EFFICIENT WORKER ASSIGNMENT IN CROWDSOURCED DATA LABELING USING GRAPH SIGNAL PROCESSING

Javier Maroto and Antonio Ortega

Ming Hsieh Department of Electrical Engineering
University of Southern California
Los Angeles, CA 90089

ABSTRACT

The first step in solving a classification problem is to collect and label a sufficient amount of training data. Given the time and cost associated to data labeling, crowdsourcing systems (e.g., Amazon Mechanical Turk) are often used. However, one of the key disadvantages of crowdsourcing systems is the presence of spammers or workers who are not as skilled or careful, thus leading to many false labels being assigned. This paper addresses this problem by proposing a novel algorithm based on graph signal sampling theory, which optimally assigns data to different workers for labeling by taking into account the expected quality of labeling provided by each worker. Our simulation of the labeling process using these schemes shows that the classification error can be reduced significantly with respect to a random assignment of workers.

***Index Terms*—** Crowdsourcing, resource allocation, labeling, Graph Signal Processing

1. INTRODUCTION

One of the key problems in semi-supervised learning is that of determining which items to label. Substantial progress has been made in this area with a focus on active learning, where the main objective is to select for labeling those data points that are most informative. State of the art studies [1, 2, 3] address the problem of selecting data points to label in order to minimize the class prediction error achievable with a given number of labels.

Our work is motivated by the observation that none of these works take into consideration potential errors in the labeling. Assuming that there are no labeling errors, or that their number is negligible, is reasonable when labeling is conducted by experts [4], or when there are guarantees that a sufficient amount of time is being invested in the labeling task [5]. In practice, however, this is rarely the case. Manual labeling is subject to significant constraints, including cost, time and reliability. Other more subjective factors, such as honesty of the workers, diversity of the group of labelers or the wording of the questions, influence the labels obtained [6].

Ideally, one would like to maximize the quality of labels obtained, but it is easy to see that hiring experts or supervising people to ensure they are focused on the task will be expensive and time-consuming. Therefore it is becoming increasingly common to use crowd-sourcing platforms such as Amazon's Mechanical Turk [7] or Crowdflower [8] in order to obtain labels. Crowdsourcing can be much cheaper, and some studies have shown that acceptable label quality can be achieved in some cases [6]. Crowdsourcing has additional advantages for subjective classification tasks. First, use of a standardized process allows no bias from the people that request the tasks. Second, because each worker is anonymous and is not supervised, more honest answers are likely to be provided. However, because of this anonymity, it is more likely that there may be some low-performing workers, or spammers, that could be a significant source of error [9].

In this paper we use the label set selection approach of [3] as a starting point, but then consider the problem of assigning workers to tasks in a situation where workers can provide erroneous labels. We assume that estimates of the reliability of each worker are available before we start the worker assignment process. The approach in [3], based on graph signal processing (GSP) concepts, has promising performance gains and is based on a metric that quantifies the relative importance of different data points for approximating the true label "signal". In our proposed method, we assign those samples previously selected to only one worker (i.e., single labeling) depending on that worker's reliability. Then, we use the (possibly incorrect) labels given by the workers to compute estimated labels for unlabeled data using the iterative distributed algorithm based on projection onto convex sets (POCS) from [3].

Several approaches have been proposed to address lack of worker reliability in crowd-sourcing problems. Unlike our proposed method, these approaches do not restrict themselves to finding one label per sample [10, 11, 12]. For instance, [10] introduces modifications to support vector machines in order to make them more robust to labeling noise, but does not consider that different workers could have different levels of reliability;[11] uses proactive learning to address the unreliable labeling problem and similarly considers several types of unreliable or "non-ideal" workers (e.g., they can fail or be slower), and chooses between them based on a cost metric, but does not study how to reduce the effect of unreliable workers as we do; finally [12] uses active learning as well and estimates both the accuracy of the workers and the most important samples to label, but assumes that workload can be adapted (i.e., more work can be requested from the best workers, while unreliable workers can be discarded). Instead, in our work we assume a fixed number of workers and a fixed amount of work, so that we will have a constraint on how much reliable labeling can be performed.

