Learning from Inconsistent and Unreliable Annotators by a Gaussian Mixture Model and Bayesian Information Criterion

Ping Zhang and Zoran Obradovic

Center for Data Analytics and Biomedical Informatics, Temple University,
Philadelphia, PA 19122, USA
{ping,zoran.obradovic}@temple.edu

Abstract. Supervised learning from multiple annotators is an increasingly important problem in machine leaning and data mining. This paper develops a probabilistic approach to this problem when annotators are not only unreliable, but also have varying performance depending on the data. The proposed approach uses a Gaussian mixture model (GMM) and Bayesian information criterion (BIC) to find the fittest model to approximate the distribution of the instances. Then the maximum a posterior (MAP) estimation of the hidden true labels and the maximum-likelihood (ML) estimation of quality of multiple annotators are provided alternately. Experiments on emotional speech classification and CASP9 protein disorder prediction tasks show performance improvement of the proposed approach as compared to the majority voting baseline and a previous data-independent approach. Moreover, the approach also provides more accurate estimates of individual annotators performance for each Gaussian component, thus paving the way for understanding the behaviors of each annotator.

Keywords: multiple noisy experts, data-dependent experts, Gaussian mixture model, Bayesian information criterion.

1 Introduction

In supervised learning, it is usually assumed that true labels are readily available from a single annotator or source. However, recent advances in corroborative technology have given rise to situations where the true label of the target is unknown. In such problems, multiple sources or annotators are often available that provide noisy labels of the targets. For example, in the area of computer-aided diagnosis (CAD) the actual gold standard (whether the suspicious region is malignant or not) can only be obtained from a biopsy of the tissue. Since it is an expensive, invasive, and potentially dangerous process, often CAD systems are built from labels assigned by multiple radiologists who provide subjective and possibly noisy version of the gold standard. Very often there is a lot of disagreement among the labels. Another example is Amazon Mechanical Turk (AMT) [1] which allows the requesters to publish any Human

D. Gunopulos et al. (Eds.): ECML PKDD 2011, Part III, LNAI 6913, pp. 553–568, 2011.

Intelligence Tasks (HIT) on the website, such as writing essays, filling out certain questionnaires, or just collecting and labeling data. Any user of AMT can finish the tasks he is interested in and get paid. Therefore, acquiring non-expert labels is now easy, fast and inexpensive. On the other hand, since there is little control for the annotators, there is no guarantee for labeling quality: there could be careless, fallible, irresponsible or even malicious annotators.

In these multi-annotator problems, building a classifier in the traditional single annotator manner, without regard for the annotator properties may not be effective in general. The reasons for this include: some annotators may be more reliable than others, some may be malicious, some may be correlated with others, and in particular annotator effectiveness may vary depending on the data instance presented. In recent years, how to make the best use of the labeling information provided by multiple annotators to approximate the hidden true concept has drawn the attention of researchers in machine learning and data mining.

There has already been some literature for dealing with the multi-annotator setting. One popular strategy is to assign each sample to multiple annotators for labeling [2-6]. This repeated labeling strategy relies on the identification of what labels should be reacquired in order to improve classification performance or data quality. This form of active learning can be well suited when we can control assignments of samples to labelers. However, there are many cases that we have no access in doing so. Even we have, getting multiple labels for one sample could be a great waste of resources. As a result, research is conducted on the methods without using repeated labeling. These include techniques where labeler similarities are used to identify what samples should be used to estimate classification models for each labeler [7], and where low-quality annotators are pruned out by using the model trained from the entire dataset with all annotators as a ground truth [8]. Application areas for multi-annotator learning vary widely. These include natural language processing [9], computer-aided diagnosis [10, 11], computer vision [12, 13], speech technology [14] and bioinformatics [15, 16].

Among these papers, an elegant probabilistic framework of iteratively evaluating the different annotators and giving an estimate of the hidden true labels is developed [11]. However, the approach assumes the error rate of each annotator is consistent across all the input data. This is an impractical assumption in many cases since annotator knowledge can fluctuate considerably depending on the groups of input instances. For example, radiologists specialized in heart images will be better at labeling lesions of the heart compared to radiologists with lung expertise, who on the other hand would label instances of lung diseases better. In this paper, our proposed approach follows prior work [11] but relaxes the data-independent assumption, i.e., we assume an annotator may not be consistently accurate across the entire feature space. A very recent paper [17] also developed a data-dependent probabilistic model by assuming each annotator provides a Bernoulli noisy version or Gaussian distorted version of the true label. Compared to [17] our proposed approach first uses GMM and BIC to find a fittest model to approximate the distribution of the instances. Then, it alternately provides the maximum a posterior (MAP) estimation of the hidden true labels and the maximum-likelihood (ML) estimation of quality of multiple annotators at each mixture component.

The remaining part of this paper consists of the background on GMM and BIC, followed by a description of the approach, the summary of experimental results, conclusions and discussions of future work.

2 Background

In this section, we introduce the way of using GMM to approximate the distribution of the instances and using BIC to find the fittest GMM, and summarize the background information into Algorithm 1 and Algorithm 2. Our proposed approach (in Section 3) used these two algorithms to build a data-dependent probabilistic model of multiple noisy annotators.

