Learning Medical Diagnosis Models from Multiple Experts

Hamed Valizadegan, Quang Nguyen, Milos Hauskrecht[1]
Department of Computer Science, University of Pittsburgh,
email: ***hamed, quang, milos@cs.pitt.edu***

Abstract

Building classification models from clinical data often requires labeling examples by human experts. However, it is difficult to obtain a perfect set of labels everyone agrees on because medical data are typically very complicated and it is quite common that different experts have different opinions on the same patient data. A solution that has been recently explored by the research community is learning from multiple experts/annotators. The objective of learning from multiple experts is to model different characteristics of the human experts and combine them to obtain a consensus model. In this work, we study and develop a new probabilistic approach for learning classification models from labels provided by multiple experts. Our method explicitly models and incorporates three characteristics of annotators into the learning process: their specific prediction model, consistency and bias. We show that in addition to building a superior classification model, our method also helps to model behavior of annotators. We applied the proposed method to learn different characteristics of Physicians labeling clinical records for Heparin Induced Thrombocytopenia (HIT) and combine them in order to obtain a final classifier.

1 Introduction

The vast amount of clinical data collected everyday from Electronic Health Records (EHR) gives us a unique opportunity to study different aspects of such data and obtain better insights into different diseases, their treatments, and their dynamics. The knowledge obtained by such studies may lead to health care cost reduction, health care quality improvement, better understanding of the disease processes, and drug-development integration with genetic studies.

The EHR data collected in the medical centers are not complete and ready to use, and require additional human labeling and annotations, a task which is not only very difficult and time-consuming but also subjective. Since medical data are typically very complicated, it is quite common that different experts have different opinions on the same patient data and it is difficult to obtain a perfect set of labels everyone agrees on. As an example, consider diagnosing a patient to find out if he has a disease or not given available observations. Different physicians may differ in their opinions and label assessments given their level of expertise and knowledge.

The question we aim to answer in this paper is how to learn from the labeled data when the labels are provided by different experts. We would like to model different sources of disagreement in labeling and use them for the construction of better diagnosing medical models. Our assumption is that there is a true (consensus) model behind different medical conditions from which the models of different experts are generated. To find such a consensus model, we need to grasp the causes for the labeling disagreement of different annotators. The labeling disagreement may be rooted in

- **Differences in the risks annotators associate with each class label assignment:** diagnosing a patient as not having a disease when the patient has disease, carries a cost due to, for example, a missed opportunity to treat the patient, or longer patient discomfort and suffering. A similar, but different cost is caused by incorrectly diagnosing a patient. The differences in the (annotator-specific) utilities (or costs) may easily explain differences in their label assessments. Hence our goal is to develop a learning framework that seeks a model consensus that at the same time permits annotators who have different utility biases.
- **Differences in the knowledge (or model) experts use to label examples:** while diagnoses provided by different experts may be often consistent, the knowledge they have and features they consider when making the disease decision may differ, potentially leading to differences in labeling. It is not rare when two expert physicians disagree on a complex patient case due to differences firmly embedded in their knowledge and understanding of

[1]corresponding author, email: milos@cs.pitt.edu

the disease. These differences are best characterized as differences in their knowledge or model they used to diagnose the patient.

- **Differences in the amount of effort annotators spend for labeling each case:** different experts may spend different amount of time and care to diagnose the same case. This leads to labeling inconsistency within the reviewer's own model.

Our objective is to develop a learning framework that would embrace and accept these types of differences and find a consensus model. We aim to provide information about different reviewers and the quality of labels they provide. This can be used for active learning where we ask more reliable annotators to provide the label of examples in the later stage of labeling process or it could be used to provide feedback to human labelers, to assign credit or compensation for labeling tasks, etc.

In this paper, we develop a new multiple-expert learning approach that takes into account the reviewers' reliability as well as differences in the expert-specific models and biases when learning a consensus model. We study our framework on real-medical data representing experts assessment of the risk of the Heparin Induced Thrombocytopenia (HIT)[1] given a set of patients' observations and labs.

2 Problem description

Before describing the paper, let us introduce the general notational style used in this paper: we denote matrices by capital letters, vectors by boldface lowercase letters and scalars by lowercase letters.

In the standard supervised binary classification framework the training data set $D = \{(x_i, y_i)\}_{i=1}^{n}$ consists of n data examples, where $x_i \in R^d$ is a d-dimensional feature vector and $y_i \in \{-1, 1\}$ is a corresponding binary class label. The objective is to learn a function: $f : R^d \rightarrow \{-1, 1\}$ that generalizes well to unseen (future) data.