The main difference between prior work and our proposed method is that we consider single labeling instead of multiple labeling. In a labeling task, the basic decision is whether to label new samples that have never been labeled or relabel samples. A key assumption in our work is that we operate in a highly constrained environment where the total number of labels that we can obtain is much smaller than the size of the dataset. In this situation, we can only label a small part of our dataset, and therefore adding more samples is in general better than relabeling already labeled data points: As will be seen in our experiments, when the labeled set is very small reductions in prediction error for each new sample labeled

is high, while the benefits decrease as the labeled set size increases. Note that our interpolation algorithms use hard decisions on the labels for each sample as a starting point. Moreover, we tend to use "better" workers early in the labeling process. Thus, if the first label by a better worker is wrong, it would take two additional labels by less reliable workers to change the labeling decision. This suggests that obtaining multiple labels per sample may not be as efficient as labeling new samples.

Papers such as[13] and [14] consider the crowdsourcing problem with unreliable workers, but do not consider the problem of selecting which data should be labeled. Methods such as [10, 11, 12] consider sample selection, but do not take into account the relative importance of each sample when assigning workers.

Thus, a key contribution of our work is to use both sampling set selection and optimized worker assignment based on sample importance and worker reliability. We also challenge the idea that the only way to reduce the error in crowdsourcing systems is through adding redundancy by using multiple measurements per sample. Instead we improve single-label approaches by selecting the samples using active learning and taking into account the redundancy between samples with similar features. For active learning we use GSP methods [3], which give better results than other state-of-the-art methods. To the best of our knowledge, we are the only ones to consider single labeling along with optimized sample selection and worker assignment in the low-budget regime. Our method shows significant prediction error reduction independently of the active learning method used.

2. ASSIGNMENT USING THE COVARIANCE DISPERSION ALGORITHM

2.1. Notation and preliminaries

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a connected, undirected and weighted graph of size N, where $\mathcal{V}$ is the set of nodes and $\mathcal{E}$ is the set of edges. If two nodes have points close to each other in the Euclidean space of features, then the edge between these two nodes will have a larger positive weight. In the opposite case, the edge will have low or zero weight (no edge). A typical example of this would be the Gaussian kernel weights, which are defined as $w_{ij} = \exp(-||\mathbf{x}_i - \mathbf{x}_j||^2/2\sigma^2)$, where $\mathbf{x}_i$, $\mathbf{x}_j$ are the feature vectors of data points i and j. If the distance between $\mathbf{x}_i$ and $\mathbf{x}_j$ is small, i.e., the data points are close in feature space, then the corresponding edge weight w_{ij} will be high.

The adjacency matrix $\mathbf{W}$ is given by the weights of the edges between the nodes of the graph. That is, we have that $\mathbf{W}_{ij} = w_{ij}$ where w_{ij} is the weight of the edge that connects i and j. We also know that $\mathbf{W} = \mathbf{W}^\top$ because the graph is undirected and that $\mathbf{W}_{ii} = 0$ because there are no self-loops. The degree matrix $\mathbf{D}$ is a diagonal matrix where $\mathbf{D}_{ii} = \sum_{j=1}^{N} w_{ij}$. The expression of the Laplacian matrix is $\mathbf{L} = \mathbf{D} - \mathbf{W}$. The Laplacian matrix can be shown to be positive and semi-definite. Hence, it has real eigenvalues $0 = \lambda_1 < \lambda_2 \leq ... \leq \lambda_N$ with corresponding set of eigenvectors $\{\mathbf{u}_1, \mathbf{u}_2, ..., \mathbf{u}_N\}$ and can be written as $\mathbf{L} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^\top$, where $\mathbf{U} = (\mathbf{u}_1, \mathbf{u}_2, ..., \mathbf{u}_N)$ and $\mathbf{\Lambda} = diag(\lambda_1, \lambda_2, ..., \lambda_N)$.