2.1 The Gaussian Mixture Model

Given observations $\boldsymbol{x} = (\boldsymbol{x}_1, ..., \boldsymbol{x}_N)$, in a Gaussian mixture each component is modeled by a multivariate normal distribution. The parameters of component k comprise the mean vector μ_k, the covariance matrix Σ_k, and the probability density function

$$f_k(\boldsymbol{x}_i \mid \mu_k, \Sigma_k) = \frac{\exp\{-\frac{1}{2}(\boldsymbol{x}_i - \mu_k)^T \Sigma_k^{-1}(\boldsymbol{x}_i - \mu_k)\}}{(2\pi)^{d/2} \mid \Sigma_k \mid^{1/2}}.$$

Let K be the number of components in the mixture, and let π_k be mixing proportions: $0 < \pi_k < 1, \sum_{k=1}^{K} \pi_k = 1$. We wish to estimate the parameters $\theta = \pi_1, ..., \pi_K, \mu_1, ..., \mu_K, \Sigma_1, ..., \Sigma_K$. Then the log likelihood of the mixture is

$$L(\theta \mid \boldsymbol{x}_1, ..., \boldsymbol{x}_N) = \sum_{i=1}^{N} \ln\{\sum_{k=1}^{K} \pi_k f_k(\boldsymbol{x}_i \mid \mu_k, \Sigma_k)\}. \tag{1}$$

Here, Σ_k determines the geometric properties of component k. In [18] a general framework is proposed for exploiting the representation of the covariance matrix in terms of its eigenvalue decomposition $\Sigma_k = \lambda_k D_k A_k D_k^T$, where D$_k$ is the orthogonal matrix of eigenvectors, A$_k$ is a diagonal matrix whose elements are proportional to the eigenvalues of Σ_k, and λ_k is a scalar. The matrix D$_k$ determines the orientation of the component, A$_k$ determines its shape, and λ_k determines its volume. Allowing some but not all of the parameters in the decomposition to vary results in a set of models within this general framework. Such an approach is sufficiently flexible to accommodate data with widely varying characteristics. In this paper, by following the discussion of [19] we used 9 parameterizations which have a closed form update for the covariance matrix.

To estimate the parameters of the Gaussian mixture we used the Expectation-Maximization (EM) algorithm [20] which is introduced in Algorithm 1.

Algorithm 1 (EM for Gaussian mixtures).

Input: Observed data $\boldsymbol{x} = (\boldsymbol{x}_1,...,\boldsymbol{x}_N)$, the number of Gaussian components K, and the form of the covariance matrix.

Output: The model parameters $\theta = \pi_1,...,\pi_K, \mu_1,...,\mu_K, \Sigma_1,...,\Sigma_K$, the responsibilities τ_{ik} which are the probabilities that $\boldsymbol{x}_i$ is generated by component k, $i = 1,...,N, k = 1,...,K$

Step 1. Use K-means algorithm to initialize the means μ_k, covariances Σ_k and mixing coefficients π_k, and evaluate the initial value of the log likelihood by equation (1).

Step 2. (E-step) Evaluate the responsibility of the current model parameters, i.e. the probability that an observation $\boldsymbol{x}_i$ belongs to component k as

$$\tau_{ik} = \frac{\pi_k f_k(\boldsymbol{x}_i \mid \mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j f_j(\boldsymbol{x}_i \mid \mu_j, \Sigma_j)}$$

Step 3. (M-step) Re-estimate the parameters using the current estimated probability as follows

$$\pi_k^{new} = \frac{1}{N}\sum_{i=1}^{N} \tau_{ik}$$

$$\mu_k^{new} = \frac{1}{N}\sum_{i=1}^{N} \frac{\tau_{ik}\boldsymbol{x}_i}{\pi_k^{new}}$$

$$\Sigma_k^{new} = \frac{1}{N}\sum_{i=1}^{N} \frac{\tau_{ik}(\boldsymbol{x}_i - \mu_k^{new})(\boldsymbol{x}_i - \mu_k^{new})^T}{\pi_k^{new}}$$ [1]

Step 4. Evaluate the log likelihood by (1) using the updated parameters and check for convergence of either parameters or the log likelihood. If the convergence criterion is not satisfied return to Step 2.

2.2 Bayesian Model Selection

Each combination of a different specification of covariance matrices and a different number of components corresponds to a separate probability model. One advantage of the Gaussian mixture model is that it allows the use of approximate Bayes factors to compare models. This gives a systematic means of selecting not only the parameterization of the model, but also the number of components.

Let X be the observed data, M$_1$ and M$_2$ be two models with parameters θ_1 and θ_2 respectively. The integrated likelihood is defined as $p(X \mid M_g) =$

[1] The update for covariance matrix is only for the unconstrained case. See [19] for a complete description of the update equations for all 9 models.

$\int p(X \mid \theta_g, M_g) p(\theta_g \mid M_g) d\theta_g$ where $g = 1,2$ and $p(\theta_g \mid M_g)$ is the prior distribution of θ_g. The integrated likelihood represents the probability that data X is observed given that the underlying model is M$_g$. The Bayes factor is defined as the ratio of the integrated likelihoods of the two models, i.e. $B_{12} = P(X \mid M_1) / P(X \mid M_2)$. In other words, the Bayes factor B$_{12}$ represents the posterior odds that the data were distributed according to M$_1$ against model M$_2$ assuming that neither model is favored a priori. If $B_{12} > 1$, model M$_1$ is favored over M$_2$. The method can be generalized to more than two models. The main difficulty in using the Bayes factor is the evaluation of the integrated likelihood. By following the discussion of [21], we used an approximation called the Bayesian Information Criterion (BIC), given by

$$p(X \mid M_g) \approx BIC_g = 2\log(X \mid \widehat{\theta_g}, M_g) - m_g \log(N) \qquad (2)$$

where m$_g$ is the number of independent parameters that must be estimated for model M$_g$, and $\widehat{\theta_g}$ is the maximum-likelihood estimate for parameter θ_g. A large BIC score indicates strong evidence for the corresponding model. Hence, the BIC score can be used to compare models with different covariance matrix parameterizations and different numbers of components.

Here, we summarize the procedure of selecting the best Gaussian mixture model in Algorithm 2.

Algorithm 2 (Bayesian Model Selection for Gaussian mixtures).