In the supervised binary classification with multiple experts, we have m different reviewers who assign labels to examples for the binary classification problem. Let $D^k = \{(\mathbf{x}_i^k, y_i^k)\}_{i=1}^{n_k}$ denotes data specific for the reviewer k, such that $\mathbf{x}_i^k \in R^d$ is a d-dimensional input example and y_i^k is it binary label ($y_i^k \in \{-1, 1\}$) assigned by reviewer k. In general, input examples can be labeled by one, or multiple reviewers, and hence the examples they label may or may not overlap.

Given the data from multiple reviewers, our main objective is to learn a function $f : R^d \rightarrow \{-1, 1\}$ that represents a good consensus model (possibly the true model). This is a difficult problem because (1) the true labels and the true model are unknown, (2) the reviewers' knowledge and reliability could vary widely and are unknown, and (3) Each reviewer can have different preferences (or utilities) for different labels, leading to different biases towards negative or positive class. Therefore, even if two reviewers have the same relative understanding of a patient case their assigned labels may be different. Under these conditions, we aim to combine the subjective labels from different reviewers to learn a good consensus model.

3 Existing Approaches

Majority Voting

Majority voting is the most commonly used approach. For each example $i \in \{1 \dots L\}$ from the set of L distinct examples, the "true" label t_i is estimated by voting: $t_i = 1$ if $\sum_{k=1}^{m} y_i^k \geq 0$, otherwise $t_i = -1$. The majority vote then defines a dataset D = $\{(x_i, t_i)\}_{i=1}^{L}$ that can be used to train a model representing the reviewer's consensus. The majority voting comes with two drawbacks. First, it assumes that all reviewers are equally reliable when labeling examples. However, when one reviewer is very reliable, and the other ones are unreliable, the majority vote may sway the final labels and assign incorrect labels to some examples. Second, reviewers may differ in their label due to differences in their preferences leading to split or close votes. These close votes are ignored when learning the majority-based consensus model.

Other Related Works

There has been a number of works on learning with multiple reviewers/annotators.[2] showed the benefits of obtaining labels from multiple annotators and the need of an efficient learning algorithm.[3] proposed a learning framework, where biases and skills of annotators were considered and modeled using a confusion matrix. This work was later generalized and extended by[4] and[5] by modeling difficulty of examples.[6] used an expectation-maximization (EM) algorithm to iteratively learn the reliability of annotators and the first stage was initialized by majority voting.

The current state-of-the-art methods are[6] and[5]. The first work shows superior performance over majority voting when the number of annotators is large (> 40). This is practical when the labeling task is easy so the crowd-sourcing services like Amazon Mechanical Turk can be utilized. However, it is not practical in domains in which the annotation is time-consuming and requires the work of experts who are scarce resource and whose annotation effort can be extremely costly. Therefore it is infeasible to hire a large number of reviewers/annotators. An example is disease modeling and labeling of examples by physicians. The second work[5] is optimized for a slightly different setting than ours. The algorithm learns reviewer specific models however it does not attempt to learn a consensus model one can use to label future examples.

4 The proposed approach

We aim to combine annotations of all reviewers and build a unified consensus classification model that can be applied to future data. Our assumption is that (1) there is a true underlying model that helps us to label examples (past and future) relative to each other (2) this model let us generate label predictions consistent with reviewer-specific preferences, and (3) all deviations from the reviewer-specific models are adequately modeled by reviewer-specific reliability parameters.

Model

To illustrate our approach, let us assume a linear model $\mathbf{u}$ representing the consensus[2]. In this case, $\mathbf{u}$ is a vector of parameters such that $sign(\mathbf{u}^\intercal \mathbf{x} + b)$ defines the true labeling of example $\mathbf{x}$. Our assumption is that reviewer k has its own model parameters $\mathbf{w}_k$ and parameter bias b_k for making calls on positive and negative labels. Under this model, the reviewer k would assign a positive label to example $\mathbf{x}$ if $\mathbf{w}_k^\intercal \mathbf{x} + b_k \geq 0$, otherwise the label is negative. In addition, we assume each reviewer is characterized by two reliability parameters: (1) the self-consistency parameter α_k that models how reliable the labeling of reviewer k is; it is the amount of consistency of reviewer k within its own model $\mathbf{w}_k$, (2) the consensus-consistency parameter β_k that models how consistent the model of reviewer k is with respect to the underlying consensus model $\mathbf{u}$. This parameter conceptually models the differences in the knowledge or expertise of the reviewers. Notice that the consensus model $\mathbf{u}$ is obtained from the reviewer specific model $\mathbf{w}_k$s. Thus, the intuitive meaning of this parameter is how consistent the labeling of each reviewer is with the labelings of other reviewers.

