) = \mathrm{KL}\left(\mathbb{Q}\left(\boldsymbol{S}|\boldsymbol{\theta}\right) \middle\| \mathbb{P}\left(\boldsymbol{S}\right)\right) - \log(Z)$$

where $\mathbb{P}\left(\boldsymbol{S}\right)$ and Z are given in equation (1) and $\mathrm{KL}\left(\mathbb{Q}\left(\boldsymbol{S}|\boldsymbol{\theta}\right) \middle\| \mathbb{P}\left(\boldsymbol{S}\right)\right)$ is the KullbackLeibler (KL) divergence given by

$$\mathrm{KL}\left(\mathbb{Q}\left(\boldsymbol{S}|\boldsymbol{\theta}\right) \middle\| \mathbb{P}\left(\boldsymbol{S}\right)\right) = \sum_{\boldsymbol{S}} \mathbb{Q}\left(\boldsymbol{S}|\boldsymbol{\theta}\right) \log\left(\frac{\mathbb{Q}\left(\boldsymbol{S}|\boldsymbol{\theta}\right)}{\mathbb{P}\left(\boldsymbol{S}\right)} \right).$$

[3] Since $\log(Z)$ does not depend on the variational parameters, $\boldsymbol{\theta}$, minimising $\Phi(\boldsymbol{\theta})$ is equivalent to minimising the KL-divergence. Thus, in minimising the variational free energy we are choosing the parameters $\boldsymbol{\theta}$ so that $\mathbb{Q}\left(\boldsymbol{S}|\boldsymbol{\theta}\right)$ is as close as possible (as measured by the KL-divergence) to $\mathbb{P}\left(\boldsymbol{S}\right)$. Furthermore as the KL-divergence is non-negative we obtain a bound that $-\log(Z) \leq \Phi(\boldsymbol{\theta})$ (in classical physics $-\beta \log(Z)$ is known as the free energy). We can also rewrite the variational free energy as

$$\Phi(\boldsymbol{\theta}) = -H(\mathbb{Q}) + \beta\, U(\mathbb{Q})$$

where $H(\mathbb{Q})$ is the entropy of $\mathbb{Q}\left(\boldsymbol{S}|\boldsymbol{\theta}\right)$

$$\begin{aligned} H(\mathbb{Q}) &= -\sum_{\boldsymbol{S}} \mathbb{Q}\left(\boldsymbol{S}|\boldsymbol{\theta}\right) \log(\mathbb{Q}\left(\boldsymbol{S}|\boldsymbol{\theta}\right)) \\ &= -\sum_{\boldsymbol{S}} \mathbb{Q}\left(\boldsymbol{S}|\boldsymbol{\theta}\right) \sum_{i\in\mathcal{U}} \log\left(\sum_{\mu\in\mathcal{C}} \theta_i^\mu \; [\![S_i = \mu]\!] \right) \\ &= -\sum_{i\in\mathcal{U}} \sum_{\mu\in\mathcal{C}} \theta_i^\mu \; \log(\theta_i^\mu) \end{aligned}$$

and $U(\mathbb{Q})$ is the "mean energy" with respect to the probability distribution $\mathbb{Q}\left(\boldsymbol{S}|\boldsymbol{\theta}\right)$

$$\begin{aligned} U(\mathbb{Q}) &= \sum_{\boldsymbol{S}} \mathbb{Q}\left(\boldsymbol{S}|\boldsymbol{\theta}\right) \; E(\boldsymbol{S}, \boldsymbol{S}^o) \\ &= \sum_{\boldsymbol{S}} \mathbb{Q}\left(\boldsymbol{S}|\boldsymbol{\theta}\right) \; \frac{1}{2} \sum_{(i,j)\in\mathcal{E}} w_{ij} \; [\![S_i \neq S_j]\!] \\ &= \frac{1}{2} \sum_{i\in\mathcal{U}} \sum_{j\in\mathcal{U}} w_{ij} \sum_{\mu,\nu\in\mathcal{C}} \theta_i^\mu \theta_j^\nu \; [\![\mu \neq \nu]\!] \\ &\quad + \sum_{i\in\mathcal{U}} \sum_{j\in\mathcal{L}} w_{ij} \sum_{\mu\in\mathcal{C}} \theta_i^\mu \; [\![\mu \neq S_j]\!] \\ &\quad + \frac{1}{2} \sum_{i\in\mathcal{L}} \sum_{j\in\mathcal{L}} w_{ij} \; [\![S_i \neq S_j]\!] \, . \end{aligned}$$

[2] where $[\![$predicate$]\!]$ denotes an indicator function which is equal to 1 if the predicate is true and 0 otherwise. It is important to note that in label propagation [12], the authors use the complement of this function.

