Published in final edited form as:
J Biomed Inform. 2013 December ; 46(6): 1125–1135. doi:10.1016/j.jbi.2013.08.007.

Learning Classification Models from Multiple Experts

Hamed Valizadegan, **Quang Nguyen**, and **Milos Hauskrecht**[1]
Department of Computer Science, University of Pittsburgh

Abstract

Building classification models from clinical data using machine learning methods often relies on labeling of patient examples by human experts. Standard machine learning framework assumes the labels are assigned by a homogeneous process. However, in reality the labels may come from multiple experts and it may be difficult to obtain a set of class labels everybody agrees on; it is not uncommon that different experts have different subjective opinions on how a specific patient example should be classified. In this work we propose and study a new multi-expert learning framework that assumes the class labels are provided by multiple experts and that these experts may differ in their class label assessments. The framework explicitly models different sources of disagreements and lets us naturally combine labels from different human experts to obtain: (1) a consensus classification model representing the model the group of experts converge to, as well as, and (2) individual expert models. We test the proposed framework by building a model for the problem of detection of the Heparin Induced Thrombocytopenia (HIT) where examples are labeled by three experts. We show that our framework is superior to multiple baselines (including standard machine learning framework in which expert differences are ignored) and that our framework leads to both improved consensus and individual expert models.

1. Introduction

The availability of patient data in Electronic Health Records (EHR) gives us a unique opportunity to study different aspects of patient care, and obtain better insights into different diseases, their dynamics and treatments. The knowledge and models obtained from such studies have a great potential in health care quality improvement and health care cost reduction. Machine learning and data mining methods and algorithms play an important role in this process.

The main focus of this paper is on the problem of building (learning) classification models from clinical data and expert defined class labels. Briefly, the goal is to learn a classification model $f : x \rightarrow y$ that helps us to map a patient instance x to a binary class label y, representing, for example, the presence or absence of an adverse condition, or the diagnosis of a specific disease. Such models, once they are learned can be used in patient monitoring, or disease and adverse event detection.

The standard machine learning framework assumes the class labels are assigned to instances by a uniform labeling process. However, in the majority of practical settings the labels come

[1]corresponding author, milos@cs.pitt.edu.
hamed@cs.pitt.edu quang@cs.pitt.edu

from multiple experts. Briefly, the class labels are either acquired (1) during the patient management process and represent the decision of the human expert that is recorded in the EHR (say diagnosis), or (2) retrospectively during a separate annotation process based on past patient data. In the first case, there may be different physicians that manage different patients, hence the class labels naturally originate from multiple experts. Whilst in the second (retrospective) case, the class label can in principle be provided by one expert, the constraints on how much time a physician can spend on patient annotation process often requires to distribute the load among multiple experts.

Accepting the fact that labels are provided by multiple experts, the complication is that different experts may have different subjective opinion about the same patient case. The differences may be due to experts' knowledge, subjective preferences and utilities, and expertise level. This may lead to disagreements in their labels, and variation in the patient case labeling due to these disagreements. However, we would like to note that while we do not expect all experts to agree on all labels, we also do not expect the expert's label assessment to be random; the labels provided by different experts are closely related by the condition (diagnosis, an adverse event) they represent.

Given that the labels are provided by multiple experts, two interesting research questions arise. The first question is whether there is a model that would represent well the labels the group of experts would assign to each patient case. We refer to such a group model as to the (group) consensus model. The second question is whether it is possible to learn such a consensus model purely from label assessments of individual experts, that is, without access to any consensus/meta labels, and this as efficiently as possible.

To address the above issues, we propose a new multi-expert learning framework that starts from data labeled by multiple experts and builds: (1) a *consensus model* representing the classification model the experts collectively converge to, and (2) *individual expert models* representing the class label decisions exhibited by individual experts. Figure 1 shows the relations between these two components: the experts' specific models and the consensus model. We would like to emphasize again that our framework builds the consensus model without access to any consensus/meta labels.

To represent relations among the consensus and expert models, our framework considers different sources of disagreement that may arise when multiple experts label a case and explicitly represents them in the combined multi-expert model. In particular our framework assumes the following sources for expert disagreements:

- Differences in the risks annotators associate with each class label as signment: 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 expert-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, and that, at the same time, permits experts 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 the disease. These differences are best

characterized as differences in their knowledge or model they used to diagnose the patient.

- Differences in time annotators spend when labeling each case: different experts may spend different amount of time and care to analyze the same case and its subtleties. This may lead to labeling inconsistency even within the expert's own model.

We experiment with and test our multi-expert framework on the Heparin Induced Thrombocytopenia (HIT) [23] problem where our goal is to build a predictive model that can, as accurately as possible, assess the risk of the patient developing the HIT condition and predict HIT alerts. We have obtained the HIT alert annotations from three different experts in clinical pharmacy. In addition we have also acquired a meta-annotation from the fourth (senior) expert who in addition to patient cases have seen the annotations and assessments given by other three experts. We show that our framework outperforms other machine learning frameworks (1) when it predicts a consensus label for future (test) patients, and (2) when it predicts individual future expert labels.

2. Background

The problem of learning accurate classification models from clinical data that are labeled by human experts with respect to some condition of interest is important for many applications such as diagnosis, adverse event detection, monitoring and alerting, the design of recommender systems, etc.

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

The key assumption for learning the classification function f in the standard framework is that examples in the training data D are independent and generated by the same (identical) process, hence there are no differences in the label assignment process. However, in practice, especially in medicine, the labels are provided by different humans. Consequently, they may vary and are subject to various sources of subjective bias and variations. We develop and study a new *multi-expert classification learning framework* for which labels are provided by multiple experts, and that accounts for differences in subjective assessments of these experts when learning the classification function.

Briefly, we have m different experts who assign labels to examples. Let $D^k=\left\{\left(\mathbf{x}_i^k, y_i^k\right)\right\}_{i=1}^{n_k}$ denotes training data specific for the expert k, such that $\mathbf{x}_i^k$ is a d-dimensional input example and y_i^k is binary label assigned by expert k. Given the data from multiple experts, our main goal is to learn the classification mapping: $f : x \rightarrow y$ that would generalize well to future examples and would represent a good consensus model for all these experts. In addition, we can learn the expert specific classification functions $g_k : x \rightarrow y^k$ for all $k = 1, \cdots, m$ that predicts as accurately as possible the label assignment for that expert. The learning of f is a difficult problem because (1) the experts' knowledge and reliability could vary, and (2) each expert can have different preferences (or utilities) for different labels, leading to different biases towards negative or positive class. Therefore, even if two experts 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 experts to learn a good consensus model.

2.1. Related work

Methodologically our multi-expert framework builds upon models and results in two research areas: *multi-task learning* and *learning-from-crowds*, and combines them to achieve the above goals.

The multi-task learning framework [9, 27] is applied when we want to learn models for multiple related (correlated) tasks. This framework is used when one wants to learn more efficiently the model by borrowing the data, or model components from a related task. More specifically, we can view each expert and his/her labels as defining a separate classification task. The multi-task learning framework then ties these separate but related tasks together, which lets us use examples labeled by all experts to learn better individual expert models. Our approach is motivated and builds upon the multi-task framework proposed by Evgeniou et al. [9] that ties individual task models using a shared task model. However, we go beyond this framework by considering and modeling the reliability and biases of the different experts.

The learning-from-crowds framework [17, 18] is used to infer consensus on class labels from labels provided jointly by multiple annotators (experts). The existing methods developed for the problem range from the simple majority approach to more complex consensus models representing the reliability of different experts. In general the methods developed try to either (1) derive a consensus of multiple experts on the label of individual examples, or (2) build a model that defines the consensus for multiple experts and can be applied to future examples. We will review these in the following.