To simplify the notation in the rest of the paper, we include the bias term in the weights vector, and extend the input vector $\mathbf{x}$ with constant 1.

SVMCrowd

We assume that there is a consensus model $\mathbf{u}$ from which all the reviewer specific models are generated. We consider a Gaussian distribution with mean zero and precision η for this consensus model $\mathbf{u}$:

$$p(\mathbf{u}|\mathbf{0}_d, \beta_k) \quad = \quad \mathcal{N}(\mathbf{u}|\mathbf{0}_d, \eta^{-1}\mathbf{I}_d) \tag{1}$$

where $\mathbf{I}_d$ is a vector of size d with all elements equal to 1, and $\mathbf{0}_d$ is a vector of size d with all elements equal to 0. The reviewer k has its own specific model $\mathbf{w}_k$. Since these reviewer specific models are generated based on the consensus model $\mathbf{u}$, the average of such specific models are $\mathbf{u}$. We assume a Gaussian distribution for $\mathbf{u}$ with specific reviewer precision β_k; i.e.

$$p(\mathbf{w}_k|\mathbf{u}, \beta_k) \quad = \quad \mathcal{N}(\mathbf{w}_k|\mathbf{u}, \beta_k^{-1}\mathbf{I}_d) \tag{2}$$

[2]Notice we are using the term 'consensus model' instead of 'true model' here because in medical, talking about a true labeling model is not informative.

The precision β_k for reviewer k determines the expertise of a reviewer. The model $\mathbf{w}_k$ for an unreliable reviewer tends to be very different from the consensus model $\mathbf{u}$ while that for a competitive reviewer will be very similar to the consensus model $\mathbf{u}$. The precision of the reviewer k, i.e. β_k is based on several factors such as the difficulty of problem and the expertise of reviewers. We consider similar Gamma distributions with shape parameter θ_β and inverse scale parameter τ_β as the conjugate prior for the precision β_k of each reviewer.

$$p(\beta_k|\theta_\beta, \tau_\beta) = \mathcal{G}(\beta_k|\theta_\beta, \tau_\beta) \tag{3}$$

Given the assumption that reviewer k utilizes the linear model $\mathbf{w}_k$ subject to a Gaussian noise with mean zero and precision (inverse variance) α_k and that α_k has a Gamma prior with shape parameter θ_α and inverse scale parameter τ_α, we have the following posterior for the example $\mathbf{x}_i^k$ labeled y_i^k by reviewer k:

$$p(y_i^k|\mathbf{x}_i^k, \mathbf{w}_k, \alpha_k) = \mathcal{N}(y_i^k|\mathbf{w}_k^\intercal \mathbf{x}_i^k, \alpha_k^{-1}) \tag{4}$$

$$p(\alpha_k|\theta_\alpha, \tau_\alpha) = \mathcal{G}(\alpha_k|\theta_\alpha, \tau_\alpha) \tag{5}$$

Now, we can write the posterior probability for matrix $W = [\mathbf{w}_1; \ldots; \mathbf{w}_m]$ as:

$$p(W|X, \mathbf{y}, \alpha, \mathbf{u}, \beta, \eta) \propto \mathcal{N}(\mathbf{u}|\mathbf{0}, \eta^{-1}\mathbf{I}_d) \times \mathcal{G}(\beta_k|\theta_\beta, \tau_\beta)\mathcal{G}(\alpha_k|\theta_\alpha, \tau_\alpha) \times \prod_{k=1}^{m} \left(\mathcal{N}(\mathbf{w}_k|\mathbf{u}, \beta_k^{-1}\mathbf{I}_d) \prod_{i=1}^{n_k} \mathcal{N}(y_i^k|\mathbf{w}_k^\intercal \mathbf{x}_i^k, \alpha_k^{-1}) \right)$$

where $X = [\mathbf{x}_1^1; \ldots; \mathbf{x}_{n_1}^1; \ldots; \mathbf{x}_m^1; \ldots; \mathbf{x}_{n_m}^m]$ is the matrix of all the examples labeled by all the reviewers, and $\mathbf{y} = [y_1^1; \ldots; y_{n_1}^1; \ldots; y_m^1; \ldots; y_{n_m}^m]$ is the label vector for examples in X given by their reviewers. Taking the negative logarithm of this posterior, we have:

$$\begin{aligned} &\frac{\eta}{2}\|\mathbf{u}\|^2 + \frac{1}{2}\sum_{k=1}^{m} \beta_k \left(\|\mathbf{w}_k - \mathbf{u}\|^2\right) + \tau_\beta\big) \\ &- (\theta_\beta - \frac{1}{2})\sum_{k=1}^{m} \ln(\beta_k) - \frac{1}{2}\sum_{k=1}^{m} (2(\theta_\alpha - 1) + n_k)\ln(\alpha_k) \\ &+ \frac{1}{2}\sum_{k=1}^{m} \alpha_k(2\tau_\alpha + \sum_{i=1}^{n_k} \|y_i^k - \mathbf{w}_k^\intercal \mathbf{x}_i^k\|^2) \end{aligned} \tag{6}$$