[3] The KL-divergence between two probability distributions can be viewed as a measurement of their difference. It is non-negative and reaches it minimum value of zero when the two distributions are identical (at least, the distributions can only differ on sets of measure zero).

To find the minimum of the variational free energy subject to $\sum_{\mu \in \mathcal{C}} \theta_i^\mu = 1$ at each vertex we minimise the Lagrangian

$$L(\boldsymbol{\theta}) = \Phi(\boldsymbol{\theta}) + \sum_{i \in \mathcal{U}} \lambda_i \sum_{\mu \in \mathcal{C}} \theta_i^\mu$$

where the λ_i's are a set of Lagrange multipliers that are chosen to ensure the constraints are satisfied. The "mean-field equations" which are satisfied at the minima of the variational free energy are given by

$$\frac{\partial L(\boldsymbol{\theta})}{\partial \theta_i^\mu} = \log(\theta_i^\mu) + 1 + \beta \sum_{j \in \mathcal{U}} w_{ij} \sum_{\substack{\nu \in \mathcal{C} \\ \nu \neq \mu}} \theta_j^\nu + \beta \sum_{j \in \mathcal{L}} w_{ij} \, [\![\mu \neq S_j]\!] + \lambda_i = 0.$$

These equations are not in general solvable in closed form. Instead we can attempt to solve these equations iteratively by setting

$$\theta_i^\mu(t+1) = \frac{\mathrm{e}^{-\beta \tilde{E}_i^\mu(\boldsymbol{\theta}(t), \boldsymbol{S}^o)}}{\sum_{\nu \in \mathcal{C}} \mathrm{e}^{-\beta \tilde{E}_i^\nu(\boldsymbol{\theta}(t), \boldsymbol{S}^o)}}$$

where

$$\tilde{E}_i^\mu(\boldsymbol{\theta}, \boldsymbol{S}^o) = \sum_{j \in \mathcal{U}} w_{ij} \sum_{\substack{\nu \in \mathcal{C} \\ \nu \neq \mu}} \theta_j^\nu + \sum_{j \in \mathcal{L}} w_{ij} \, [\![\mu \neq S_j]\!]$$

(this is a self-consistent soft-max solution). Note that we have chosen λ_i so that the constraint is satisfied (this gives the normalisation term). There can be many local solutions to the mean-field equations so that the result will depend on the initial conditions. To prevent finding very poor solutions we can anneal the temperature (start from a low value of β and increase it to the required value) at each iteration. [4]

For the prediction of the class label, we allow to slowly anneal the value of β until stabilized Once stabilized, we predict the label of the query point by the class label that maximizes θ for that vertex. Once the true label is revealed, we update the partial labelling vector $\boldsymbol{S}^o$, increment mistake count (if a mistake has been made) and continue with the annealing process. Since we assume the adversary cannot increase the minimum cut, it cannot force a mistake on the learner; the cost of the feedback from the environment is trivial. This is especially important in autonomous experimentation systems that interact with biological systems, where the feedback can be uncharacteristic. The main focus of our work is in minimizing the total number of mistakes and not the computational efficiency of the method. The inverse temperature β controls the amount of uncertainty in our model. We could choose it through cross-validation. Alternatively, we can take a Bayesian interpretation in which we take the joint probability of the unobserved spins $\boldsymbol{S}$ and the observed spins $\boldsymbol{S}^o$ to be

$$\mathbb{P}(\boldsymbol{S}, \boldsymbol{S}^o) = \frac{\mathrm{e}^{-\beta E(\boldsymbol{S}, \boldsymbol{S}^o)}}{Z'}, \quad Z' = \sum_{\boldsymbol{S}, \boldsymbol{S}^o} \mathrm{e}^{-\beta E(\boldsymbol{S}, \boldsymbol{S}^o)}$$

where Z' is the partition function under the assumption that no labels are observed. The probability of the observed spins is

$$\mathbb{P}(\boldsymbol{S}^o) = \sum_{\boldsymbol{S}} \frac{\mathrm{e}^{-\beta E(\boldsymbol{S}, \boldsymbol{S}^o)}}{Z'} = \frac{Z}{Z'}.$$

We have seen that the variational free energy $\Phi(\boldsymbol{\theta}^*)$ (where $\boldsymbol{\theta}^*$ is our solution to the mean-field equation) acts as an approximation for $-\log(Z)$ (the evidence). We can similarly introduce a variational free energy to compute $-\log(Z')$ (we repeat the calculation except with no observed spins). Choosing the value of β which maximises the difference $\log(Z) - \log(Z')$ would provide an approximation to the best value of β (i.e. it is the value which maximises the probability of the data). [5] [6]

