Active Learning of Multi-class Classification Models from Ordered Class Sets

Yanbing Xue
Department of Computer Science
University of Pittsburgh
yax14@pitt.edu

Milos Hauskrecht
Department of Computer Science
University of Pittsburgh
milos@pitt.edu

Abstract

In this paper, we study the problem of learning multi-class classification models from a limited set of labeled examples obtained from human annotator. We propose a new machine learning framework that learns multi-class classification models from ordered class sets the annotator may use to express not only her top class choice but also other competing classes still under consideration. Such ordered sets of competing classes are common, for example, in various diagnostic tasks. In this paper, we first develop strategies for learning multi-class classification models from examples associated with ordered class set information. After that we develop an active learning strategy that considers such a feedback. We evaluate the benefit of the framework on multiple datasets. We show that class-order feedback and active learning can reduce the annotation cost both individually and jointly.

Introduction

In recent years, the world has witnessed remarkable increase in the number and quality of classification models built from data. One important factor contributing to this progress and improvement is the amount of labeled data instances available to train these models. However, this improvement may not be possible when the original data are unlabeled and when the labels are obtained through additional human annotation effort. To alleviate this problem, we study various ways of reducing the annotation effort, while keeping the quality of the classification models built from data high. Our specific focus in this work is on construction of multi-class classification models.

Multi-class classification models are typically learned from annotated data in which every data instance is associated with one class label indicating the top class choice assigned to it by a human annotator. However, human annotators can often express and provide additional information about the top class and its relation to other class choices. For example, when the instance is not a clearcut case, there are other likely class choices the annotator may have in mind. Associating multiple competing classes with one instance is common in various diagnostic tasks. For example, in medical domain, a list of competing diagnostic classes is referred to as a differential diagnosis. Briefly, given the features (symptoms, observations, etc.) of a patient, the physician considers not only the leading diagnosis (class), but also other alternative diagnoses (classes) that are possible and may fit the patient's case. The gist of our approach is to utilize such information to learn multi-class classifiers. More specifically, apart from the top class label for each data instance, we let the annotator provide also information about other alternative classes, and express these in terms of the ordered set of classes representing the descending priorities (or confidence) in these classes.

To translate this idea into a working framework we first develop and present a new multi-class support vector classifier method that lets us incorporate the ordered class set (OCS) information into the model learning process. Since data instances may not be initially labeled, we then explore active learning strategies to further reduce the annotation effort. Briefly, active learning (Lewis and Gale 1994; Settles 2010; Roy and McCallum 2001) helps us reduce the number of examples necessary to train a high quality classifier by repeatedly identifying and annotating the data instance with the greatest potential to improve the quality of the classification model. The effectiveness of active learning is, however, highly dependent on the instance-selection strategy, and such a strategy is highly related to the format of the labels in the data. Hence, the instance-selection strategy for ordered class set feedback may not be a straightforward modification of strategies for a single class feedback. In this paper, we develop a new active learning strategy that considers ordered class set feedback. Specifically, our active learning strategy calculates the expected prediction change for an unlabeled instance by calculating and combining the estimate of the prediction change for the different class-order sets one can assign to the instance. Since the estimate of the expected change requires one to repeat the retraining by considering each unlabeled instance, we propose new approximation strategies that can reduce the running time of the estimate.

We experiment with our new framework on both synthetic and real-world datasets with class-order feedback. We show the effectiveness of the ordered class set feedback and active learning for reducing the annotation effort both individually and jointly. We also show that our solution outperforms existing multi-class classification methods that consider one-class-per-example labels.

Related Work

In this section, we briefly review literature related to our work. We divide it to three different topics: learning with auxiliary information, multi-class support vector machine, and active learning.