Equation 6 results the maximum a posterior (MAP) estimation of W, $\boldsymbol{\beta}$, $\boldsymbol{\alpha}$, and $\mathbf{u}$. Although we can solve the objective function in Equation 6 directly, we replace the squared error function in Equation 6 with the Hinge loss function to obtain sparse kernel solution[7]:

$$\begin{aligned} &\frac{\eta}{2}\|\mathbf{u}\|^2 + \frac{1}{2}\sum_{k=1}^{m} \beta_k \left(\|\mathbf{w}_k - \mathbf{u}\|^2\right) + \tau_\beta\big) \\ &- \frac{\delta_\beta}{2}\sum_{k=1}^{m} \ln(\beta_k) - \frac{1}{2}\sum_{k=1}^{m} (\delta_\alpha + n_k)\ln(\alpha_k) \\ &+ \frac{1}{2}\sum_{k=1}^{m} \alpha_k(2\tau_\alpha + \sum_{i=1}^{n_k} \max\left(0, 1 - y_i^k \mathbf{w}_k^\intercal \mathbf{x}_i^k\right)) \end{aligned} \tag{7}$$

where we used $\delta_\beta = 2\theta_\beta - 1$ and $\delta_\alpha = 2(\theta_\alpha - 1)$. We need to minimize the above objective function with regards to the consensus model $\mathbf{u}$, the reviewer specific model $\mathbf{w}_k$, , and reviewer specific reliability parameters α_k and β_k. Note that α_k measures the consistency of labels provided by reviewer k with its own model $\mathbf{w}_k$ and α_k measures the

consistency of the labeled provided by reviewer k with the consensus model $\mathbf{u}$. A reviewer with large values of α_k and β_k is considered as a good reviewer.

Optimization

Here we provide a brief description on how to optimize the objective function in Equation 7 with regards to the consensus model $\mathbf{u}$, the reviewer specific model $\mathbf{w}_k$, , and reviewer specific reliability parameters α_k and β_k.

Similar to SVM, the term related to the the hinge loss can be written as the constraints of the optimization which results the following equivalent form:

$$\min_{\mathbf{w},\boldsymbol{\epsilon},\boldsymbol{\alpha},\boldsymbol{\beta}} \quad \frac{\eta}{2}\|\mathbf{u}\|^2 + \frac{1}{2}\sum_{k=1}^{m}\beta_k\left(\|\mathbf{w}_k - \mathbf{u}\|^2 + 2\tau_\beta\right) \tag{8}$$

$$-\frac{\delta_\beta}{2}\sum_{k=1}^{m}\ln(\beta_k) - \frac{1}{2}\sum_{k=1}^{m}(\delta_\alpha + n_k)\ln(\alpha_k)$$

$$+\frac{1}{2}\sum_{k=1}^{m}\alpha_k(2\tau_\alpha + \sum_{i=1}^{n_k}\epsilon_i^k) \tag{9}$$

$$\text{s.t.} \quad y_i^k\mathbf{w}_k^\intercal\mathbf{x}_i^k \geq 1 - \epsilon_i^k,\ k = 1\ldots m,\ i = 1\ldots n_k$$

$$\epsilon_i^k \geq 0, \qquad k = 1\ldots m,\ i = 1\ldots n_k$$

In order to optimize the above objective function, we use alternating optimization: we initialize the reliability parameters $\alpha_k = 1$ and $\beta_k = 1$ and then iterate on performing the following two steps:

- **Learning $\mathbf{u}$ and $\mathbf{w}_k$:** In order to learn the consensus model $\mathbf{u}$ and reviewer specific model $\mathbf{w}_k$, we consider the reliability parameters α_k and β_k as constants. This will lead to a SVM form optimization to obtain $\mathbf{u}$ and $\mathbf{w}_k$.
- **Learning α_k and β_k:** By fixing $\mathbf{u}$, $\mathbf{w}_k$ for all reviewers, and $\boldsymbol{\epsilon}$, we can minimize the objective function in 8 by computing the derivative w.r.t. $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$. This results the following closed form solutions for α_k and β_k:

$$\alpha_k = \frac{\delta_\alpha + n_k}{2\tau_\alpha + \sum_i \epsilon_i^k}, \tag{10}$$

$$\beta_k = \frac{\delta_\beta}{2\tau_\beta + \|\mathbf{w}_k - \mathbf{u}\|^2}, \tag{11}$$