3 Empirical Evaluation

We perform empirical tests to evaluate our method on standard machine learning datasets from UCI. We use our own implementation of the `meanField` method and the competitor algorithm `labelProp` [12, 11], while we adapt the code for `treeOpt` given to us by the authors in [10]. In general, for our experiments we use an uniform way of sampling instances and building the graphs from the datasets. For datasets `ISOLET` (UCI), `webSpam` [1] and a subset `20 newsGroups` [9], we randomly sample instances from the entire dataset. The sampling also ensures that both the classes are equally represented. Each instance is represented as a vertex in the graph. An Euclidean distance matrix is constructed using the pairwise distances between the instances. In the case of `20 newsGroups`, instead of using the Euclidean distance matrix, we use the cosine distance matrix for binary valued instances. We build a $3-NN$ nearest neighbour graph

[4] We may further wish to randomly choose the order of the variables we are updating to reduce the bias caused by the order of updating (alternatively we can update all the variables at once).

[5] However, for very large problems this may not be feasible and we may just have to use some guess for β based on experimentation. A nice feature of this framework is that we obtain the marginal distribution for the labels. This can be used in any decisions theoretic framework.

[6] The quality of the mean-field approximation is hard to determine a priori. The separable probability distribution $\mathbb{Q}(\boldsymbol{S}|\boldsymbol{\theta})$ will not capture the strong dependencies between many of the labels. These are not entirely ignored in the approximation since they come in through the $U(\mathbb{Q})$ term. One can investigate the quality of the approximation by considering a small system where we can compute the sum over all labellings exactly.

	$\ell = 250$	$\ell = 450$	$\ell = 650$	$\ell = 850$	$\ell = 1050$
`labelProp`	$.842 \pm .010$	$\mathbf{.908} \pm .005$	$\mathbf{.940} \pm .005$	$\mathbf{.956} \pm .004$	$\mathbf{.969} \pm .003$
`meanField`	$\mathbf{.846} \pm .014$	$.894 \pm .006$	$.925 \pm .006$	$.942 \pm .005$	$.955 \pm .005$

Figure 1: [Experiment `Squares`] Classification of 0 intensity pixels against 1 intensity pixels. `labelProp` and `meanField` are extremely competitive, with `labelProp` eventually outperforms `meanField`

	$\ell = 800$	$\ell = 1000$	$\ell = 2000$	$\ell = 4000$	$\ell = 6000$
`labelProp`	$\mathbf{.826} \pm .005$	$\mathbf{.826} \pm .006$	$\mathbf{.845} \pm .004$	$\mathbf{.871} \pm .002$	$.868 \pm .003$
`treeOpt`	$.753 \pm .006$	$.772 \pm .01$	$.805 \pm .002$	$.858 \pm .004$	$.868 \pm .002$
`meanField`	$.800 \pm .006$	$.805 \pm .009$	$.833 \pm .003$	$.865 \pm .002$	$\mathbf{.890} \pm .002$

Figure 2: [Experiment `20 newsGroups`] It is a sparse dataset. Label distribution is 8124:8118 on classifying (comp.* and rec.* Vs. sci.* and talk.*) newsgroups. `meanField` beats `labelProp` with enough information. `treeOpt` under-performs until the labels are sufficiently large for comparable performance.

	$\ell = 200$	$\ell = 400$	$\ell = 600$	$\ell = 800$	$\ell = 900$
`labelProp`	$\mathbf{.729} \pm .075$	$.698 \pm .015$	$\mathbf{.803} \pm .001$	$\mathbf{.805} \pm .0001$	$.806 \pm .0001$
`treeOpt`	$.728 \pm .018$	$\mathbf{.734} \pm .007$	$.71 \pm .013$	$.747 \pm .003$	$.749 \pm .001$
`meanField`	$.677 \pm .036$	$.645 \pm 0.002$	$.770 \pm .008$	$.800 \pm .003$	$\mathbf{.808} \pm .0004$

Figure 3: [Experiment `webSpam`] `webSpam` is a sparse dataset of a computer hosts network. Classifying spam Vs. non-spam hosts, we see that `meanField` eventually surpasses `labelProp`, with `treeOpt` being competitive.