The simplest and most commonly used approach for defining the label consensus on individual examples is the majority voting. Briefly, the consensus on the labels for an example is the label assigned by the majority of reviewers. The main limitation of the majority voting approach is that it assumes all experts are equally reliable. The second limitation is that although the approach defines the consensus on labels for existing examples, it does not directly define a consensus model that can be used to predict consensus labels for future examples; although one may use the labels obtained from majority voting to train a model in a separate step.

Improvements and refinements of learning a consensus label or model take into account and explicitly model some of the sources of annotator disagreements. Sheng et al. [17] and Snow et al. [18] showed the benefits of obtaining labels from multiple non-experts and unreliable annotators. Dawid and Skene [8] proposed a learning framework in which biases and skills of annotators were modeled using a confusion matrix. This work was later generalized and extended in [25], [24], and [26] by modeling difficulty of examples. Finally, Raykar et al. [14] used an expectation-maximization (EM) algorithm to iteratively learn the reliability of annotators. The initial reliability estimates were obtained using the majority vote.

The current state-of-the-art learning methods with multiple human annotators are the works of Raykar et al. [14], Welinder et al. [24], and Yan et al. [26]. Among these, only Raykar et al. [14] uses a framework similar to the one we use in this paper; that is, it assumes (1) not all examples are labeled by all experts, (2) the objective is to construct a good classification model. However, the model differs from our approach in how it models the skills and biases of the human annotators. Also the authors in [14] show that their approach improves over simple baselines only when the number of annotators is large (more than 40). This is practical when the labeling task is easy so crowd-sourcing services like Amazon Mechanical Turk can be utilized. However, it is not practical in domains in which the annotation is time consuming. In real world or scientific domains that involve uncertainty, including medicine, it is infeasible to assume the same patient case is labeled in parallel by many different

experts. Indeed the most common cases is when every patient instance is labeled by just one expert.

The remaining state-of-the-art learning from crowds methods, i.e. the works of Welinder et al. [24] and Yan et al. [26], are optimized for different settings than ours. Welinder et al. [24] assumes that there is no feature vector available for the cases; it only learns expert specific models g_ks, and it does not attempt to learn a consensus model f. On the other hand, Yan et al. [26] assumes that each example is labeled by all experts in parallel. As noted earlier, this is unrealistic, and most of the time each example is labeled only by one expert. The approach we propose in this paper overcomes these limitations and is flexible in that it can learn the models when there is one or more labels per example. In addition, our approach differs from the work of Yan et al. [26] in how we parameterize and optimize our model.

3. Methodology

We aim to combine data labeled by multiple experts and build (1) a unified consensus classification model f for these experts and (2) expert-specific models g_k, for all $k = 1, \cdots, m$ that can be applied to future data. Figure 2 illustrates the idea of our framework with linear classification models. Briefly, let us assume a linear consensus model f with parameters (weights) $\mathbf{u}$ and b from which linear expert-specific models g_ks with parameters $\mathbf{w}_k$ and b_k are generated. Given the consensus model, the consensus label on example $\mathbf{x}$ is positive if $\mathbf{u}^{\mathrm{T}}\mathbf{x} + b \geq 0$, otherwise it is negative. Similarly, the expert model g_k for expert k assigns a positive label to example $\mathbf{x}$ if $\mathbf{w}_k^{\mathrm{T}}\,\mathbf{x}+b_k \geq 0$, otherwise the label is negative. To simplify the notation in the rest of the paper, we include the bias term b for the consensus model in the weights vector $\mathbf{u}$, the biases b_k in w_ks, and extend the input vector $\mathbf{x}$ with constant 1.

The consensus and expert models in our framework and their labels are linked together using two reliability parameters:

1. α_k: the self-consistency parameter that characterizes how reliable the labeling of expert k is; it is the amount of consistency of expert k within his/her own model $\mathbf{w}_k$.
2. β_k: the consensus-consistency parameter that models how consistent the model of expert k is with respect to the underlying consensus model $\mathbf{u}$. This parameter models the differences in the knowledge or expertise of the experts.

We assume, all deviations of the expert specific models from the consensus model are adequately modeled by these expert-specific reliability parameters. In the following we present the details of the overall model and how reliability parameters are incorporated into the objective function.

3.1. Multiple Experts Support Vector Machines (ME-SVM)

Our objective is to learn the parameters $\mathbf{u}$ of the consensus model and parameters $\mathbf{w}_k$ for all expert-specific models from the data. We combine this objective with the objective of learning the expert specific reliability parameters α_k and β_k. We have expressed the learning problem in terms of the objective function based on the max-margin classification framework [16, 19] which is used, for example, by Support Vector Machines. However, due to its complexity we motivate and explain its components using an auxiliary probabilistic graphical model that we later modify to obtain the final max-margin objective function.

Figure 3 shows the probabilistic graphical model representation [5, 13] that refines the high level description presented in Figure 2. Briefly, the consensus model **u** is defined by a Gaussian distribution with zero mean and precision parameter η as:

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

where $\mathbf{I}_d$ is the identity matrix of size *d*, and 0_d is a vector of size *d* with all elements equal to 0. The expert-specific models are generated from a consensus model **u**. Every expert k has his/her own specific model $\mathbf{w}_k$ that is a noise corrupted version of the consensus model **u**; that is, we assume that expert *k*, $\mathbf{w}_k$, is generated from a Gaussian distribution with mean **u** and an expert-specific precision β_k:

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

The precision parameter β_k for the expert *k* determines how much $\mathbf{w}_k$ differs from the consensus model. Briefly, for a small β_k, the model $\mathbf{w}_k$ tends to be very different from the consensus model **u**, while for a large β_k the models will be very similar. Hence, β_k represents the consistency of the reviewer specific model $\mathbf{w}_k$ with the consensus model **u**, or, in short, consensus-consistency.

The parameters of the expert model $\mathbf{w}_k$ relate examples (and their features) **x** to labels. We assume this relation is captured by the regression model:

$$p\left(y_i^k|\mathbf{x}_i^k, \mathbf{w}_k, \alpha_k\right) = \mathcal{N}\left(\mathbf{w}_k^{\mathsf{T}}\,\mathbf{x}_i^k, \alpha_k^{-1}\right),$$

where a_k is the precision (inverse variance) and models the noise that may corrupt expert's label. Hence a_k defines the self-consistency of expert *k*. Please also note that although y_i^k is binary, similarly to [9] and [27], we model the label prediction and related noise using the Gaussian distribution. This is equivalent to using the squared error loss as the classification loss.

We treat the self-consistency and consensus-consistency parameters a_k and β_k as random variables, and model their priors using Gamma distributions. More specifically, we define:

$$\begin{aligned} p(\beta_k|\theta_\beta, \tau_\beta) &= \mathcal{G}(\theta_\beta, \tau_\beta), \\ p(\alpha_k|\theta_\alpha, \tau_\alpha) &= \mathcal{G}(\theta_\alpha, \tau_\alpha), \end{aligned} \quad (2)$$

where hyperparameters θ_{β_k} and τ_{β_k} represent the shape and the inverse scale parameter of the Gamma distribution representing β_k. Similarly, θ_{a_k} and τ_{a_k} are the shape and the inverse scale parameter of the distribution representing a_k.