Input: Observed data $\boldsymbol{x} = (\boldsymbol{x}_1, ..., \boldsymbol{x}_N)$.

Output: The optimal number of components, the optimal form of the component densities, and the corresponding model parameters and components responsibilities for each instance.

Step 1. Choose a form of model M from the 9 candidate models [19].

Step 2. Choose a number of components k. Here check from 1 to 6 (the maximum number of components).

Step 3. Use Algorithm 1 to obtain model parameters and log likelihood for this M and k.

Step 4. Calculate the value of BIC for this M and k by using equation (2).

Step 5. Go to Step 2 to choose another value of k.

Step 6. Go to Step 1 to choose another form of model M.

Step 7. Choose the optimal configuration (number of components and form of the covariance matrices) that corresponds to the highest BIC.

3 Method

Given a dataset $D = \{\boldsymbol{x}_i, y_i^1, ..., y_i^R\}_{i=1}^N$ containing N instances, where $\boldsymbol{x}_i$ is an instance (typically a d-dimensional feature vector), $y_i^j \in \{0,1\}$ is the corresponding binary label assigned to the instance $\boldsymbol{x}_i$ by the j-th annotator and R is the number of annotators.

Based on the intuition that real world annotators have different sensitivity and specificity for different regions of the entire feature space, we introduced a new data-dependent model in this paper. By using Algorithm 2, a fittest K-mixture-component GMM is used to approximate the distribution of the instances. To model the data-dependent behavior of annotators, we hypothesize that each annotator has its own sensitivity and specificity for each mixture component. The sensitivity α_k^j and specificity β_k^j are defined as follows:

$$\alpha_k^j = \Pr(y_i^j = 1 \mid y_i = 1, \text{k-th Gaussian mixture component generates } \boldsymbol{x}_i) \tag{3}$$

$$\beta_k^j = \Pr(y_i^j = 0 \mid y_i = 0, \text{k-th Gaussian mixture component generates } \boldsymbol{x}_i) \tag{4}$$

where $j = 1,...,R$; $k = 1,...,K$. We hypothesize that annotators generate labels as follows: given an instance $\boldsymbol{x}_i$ to label, the annotators find the mixture component which most possible generates that instance. Then the annotators generate labels with their sensitivities and specificities at that most possible component.

Our task is not only to get an estimation of the unknown true labels $y_1,..., y_N$, but also to estimate the sensitivity (i.e. true positive rate) $\boldsymbol{\alpha} = [\alpha_1^1,...,\alpha_k^j,...,\alpha_K^R]$ and the specificity (i.e. true negative rate) $\boldsymbol{\beta} = [\beta_1^1,...,\beta_k^j,...,\beta_K^R]$ of the R annotators at K Gaussian mixture components.

To fulfill the task defined before, we propose an iterative algorithm that we will call GMM-MAPML. Given dataset D , we use Algorithm 2 to get parameters of the fittest GMM and its mixture components' responsibilities for each instance. Also, we use majority voting to initialize the probabilistic labels z_i (i.e., the probability when the hidden true label is 1). Then, the algorithm alternately carries out the ML estimation and the MAP estimation which described in details in the following subsections. Given the current estimates of probabilistic labels z_i , the ML estimation measures annotators' performance (i.e., their sensitivity $\boldsymbol{\alpha}$ and specificity $\boldsymbol{\beta}$) at each mixture component and learns a classifier with parameter $\boldsymbol{w}$. Given the estimated sensitivity $\boldsymbol{\alpha}$, specificity $\boldsymbol{\beta}$, and the prior probability which is provided by the learned classifier, the MAP estimation gets the updated probabilistic labels z_i based on the Bayesian rule. After the two estimations converge, we get the algorithm outputs which include both the probabilistic labels z_i and the model parameters $\phi = \{\boldsymbol{w}, \boldsymbol{\alpha}, \boldsymbol{\beta}\}$.

3.1 ML Estimation of the Model Parameters

Given a dataset D and the current estimates of z_i , we estimate the model parameters $\phi = \{\boldsymbol{w}, \boldsymbol{\alpha}, \boldsymbol{\beta}\}$ by maximizing the conditional likelihood.

Denote $z_{ik} = z_i \tau_{ik}$, where τ_{ik} is the probability that $\boldsymbol{x}_i$ is generated by component k, according to (3) and (4) we can get the sensitivity of j-th annotator at k-th component and the specificity of j-th annotator at k-th component as

$$\alpha_k^j = \sum_{i=1}^N z_{ik} y_i^j \Big/ \sum_{i=1}^N z_{ik}$$
$$\beta_k^j = \sum_{i=1}^N (\tau_{ik} - z_{ik})(1 - y_i^j) \Big/ \sum_{i=1}^N (\tau_{ik} - z_{ik}) \tag{5}$$

Given probabilistic labels z_i, we can learn any classifier using ML estimation. However, in this section for convenience, we will explain it with a logistic regression classifier. By using that classifier, the probability for the positive class is modeled as a sigmoid acting on the linear discriminating function, that is,

$$\Pr[y = 1 \mid \boldsymbol{x}, \boldsymbol{w}] = \sigma(\boldsymbol{w}^T \boldsymbol{x}) \tag{6}$$

where the logistic sigmoid function is defined as $\sigma(x) = 1/(1+e^{-x})$. To estimate the classifier's parameter $\boldsymbol{w}$, we use a gradient descent method, that is, the Newton-Raphson method [22]

$$\boldsymbol{w}^{t+1} = \boldsymbol{w}^t - \eta \boldsymbol{H}^{-1} \boldsymbol{g} \tag{7}$$

where **g** is the gradient vector, **H** is the Hessian matrix, and η is the step length. The gradient vector is given by $\boldsymbol{g}(\boldsymbol{w}) = \sum_{i=1}^N [z_i - \sigma(\boldsymbol{w}^T \boldsymbol{x}_i)] \boldsymbol{x}_i$, and the Hessian matrix is given by $\boldsymbol{H}(\boldsymbol{w}) = -\sum_{i=1}^N [\sigma(\boldsymbol{w}^T \boldsymbol{x}_i)][1 - \sigma(\boldsymbol{w}^T \boldsymbol{x}_i)] \boldsymbol{x}_i \boldsymbol{x}_i^T$.