The sets of eigenvalues and eigenvectors of the Laplacian matrix gives us a notion of frequency in the spectral graph domain. Eigenvectors associated with eigenvalues with low value have similar values at nodes that are strongly connected, and while variation is greater when nodes are not strongly connected. Thus, in GSP, low frequency is related with closeness of values in neighboring nodes and high frequency would be equivalent to having very different values in strongly connected nodes. Any graph signal can be projected onto the basis formed by the eigenvectors of the Laplacian and the resulting coefficients correspond to the spectral components of the signal. This projection operation is called Graph Fourier Transform or GFT. See [15] for a more detailed introduction.

2.2. Gaussian Random Field assumption

We will assume that the vector of class labels associated to the nodes is generated by thresholding a graph signal generated by a Gaussian Random Field (GRF). Thus, for a given graph, typical label signals are likely to be low frequency signals in the graph spectrum. This emulates the cluster-like behavior typical in machine learning problems and is a reasonable assumption leading to neighboring nodes (data points) being likely to have the same label.

Let S be the subset of nodes chosen by an active learning algorithm in order to be labeled by workers, and let $S^C = \mathcal{V} \backslash S$. Consider $\mathbf{f} = (f_1, f_2, ..., f_N)^\top$ to be a graph signal generated by a GRF. For the sake of simplicity, let us assume binary classification and the prior $E\{f_i\} = 0$. The probability of obtaining a given graph signal will then be proportional to:

$$p(\mathbf{f}) \propto e^{-\mathbf{f}^\top(\mathbf{L}+\delta\mathbf{I})\mathbf{f}} = e^{-\mathbf{f}^\top\mathbf{K}^{-1}\mathbf{f}}, \tag{1}$$

where $\mathbf{K}$ denotes the covariance matrix. In the general case $E\{f_i\} \neq 0$, we would have to make the substitution $f_i \longrightarrow (f_i - E\{f_i\})$ in (1). In multiclass classification, we would have a GRF generated signal for each of the classes.

As in any other multivariate Gaussian process, here the covariance matrix is given by $\mathbf{K} = (\mathbf{L} + \delta\mathbf{I})^{-1}$ where $\mathbf{I}$ is the identity matrix of size N x N and δ is a real positive scalar with an arbitrarily small value. It is easy to see that the covariance matrix $\mathbf{K}$ has the same eigenvectors as the Laplacian matrix $\mathbf{L}$ but their corresponding eigenvalues are $\sigma_i = 1/(\lambda_i + \delta)$. The $\delta\mathbf{I}$ term in the covariance matrix expression is added to avoid having a singular σ_1 eigenvalue ($\lambda_1 = 0$). Because $\mathbf{L}$ is a positive semi-definite, real and symmetric matrix, $\mathbf{K}$ and $\mathbf{K}^{-1}$ will be positive definite, real and symmetric matrices, and thus, by the spectral theorem they are diagonalizable. Then, the covariance matrix can be written as $\mathbf{K} = \mathbf{U}\mathbf{\Sigma}\mathbf{U}^\top$, where $\mathbf{\Sigma} = diag(\sigma_1, \sigma_2, ..., \sigma_N)$.

Assume we have selected a subset of nodes $S \subset \mathcal{V}$ to be labeled (different algorithms can be used for this, as will be shown in the experiments). By simple permutation we can write without loss of generality $\mathbf{f} = [\mathbf{f}_S^\top \ \mathbf{f}_{S^C}^\top]^\top$, where $\mathbf{f}_S$ is the graph signal corresponding to only the vertices in S. It is well known that, given $E\{f_i\} = 0$, the conditional distribution of $\mathbf{f}_{S^C}$ given $\mathbf{f}_S$ will be another multivariate Gaussian [16, 17] with mean $\boldsymbol{\mu}_{S^C|S}$ and covariance matrix $\mathbf{K}_{S^C|S}$:

$$\boldsymbol{\mu}_{S^C|S} = \mathbf{K}_{S^C S}(\mathbf{K}_S)^+\mathbf{f}_S \tag{2}$$

$$\mathbf{K}_{S^C|S} = \mathbf{K}_{S^C} - \mathbf{K}_{S^C S}(\mathbf{K}_S)^+\mathbf{K}_{SS^C} \tag{3}$$

where $\mathbf{K}_{S^C S}$ denotes the sub-matrix of K with rows indexed by S^C and columns indexed by S and $(\mathbf{K}_S)^+$ is the Moore-Penrose pseudoinverse of $\mathbf{K}_S$. For simplicity in the notation, $\mathbf{K}_S$ is equivalent to $\mathbf{K}_{SS}$. For the general case $E\{f_i\} \neq 0$, we just have to add a bias term to the mean $\boldsymbol{\mu}_{S^C|S}$ and to the graph signal $\mathbf{f}_S$.

This model gives us a probabilistic interpretation of the interpolated labels, which are the labels we will use for the samples that have not been labeled. The influence that the choice of S has on the predicted classes and the uncertainty of each prediction, which is given by the covariance matrix, is well defined by this model. Each prediction is given as a Gaussian distribution, where $sign(\boldsymbol{\mu}_{S^C|S})$ gives us the most probable combination of positive or negative class labels for S^C. The integral of the multivariate conditional density

function in one of the $2^{|S^C|}$ quadrants gives us the probability that the corresponding particular combination of class labels is correct.

2.3. Covariance dispersion algorithm

We now present the *covariance dispersion algorithm* (CDA), which we will use to assign one worker to each node of S efficiently, taking into account the quality of the workers. Although multiclass classification follows pretty closely the two class case, for the sake of simplicity in the notation we will show the CDA only for binary classification.

We face two problems because of the nature of labeling. First, the graph signal obtained from labeling the samples of S is not $\mathbf{f}_S$, but $\boldsymbol{l}_S = \{-1, 1\}$. Second, the predicted graph signal $\mathbf{f}_{S^C}$ has to be thresholded. That is, we have to make a hard decision on the class of each sample of S^C from the graph signal obtained: $\boldsymbol{l}_{S^C} = \text{sign}(\mathbf{f}_{S^C})$. The prediction error will be assessed based on $\boldsymbol{l}_{S^C}$, rather than $\mathbf{f}_{S^C}$.

We will assume that we are still able to use the GRF model although we will estimate the prediction error from thresholded labels. This simplification of using a thresholded signal as if it were the GRF-generated non-thresholded signal will introduce some error that we cannot eliminate.

Using the GRF model, in case of no labeling errors, f_{S^C} can be modeled as a normal distribution whose mean $\boldsymbol{\mu}_{S^C|S}$ is obtained from substituting $\mathbf{f}_S$ by $\boldsymbol{l}_S$ in (2), with its covariance matrix given by (3). In this case, $\boldsymbol{\mu}_{S^C|S}$ would be the MAP estimation of $\mathbf{f}_{S^C}$ given $\mathbf{f}_S$. That is, it is the best estimation possible.

Now consider errors in the labeling. Let $\tilde{\boldsymbol{l}}_S = \boldsymbol{l}_S - 2\boldsymbol{\phi}_S \cdot \boldsymbol{l}_S$, be the noisy labels. $\boldsymbol{\phi}_S$ is a column vector in which each of its components is a random variable that has Bernoulli distribution valued 1 with probability ε_i and 0 with probability $(1-\varepsilon_i)$ with $i \subset S$, where ε_i is the probability that the worker assigned to sample i makes a mistake when labeling. Adding errors shifts the mean of the conditional normal distribution. The shifted mean will be

$$\tilde{\boldsymbol{\mu}}_{S^C|S} = \mathbf{K}_{S^C S}(\mathbf{K}_S)^+ \boldsymbol{l}_S \cdot (\mathbf{1}_S - 2\boldsymbol{\phi}_S).$$