Using the above probabilistic model we seek to learn the parameters of the consensus **u** and expert-specific models *W* from data. Similarly to Raykar at al [14] we optimize the parameters of the model by maximizing the posterior probability $p(\mathbf{u}, W, a, \beta|X, y, \xi)$, where ξ is the collection of hyperparameters η, θ_{β_k}, τ_{β_k}, θ_{a_k}, τ_{a_k}. The posterior probability can be rewritten as follows:

$$p(\mathbf{u}, W, , \alpha, \beta|X, \mathbf{y}, \xi) \propto p(\mathbf{u}|\mathbf{0}_d, \eta)\left(\prod_{k=1}^{m} p(\beta_k|\theta_\beta, \tau_\beta)\, p(\alpha_k|\theta_\alpha, \tau_\alpha)\, p(\mathbf{w}_k|\mathbf{u}, \beta_k) \prod_{i=1}^{n_k} p\left(y_i^k|\mathbf{x}_i^k, \alpha_k, \mathbf{w}_k\right)\right) \quad (3)$$

where $X=\left[\mathbf{x}_1^1;\ldots;\mathbf{x}_{n_1}^1;\ldots;\mathbf{x}_1^m;\ldots;\mathbf{x}_{n_m}^m\right]$ is the matrix of examples labeled by all the experts, and $\mathbf{y}=\left[y_1^1;\ldots;y_{n_1}^1;\ldots;y_m^1;\ldots;y_{n_m}^m\right]$ are their corresponding labels. Similarly, X^k and $\mathbf{y}^k$ are the examples and their labels from expert k. Direct optimization (maximization) of the above function is difficult due to the complexities caused by the multiplication of many terms. A common optimization trick to simplify the objective function is to replace the original complex objective function with the logarithm of that function. This conversion reduces the multiplication to summation [5]. Logarithm function is a monotonic function and leads to the same optimization solution as the original problem. Negative logarithm is usually used to cancel many negative signs produced by the logarithm of exponential distributions. This changes the maximization to minimization. We follow the same practice and take the negative logarithm of the above expression to obtain the following problem (see Appendix A for the details of the derivation):

$$\begin{aligned}\min_{\mathbf{u},\mathbf{w},\alpha,\beta} \frac{\eta}{2}\|\mathbf{u}\|^2 + & \ \frac{1}{2}\sum_{k=1}^{m}\alpha_k\sum_{i=1}^{n_k}\|y_i^k-\mathbf{w}_k^{\mathsf{T}}\,\mathbf{x}_i^k\|^2 \\ + & \ \frac{1}{2}\sum_{k=1}^{m}\beta_k\|\mathbf{w}_k-\mathbf{u}\|^2 \\ + & \ \sum\left(-\ln\left(\beta_k\right)-n_k\ln\left(\alpha_k\right)\right) \\ + & \ \sum\left(-\left(\theta_{\beta_k}-1\right)\ln\left(\beta_k\right)+\tau_{\beta_k}\beta_k\right) \\ + & \ \sum\left(-\left(\theta_{\alpha_k}-1\right)\ln\left(\alpha_k\right)+\tau_{\alpha_k}\alpha_k\right)\end{aligned} \tag{4}$$

Although we can solve the objective function in Equation 4 directly, we replace the squared error function in Equation 4 with the hinge loss[2] for two reasons: (1) the hinge loss function is a tighter surrogate for the zero-one (error) loss used for classification than the squared error loss[15], (2) the hinge loss function leads to the sparse kernel solution[5]. Sparse solution means that the decision boundary depends on a smaller number of training examples. Sparse solutions are more desirable specially when the models are extended to non-linear case where the similarity of the unseen examples needs to be evaluated with respect to the training examples on which the decision boundary is dependent. By replacing the squared errors with the hinge loss we obtain the following objective function:

$$\begin{aligned}\min_{\mathbf{u},\mathbf{w},\alpha,\beta} \frac{\eta}{2}\|\mathbf{u}\|^2 + & \ \frac{1}{2}\sum_{k=1}^{m}\alpha_k\sum_{i=1}^{n_k}\max\left(0,1-y_i^k\mathbf{w}_k^{\mathsf{T}}\,\mathbf{x}_i^k\right)\Big) \\ + & \ \frac{1}{2}\sum_{k=1}^{m}\beta_k\|\mathbf{w}_k-\mathbf{u}\|^2 \\ + & \ \sum\left(-\ln\left(\beta_k\right)-n_k\ln\left(\alpha_k\right)\right) \\ + & \ \sum\left(-\left(\theta_{\beta_k}-1\right)\ln\left(\beta_k\right)+\tau_{\beta_k}\beta_k\right) \\ + & \ \sum\left(-\left(\theta_{\alpha_k}-1\right)\ln\left(\alpha_k\right)+\tau_{\alpha_k}\alpha_k\right)\end{aligned} \tag{5}$$

We minimize the above objective function with respect to the consensus model **u**, the expert specific model $\mathbf{w}_k$, and expert specific reliability parameters a_k and β_k.

3.2. Optimization

We need to optimize the objective function in Equation 5 with regard to parameters of the consensus model **u**, the expert-specific models $\mathbf{w}_k$, and expert-specific parameters a_k and β_k.

[2]Hinge loss is a loss function originally designed for training large margin classifiers such as support vector machines. The minimization of this loss leads to a classification decision boundary that has the maximum distance to the nearest training example. Such a decision boundary has interesting properties, including good generalization ability. [15, 21]

Similarly to the SVM, the hinge loss term: $\max\left(0, 1 - y_i^k \mathbf{w}_k^\mathsf{T} \mathbf{x}_i^k\right)$ in Equation 5 can be replaced by a constrained optimization problem with a new parameter ϵ_i^k. Briefly, from the optimization theory, the following two equations are equivalent [6]:

$$\min_{\mathbf{w}_k} \max\left(0, 1 - y_i^k \mathbf{w}_k^\mathsf{T} \mathbf{x}_i^k\right)$$

and

$$\min_{\epsilon_i^k, \mathbf{w}_k} \epsilon_i^k \quad s.t. \quad y_i^k \mathbf{w}_k^\mathsf{T} \mathbf{x}_i^k > 1 - \epsilon_i^k$$

Now replacing the hinge loss terms in Equation 5, we obtain the equivalent optimization problem:

$$\begin{aligned} \min_{\mathbf{u},\mathbf{w},\epsilon,\alpha,\beta} \frac{\eta}{2}\|\mathbf{u}\|^2 + & \quad \frac{1}{2}\sum_{k=1}^{m} \alpha_k \sum_{i=1}^{n_k} \epsilon_i^k \\ + & \quad \frac{1}{2}\sum_{k=1}^{n} \beta_k \|\mathbf{w}_k - \mathbf{u}\|^2 \\ + & \quad \sum\left(-\ln\left(\beta_k\right) - n_k \ln\left(\alpha_k\right)\right) \\ + & \quad \sum\left(-\left(\theta_{\beta_k} - 1\right) \ln\left(\beta_k\right) + \tau_{\beta_k}\beta_k\right) \\ + & \quad \sum\left(-\left(\theta_{\alpha_k} - 1\right) \ln\left(\alpha_k\right) + \tau_{\alpha_k}\alpha_k\right) \\ \text{s.t.} & \quad y_i^k \mathbf{w}_k^\mathsf{T} \mathbf{x}_i^k \geq 1 - \epsilon_i^k, k{=}1 \ldots m, \quad i{=}1 \ldots n_k \\ & \quad \epsilon_i^k \geq 0, \qquad\qquad k{=}1 \ldots m, \quad i{=}1 \ldots n_k \end{aligned} \tag{6}$$

where $\boldsymbol{\epsilon}$ denote the new set of ϵ_i^k parameters.