Learning with auxiliary information

Learning with auxiliary class information is a relatively new approach for improving classification learning process. In general, auxiliary class information covers additional information provided by a human annotator related to the class choice. The idea of learning with auxiliary class information is based on a simple premise: auxiliary class information can often be provided by human annotators at an insignificant cost when compared to the cost of instance review and label assessment. Perhaps the most intuitive format is a probabilistic score, which has been explored in context of binary classification problems. The probabilistic score indicates the confidence with which the annotator believes the class label is true. The problem of learning classification models from auxiliary probabilistic scores was formulated first by (Nguyen, Valizadegan, and Hauskrecht 2011a; 2011b; 2013). The authors developed a method that focuses on the pairwise orderings among all data examples, which is robust against the noise in human's probabilistic score estimate (Juslin, Olsson, and Winman 1998; Griffin and Tversky 1992; O'Hagan et al. 2007). This method learns a parametric model trying to satisfy the constraints from pairwise orderings among all data examples. The approach has been tested on data based on electronic health records (Hauskrecht et al. 2013; 2016) and lead to improved sample efficiency compared to binary class feedback. A probabilistic score obtained by averaging labels from multiple annotators (with potential disagreements) was explored by (Thiel 2008). This work, however, does not consider any noise. More recent approaches for building classification models from probabilistic label feedback include a non-parametric approach based on the Gaussian process regression by (Peng and Wong 2014; Peng, Wong, and Yu 2014) and an ordinal regression approach with a reduced set of constraints by (Xue and Hauskrecht 2017b).

All of the above works show that additional label-related information can improve model learning in binary classification settings. This raises the following questions: are there other formats of auxiliary class-related information one can use for training the models, and, is it possible to use such an information in multi-class classification problems? In this work, we investigate auxiliary class-related information expressed in terms of the ordered class set (OCS), which makes sense in multi-class classification scenario. Basically, an OCS defines an ordered subset of classes that represent choices that are likely (considered) for labeling the instance and their priority. An OCS may vary in size and includes classes that are considered to be viable class alternatives. Classes not in the OCS are considered to be unimportant or negligible. For example, in a four-class scenario, the OCS $\langle 3, 4\rangle$ indicates that the annotator believes class 3 to be the most likely and class 4 to be the second most likely choice, while other two classes 1 and 2, are unimportant. The problem of learning multi-class classification models from OCS is a new open problem. In this paper, we first propose a multi-class classifier that learns from OCS in addition to class labels. That is, each data instance is associated with an OCS likely classes in descending order. In the experiments section we show that our class-order multi-class classifier substantially reduces the number of instances one needs to label when compared to existing multi-class classifiers based only on the top class label.

Multi-class support vector machine

Multi-class support vector machine (MCSVM) was first proposed by (Vapnik 1998; Weston et al. 1999). Basically, for a K-classification problem, MCSVM trains K one-vs-rest binary classifiers in *one* optimization problem that consist of the sum of the regularization term and the penalty for slack variables of all the K one-vs-rest classifiers is minimized jointly. Compared with previous methods, one-vs-rest, that trains K one-vs-rest classifiers independently and, one-vs-one, that trains $\frac{K(K-1)}{2}$ one-vs-one classifiers *independently*, MCSVM achieves higher performance especially when the labeled data are limited. Recently, (He et al. 2012) proposed an approximate multi-class support vector machine (AMSVM) approach that reduces the number of constraints one has to satisfy via averaging. Compared with the original MCSVM, AMSVM uses an approximation that relaxes the constraints by only enforcing the comparison on projections between the labeled class and the average of all other classes. With such an approximation, AMSVM significantly reduces the number of constraints it needs to solve while still reaching performance comparable to MCSVM.

In this work, we first show how to adapt AMSVM to incorporate OCS. Then, we design and implement an active learning strategy that is compatible with the new AMSVM with OCS.

Active learning

In active learning, model training and data instance annotation process are interleaved. Active learning sequentially selects and labels originally unlabeled instances that are most informative and believed to have the greatest potential to improve the model. There are multiple ways to assess the "informativeness" of an unlabeled instance. Perhaps the most popular strategy is *uncertainty sampling* (Lewis and Gale 1994). In multi-class classification scenarios, three different standards are applied to measure uncertainty: (1) lowest confidence, that queries the unlabeled instance with lowest maximum in predictions over all classes, and (2) marginal confidence, that queries the unlabeled instance with lowest discrepancy in its top two class predictions, and (3) information entropy, that queries the unlabeled instance with highest information entropy over predictions of all classes. However, uncertainty sampling is incompatible with class-order information as the ordering of classes indirectly reflects the uncertainty associated with all probable classes. Another popular strategy is *query-by-committee* (Seung, Opper, and Sompolinsky 1992) that trains a committee of models and selects the unlabeled instance on which the models disagree the most. The models in the committee can be acquired from different training sets via, for example, bootstrapping all data instances (Breiman 1996). The limitation of query-by-committee is a potential bias introduced by the trained models.