Notice that ϵ_i^k is the amount of violation of label constrain for example $\mathbf{x}_i^k$, i.e. the i^{th} example labeled by reviewer k so $\sum_{i=1}\epsilon_i^k$ is the summation of all labeling violations for model of reviewer k. This implies that α_k is inversely proportional to the amount of misclassification of examples by reviewer k according to its specific model $\mathbf{w}_k$. β_k is inversely related to the difference of the model of reviewer k, i.e. $\mathbf{w}_k$, with the consensus model $\mathbf{u}$. Thus it is the consistency of the model learned for reviewer k from the consensus model $\mathbf{u}$. τ_α, δ_α, τ_β, and δ_β prevents α and β rely too much on the data; as we saw earlier in this section they are the Bayesian prior on $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$.

5 Experiments

We test the performance of our methods on clinical data obtained from EHRs for post-surgical cardiac patients and the problem of monitoring and detection of the Heparin Induced Thrombocytopenia (HIT) [1,9]. HIT is an adverse immune reaction that may develop if the patient is treated for a longer time with heparin, the most common anticoagulation treatment. If the condition is not detected and treated promptly it may lead to further complications (such as thrombosis) and even to patient's death. An important clinical problem is the monitoring and detection of patients who are at risk of developing the condition. Alerting when this condition becomes likely prevents the aggravation of the condition and appropriate countermeasures (discontinuation of the heparin treatment or switch to an alternative anticoagulation

treatment) may be taken. In this work, we investigate the possibility of building a detector from patient data and human expert assessment of patient cases with respect to HIT and the need to raise the HIT alert. This corresponds to the problem of learning a classification model from data where expert's alert or no-alert assessments define class labels.

Data Collection

We extracted data from electronic health records that consisted of over $50,000$ patient-state instances. Out of these we have selected 377 instances using a special stratified sampling approach where individual strata were defined to increase or decrease the chance the patient-state instance is associated with Heparin Induced Thrombocytopenia (HIT). We asked three experts to provide us with a label showing if the patient is at the risk of HIT. We further asked a senior physician who was known to be very reliable in labeling HIT data. We utilized the labels provided by this senior expert as the true labels and evaluated different approaches.

Data Instances

Since the data in medical records are high dimensional, we have selected 50 features derived from the patient health record and clinical variables important for the detection of HIT for this study. These features represent time series of labs, medications and procedures as follows:

- From labs we used Platelet counts, Hemoglobin levels and White Blood Cell Counts and their time series. The features generated for labs in the experiment included: last values observed, time elapsed since the last value was observed, quantitative value trends, apex and nadir values and differences of last values from the nadir and apex values.
- From medications we used Heparin and its administration record. The medication related features generated for all patient instances reflect whether the patient is currently on the heparin or not, the time elapsed since the medication was started and the time since last change in its administration.
- The procedure features included in the data were the indicator of a major heart procedure and the time elapsed since such a procedure.

All these features were used to define the patient case. The alert decision by the expert was used as a class label.

Experimental Set-up We compare the following algorithms:

- **SVMCrowd:** This is the proposed method in this paper. We set the parameters $\eta = \tau_\alpha = \tau_\beta = 1$, $\theta_\alpha = 1$ ($\delta_\alpha = 0$), and $\theta_\beta = 1$ ($\delta_\beta = 1$).
- **Majority:** This is the majority voting algorithm that we described in Section 3. For the majority voting approach, we first obtain the majority label t_i for each example $\mathbf{x}_i$ using the procedure introduced in Section 3 and then use the standard SVM to construct the majority-based consensus model. We also set the trade-off parameter $C = 1$ in SVM (default value in LIBSVM[11]).
- **Raykar:** This is the algorithm developed by Raykar et. al.[6]. We used the same setting as discussed in[6].

The HIT data was randomly divided into $2/3$ training and $1/3$ test set. We trained models on the training set and evaluated performance of the different methods by calculating the area under the ROC curve on the test set. We repeated all experiments 100 times and reported the average and 95% confidence interval.

We investigate three aspects of the proposed method: 1) the performance of the consents model on the test data when evaluated by the labels of examples provided by the senior expert. 2) the performance of the reviewer specific model $\mathbf{w}_k$ for reviewer k when evaluated by the examples labeled by that reviewer 3) the accuracy of the reliability parameters.