We optimize the above objective function using the alternating optimization approach [4]. Alternating optimization splits the objective function into two (or more) easier subproblems, each depends only on a subset of (hidden/learning) variables. After initializing the variables, it iterates over optimizing each set by fixing the other set until there is no change of values of all the variables. For our problem, diving the learning variables into two subsets, $\{\alpha, \beta\}$ and $\{\mathbf{u}, \mathbf{w}\}$ makes each subproblem easier, as we describe below. After initializing the first set of variables, i.e. $\alpha_k = 1$ and $\beta_k = 1$, we iterate by performing the following two steps in our alternating optimization apparoach:

- **Learning u and $\mathbf{w}_k$**: In order to learn the consensus model **u** and expert specific model $\mathbf{w}_k$, we consider the reliability parameters α_k and β_k as constants. This will lead to an SVM form optimization to obtain **u** and $\mathbf{w}_k$. Notice that ϵ_i^k is also learned as part of SVM optimization.
- **Learning α_k and β_k**: By fixing **u**, $\mathbf{w}_k$ for all experts, and $\boldsymbol{\epsilon}$, we can minimize the objective function in Equation 6 by computing the derivative with respect to $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$. This results in the following closed form solutions for α_k and β_k:

$$\alpha_k = \frac{2\left(n_k + \theta_{\alpha_k} + 1\right)}{\sum_{y_i^k = 1} \epsilon_i^k + 2\tau_{\alpha_k}} \tag{7}$$

$$\beta_k = \frac{2\theta_{\beta_k}}{\|\mathbf{w}_k - \mathbf{u}\|^2 + 2\tau_{\beta_k}}. \quad (8)$$

Notice that ϵ_i^k is the amount of violation of label constraint for example $\mathbf{x}_i^k$ (i.e. the i^{th} example labeled by expert k) thus $\sum_{i=1} \epsilon_i^k$ is the summation of all labeling violations for model of expert k. This implies that α_k is inversely proportional to the amount of misclassification of examples by expert k according to its specific model $\mathbf{w}_k$. As a result, α_k represents the consistency of the labels provided by expert k with his/her own model. β_k is inversely related to the difference of the model of expert k (i.e. w_k) with the consensus model $\mathbf{u}$. Thus it is the consistency of the model learned for expert k from the consensus model $\mathbf{u}$.

4. Experimental evaluation

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) [23, 22]. 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.

4.1. Data

The data used in the experiments were extracted from over 4, 486 electronic health records (EHRs) in Post-surgical Cardiac Patient (PCP) database [11, 20, 12]. The initial data consisted of over 51, 000 unlabeled patient-state instances obtained by segmenting each EHR record in time with 24-hours period. Out of these we have selected 377 patient instances using a stratified sampling approach that were labeled by clinical pharmacists who attend and manage patients with HIT. Since the chance of observing HIT is relatively low, the stratified sampling was used to increase the chance of observing patients with positive labels. Briefly, a subset of strata covered expert-defined patterns in the EHR associated with the HIT or its management, such as, the order of the HPF4 lab test used to confirm the condition [22]. We asked three clinical pharmacists to provide us with labels showing if the patient is at the risk of HIT and if they would agree to raise an alert on HIT if the patient was encountered prospectively. The assessments were conducted using a web-based graphical interface (called PATRIA) we have developed to review EHRs of patients in the PCP database and their instances. All three pharmacists worked independently and labeled all 377 instances. After the first round of expert labeling (with three experts) we asked a (senior) expert on HIT condition to label the data, but this time, the expert in addition to information in the EHR also had access to the labels of the other three experts. This process led to 88 positive and 289 negative labels. We used the judgement and labels provided by this expert as consensus labels.

We note that alternative ways of defining consensus labels in the study would be possible. For example, one could ask the senior expert to label the cases independent of labels of other reviewers and consider expert's labels as surrogates for the consensus labels. Similarly one can ask all three experts to meet and resolve the cases they disagree on. However, these alternative designs come with the different limitations. First, not seeing the labels of other reviewers the senior expert would make a judgment on the labels on her own and hence it would be hard to speak about consensus labels. Second, the meeting of the experts and the resolution of the differences on every case in the study in person would be hard to arrange and time consuming to undertake. Hence, we see the option of using senior expert's opinion to break the ties as a reasonable alternative that (1) takes into account labels from all experts, and, (2) resolves them without arranging a special meeting of all experts involved.

In addition, we would like to emphasize that the labels provided by the (senior) expert were only used to evaluate the quality of the different consensus models. That is, we we did not use the labels provided by that expert when training the different consensus models, and only applied them in the evaluation phase.

4.2. Temporal feature extraction

The EHR consists of complex multivariate time series data that reflect sequences of lab values, medication administrations, procedures performed, etc. In order to use these for building HIT prediction models, a small set of temporal features representing well the patient state with respect to HIT for any time t is needed. However, finding a good set of temporal features is an extremely challenging task [10, 2, 7, 3, 1]. Briefly, the clinical time series, are sampled at irregular times, have missing values, and their length may vary depending on the time elapsed since the patient was admitted to the hospital. All these make the problem of summarizing the information in the time series hard. In this work, we address the above issues by representing the patient state at any (segmentation) time *t* using a subset of pre-defined temporal feature mappings proposed by Hauskrecht et al [11, 20, 12] (Table 1 in Appendix B) that let us convert patient's information known at time *t* to a fixed length feature vector. The feature mappings define temporal features such as last observed platelet count value, most recent platelet count trend, or, the length of time the patient is on medication, etc. Figure 4 illustrates a subset of 10 feature mappings (out of 14) that we applied to summarize time series for numeric lab tests. We used feature mappings for five clinical variables useful for the detection of HIT: Platelet counts, Hemoglobin levels, White Blood Cell Counts, Heparin administration record, Major heart procedure. The full list of features generated for these variables is listed in Table 1 in Appendix B. Briefly, temporal features for numeric lab tests: Platelet counts, Hemoglobin levels and White Blood Cell Counts used feature mappings illustrated in Figure 4 plus additional features representing the presence of last two values, and pending test. The heparin features summarize if the patient is currently on the heparin or not, and the timing of the administration, such as the time elapsed since the medication was started, and the time since last change in its administration. The heart procedure features summarize whether the procedure was performed or not and the time elapsed since the last and first procedure. The feature mappings when applied to EHR data let us map each patient instance to a vector of 50 features. These features were then used to learn the models in all subsequent experiments. The alert labels assigned to patient instances by experts were used as class labels.

4.3. Experimental Set-up

To demonstrate the benefits of our multi-expert learning framework we used patient instances labeled by four experts as outlined above. The labeled data were randomly split into the training and test sets, such that 2/3 of examples were used for training examples and 1/3 for testing. We trained all models in the experimental section on the training set and

evaluated on the test set. We used the Area Under the ROC Curve (AUC) on the test set as the main statistic for all comparisons. We repeated train/test split 100 times and report the average and 95% confidence interval. We compare the following algorithms:

- **SVM-baseline:** This is a model obtained by training a linear SVM classifier that considers examples and their labels and ignores any expert information. We use the model as a baseline.
- **Majority:** This model selects the label in the training data using the majority vote and learns a linear SVM classifier on examples with the majority label. This model is useful only when multiple experts label the same patient instance. Notice that SVM and Majority performs exactly the same if each example is labeled by one and only one expert.
- **Raykar:** This is the algorithm and model developed by Raykar et. al. [14]. We used the same setting as discussed in [14].
- **ME-SVM:** This is the new method we propose in this paper. We set the parameters $\eta = \tau_\alpha = \tau_\beta = 1$, $\theta_\alpha = \theta_\beta = 1$.
- **SE-SVM:** Senior-Expert-SVM (SE-SVM) is the SVM model trained using the consensus labels provided by our senior pharmacist. Note that this method does not derive a consensus model from labels given by multiple experts; instead, it 'cheats' and learns consensus model directly from consensus labels. This model and its results are used for comparison purposes only and serve as the reference point.