Other more sophisticated querying strategies estimate the expected change in the model the specific query may lead to. Briefly, the strategy calculates the change in the model due to an unlabeled instance being assigned to one of the possible labels and weights the change by an estimate of its probability. The first expectation-based querying strategy is *expected model change* (EMC) (Tong and Koller 2000; Settles, Craven, and Ray 2008). The model change is measured in terms of the change of the model parameters. However, a big change of the parameters does not necessarily imply a big change in model's predictions. Therefore, this strategy typically overestimates the "informativeness" of each unlabeled instance. The *expected error reduction* (Roy and McCallum 2001) measures the change based on the generalization error when an unlabeled instance is assumed labeled. More recently, *expected performance change* (EPC) (Xue and Hauskrecht 2017a) has been proposed for binary classifiers with auxiliary Likert-scale information. Such a strategy measures the change in the Likert-scale prediction due to labeling.

Recent active learning work focuses on more sophisticated querying strategies that go beyond standard instance-based label-oriented queries. For example, *active group learning* (AGL) (Luo and Hauskrecht 2018a; 2018b) constructs queries for subpopulations (groups of examples) the human annotator labels with class proportions. The advantage of the approach is that multiple instances are labeled jointly with just one query. Another approach is *structural query-by-committee* (SQBC) (Tosh and Dasgupta 2018). It is a generalization of the query-by-committee (QBC) strategy (Seung, Opper, and Sompolinsky 1992), that attempts to learn the best structure defined on some space $\mathcal{X}$ by constructing queries that represent a snapshot of the most uncertain part of the structure that is then either confirmed or corrected via human feedback.

In this work, we propose to use the OCS to improve the learning. An open question is how to combine it with active learning. To address the problem, we propose a new active learning strategy based on the expected model change (EMC) that is compatible with the multi-class classifiers with OCS. Briefly, when adding an unlabeled instance and a possible OCS, it calculates the change in the ordering induced by all one-vs-rest classifiers over all unlabeled instances. We propose several techniques and approximations to accelerate the retraining of the models and to reduce the number of ordered class set assignments. In experiments, we show our active learning strategy can substantially reduce the number of examples it needs to query.

Methodology

In this part, we develop an active learning framework that builds a multi-class classification model by actively querying an annotator who provides feedback to the framework by assessing instances with OCS. We start by first defining and formalizing the problem of learning from OCS in a multi-class settings. After that, we present an algorithm for learning the multi-class classification model from such feedback. Second, we show how this algorithm can be included in the active learning framework that aims to improve the model by wisely selecting the examples to be assessed next. The criterion used to choose from among unlabeled candidate instances is based on the highest expected change in OCS. Since the calculation of the expected change in OCS is non-trivial, we present our solutions to the following problems: (1) how to model the distribution of OCS for calculating the expected change, and (2) how to speed up training via incremental solver when adding one unlabeled instance and an OCS.

Multi-class classifier with ordered class sets (OCS)

Problem Our objective is to learn a multi-class classifier $f : X \rightarrow Y$, where $X \in \mathbb{R}^d$ is the input space and $Y = \{1, 2, \ldots, K\}$ represents possible (mutually exclusive) classes one can assign to an example. Standard way to learn such a model is to use input-output pairs $\langle \mathbf{x}_i, y_i \rangle$. In this work we learn from the input-OCS pairs $\langle \mathbf{x}_i, S_i \rangle$, where the input $\mathbf{x}_i$ is associated with the ordered class set (or OCS) S_i reflecting the annotator's class preferences. The ordered class set S_i is formed by a non-empty subset of classes defining Y. Please note that the information in the input-OCS pair subsumes the information provided in the standard input-output data format. Briefly, we assume $y_i = S_{i1}$, that is, the class label y_i is identical to the first class in ordered class set S_i. In general S_i may contain any number of classes: an ordered set of only one class only indicates the annotator's top class choice; an ordered set of all K classes indicates the annotator provides the complete ordering of all alternative classes. For example, in a 4-class setting, an OCS $\langle 3, 2 \rangle$ indicates this data instance most probably belongs to class 3, then class 2 and is not likely to belong to any other class. Since the class label is identical to the first class in the OCS, the output (class label) of this instance should be 3.