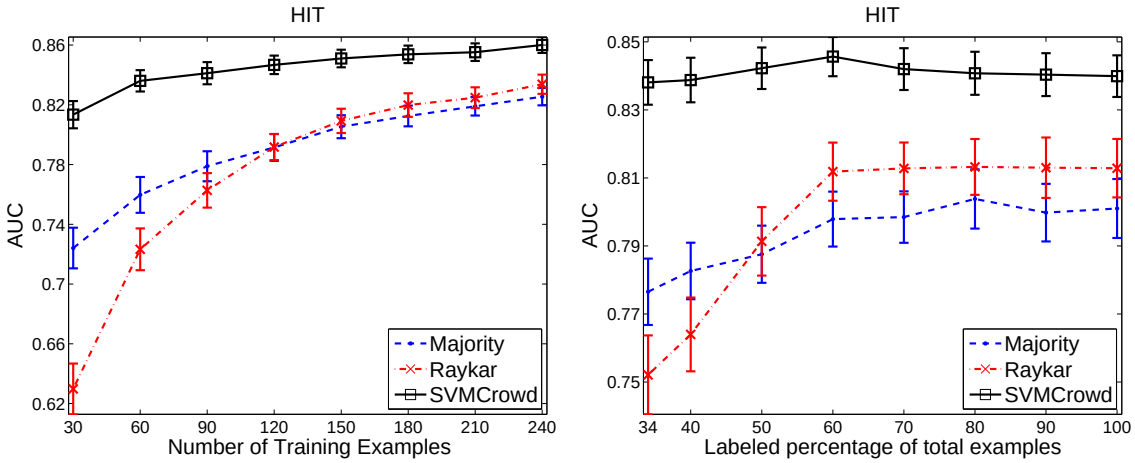

Figure 1: (a) Effect of the number of training examples; (b) Effect of the amount of overlapping examples

6 Results and Discussion

Effect of Different Numbers of Training Examples

The cost of labeling examples in medical domain is typically very high, so in practice we may have a very limited amount of training data. Therefore, it is important to have a model that can efficiently learn from a small number of training examples. We investigate how the different methods perform when the size of training data varies. For this experiment we randomly sample from 30 to 240 examples from the training set to feed the models and evaluate on the test set. Notice that in this experiment, each example is labeled by all the reviewers. The result is shown in figure 1(a). As we can see our method consistently outperforms the other methods. Since our method explicitly models different characteristics of different reviewers, it performs very well even with a small number of labeled examples.

Effect of Reviewing Overlap Since the labeling process is costly, it can be a good idea to give separate set of examples labeled by different reviewers to increase the total number of labeled examples. The question here is how much this will affect the performance of the final consensus classifier. In this section, we investigate how the amount of overlap in the set of examples labeled by different reviewers affects the performance of different learning techniques. Notice that the number of (distinct) examples being labeled will increase if we give reviewers separate sets. To perform this experiment, we set the labeling budget to the total number of training examples (e.g. 377 * 2/3 = 251). We then sampled a percentage (from 34% to 100%) of examples from the training set and fed a random subset from this sample to each reviewer. Since there are three reviewers, the case of 34% leads to a complete overlap and the case of 100% leads to almost separate sets, the two extreme cases for the reviewers overlap study. To understand this for the case of 34%, notice that $34 * 251/100 \approx 85$ and since we have the budget to label 251 examples, we need to ask each reviewer to label the same 85 examples. For the case of 100%, notice that $100 * 251/100 = 251$ and since we have the budget to label 251 examples, we can ask each reviewer to label all 85 distinct examples.

Figure 1(b) shows the result of this experiment, where the x-axis shows the percentage of examples being labeled. It is clear that SVMCrowd consistently outperforms competitors and is very robust to the changes in the reviewing overlap. This is unlike Majority and Raykar methods, which are sensitive to the changes: their performance increases when the amount of overlap decreases which is because they can obtain more number of examples and generalize better. Our method efficiently incorporates different characteristics of reviewers through learning from their labeled examples, so it does not depend on how many distinct examples being labeled.

Modeling Individual Reviewers In this section, we investigate if the model learned for individual reviewer and the reliability parameters learned for each reviewer are accurate.