We investigate two aspects of the proposed ME-SVM method:

1. The performance of the consensus model on the test data when it is evaluated on the labels provided by the senior expert on HIT.
2. The performance of the expert-specific model $\mathbf{w}_k$ for expert k when it is evaluated on the examples labeled by that expert.

4.4. Results and Discussion

4.4.1. Learning consensus model—The cost of labeling examples in medical domain is typically very high, so in practice we may have a very limited number 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 different methods perform when the size of training data varies. For this experiment we randomly sample examples from the training set to feed the models and evaluate them on the test set. We simulated and tested two different ways of labeling the examples used for learning the model: (1) every example was given to just one expert, and every expert labeled the same number of examples, and (2) every example was given to all experts, that is, every example was labeled three times. The results are shown in Figure 5. The x-axis shows the total number of cases labeled by the experts. The left and right plots respectively show the results when labeling options 1 and 2 are used.

First notice that our method that explicitly models experts' differences and their reliabilities consistently outperforms other consensus methods in both strategies, especially when the number of training examples is small. This is particularly important when labels are not recorded in the EHRs and must be obtained via a separate post-processing step, which can turn out to be rather time-consuming and requires additional expert effort. In contrast to our method the majority voting does not model the reliability of different experts and blindly considers the consensus label as the majority vote of labels provided by different experts. The SVM method is a simple average of reviewer specific models and does not consider the reliability of different experts in the combination. The Raykar method, although modeling

the reliabilities of different experts, assumes that the experts have access to the label generated by the consensus model and report a perturbed version of the consensus label. This is not realistic because it is not clear why the expert perturb the labels if they have access to consensus model. In contrary, our method assumes that different experts aim to use a model similar to consensus model to label the cases however their model differs from the label of the consensus model because of their differences in the domain knowledge, expertise and utility functions. Thus, our method uses a more intuitive way and realistic approach to model the label generating process.

Second, by comparing the two strategies for labeling patient instances we see that option 1, where each reviewer labels different patient instances, is better (in terms of the total labeling effort) than option 2 where all reviewers label the same instances. This shows that the diversity in patient examples seen by the framework helps and our consensus model is improving faster, which is what we intuitively expect.

Finally, note that our method performs very similarly to the SE-SVM – the model that 'cheats' and is trained directly on the consensus labels given by the senior pharmacist. This verifies that our framework is effective in finding a good consensus model without having access to the consensus labels. .

4.4.2. Modeling individual experts—One important and unique feature of our framework when compared to other multi-expert learning frameworks is that it models explicitly the individual experts' models $\mathbf{w_k}$, not just the consensus model **u**. In this section, we study the benefit of the framework for learning the expert specific models by analyzing how the model for any of the experts can benefit from labels provided by other experts. In other words we investigate the question: *Can we learn an expert model better* by borrowing the knowledge and labels from other experts? We compared the expert specific models learned by our framework with the following baselines:

- **SVM:** We trained a separate SVM model for each expert using patient instances labeled only by that expert. We use this model as a baseline.
- **Majority*:** This is the Majority model described in the previous section. However, since Majority model does not give expert specific models, we use the consensus model learned by the Majority method in order to predict the labels of each expert.
- **Raykar*:** This is the model developed by Raykar et. al. [14], as described in the previous section. Similarly to Majority, Raykar's model does not learn expert specific models. Hence, we use the consensus model it learns to predict labels of individual experts.
- **ME-SVM:** This is the new method we propose in this paper, that generates expert specific models as part of its framework.

Similarly to Section 4.4.1, we assume two different ways of labeling the examples: (1) every example was given to just one expert, and every expert labeled the same number of examples, and (2) every example was given to all experts, that is every example was labeled three times.

We are interested in learning individual prediction models for three different experts. If we have a budget to label some number of patient instances, say, 240, and give 80 instances to each expert, then we have can learn an individual expert model from: (1) all 240 examples by borrowing from the instances labeled by the other experts, or (2) only its own 80 examples. The hypothesis is that learning from data and labels given by all three experts collectively is better than learning each of them individually. The hypothesis is also closely

related to the goal of multi-task learning, where the idea is to use knowledge, models or data available for one task to help learning of models for related domains.

The results for this experiment are summarized in Figure 6, where x-axis is the number of training examples fed to the models and y-axis shows how well the models can predict individual experts' labels in terms of the AUC score. The first (upper) line of sub-figures shows results when each expert labels a different set of patient instances, whereas the second (lower) line of sub-figures shows results when instances are always labeled by all three experts. The results show that our ME-SVM method outperforms the SVM trained on experts' own labels only. This confirms that learning from three experts collectively helps to learn expert-specific models better than learning from each expert individually and that our framework enables such learning. In addition, the results of Majority* and Raykar* methods show that using their consensus models to predict expert specific labels is not as effective and that their performance falls bellow our framework that relies on expert specific models.

4.4.3. Self-consistency and consensus-consistency—As we described in Section 3, we model self-consistency and consensus-consistency with parameters α_k and β_k. α_k measures how consistent the labeling of expert k is with his/her own model and β_k measures how consistent the model of expert k is with respect to the consensus model. The optimization problem we proposed in Equation 6 aims to learn not just the parameters $\mathbf{u}$ and $\mathbf{w}_k$ of the consensus and experts' models, but also the parameters α_k and β_k, and this without having access to the labels from the senior expert.

In this section, we attempt to study and interpret the values of the reliability parameters as they are learned by our framework and compare them to empirical agreements in between the senior (defining the consensus) and other experts. Figure 7(a) shows the agreements of labels provided by the three experts with labels given by the senior expert, which we assumed gives the consensus labels. From this figure we see that Expert 2 agrees with the consensus labels the most, followed by Expert 3 and then Expert 1. The agreement is measured in terms of the absolute agreement, and reflects the proportion of instances for which the two labels agree.

Figures 7(b) and 7(c) show the values of the reliability parameters α and β, respectively. The x-axis in these figures shows how many training patient instances per reviewer are fed to the model. Normalized Self-Consistency in Figure 7(b) is the normalized value of α_k in Equation 6. Normalized Consensus-Consistency in Figure 7(c) is the normalized inverse value of Euclidean distance between an expert specific model and consensus model: $1/\|\mathbf{w}_k - \mathbf{u}\|$, which is proportional to β_k in Equation 6. In Figure 7(d) we add the two consistency measures in an attempt to measure the overall consistency in between the senior expert (consensus) and other experts.

As we can see, at the beginning when there is no training data all experts are assumed to be the same (much like the majority voting approach). However, as the learning progresses with more training examples available, the consistency measures are updated and their values define the contribution of each expert to the learning of consensus model: the higher the value the larger the contribution. Figure 7(b) shows that expert 3 is the best in terms of self-consistency given the linear model, followed by expert 2 and then expert 1. This means expert 3 is very consistent with his model, that is, he likely gives the same labels to similar examples. Figure 7(c) shows that expert 2 is the best in terms of consensus-consistency, followed by expert 3 and then expert 1. This means that although expert 2 is not very consistent with respect to his own linear model his model appears to converge closer to the consensus model. In other words, expert 2 is the closest to the expected consensus in terms

of the expertise but deviates with some labels from his own linear model than expert 3 does[3].

Figure 7(d) shows the summation of the two consistency measures. By comparing Figure 7(a) and Figure 7(d) we observe that the overall consistency mimics well the agreements in between the expert defining the consensus and other experts, especially when the number of patient instances labeled and used to train our model increases. This is encouraging, since the parameters defining the consistency measures are learned by our framework only from the labels of the three experts and hence the framework never sees the consensus labels.