Approximate multi-class SVM (AMSVM) To learn a multi-class classifier for instances with OCS, we build upon the approximate multi-class SVM method (AMSVM) proposed by (He et al. 2012). The AMSVM is an approximation of the standard multi-class SVM (MCSVM) method. Briefly, MCSVM works by trying to assure for every training data instance the projection of its assigned class label is higher than the projection of any other class. Therefore, $(K - 1)$ constraints are derived for each labeled data instance, one for each class, except for the assigned class label. The total number of constraints in MCSVM is thus $O(KN)$, where N is the number of labeled data instances. In AMSVM the set of the constraints is merged and replaced with one constraint that assumes that for each data instance the projection of the class label is higher than the average projection for all other classes. Via such averaging, the number of constraints is significantly reduced: only one constraint is derived for each labeled data instance. Therefore, the total number of constraints in AMSVM is reduced to $O(N)$. Formally, in the AMSVM with k classes, k binary SVMs $f_1(\cdot), f_2(\cdot), \ldots, f_k(\cdot)$ are trained jointly. For every labeled instance $\langle \mathbf{x}_i, y_i \rangle$, we try to assure the projection $f_{y_i}(\mathbf{x}_i)$ of the class label y_i should be greater than the average projection $\frac{1}{k-1}\sum_{j \neq y_i} f_j(\mathbf{x}_i)$. The optimization of AMSVM can be formalized as:

$$\min_{W,\Xi} \frac{1}{2}\sum_{l=1}^{k} \mathbf{w}_l^T \mathbf{w}_l + C\sum_{i=1}^{N} \xi_i$$
$$(\mathbf{w}_{y_i} - \frac{1}{k-1}\sum_{j \neq y_i} \mathbf{w}_j)^T \phi(\mathbf{x}_i) \geq 1 - \xi_i$$
$$\xi_i \geq 0 \qquad\qquad \forall i; \tag{1}$$

where y_i is the class label of $\mathbf{x}_i$ and $\phi(\cdot)$ is the projection of kernel space. $W = \{\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_k\}$ are parameters of the k binary one-vs-rest classifiers. N is the number of labeled instances. $\Xi = \{\xi_1, \xi_2, \ldots, \xi_N\}$ are the slack variables for each constraint. For prediction, the class with the highest projection value is selected as the predicted class. As shown in (He et al. 2012) the performance of AMSVM is often comparable to the standard multi-class SVM (MCSVM).

AMSVM with ordered class sets (OCS) Next we show how we can combine AMSVM with ordered class set (OCS). One straight forward solution to incorporate OCS to the framework is to enforce the pairwise ordering for each pair of classes for each data instance; that is, the ordering of a pair of classes should conform to their ranking in the OCS. However, such intuition suffers from high time complexity: $\frac{K(K-1)}{2}$ constraints are derived for each labeled data instance, one for each pair of classes. Therefore, the total number of constraints of this intuition is thus $O(K^2N)$. To reduce the time complexity, we instead incorporate the OCS via constraints derived from the ordinal regression (Chu and Keerthi 2005). The gist of the approach is that, for every data instance, we split the classes in its OCS into two subsets: a "higher" subset and a ”lower" subset. Each class in the “lower” subset must satisfy one of the two conditions: (1) it is not included in the OCS, or (2) if in the OCS, it comes after all the classes from the higher subset. In other words, each class in the “higher” subset should have higher priority than all the classes from the “lower” subset in their projections. If such condition is guaranteed, we may enforce that the average projection of the “higher” subset is higher than the average projection of the “lower” subset. Since there are at most $(K-1)$ splits of “higher” and “lower” subsets for each labeled instance and each split corresponds to one constraint, the total number of constraints is reduced to $O(KN)$. Formally, for every labeled instance $\langle \mathbf{x}_i, S_i \rangle$ and $j \in \{1, 2, \ldots, |S_i|\}$, the “higher” subset can be constructed as $\{S_{i1}, S_{i2}, \ldots, S_{ij}\}$, where S_{ij} indicates the jth class in S_i, and the “lower” subset consists of all other classes. Then the goal is to try to enforce the average projection $\frac{1}{j}\sum_{a \in \{S_{i1}, S_{i2}, \ldots, S_{ij}\}} f_a(\mathbf{x}_i)$ of the “higher” subset should be greater than the average projection $\frac{1}{k-j}\sum_{b \notin \{S_{i1}, S_{i2}, \ldots, S_{ij}\}} f_b(\mathbf{x}_i)$ of the “lower” subset. Therefore, the optimization of AMSVM with OCS can be formulated as:

$$\min_{W,\Xi} \frac{1}{2}\sum_{l=1}^{k} \mathbf{w}_l^T \mathbf{w}_l + C\sum_{i=1}^{N}\sum_{j=1}^{|S_i|} \xi_{ij}$$
$$(\frac{1}{j}\sum_{a \in \{S_{i1}, \ldots, S_{ij}\}} \mathbf{w}_a - \frac{1}{k-j}\sum_{b \notin \{S_{i1}, \ldots, S_{ij}\}} \mathbf{w}_b)^T \phi(\mathbf{x}_i) \geq 1 - \xi_{ij}$$
$$\xi_{ij} \geq 0 \qquad\qquad \forall ij; \tag{2}$$

where S_i is the OCS of $\mathbf{x}_i$ and $\phi(\cdot)$ is the projection of kernel space. $W = \{\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_k\}$ are parameters of the k binary one-vs-rest classifiers. N is the number of labeled instances. $\Xi = \{\xi_{ij}\}$ for all $i = 1, 2, \ldots, N$ and $j = 1, 2, \ldots, |S_i|$ index the slack variables for each constraint. For prediction, the class with the highest projection value is selected as the predicted class. Please notice the prediction from our AMSVM with OCS is still a class label.

Active learning with OCS

The next challenge is to embed the above multi-class classifier with OCS in a compatible active learning strategy. The core of any active learning strategy is a schema to select examples to be queried next. In this work, we propose and experiment with a strategy called expected model change that measures the potential of an unlabeled data instance to change the model by estimating its impact on predictions. In this section, we first show how the expected model change of an unlabeled instance can be calculated by considering all OCS of this instance. After that we tackle two related problems: (1) how to obtain the probability of a specific OCS, and (2) how to measure the change of the model given an unlabeled instance and one of its OCS.

Expected model change (EMC)

Let $\mathbf{f}_L$ denotes a multi-class classification model trained on all currently labeled data. Our objective is to assess how much impact the annotation of a currently unlabeled example $\mathbf{x}_0$ with an OCS can make. Let $\Delta(\mathbf{f}_L, \mathbf{x}_0)$ be a measure of this impact. In this work, we assess the impact in terms of the expected model change and an unlabeled instance with the highest expected model change is selected for the labeling first. We define the expected model change for the OCS feedback as:

$$\Delta(\mathbf{f}_L, \mathbf{x}_0) = \sum_{S_0 \in \mathbf{S}} P(S_0|\mathbf{x}_0)\delta(\mathbf{f}_L, \mathbf{f}_{L \cup \langle \mathbf{x}_0, S_0 \rangle}) \tag{3}$$

where $\delta(\mathbf{f}_L, \mathbf{f}_{L \cup \langle \mathbf{x}_0, S_0 \rangle})$ denotes a model change induced by assigning an ordered class set (OCS) S_0 to example $\mathbf{x}_0$. Intuitively, the expected change is a weighted average of model changes for all possible ordered class sets $\mathbf{S}$ where the weight is a probability of the instance $\mathbf{x}_0$ being assigned an OCS S_0. To simplify the model of $P(S_0|\mathbf{x}_0)$ and its construction we express it in terms of two probabilities: $P(S_0|\mathbf{x}_0) = P(S_0|A_0, \mathbf{x}_0)P(A_0|\mathbf{x}_0)$, where $P(A_0|\mathbf{x}_0)$ is the probability of an unordered class-set A_0 defining S_0, and $P(S_0|A_0, \mathbf{x}_0)$ is the probability of the specific class-order for a fixed A_0. In order to calculate the expected model change three quantities defining it need to be estimated: (1)

the probability $P(A_0|\mathbf{x}_0)$ of each unordered class set A_0, (2) the conditional probability $P(S_0|A_0, \mathbf{x}_0)$ for each OCS S_0 given its corresponding unordered class set A_0, and (3) the model change $\delta(\mathbf{f}_L, \mathbf{f}_{L\cup\langle\mathbf{x}_0, S_0\rangle})$. We present our solutions to these next.