Our objective will be to choose which samples to allocate to the less reliable workers so that the resulting shifted distribution is as close as possible to the best estimation. We will measure this closeness by calculating the Kullback-Leibler (KL) divergence between the shifted and original multivariate normal distributions:

$$D_{KL} = \frac{1}{2}(\tilde{\boldsymbol{\mu}}_{S^C|S} - \boldsymbol{\mu}_{S^C|S})^\top \mathbf{K}^{-1}_{S^C|S}(\tilde{\boldsymbol{\mu}}_{S^C|S} - \boldsymbol{\mu}_{S^C|S}) \quad (4)$$

The shift between the means is given by:

$$\begin{aligned}\tilde{\boldsymbol{\mu}}_{S^C|S} - \boldsymbol{\mu}_{S^C|S} &= \mathbf{K}_{S^C S}(\mathbf{K}_S)^+ \boldsymbol{l}_S \cdot ((\mathbf{1}_S - 2\boldsymbol{\phi}_S) - \mathbf{1}_S) \\ &= -2\mathbf{K}_{S^C S}(\mathbf{K}_S)^+ \boldsymbol{\phi}_S \cdot \boldsymbol{l}_S\end{aligned} \quad (5)$$

Applying (5) in (4) and using the fact that $(\boldsymbol{l}_S \cdot \boldsymbol{\phi}_S)^2 = \boldsymbol{\phi}_S^2$, we can make the approximation of substituting $\boldsymbol{\phi}_S$ by $\boldsymbol{\varepsilon}_S$ to obtain the averaged KL divergence:

$$\bar{D}_{KL} \approx 2\boldsymbol{\varepsilon}_S^\top (\mathbf{K}_S)^+ \mathbf{K}_{SS^C} \mathbf{K}^{-1}_{S^C|S} \mathbf{K}_{S^C S} (\mathbf{K}_S)^+ \boldsymbol{\varepsilon}_S \quad (6)$$

$$\nabla \bar{D}_{KL} \approx 4\boldsymbol{\varepsilon}_S^\top (\mathbf{K}_S)^+ \mathbf{K}_{SS^C} \mathbf{K}^{-1}_{S^C|S} \mathbf{K}_{S^C S} (\mathbf{K}_S)^+ \quad (7)$$

To minimize the average KL divergence value (6), we use an iterative process where in each step we compute its gradient (7) and randomly swap two workers so that we reduce the error in the sample with bigger gradient.

To sum up, the CDA (see Algorithm 1) allows us to reduce the KL divergence between the distribution of the graph signal when there are no labeling errors and our noisy graph signal. We will see in next section that this reduces significantly the prediction error.

Algorithm 1 Covariance dispersion algorithm

Input: Adjacency matrix $\mathbf{W}$, optimal subset of nodes S, worker probability errors $\boldsymbol{\varepsilon}^{(0)}$
Output: Optimal assignment of workers $\boldsymbol{\varepsilon}^{(numIter)}$

```
K ← createCovarianceMatrix(W)
K_c ← createCondCovMatrix(K, S)
Q ← createQMatrix(K, S)
[U, Λ] ← getEigenvectors(K_c)
for i = 1 : numIter do
    ε^(i) ← swapWorkers(U, Λ, Q, ε^(i−1))
end for
```

Time Complexity: $O(|S^C|^3 + |S|^2 numIter)$

3. EXPERIMENTAL RESULTS

To test our previous method we have designed software using Matlab R2014a. The code is available on GitHub[1]. The datasets used for the following experiments are the USPS handwritten digits dataset[2] and the Isolet dataset[3]. Their graphs were constructed using Gaussian kernel weights, with σ and K fixed accordingly.