3.2 MAP Estimation of the Unknown True Labels

Given a dataset D and the model parameters $\phi = \{\boldsymbol{w}, \boldsymbol{\alpha}, \boldsymbol{\beta}\}$, the probabilistic labels are $z_i = \Pr[y_i = 1 \mid y_i^1, ..., y_i^R, \boldsymbol{x}_i, \phi]$. Using the Bayesian rule we have

$$z_i = \frac{\Pr[y_i^1, ..., y_i^R \mid y_i = 1, \phi] \cdot \Pr[y_i = 1 \mid \boldsymbol{x}_i, \phi]}{\Pr[y_i^1, ..., y_i^R \mid \phi]} \tag{8}$$

which is a MAP estimation problem.

Conditioning on the true label $y_i \in \{1, 0\}$, the denominator of formula (8) is decomposed as

$$\begin{aligned} &\Pr[y_i^1, ..., y_i^R \mid \phi] = \\ &\Pr[y_i^1, ..., y_i^R \mid y_i = 1, \boldsymbol{\alpha}] \Pr[y_i = 1 \mid \boldsymbol{x}_i, \boldsymbol{w}] \\ &+ \Pr[y_i^1, ..., y_i^M \mid y_i = 0, \boldsymbol{\beta}] \Pr[y_i = 0 \mid \boldsymbol{x}_i, \boldsymbol{w}] \end{aligned} \tag{9}$$

In our data-dependent model, given an instance $\boldsymbol{x}_i$ to label, the j-th annotator finds the q-th mixture component which most possible generates that instance. Then the annotator generates a label with the sensitivity α_q^j and specificity β_q^j. Therefore,

$$\Pr[y_i^1,...,y_i^R \mid y_i = 1, \boldsymbol{\alpha}] = \Pr[y_i^1,...,y_i^R \mid y_i = 1, \alpha_q^1,...,\alpha_q^R] \tag{10}$$

where $q = \arg\max_{k=1,...,K}(\tau_{ik})$. At each component, given the true label y_i we assume that $y_i^1,...,y_i^R$ are independent, that is, the annotators label the instances independently. Hence,

$$\begin{aligned}&\Pr[y_i^1,...,y_i^R \mid y_i = 1, \alpha_q^1,...,\alpha_q^R] = \prod_{j=1}^{R} \Pr[y_i^j \mid y_i = 1, \alpha_q^j]\\ &= \prod_{j=1}^{R} [\alpha_q^j]^{y_i^j} [1-\alpha_q^j]^{1-y_i^j}\end{aligned} \tag{11}$$

Similarly, we have

$$\Pr[y_i^1,...,y_i^R \mid y_i = 0, \boldsymbol{\beta}] = \prod_{j=1}^{R} [1-\beta_q^j]^{y_i^j} [\beta_q^j]^{1-y_i^j} \tag{12}$$

From (6), (8), (9), (10), (11) and (12), the posterior probability z_i which is a soft probabilistic estimate of the hidden true label is computed as

$$z_i = \frac{a_i p_i}{a_i p_i + b_i(1-p_i)} \tag{13}$$

where

$$\begin{aligned}p_i &= \Pr[y_i = 1 \mid \boldsymbol{x}_i, \boldsymbol{w}] = \sigma(\boldsymbol{w}^T \boldsymbol{x}_i)\\ a_i &= \prod_{j=1}^{R} [\alpha_q^j]^{y_i^j} [1-\alpha_q^j]^{1-y_i^j}\\ b_i &= \prod_{j=1}^{R} [1-\beta_q^j]^{y_i^j} [\beta_q^j]^{1-y_i^j}\\ q &= \arg\max_{k=1,...,K}(\tau_{ik})\end{aligned}$$

3.3 The GMM-MAPML Algorithm

We summarize the iterative approach in Algorithm 3.

Algorithm 3 (Iterative GMM-MAPML Algorithm).

Input: Dataset $D = \{\boldsymbol{x}_i, y_i^1,...,y_i^R\}_{i=1}^N$ containing N instances. Each instance has binary labels from R annotators.

Output: The fittest K-mixture-component GMM for the instances; the estimated sensitivity and specificity of each annotator at each mixture component; the weight parameter of a classifier; the probabilistic labels z_i; the estimation of the hidden true label y_i.

Step 1. Find the fittest K-mixture-component GMM for the instances, and get the corresponding GMM parameters and components responsibilities for each instance by Algorithm 2.

Step 2. Use majority voting to initialize $z_i = \sum_{j=1}^{R} y_i^j \Big/ R$.

Step 3. Iterative optimization.

(a) ML estimation – Estimate the model parameters $\phi = \{\boldsymbol{w}, \boldsymbol{\alpha}, \boldsymbol{\beta}\}$ based on current probabilistic labels z_i using (5) and (7).

(b) MAP estimation – Given the model parameters ϕ, update z_i using (13).

Step 4. If ϕ and z_i do not change between two successive iterations or the maximum number of iterations is reached, go to the Step 5; otherwise, go back to the Step 3.

Step 5. Estimate the hidden true label y_i by applying a threshold γ on z_i, that is, y_i =1 if $z_i > \gamma$ and y_i =0 otherwise.