5. Conclusion

The construction of predictive classification models from clinical data often relies on labels reflecting subjective human assessment of the condition of interest. In such a case, differences among experts may arise leading to potential disagreements on the class label that is assigned to the same patient case. In this work, we have developed and investigated a new approach to combine class-label information obtained from multiple experts and learn a common (consensus) classification model. We have shown empirically that our method outperforms other state-ofthe-art methods when building such a model. In addition to learning a common classification model, our method also learns expert specific models. This addition provides us with an opportunity to understand the human experts' differences and their causes which can be helpful, for example, in education and training, or in resolving disagreements in the patient assessment and patient care.

Acknowledgments

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.

Appendix A: Taking the negative logarithm of the posterior in Equation 3

In this appendix, we give a more detailed derivation of Equation 4 from 3:

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

$$= \mathcal{N}\left(\mathbf{u}|\mathbf{0}, \beta_{\mathbf{k}}^{-1}\mathbf{I_d}\right) \left(\prod_{\mathbf{k=1}}^{\mathbf{m}} \mathcal{G}\left(\beta_{\mathbf{k}}|\theta_\beta, \tau_\beta\right) \mathcal{G}\left(\alpha_{\mathbf{k}}|\theta_\alpha, \tau_\alpha\right) \mathcal{N}\left(\mathbf{w_k}|\mathbf{u}, \beta_{\mathbf{k}}\right) \mathcal{N}\left(\mathbf{w_k}|\mathbf{u}, \beta_{\mathbf{k}}\right) \prod_{\mathbf{i=1}}^{\mathbf{n_k}} \mathcal{N}\left(\mathbf{y_i^k}|\mathbf{w_k^\top}\, \mathbf{x_i^k}, \alpha_{\mathbf{k}}\right) \right)$$

$$= \frac{\beta_k}{\sqrt{2\pi}} e^{-\frac{\eta\|\mathbf{u}\|^2}{2}} \times \left(\prod_{k=1}^{m} \frac{1}{\Gamma\left(\theta_\beta\right)} \tau_\beta^{\theta_\beta} \beta_k^{\theta_\beta - 1} e^{-\tau_\beta \beta_k} \frac{1}{\Gamma\left(\theta_\alpha\right)} \tau_\alpha^{\theta_\alpha} \alpha_k^{\theta_\alpha - 1} e^{-\tau_\alpha \alpha_k} \frac{\beta_k}{\sqrt{2\pi}} e^{-\frac{\beta_k\|\mathbf{w}_k - \mathbf{u}\|^2}{2}} \prod_{i=1}^{n_k} \frac{\alpha_k}{\sqrt{2\pi}} e^{-\frac{\alpha_k\|y_i^k - \mathbf{w}_k^\top\, \mathbf{x}_i^k\|^2}{2}} \right)$$

Getting the negative logarithm of the last statement, we will have:

[3]We would like to note that the self-consistency and consensus-consistency parameters learned by our framework are learned together and hence it is possible one consistency measure may offset or compensate for the value of the other measure during the optimization. In that case the interpretation of the parameters as presented may not be as straightforward.

$$
\begin{aligned}
&-\log(\eta) - log\left(\sqrt{2\pi}\right) \\
&\quad +\frac{1}{2}\eta\|\mathbf{u}\|^2 \\
&\quad +\sum_{k=1}^{m}\Big(\log(\Gamma(\theta_\beta)) - \theta_\beta\log(\tau_\beta) - (\theta_\beta-1)\log(\beta_k) + \tau_\beta\beta_k + \log(\Gamma(\theta_\alpha)) = \theta_\alpha\log(\tau_\alpha) - (\theta_\alpha-1)\log(\alpha_k) + \tau_\alpha\alpha_k - \log(
\end{aligned}
$$

Removing the constants terms (i.e. those related to η, θ_α, τ_α, θ_β and τ_β, we will have:

$$
\begin{aligned}
&\frac{1}{2}\eta\|\mathbf{u}\|^2 \\
&\quad +\sum_{k=1}^{m}\Big(-(\theta_\beta-1)\log(\beta_k) + \tau_\beta\beta_k - (\theta_\alpha-1)\log(\alpha_k) + \tau_\alpha\alpha_k - \log(\beta_k) + \frac{1}{2}\beta_k\|_{\mathbf{w}_k} - \mathbf{u}\|^2 + \sum_{i=1}^{n_k} - \log(\alpha_k) + \frac{1}{2}\alpha_k y_i^k - \mathbf{w}_k^{\mathsf{T}}
\end{aligned}
$$

Rearranging the terms in the above equation, we obtain Equation 4.

Appendix B: Features used for constructing the predictive models

Table 1

Features used for constructing the predictive models. The features were extracted from time series data in electronic health records using methods from Hauskrecht et al [11, 20, 12]

Clinical variables	Features	
Platelet count (PLT)	1	last PLT value measurement
	2	time elapsed since last PLT measurement
	3	pending PLT result
	4	known PLT value result indicator
	5	known trend PLT results
	6	PLT difference for last two measurements
	7	PLT slope for last two measurements
	8	PLT % drop for last two measurements
	9	nadir HGB value
	10	PLT difference for last and nadir values
	11	apex PLT value
	12	PLT difference for last and apex values
	13	PLT difference for last and baseline values
	14	overall PLT slope
Hemoglobin (HGB)	15	last HGB value measurement
	16	time elapsed since last HGB measurement
	17	pending HGB result
	18	known HGB value result indicator
	19	known trend HGB results
	20	HGB difference for last two measurements
	21	HGB slope for last two measurements
	22	HGB % drop for last two measurements
	23	nadir HGB value
	24	HGB difference for last and nadir values
	25	apex HGB value
	26	HGB difference for last and apex values
	27	HGB difference for last and baseline values
	28	overall HGB slope

Clinical variables	Features	
White Blood Cell count (WBC)	29	last WBC value measurement
	30	time elapsed since last WBC measurement
	31	pending WBC result
	32	known WBC value result indicator
	33	known trend WBC results
	34	WBC difference for last two measurements
	35	WBC slope for last two measurements
	36	WBC % drop for last two measurements
	37	nadir WBC value
	38	WBC difference for last and nadir values
	39	apex WBC value
	40	WBC difference for last and apex values
	41	WBC difference for last and baseline values
	42	overall WBC slope
Heparin	43	Patient on Heparin
	44	Time elapsed since last administration of Heparin
	45	Time elapsed since first administration of Heparin
	46	Time elapsed since last change in Heparin administration
Major heart proce-dure	47	Patient had a major heart procedure in past 24 hours
	48	Patient had a major heart procedure during the stay
	49	Time elapsed since last major heart procedure
	50	Time elapsed since first major heart procedure