To select the subset that best represents the graph (S), given its size, we have used two different methods: the selection scheme described in [3], which maximizes the cut-off graph frequency of the graph signals generated by the subset of nodes chosen, and a simpler algorithm based on k-medoids [18] that minimizes the mean graph distance to any node in S.

To predict the graph signal of the subset S^C we interpolate the graph signal that has the values of the worker labels in S and zeros in the remaining nodes using an iterative algorithm developed in [19] based on projection onto convex sets (POCS).

3.1. Assignment problem

We simulate and compare the prediction error of the CDA with assigning the workers at random and with assigning the best workers using the sample selection order, for different sized subsets S_n (see Figure 1). We cannot compare to other baselines because no works consider optimizing assignment order and we do. The objective is to obtain insights about the labeling process while also proving that worker assignment matters.

We have chosen a simple way of modeling the worker behavior. Any worker i has associated a probability labeling error ε_i. When that worker labels any given sample, the probability of giving the correct class is $1-\varepsilon_i$, and in case of error all the wrong classes are equally probable. Without loss of generality we create as many workers as samples, and assign each worker to one sample. The parameter ε_i for each worker is generated using Beta(2,8) probability density function, but we obtain similar conclusions for different distributions. We now proceed to explain the different insights given by the simulations.

First, labeling more samples yields little improvement when we already have between 8 to 10% of the samples labeled in each dataset. Of course the point where this effect occurs will depend on the problem we consider. The reason is that when the signal is bandlimited with very low cut-off frequency, we only need a small subset of samples carefully chosen to recover most of the signal energy as shown in [3], and thus, adding more samples after that