3.4 Analysis of the Model

To explain how the model works, we apply the logit function to the posterior probability z_i. From equation (13), the logit of z_i is written as

$$\begin{aligned} \text{logit}(z_i) &= \ln\frac{z_i}{1-z_i} = \ln\frac{\Pr[y_i = 1 \mid y_i^1, ..., y_i^R, \boldsymbol{x}_i, \phi]}{\Pr[y_i = 0 \mid y_i^1, ..., y_i^R, \boldsymbol{x}_i, \phi]} \\ &= w^T \boldsymbol{x}_i + \sum_{j=1}^{R} y_i^j [\text{logit}(\alpha_q^j) + \text{logit}(\beta_q^j)] + c \end{aligned} \tag{14}$$

where $c = \sum_{j=1}^{R} \ln[(1-\alpha_q^j)\big/\beta_q^j]$, and $q = \arg\max_{k=1,...,K}(\tau_{ik})$. The first term of (14) is a linear combination (provided by the learned classifier) of features of instance $\boldsymbol{x}_i$. The second term of (14) is a weighted linear combination of the labels from all annotators. The weight of each annotator is the sum of the logit of the estimated sensitivity and specificity at the q-th component, i.e., the most possible component for generating $\boldsymbol{x}_i$. Therefore, our proposed model is data-dependent. Also, from equation (14) we can infer that the estimates of the hidden true labels depend both on observations and on the labels from all annotators.

4 Experimental Results

In this section we experimentally validate the proposed approach on two real-life datasets.

4.1 Emotional Speech Classification Experiment

Emotion recognition is an area which attracts interest from the speech research community. A wide area of applications such as interface optimization and expressive voice synthesis are related to the classification of speech into emotional states. In this experiment, we used a publicly available dataset from the EMA database [23]. This dataset has 3 speakers: a male native speaker read 14 sentences, and two female native speakers read 10 sentences. Each sentence was produced five times for four acted emotions, i.e., neutral, angry, sad, and happy. In this dataset, each utterance was evaluated by at least 3 expert listeners and 568 utterances were chosen as best emotion utterances. In our experiment, we used these 568 utterances as instances and their experts verified target emotions as the ground truth labels. Following the experiments in [14], {happy, neutral} were assigned to class 0 as positive emotion, and {sad, angry} were assigned to class 1 as negative emotion. To get the labels from multiple annotators, we sent the raw audio files (in WAV format) to 5 inexperienced listeners and asked them to provide binary labels (0 for positive emotion, 1 for negative emotion) for each instance. The 5 annotators have different academic backgrounds, and most of them are non-native speakers. By using VOICEBOX [24], we extracted 13 static features (12 MFCCs computed from 24 filter banks and log energy), 13 delta coefficients (first derivatives of static features) and 13 delta-delta coefficients (second derivatives of static features) from the speech signal over 25 ms frames with 10 ms overlap. The feature-wise mean is computed over the entire utterance, resulting in a 39-element feature vector for each instance.

In our comparisons, we considered three multiple-annotator methods: (1) Majority Voting that uses the average of annotators’ votes as the estimation of the hidden true label; (2) MAP-ML that estimates the hidden true labels and annotators' constant accuracy across all the input data using a data-independent model [11, 16]; and (3) GMM-MAPML that uses our proposed data-dependent model as described in Section 3. For further comparisons, we also learned two additional logistic regression classifiers: (4) LR Concatenation that concatenates all annotators’ labels as a training set, and (5) LR Ground Truth that uses the actual ground truth as a training set. For both LR Concatenation and LR Ground Truth, we randomly divided the whole dataset into five equally sized folds (20% of the dataset each). We repeated five times the logistic regression model training where we used four of the folds (80% of the dataset) for training and one fold for testing.

The ROC comparisons for three multiple-annotator methods and two additional logistic regression classifiers are shown in Fig. 1. The figure demonstrates the power of our proposed GMM-MAPML approach: GMM-MAPML significantly outperforms baseline methods (Majority Voting and MAP-ML) where information from all annotators is taken into account in a more naïve way. In addition, GMM-MAPML successfully approximates the LR Ground Truth classifier which is trained by the actual true labels. The Fig. 1 also shows that building a classifier in the traditional single annotator manner (simply concatenates all annotators’ labels as LR Concatenation) without regard for the annotator properties may not be effective for the multi-annotator problems.

As an output of our proposed GMM-MAPML method, a two-Gaussian-component model with an unconstrained covariance matrix has been selected for the emotional speech data. For each component, estimated sensitivity and specificity of 5 listeners using GMM-MAPML are shown in Table 1. The table shows that the 5 listeners have different sensitivity and specificity at the two components. Taking a closer look at the emotional speech data, we found that 75.35% of the utterances produced by the male speaker have the first Gaussian component as their principle component (i.e., the most possible component) and 64.31% of the utterances produced by the female speakers have the second Gaussian component as their principle component. It seems like some listeners (e.g., Listener 2 and Listener 4 in our experiment) are good at labeling one gender's utterance, but not at other gender's. The analysis shows that our proposed GMM-MAPML approach can be used for selecting the best annotators for instances at different components (or in different regions) of the feature space and for the training of annotators by informing them about the set of examples which they labeled unreliably.