References

1. Batal, Iyad; Fradkin, Dmitriy; Harrison, James; Moerchen, Fabian; Hauskrecht, Milos. Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM; 2012. Mining recent temporal patterns for event detection in multivariate time series data.; p. 280-288.
2. Batal, Iyad; Sacchi, Lucia; Bellazzi, Riccardo; Hauskrecht, Milos. Multivariate time series classification with temporal abstractions.. Florida Artificial Intelligence Research Society Conference; 2009;
3. Batal, Iyad; Valizadegan, Hamed; Cooper, Gregory F.; Hauskrecht, Milos. IEEE International Conference on Bioinformatics and Biomedicine (BIBM). IEEE; 2011. A pattern mining approach for classifying multivariate temporal data.; p. 358-365.
4. Bezdek, James C.; Hathaway, Richard J. Proceedings of the 2002 AFSS International Conference on Fuzzy Systems. Calcutta: Advances in Soft Computing, AFSS '02. Springer-Verlag; London, UK, UK: 2002. Some notes on alternating optimization.; p. 288-300.
5. Christopher, M. Bishop. Pattern Recognition and Machine Learning. Springer; 2006. Information Science and Statistics..
6. Boyd, Stephen; Vandenberghe, Lieven. Convex Optimization. Cambridge University Press; New York, NY, USA: 2004.
7. Combi, Carlo; Keravnou-Papailiou, Elpida; Shahar, Yuval. Temporal information systems in medicine. Springer Publishing Company, Incorporated; 2010.
8. Dawid AP, Skene AM. Maximum likelihood estimation of observer error-rates using the em algorithm. Applied Statistics. 1979; 28(1):20–28.
9. Evgeniou, Theodoros; Pontil, Massimiliano. KDD. ACM; New York, NY, USA: 2004. Regularized multi–task learning.; p. 109-117.
10. Hauskrecht, M.; Fraser, H. Modeling treatment of ischemic heart disease with partially observable markov decision processes.. Proceedings of the AMIA Symposium; 1998; p. 538-542.
11. Hauskrecht, M.; Valko, M.; Batal, I.; Clermont, G.; Visweswaran, S.; Cooper, GF. Conditional outlier detection for clinical alerting.. AMIA Annual Symposium Proceedings; 2010; p. 286-890.
12. Hauskrecht, Milos; Batal, Iyad; Valko, Michal; Visweswaran, Shyam; Cooper, Gregory F.; Clermont, Gilles. Outlier detection for patient monitoring and alerting. Journal of Biomedical Informatics. 2013; 46(1):47–55. [PubMed: 22944172]

13. Koller, Daphne; Friedman, Nir. Probabilistic Graphical Models: Principles and Techniques. MIT Press; 2009.
14. Raykar VC, Yu S, Zhao LH, Valadez GH, Florin C, Bogoni L, Moy L. Learning from crowds. JMLR. Apr.2010 11:1297–1322.
15. Scholkopf, Bernhard; Smola, Alexander J. Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond. MIT Press; Cambridge, MA, USA: 2001.
16. Scholkopf, Bernhard; Smola, Alexander J. Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond. MIT Press; Cambridge, MA, USA: 2002.
17. Sheng, Victor S.; Provost, Foster; Ipeirotis, Panagiotis G. KDD. ACM; 2008. Get another label? improving data quality and data mining using multiple, noisy labelers.; p. 614-622.
18. Snow, Rion; O'Connor, Brendan; Jurafsky, Daniel; Ng, Andrew Y. EMNLP. Association for Computational Linguistics; Stroudsburg, PA, USA: 2008. Cheap and fast—but is it good?: evaluating non-expert annotations for natural language tasks.; p. 254-263.
19. Valizadegan, Hamed; Jin, Rong. Generalized maximum margin clustering and unsupervised kernel learning.. In: Schölkopf, B.; Platt, J.; Hoffman, T., editors. Advances in Neural Information Processing Systems 19. MIT Press; Cambridge, MA: 2007. p. 1417-1424.
20. Valko, Michal; Hauskrecht, Milos. Feature importance analysis for patient management decisions.. Proceedings of the 13th International Congress on Medical Informatics; 2010; p. 861-865.
21. Vapnik, Vladimir N. The nature of statistical learning theory. Springer-Verlag New York, Inc.; New York, NY, USA: 1995.
22. Warkentin TE. Heparin-induced thrombocytopenia: pathogenesis and management. Br J Haematology. 2003:535–555.
23. Warkentin TE, Sheppard JI, Horsewood P. Impact of the patient population on the risk for heparin-induced thrombocytopenia. Blood. 2000:1703–1708. [PubMed: 10961867]
24. Welinder, Peter; Branson, Steve; Belongie, Serge; Perona, Pietro. The multidimensional wisdom of crowds. NIPS. 2010
25. Whitehill, Jacob; Ruvolo, Paul; Wu, Ting fan; Bergsma, Jacob; Movellan, Javier. Whose Vote Should Count More: Optimal Integration of Labels from Labelers of Unknown Expertise. NIPS. 2009:2035–2043.
26. Yan, Yan; Fung, Glenn; Dy, Jennifer; Rosales, Romer. Modeling annotator expertise: Learning when everybody knows a bit of something. AISTATS. Apr.2010
27. Zhang, Yu; Yeung, Dit-Yan. A convex formulation for learning task relationships in multi-task learning. UAI. 2010

Paper highlights

- Learning of classification models when labels are provided by multiple experts
- A new multi-expert learning approach that gives: (a) consensus, and (b) experts' models
- Tests the approach on clinical data with three expert reviewers and one meta-reviewer
- The results show the improved learning of the consensus model
- The results show the improved learning of individual expert models

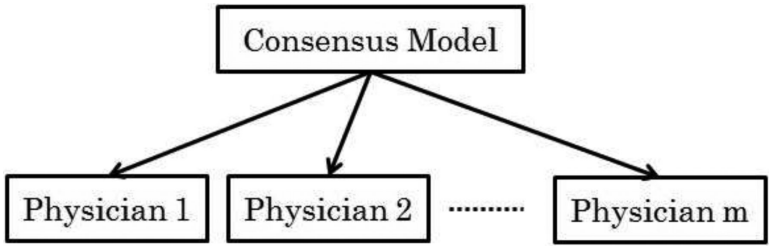

Figure 1.
The consensus model and its relation to individual expert models.

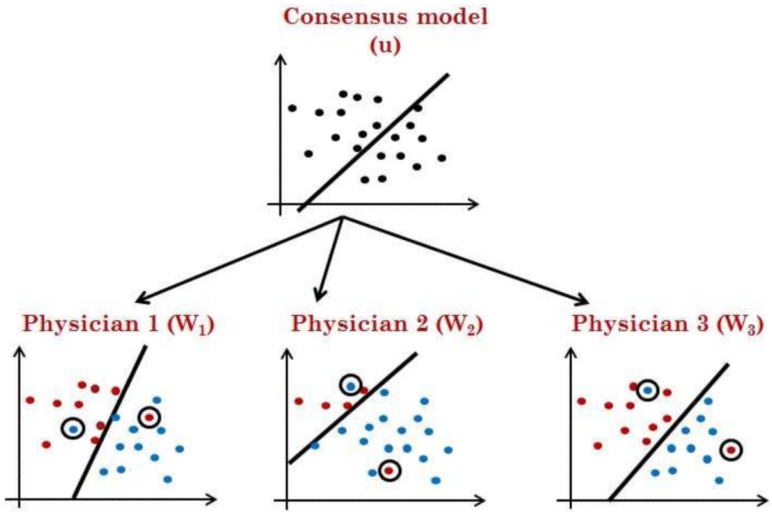

Figure 2.
The experts' specific linear models $\mathbf{w}_k$ are generated from the consensus linear model **u**. The circles show instances that are mislabeled with respect to individual expert's models and are used to define the model self consistency.

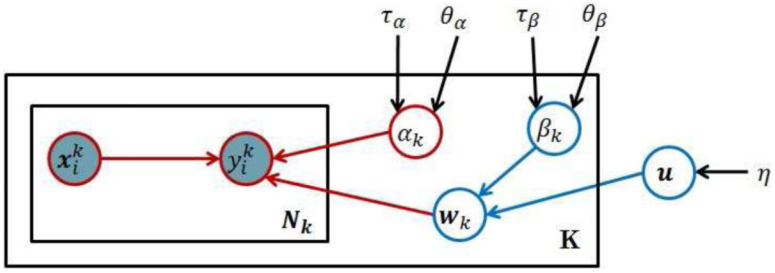

Figure 3.