First, we show that our method can predict labels that each reviewer will give on future examples. The results are

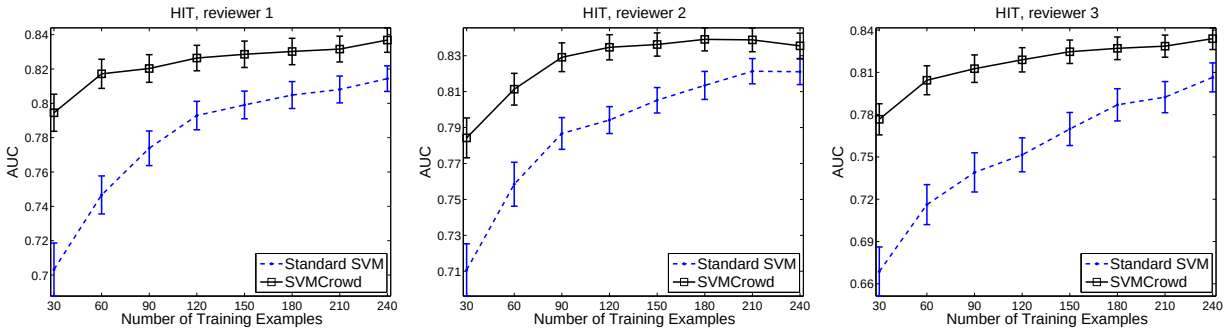

Figure 2: (a) Effect of the number of training examples; (b) Effect of the amount of overlapping examples

summarized in Figure 2, where x-asis is the number of training examples fed to the models and y-axis shows how well the models can predict reviewers' labels in terms of AUC score. Since "Majority" and "Raykar" do not produce reviewer specific models, we compare our method with standard SVM when trained only by the examples reviewed by the reviewer under investigation. The results show that our method clearly outperforms standard SVM for all reviewers. The explanation for this superior performance is that while SVM learns individual reviewers' models independently, our method learns all of them together and knowledge obtained from learning one model can benefit other models. This shows a connection of our approach with transfer learning, where the main idea is utilizing knowledge in one domain to help learning in another related domain.

Second, we give more insight into the reviewers modeling process, where we define the contributions of different reviewers. As we described in Section 4, we model self-consistency and consensus-consistency with parameters α_k and β_k. The first one measures how consistent the labeling of reviewer k is with its own model and the later measures how consistent the model of reviewer k is with the consensus model (Check Equations 10 and 11). Figure 3(a) shows how much labels of the three reviewers under investigation agree with labels given by the senior expert, which we assumed provides the 'true' labels.

According to Figure 3(a) Reviewer 2 is the best, followed by Reviewer 3 and then Reviewer 1. Thus, we expect the Reviewer 2 will have the maximum affect in learning the consensus model $\mathbf{u}$, then Reviewer 3 followed by reviewer 1. To see if this happens in our method, we draw the value of α_k and β_k in Figures 3(b) and 3(c) respectively for different reviewers. In these figures x-axis shows how many training examples are fed to the model. Normalized Self-Consistency in Figures 3(b) is the normalized value of α_k in Equation 8. Normalized Consensus-Consistency in Figure 3(c) is the normalized inverse value of Euclidean distance between a reviewer specific model and consensus model $1/||\mathbf{w}_k - \mathbf{u}||$, which is proportional to β_k in Equation 8. The y-axis of Figure 3(d) is the summation of the two consistency measures. Notice that these figures gives us some detailed insights on how each reviewer performs. First notice that Reviewer 2 is the best reviewer both in terms of self-consistency and consensus-consistency. Second, notice that according to consensus-consistency, Reviewer 3 is better than Reviewer 1, and according to self-consistency, Reviewer 3 is better than Reviewer 1. This means that while Reviewer 3's model is more similar to the consensus model, it makes more random mistakes even according to his own model. The summation of these two reliability parameters in Figure 3(d) implies that the insight provided by our model about the reviewers follows the true knowledge we have about the reviewers.

This result is very promising for practical applications. For example, this can helps to decide different compensation amounts for different reviewers, or in the case we want to collaborate with them in the future we would prefer to hire reviewer 2 (who is more reliable). Moreover, the ability to model experts' behavior is very interesting for education/research purpose, especially in medical domain, where knowledge is complicated and many other factors (care, time spent etc) can affect the performance of an expert.