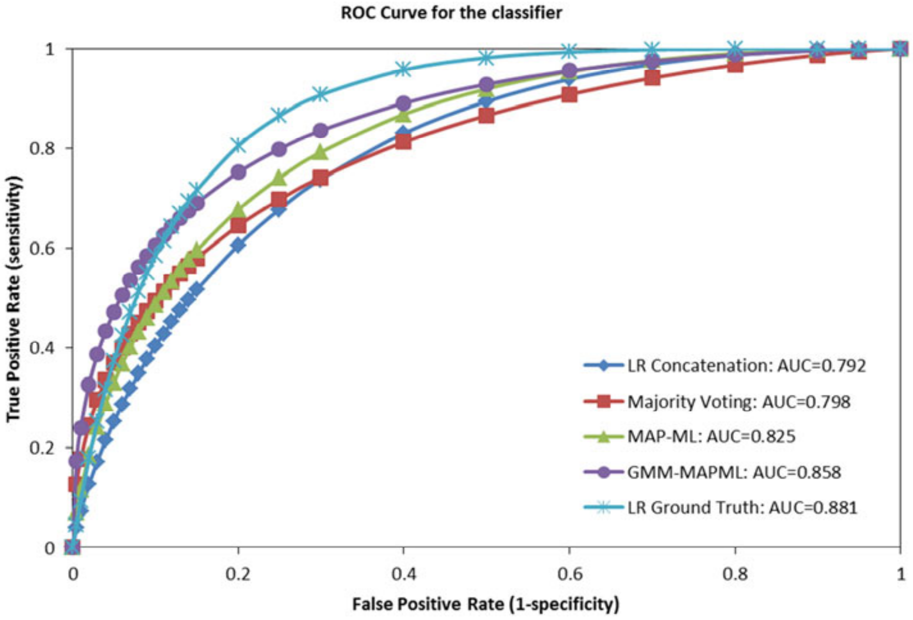

Fig. 1. The ROC comparisons for three multiple-annotator methods and two logistic regression classifiers on the emotional speech classification task. Methods are sorted in legend of the figure according to their AUC value.

Table 1. GMM-MAPML based estimates of 5 listeners' accuracy in emotional speech data (first and second component) without using golden ground truth

	First Component		Second Component	
Listeners	Estimated Sensitivity	Estimated Specificity	Estimated Sensitivity	Estimated Specificity
Listener 1	0.902	0.891	0.925	0.951
Listener 2	0.843	0.862	0.814	0.799
Listener 3	0.784	0.802	0.779	0.792
Listener 4	0.756	0.744	0.877	0.861
Listener 5	0.719	0.698	0.728	0.736

4.2 CASP9[2] Protein Disorder Prediction Experiment

Computational characterization of disorder in proteins is appealing due to the difficulties and high cost involved in experimental characterization of disorders. Treating an individual predictor as an annotator, the multiple-annotator methods can be used to build meta-predictors for protein disorder prediction. Recently, a data-independent model based on the idea of the MAP-ML method discussed in section 4.1 is used to integrate the prediction labels from multiple predictors [16]. This is shown to improve accuracy in performed experiments as compared to using individual component predictors. Following the experiments in [16], to characterize the method proposed in our study we used CASP9 data [25] consisting of 117 protein sequences with 26,083 amino-acid residues. For each residue, the golden ground truth (i.e. the residue is either in ordered state or in disordered state) was obtained by either X-ray or NMR experimental characterization. We have also obtained prediction labels (1 represents a disordered state while 0 represents an ordered state) with disorder probabilities (values in the range of 0–1) of all predictors which participated in CASP9 from the contest's official website [25]. We selected 15 predictors developed by groups at different institutions assuming that their errors are independent. By following the method proposed in [16], we extracted a 20-dimensional feature vector (19 amino acid composition features and 1 sequence complexity feature) for each residue.

In the experiment, as the input of our GMM-MAPML algorithm we used the 26,083 amino-acid instances and the prediction labels from the 15 individual predictors. After the algorithm had converged, we used the estimation of the hidden true labels y_i (given the threshold of 0.5) produced by GMM-MAPML as the binary disorder/order predictions and the probabilistic labels z_i from GMM-MAPML outputs as the disorder probability. As alternatives we also used the other two multiple-annotator methods, i.e., MAP-ML and Majority Voting to integrate the individual predictors, so that we can compare those methods with the GMM-MAPML to see which one is more effective. Following the regulation of CASPs, performance of the methods was evaluated by three criteria [26]: (1) the average of sensitivity and specificity (ACC), (2) a weighted score (S_w) that considers the rates of ordered and disordered residues in the data; (3) and the area under the ROC curve (AUC).

Comparisons of 15 individual predictors, the MAP-ML method, the Majority Voting method, and our GMM-MAPML method on CASP9 data is shown in Table 2. Our proposed GMM-MAPML method significantly outperforms the two baseline methods (Majority Voting and MAP-ML) and each individual predictor in all three criteria.

Using the BIC, our GMM-MAPML method also finds that the fittest GMM for CASP9 data is three Gaussian components with the covariance matrix in the form of $\lambda_k \boldsymbol{B}_k$ (**B** is a diagonal matrix). For each component, estimated sensitivity and specificity of 15 individual predictors using GMM-MAPML without relying on golden ground truth are shown in Fig. 2. The obtained estimates are sorted according to the average of their estimated sensitivity and specificity. The Fig. 2 clearly shows that the individual CASP9 disorder predictors have different sensitivity and specificity at

[2] CASP9 is the abbreviation of the 9th Biannual Community Wide Experiment on the Critical Assessment of Techniques for Protein Structure Prediction held in year 2010.

different components. The figure also demonstrates that the rankings of individual predictors are different at different components.

For further analysis, the relationship between amino-acid residue positions and the three Gaussian components are shown in Fig. 3. We found that the principal (most likely) component at the N-terminus (defined here as 20% of residues at the start of a protein sequence) was the first Gaussian component. In particular, about 56% of the amino-acid residues from this region belong to this component. The principal component for the C-terminus consisting of 20% of residues at the end of each protein was the third component (59% of residues from this region belong to this component). The internal 60% of residues were most likely to belong to the second Gaussian component (54% of these residues belong to this component). The results well agree with previous protein disorder work where amino-acid residues at different regions (i.e., N-terminal, C-terminal and internal) have different compositions and different tendencies for disorder [27].