[1] https://github.com/javier-maroto/GSP-Noisy-Crowdsourcing

[2] http://www.cs.nyu.edu/~roweis/data.html

[3] http://archive.ics.uci.edu/ml/datasets/ISOLET

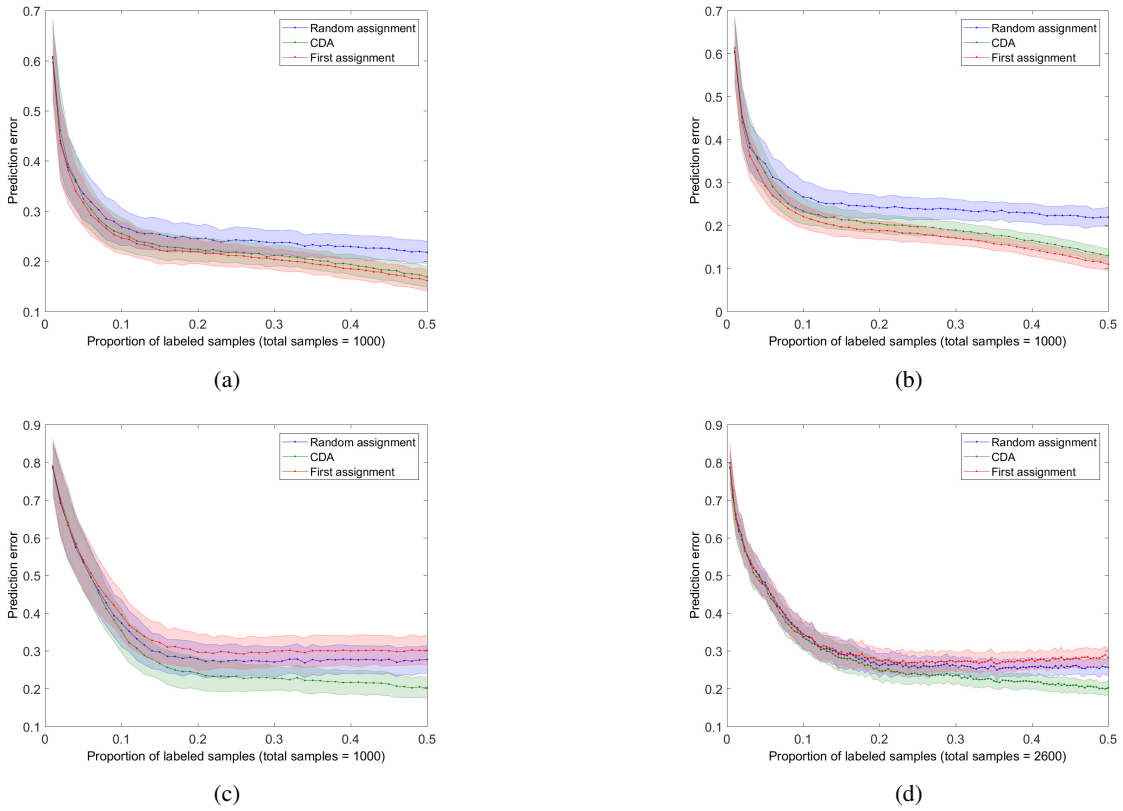

Fig. 1: Comparison between the prediction error obtained after applying our CDA algorithm, the selection order heuristic and random assignment when we vary the size of the sampling set. We use the USPS dataset for Figures 1a,1b,1c and the Isolet dataset for Figure 1d. Additionally, we use a Beta(2,8) worker error distribution in Figures 1a,1c,1d and a heterogeneous distribution of 40% of spammers (p = 0.5) and 60% of hammers (p = 0) in Figure 1b. In Figures 1a,1b we use the GSP-based active learning algorithm and in Figures 1c,1d, the k-medoids active learning algorithm.

point has almost no effect in our prediction. This supports the idea that using sampling reduces greatly the error, because most of the process information is contained in the small subset that has been carefully selected. Additionally, when we can only manually single label 3% of the dataset due to budget limitations, it is evident from the results that relabeling increments greatly the prediction error (because we have less samples labeled), which supports single labeling against multi labeling for limited budgets.

Second, changing the error distribution of the group of workers (Figures 1a, 1b) has almost no impact when we are assigning them randomly. However, the significance of assignment algorithms like CDA or the selection order heuristic is greater in heterogeneous groups. The improvement margin is dependent on the variance of the worker error distribution.

Third, it is better to use CDA than random assignment for any size of S, which proves that the minimization of the KL divergence affects the prediction error. This reduction on prediction error also proves that there is some gain from the redundancy between samples, characterized in part by the covariance matrix, which we use as the starting point in our work. We have to take into account that the CDA is a greedy algorithm that converges to a local minimum, so there could be even better assignments.

Fourth, the selection order heuristic is similar or slightly better to our proposed method when using the active learning algorithm described in [3] (Figure 1a). When using another selection approach like the k-medoids method, where the samples are given in order of importance too, our algorithm greatly outperforms the selection order heuristic (Figure1c). This shows that not in all active learning algorithms the first samples selected are always the most important. In case of using random sampling or other methods that give the selected samples in a batch, the selection heuristic cannot be applied so our algorithm is the only option.

Comparing the cost of using the complex selection algorithm with the selection order heuristic ($\sim O(|S^C|^2|S|)$) with using a faster selection method with using our proposed assignment method (inverting a matrix, $\sim O(|S^C|^3)$), we see that in the case we do not have S beforehand it is better to use the selection algorithm in [3] with the selection order heuristic, while in any other case our method is much more robust.

4. CONCLUSION

We show with this work that efficient assignment improves significantly the prediction error. We show that our proposed method is robust to different selection algorithms and gives significant improvement with respect to using nothing, which is the common approach. When using the selection algorithm given by [3], assigning in order gives significant improvement.

5. REFERENCES

[1] Simon Tong and Daphne Koller, "Support vector machine active learning with applications to text classification," *Journal of machine learning research*, vol. 2, no. Nov, pp. 45–66, 2001.

[2] Yi Yang, Zhigang Ma, Feiping Nie, Xiaojun Chang, and Alexander G Hauptmann, "Multi-class active learning by uncertainty sampling with diversity maximization," *International Journal of Computer Vision*, vol. 113, no. 2, pp. 113–127, 2015.