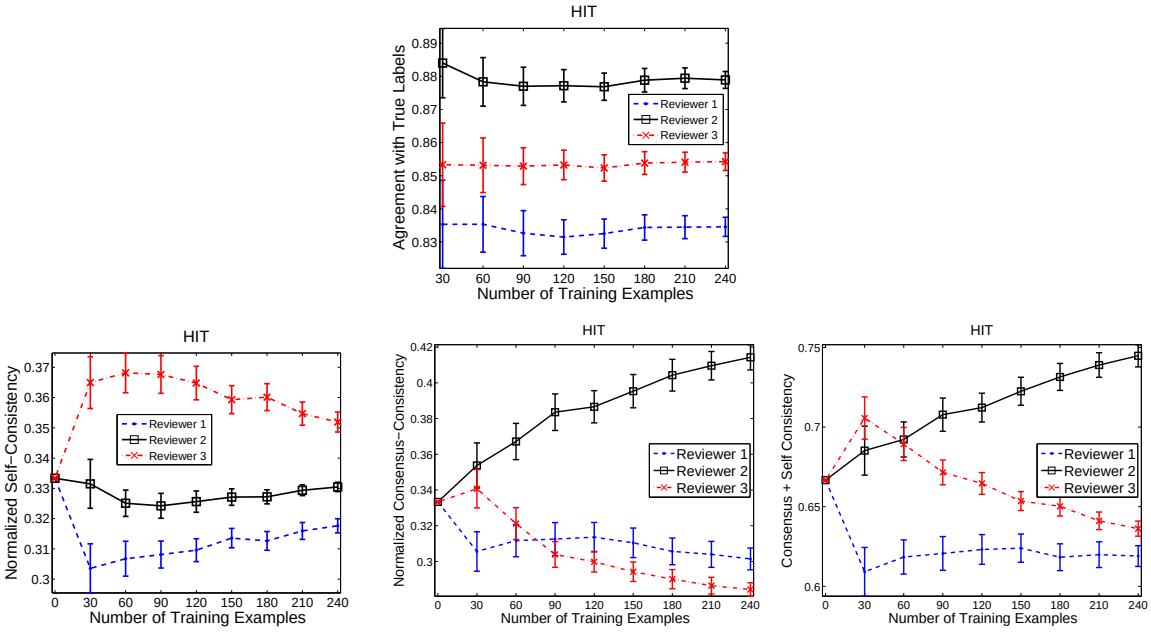

Figure 3: (a) Agreement of junior reviewers' labels with "true" labels given by the senior expert; (b Consensus-Consistency learning progress; (c) Self-Consistency learning progress; (d) Summation of Self and Consensus Consistencies

;

7 Conclusion

It is infeasible or very expensive to obtain objective reliable labels in medical domain. Instead, we may collect labels from multiple annotators/experts. The annotators may be very different in terms of knowledge, reliability and bias. In this work we have investigated a new approach to combine information obtained from different annotators and efficiently learn a common classification model. We have shown empirically that our method clearly outperforms commonly used majority voting and "Raykar" – a state-of-the art method. Our method is especially useful in the case when the number of training examples is small, which is common in practice since the cost of labeling in medical domain is typically very high. Moreover the proposed method is robust to the amount of overlap in training examples. This is important because it gives us flexibility when assigning labeling tasks to different reviewers. In addition to learning a common classification model, our method also learns reviewer specific models and identifies the most reliable reviewers. This provides us the opportunities for many practical applications, for example, fairly distributing resources/compensations, educating/training staff and research of human expert behaviour/knowledge in medical domain.

8 Acknowledgement

This research work was supported by grants R01LM010019 and R01GM088224 from the National Institutes of Health. Its content is solely the responsibility of the authors and does not necessarily represent the official views of the NIH.

References

1. TE. Warkentin, JI. Sheppard, and P. Horsewood. Impact of the patient population on the risk for heparin-induced thrombocytopenia. *Blood*, pages 1703 – 1708, 2000.

2. Victor S. Sheng, Foster Provost, and Panagiotis G. Ipeirotis. Get another label? improving data quality and data mining using multiple, noisy labelers. In *KDD*, pages 614–622. ACM, 2008.

3. A. P. Dawid and A. M. Skene. Maximum likelihood estimation of observer error-rates using the em algorithm. *Applied Statistics*, 28(1):20–28, 1979.

4. Jacob Whitehill, Paul Ruvolo, Ting fan Wu, Jacob Bergsma, and Javier Movellan. Whose Vote Should Count More: Optimal Integration of Labels from Labelers of Unknown Expertise. In *NIPS*, pages 2035–2043. 2009.

5. Peter Welinder, Steve Branson, Serge Belongie, and Pietro Perona. The multidimensional wisdom of crowds. In *NIPS*, 2010.

6. V. C. Raykar, S. Yu, L. H. Zhao, G. H. Valadez, C. Florin, L. Bogoni, and L. Moy. Learning from crowds. *JMLR*, 11:1297–1322, April 2010.

7. Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Information Science and Statistics. Springer, 2006.

8. Hamed Valizadegan, Quang Nguyen, and Milos Hauskrecht. Classification with multiple experts. In *ICML (under review)*, 2012.

9. TE. Warkentin. Heparin-induced thrombocytopenia: pathogenesis and management. *Br J Haematology*, pages 535 – 555, 2003.

10. Michal Valko and Milos Hauskrecht. Feature importance analysis for patient management decisions. *MEDINFO*, 2010.

11. Chih-Chung Chang and Chih-Jen Lin. *LIBSVM: A library for support vector machines*, 2011.