The experiment on CASP9 data shows that our proposed GMM-MAPML method can potentially be used to improve prediction of protein disorder and to provide helpful suggestion on choosing the suitable disorder predictors for each region of unknown protein sequences.

Table 2. Comparisons of GMM-MAPML vs. alternative protein disorder meta-predictors (MAP-ML and MAJORITY VOTING) and individual CASP9 predictors according to CASP9 evaluation measures using CASP9 data

Predictor Name	ACC	S_w	AUC
GMM-MAPML	0.785	0.527	0.874
MAP-ML	0.764	0.513	0.859
MAJORITY VOTING	0.735	0.496	0.776
PRDOS2	0.754	0.509	0.855
MULTICOM-REFINE	0.750	0.500	0.822
BIOMINE_DR_PDB	0.741	0.483	0.821
GSMETADISORDERMD	0.738	0.476	0.816
MASON	0.736	0.473	0.743
ZHOU-SPINE-D	0.731	0.462	0.832
DISTILL-PUNCH1	0.726	0.453	0.800
OND-CRF	0.706	0.412	0.737
UNITED3D	0.704	0.412	0.781
CBRC_POODLE	0.694	0.405	0.830
MCGUFFIN	0.688	0.402	0.817
ISUNSTRUCT	0.679	0.396	0.742
DISOPRED3C	0.670	0.391	0.853
ULG-GIGA	0.585	0.341	0.726
MEDOR	0.579	0.338	0.688

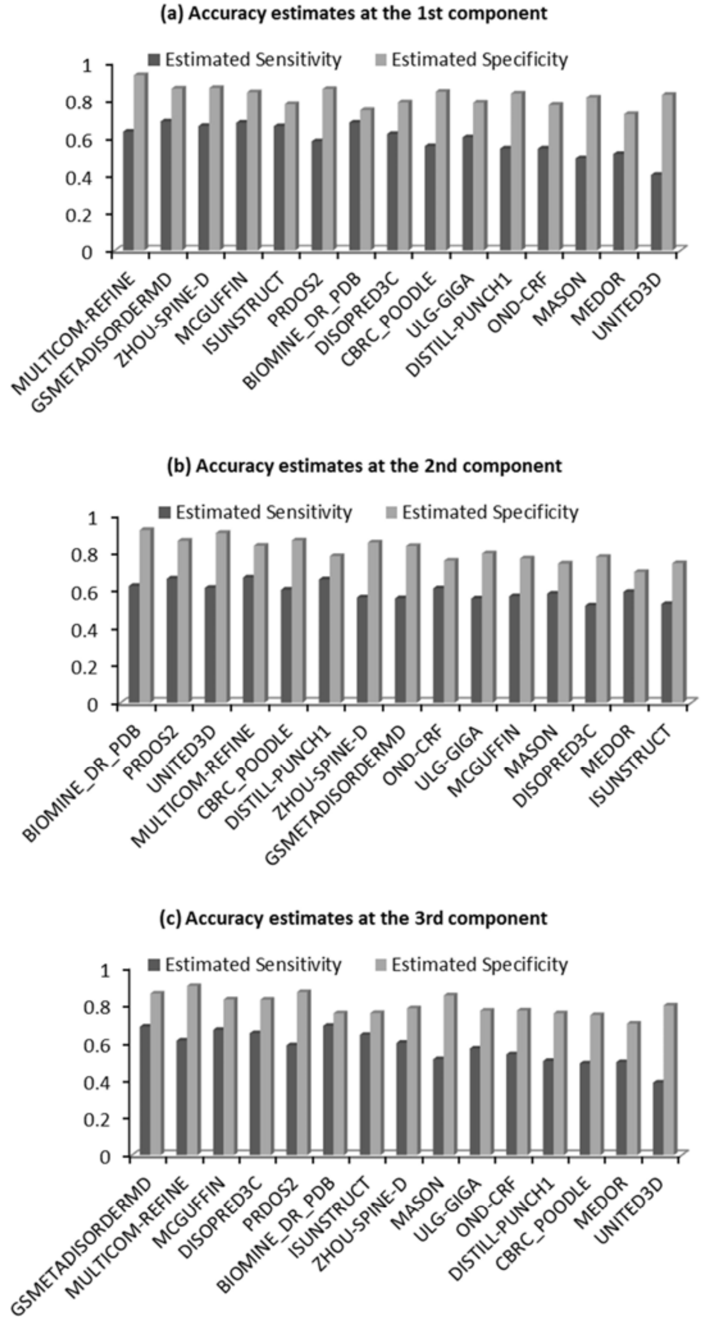

Fig. 2. Analysis of CASP9 disorder predictors at three principal components of CASP9 data identified by GMM-MAPML. In panels a, b, and c the predictors are sorted in descending order of the average of the estimated sensitivity and specificity on the corresponding component of CASP9 data.

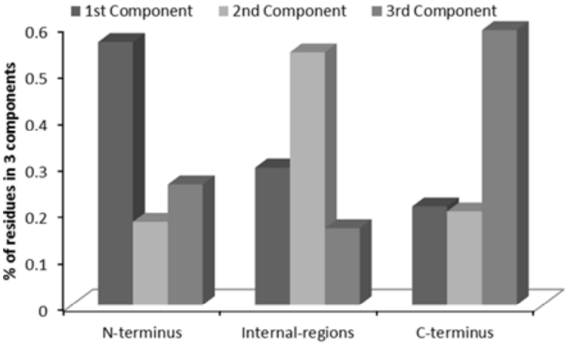

Fig. 3. Distribution of residues at N-terminus, internal-regions and at C-terminus with respect to three principal components identified by GMM-MAPML

5 Conclusion

In this paper we proposed a data-dependent probabilistic model for classification when given labels obtained by multiple noisy annotators but without any gold standard annotation. The proposed GMM-MAPML method uses a Gaussian mixture model (GMM) and Bayesian information criterion (BIC) to find the fittest model to approximate the distribution of the instances. Then the maximum a posterior (MAP) estimation of the hidden true labels and the maximum-likelihood (ML) estimation of quality of multiple annotators at each Gaussian component are provided alternately. Emotional speech classification and CASP9 protein disorder prediction experiments show a significant performance improvement of the proposed GMM-MAPML method as compared to the majority voting baseline and a previous data-independent method (MAP-ML). Moreover, GMM-MAPML also provides more accurate estimates of individual annotator performance for each Gaussian component, which can be used for active learning, feedback, and annotator selection.