[3] Akshay Gadde, Aamir Anis, and Antonio Ortega, "Active semi-supervised learning using sampling theory for graph signals," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2014, pp. 492–501.

[4] Faiza Khan Khattak and Ansaf Salleb-Aouissi, "Quality control of crowd labeling through expert evaluation," in *Proceedings of the NIPS 2nd Workshop on Computational Social Science and the Wisdom of Crowds*, 2011, vol. 2.

[5] Justin Cheng, Jaime Teevan, and Michael S Bernstein, "Measuring crowdsourcing effort with error-time curves," in *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. ACM, 2015, pp. 1365–1374.

[6] Gabriele Paolacci, Jesse Chandler, and Panagiotis G Ipeirotis, "Running experiments on amazon mechanical turk," 2010.

[7] Michael Buhrmester, Tracy Kwang, and Samuel D Gosling, "Amazon's mechanical turk a new source of inexpensive, yet high-quality, data?," *Perspectives on psychological science*, vol. 6, no. 1, pp. 3–5, 2011.

[8] JCF De Winter, Miltos Kyriakidis, Dimitra Dodou, and Riender Happee, "Using crowdflower to study the relationship between self-reported violations and traffic accidents," *Procedia Manufacturing*, vol. 3, pp. 2518–2525, 2015.

[9] Panagiotis G Ipeirotis, Foster Provost, and Jing Wang, "Quality management on amazon mechanical turk," in *Proceedings of the ACM SIGKDD workshop on human computation*. ACM, 2010, pp. 64–67.

[10] Liyue Zhao, Gita Sukthankar, and Rahul Sukthankar, "Incremental relabeling for active learning with noisy crowdsourced annotations," in *Privacy, Security, Risk and Trust (PASSAT) and 2011 IEEE Third Inernational Conference on Social Computing (SocialCom), 2011 IEEE Third International Conference on*. IEEE, 2011, pp. 728–733.

[11] Pinar Donmez and Jaime G Carbonell, "Proactive learning: cost-sensitive active learning with multiple imperfect oracles," in *Proceedings of the 17th ACM conference on Information and knowledge management*. ACM, 2008, pp. 619–628.

[12] Pinar Donmez, Jaime G Carbonell, and Jeff Schneider, "Efficiently learning the accuracy of labeling sources for selective sampling," in *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2009, pp. 259–268.

[13] Victor S Sheng, Foster Provost, and Panagiotis G Ipeirotis, "Get another label? improving data quality and data mining using multiple, noisy labelers," in *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2008, pp. 614–622.

[14] David R Karger, Sewoong Oh, and Devavrat Shah, "Iterative learning for reliable crowdsourcing systems," in *Advances in neural information processing systems*, 2011, pp. 1953–1961.

[15] David I Shuman, Sunil K Narang, Pascal Frossard, Antonio Ortega, and Pierre Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," *IEEE Signal Processing Magazine*, vol. 30, no. 3, pp. 83–98, 2013.

[16] Holger Rootzén, "Some probability and statistics," `http://www.math.chalmers.se/~rootzen/highdimensional/SSP4SE-appA.pdf`, 2013, [Online; accessed 27-November-2016].

[17] Georg Lindgren, Holger Rootzén, and Maria Sandsten, *Stationary stochastic processes for scientists and engineers*, CRC press, 2013.

[18] Leonard Kaufman and Peter J Rousseeuw, "Partitioning around medoids (program pam)," *Finding groups in data: an introduction to cluster analysis*, pp. 68–125, 1990.

[19] Sunil K Narang, Akshay Gadde, Eduard Sanou, and Antonio Ortega, "Localized iterative methods for interpolation in graph structured data," in *Global Conference on Signal and Information Processing (GlobalSIP), 2013 IEEE*. IEEE, 2013, pp. 491–494.