Graphical representation of the auxiliary probabilistic model that is related to our objective function. The circles in the graph represent random variables. Shaded circles are observed variables, regular (unshaded) circles denote hidden (or unobserved) random variables. The rectangles denote plates that represent structure replications, that is, there are k different expert models $\mathbf{w}_k$, and each is used to generate labels for N_k examples. Parameters not enclosed in circles (e.g. η) denote the hyperparameters of the model.

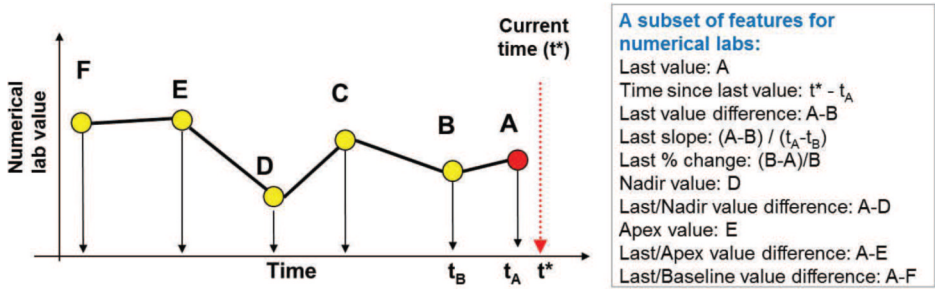

Figure 4.
The figure illustrates a subset of 10 temporal features used for mapping time-series for numerical lab tests.

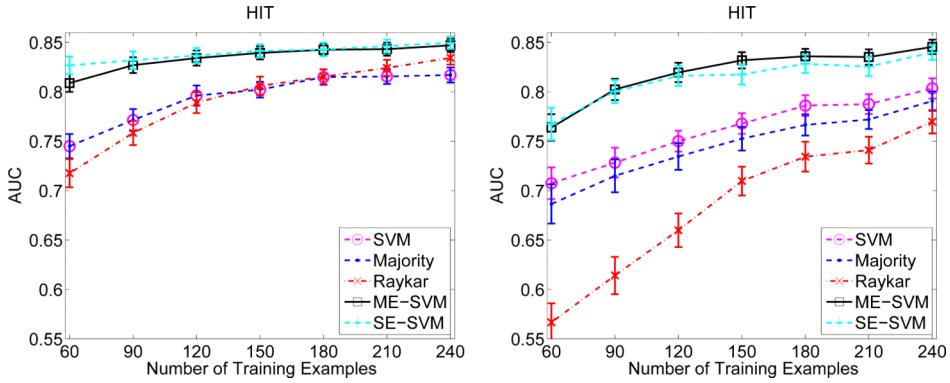

Figure 5.
Effect of the number of training examples on the quality of the model when: (Left) every example is labeled by just one expert; (Right) every example is labeled by all three experts

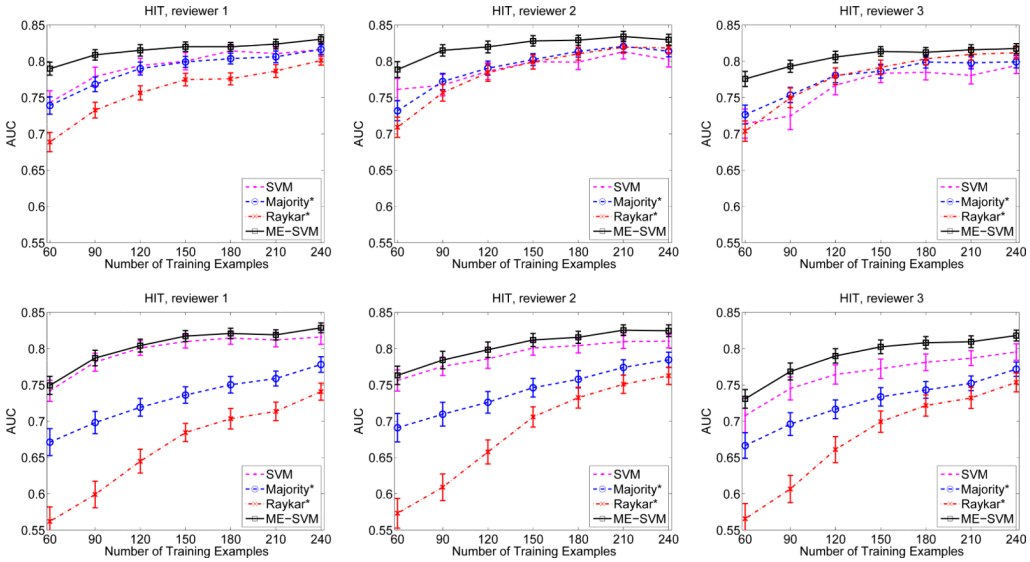

Figure 6.

Learning of expert-specific models. The figure shows the results for three expert specific models generated by the ME-SVM and the standard SVM methods, and compares them to models generated by the Majority* and Raykar* methods. First line: different examples are given to different experts; Second line: the same examples are given to all experts.

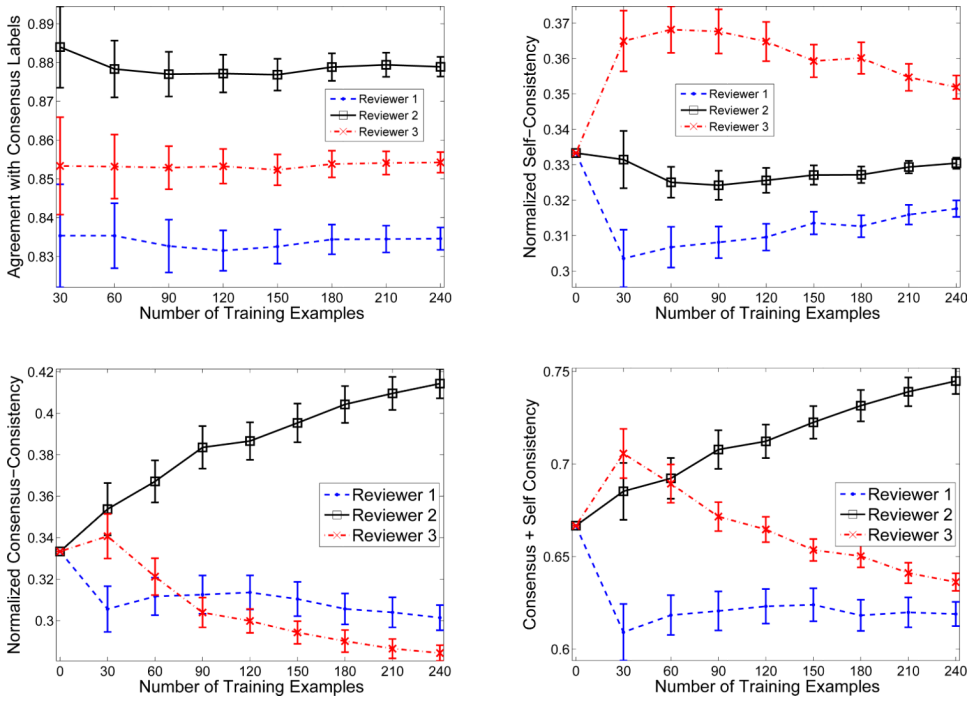

Figure 7.

(left-top) Agreement of experts with labels given by the senior expert; (right-top) Learned self-consistency parameters for Experts 1-3; (left-bottom) Learned consensus-consistency parameters for Experts 1-3; (right-bottom) Cumulative self and consensus consistencies for Expert 1-3