The proposed method assumed that the annotators make their errors independently. We emphasize that in practice the independence assumption might not be always true which is the limitation of the proposed algorithm. To relax the independence assumption and to develop a more realistic model for the multiple-annotator problems, our research in progress includes additional parameters such as the degree of correlation among the annotators.

Acknowledgments. This work is funded in part by a grant with the Pennsylvania Department of Health. They specifically disclaim responsibility for any analyses, interpretations, or conclusions.

References

1. Amazon Mechanical Turk, `http://www.mturk.com`
2. Smyth, P., Fayyad, U.M., Burl, M.C., Perona, P., Baldi, P.: Inferring ground truth from subjective labelling of venus images. In: NIPS, pp. 1085–1092 (1994)
3. Jin, R., Ghahramani, Z.: Learning with multiple labels. In: NIPS, pp. 897–904 (2002)

4. Sheng, V.S., Provost, F.J., Ipeirotis, P.G.: Get another label? Improving data quality and data mining using multiple, noisy labelers. In: KDD, pp. 614–622 (2008)
5. Donmez, P., Carbonell, J.G.: Proactive learning: cost-sensitive active learning with multiple imperfect oracles. In: CIKM, pp. 619–628 (2008)
6. Donmez, P., Carbonell, J.G., Schneider, J.: Efficiently learning the accuracy of labeling sources for selective sampling. In: KDD, pp. 259–268 (2009)
7. Crammer, K., Kearns, M., Wortman, J.: Learning from multiple sources. Journal of Machine Learning Research 9, 1757–1774 (2008)
8. Dekel, O., Shamir, O.: Vox populi: Collecting high-quality labels from a crowd. In: COLT (2009)
9. Snow, R., O'Connor, B., Jurafsky, D., Ng, A.Y.: Cheap and fast - but is it good? Evaluating non-expert annotations for natural language tasks. In: EMNLP, pp. 254–263 (2008)
10. Cholleti, S.R., Goldman, S.A., Blum, A., Politte, D.G., Don, S., Smith, K., Prior, F.: Veritas: combining expert opinions without labeled data. International Journal on Artificial Intelligence Tools 18, 633–651 (2009)
11. Raykar, V.C., Yu, S., Zhao, L.H., Jerebko, A.K., Florin, C., Valadez, G.H., Bogoni, L., Moy, L.: Supervised learning from multiple experts: whom to trust when everyone lies a bit. In: ICML, pp. 889–896 (2009)
12. Whitehill J., Ruvolo P., Wu T., Bergsma J., Movellan J.: Whose vote should count more: optimal integration of labels from labelers of unknown expertise. In NIPS (2009)
13. Welinder P., Branson S., Belongie S., Perona P.: The multidimensional wisdom of crowds. In: NIPS (2010)
14. Audhkhasi K., Narayanan S.: Data-dependent evaluator modeling and its application to emotional valence classification from speech. In: InterSpeech, pp. 2366–2369 (2010)
15. Rzhetsky, A., Shatkay, H., Wilbur, W.J.: How to get the most out of your curation effort. PLoS. Comput. Biol. 5(5), e1000391 (2009)
16. Zhang, P., Obradovic, Z.: Unsupervised integration of multiple protein disorder predictors. In: IEEE Int'l. Conf. Bioinformatics and Biomedicine, pp. 49–52 (2010)
17. Yan, Y., Rosales, R., Fung, G., Schmidt, M., Hermosillo, G., Bogoni, L., Moy, L., Dy, J.G.: Modeling annotator expertise: learning when everybody knows a bit of something. Journal of Machine Learning Research - Proceedings Track 9, 932–939 (2010)
18. Banfield, J.D., Raftery, A.E.: Model-based Gaussian and non-Gaussian clustering. Biometrics 49, 803–821 (1993)
19. Martinez, W.L., Martinez, A.R.: Exploratory data analysis with MATLAB, pp. 163–195. Chapman & Hall/CRC, Boca Raton (2004)
20. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistical Society: Series B 39(1), 1–38 (1977)
21. Fraley, C., Raftery, A.E.: How many clusters? Which clustering method? Answers via model-based cluster analysis. Comput. J., 578–588 (1998)
22. Bishop, C.: Pattern recognition and machine learning, pp. 203–213. Springer, New York (2006)
23. Lee, S., Yildirim, S., Kazemzadeh, A., Narayanan, S.: An articulatory study of emotional speech production. In: Eurospeech, pp. 497–500 (2005)
24. VOICEBOX, `http://www.ee.imperial.ac.uk/hp/staff/dmb/voicebox/voicebox.html`
25. CASP experiments, `http://predictioncenter.org/`
26. Noivirt-Brik, O., Prilusky, J., Sussman, J.L.: Assessment of disorder predictions in CASP8. Proteins 77(suppl. 9), 210–216 (2009)
27. Uversky, V.N., Dunker, A.K.: Understanding protein non-folding. Biochim. Biophys. Acta 1804, 1231–1264 (2010)