	$\ell = 32$	$\ell = 64$	$\ell = 128$	$\ell = 256$	$\ell = 512$	$\ell = 1024$
`labelProp`	$.661 \pm .039$	$\mathbf{.707} \pm .029$	$.764 \pm .024$	$.787 \pm .015$	$.82 \pm .012$	$\mathbf{.869} \pm .006$
`treeOpt`	$.658 \pm .042$	$.688 \pm .033$	$.731 \pm .012$	$.786 \pm .022$	$.824 \pm .008$	$.859 \pm .005$
`meanField`	$\mathbf{.700} \pm .022$	$.700 \pm .022$	$\mathbf{.791} \pm .014$	$\mathbf{.813} \pm .017$	$\mathbf{.837} \pm .011$	$.860 \pm .006$

Figure 4: [Experiment `ISOLET`] `meanField` beats `labelProp` in most of the settings. `ISOLET` has a 3900:3897 ratio between the labels while classifying the first 13 letters against the next 13 letters. Files `ISOLET 1` through `ISOLET 5` are used for the construction of the graph.

using the distance matrix built as the previous step. For ensuring that the graph thus constructed is connected, we always sample a minimum spanning tree (randomly) using the Euclidean distance or cosine distance as weights. We ensure that the MST edges are maintained. Having MST edges also allows for sparsity in the graph. All trials receive the same graph.[7] We choose the connectivity of $K = 3$ for our experiments, as in the literature, empirical evidences show competitive performance for 3-NN connectivity.[8] The synthetic dataset `Squares` is a 60x60 image from which a grid graph with 3600 vertices is constructed, where each pixel is represented as a vertex. We ensure that the graph has toroidal boundary properties such that each vertex is surrounded by four neighbours based on pixel locations. The results of the experiments for each dataset are discussed in Figures 1,2,3,4. The performance of the algorithms is measured as accuracy of the prediction i.e. number of correctly classified instances over all the instances; higher the better. For all datasets, results are averaged over 10 trials except for `20 newsGroups` which used 5 trials. The randomly sampled labels l are balanced. For all the experiments, the choice of β is annealed to a value of 2.4 before prediction. Due to computational feasibility, we perform the experiments in the batch setting rather than online.

4 Conclusion

Here, we use an online learning meanfield approximation technique for graph labelling in a semi-supervised setting. If we incorporate the Halving algorithm and other online graph prediction techniques [6, 4], where we predict such that maximum number of hypotheses are eliminated in case of a mistake, as we plan to do in our future work; we are sure to see `meanField` challenging `labelProp` more often. If we relax the adversary such that in can increase the minimum cut, then that could be the beginning of another online learning game, the idea then should be to minimize mistakes over all such games.

[7] We use benchmark classification datasets, graphs built are sparse; biological systems we are interested in represent sparse graphs.

[8] We use quad-core processor notebooks (@2.30 GHz each) with 8GB and 16GB RAM. We also use the Iridis 4 HPC cluster at University of Southampton, UK.

References

[1] Web Spam Challenge home page. http://webspam.lip6.fr/wiki/pmwiki.php, 2007. [Online;accessed 30-April-2015].

[2] Nicolo Cesa-Bianchi, Claudio Gentile, and Fabio Vitale. Fast and optimal prediction of a labeled tree. In *Proceedings of the 22nd Annual Conference on Learning*, 2009.

[3] Mark Herbster. Exploiting cluster-structure to predict the labeling of a graph. In *Proceedings of the 19th International Conference on Algorithmic Learning Theory*, pages 54–69, 2008.

[4] Mark Herbster, Massimiliano Pontil, and Lisa Wainer. Online learning over graphs. In *ICML '05: Proceedings of the 22nd international conference on Machine learning*, pages 305–312, New York, NY, USA, 2005. ACM.

[5] Ross D King, Kenneth E Whelan, Ffion M Jones, Philip GK Reiser, Christopher H Bryant, Stephen H Muggleton, Douglas B Kell, and Stephen G Oliver. Functional genomic hypothesis generation and experimentation by a robot scientist. *Nature*, 427(6971):247–252, 2004.

[6] Nick Littlestone. Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm. *Machine Learning*, 2:285–318, April 1988.

[7] Chris Lovell, Gareth Jones, Steve Gunn, and Klaus-Peter Zauner. Autonomous experimentation: Active learning for enzyme response characterisation. In *JMLR: Workshop and Conference Proceedings*, volume 16, pages 141–155. JMLR, 2011.

[8] Ulrike V. Luxburg, Agnes Radl, and Matthias Hein. Getting lost in space: Large sample analysis of the resistance distance. In *NIPS 23*, pages 2622–2630. 2010.

[9] S. Roweis. Data for matlab hackers. http://www.cs.nyu.edu/ roweis/data.html, 2006. [Online; accessed 29-April-2015].

[10] Fabio Vitale, Nicolo Cesa-Bianchi, Claudio Gentile, and Giovanni Zappella. See the tree through the lines: The shazoo algorithm. In *Advances in Neural Information Processing Systems*, pages 1584–1592, 2011.

[11] Xiaojin Zhu and Zoubin Ghahramani. Towards semi-supervised classification with markov random fields. Technical Report CMU-CALD-02-106, Carnegie Mellon University, 2002.

[12] Xiaojin Zhu, Zoubin Ghahramani, and John D. Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In *ICML*, pages 912–919, 2003.