Estimating the probability of an unordered class set The first quantity to be estimated is the probability $P(A_0|\mathbf{x}_0)$ for each unordered class set A_0. We approximate this quantity with the help of an auxiliary multi-label logistic regression model $\mathbf{g}_L$ we train on the data annotated with OCS. The model $\mathbf{g}_L$ maps instances to a class vector of size k indicating whether a class should be included in the unordered class set or not. We define the output of a multi-label classifier as $\mathbf{z}_i = \mathbf{g}_L(\mathbf{x}_i) = M^T\phi(\mathbf{x}_i)$. The input $\mathbf{x}_i$ of this model is a d-dimensional feature vector of a data instance, and the output $\mathbf{z}_i$ is a class vector of size k indicating whether a class should be included in the unordered class set or not. M is a $d \times k$ matrix of parameters of this model, and $\phi(\cdot)$ is the projection of the kernel space. The training of $\mathbf{g}_L$ is also intuitive: an OCS can be converted into a class vector naturally. If a class is included in the OCS, then the corresponding scalar of this class in the class vector is 1, otherwise the scalar is -1. After converting the OCS of each labeled instance, we will take the feature vector and class vector of each labeled instance for training. In this paper, we use an improved multi-label logistic regression model by (Xu, Tao, and Geng 2018). Basically, this multi-label logistic regression considers the topological information of the feature space: the data instances close to each other are more likely to share the same class vector. Formally, the optimization of the model parameter M can be formalized as follows:

$$\min_M \sum_{i=1}^{N} ||M^T\phi(\mathbf{x}_i) - \mathbf{z}_i||^2 + \lambda \sum_{ij} t_{ij} ||M^T[\phi(\mathbf{x}_i) - \phi(\mathbf{x}_j)]||^2 \tag{4}$$

where $\mathbf{x}_i$ and $\mathbf{z}_i$ are the feature vector and class vector. t_{ij} is the topological information between $\mathbf{x}_i$ and $\mathbf{x}_j$. $t_{ij} = \exp(-\frac{||\mathbf{x}_i - \mathbf{x}_j||^2}{2})$ if $\mathbf{x}_j$ is among the nearest neighbors of $\mathbf{x}_i$, and $t_{i,j} = 0$ otherwise. λ is the parameter trading off the two terms. The number of nearest neighbors is also tunable. After obtaining the optimal parameter $\hat{M}$ from the optimization, the estimate of $\hat{A}_0$ of A_0 can be obtained from the predicted class vector $\hat{\mathbf{z}_0} = \hat{M}^T\phi(\mathbf{x}_0)$.

Estimating the conditional probability of an OCS The second quantity to be estimated when calculating the expected model change is the conditional probability $P(S_0|A_0, \mathbf{x}_0)$ of each OCS S_0 given the corresponding unordered class set A_0 and an unlabeled instance $\mathbf{x}_0$. Although it is hard for us to directly estimate $P(S_0|A_0, \mathbf{x}_0)$, the class-wise conditional probability $P(c|A_0, \mathbf{x}_0)$ of a single class $c \in A_0$ can be estimated directly by applying a soft-max function: $P(c|A_0, \mathbf{x}_0) = \frac{\exp(\mathbf{w}_c^T\phi(\mathbf{x}_0))}{\sum_{i\in A_0}\exp(\mathbf{w}_i^T\phi(\mathbf{x}_0))}$. Since each OCS $S_0 \sim A_0$ is a permutation of the unordered class set A_0, the conditional probability $P(S_0|A_0, \mathbf{x}_0)$ can be constructed from the class-wise conditional probability $P(c|A_0, \mathbf{x}_0)$ for all $c \in A_0$ the same way we construct the probability of a permutation. Formally, the probability $P(S_0|A_0, \mathbf{x}_0)$ can be constructed as:

$$P(S_0|A_0, \mathbf{x}_0) = \prod_{i=1}^{|S_0|} \frac{P(S_{0i}|A_0, \mathbf{x}_0)}{1 - \sum_{j=1}^{i-1} P(S_{0j}|A_0, \mathbf{x}_0)} \tag{5}$$

where S_{0i} indicates the ith probable class in S_0.

It seems the conditional probability $P(S_0|A_0, \mathbf{x}_0)$ is perfectly calculated. However, there is an inevitable fact: each S_0 is a permutation of its corresponding unordered class set A_0. This indicates that, given an unordered class set A_0, the number of OCS S_0 such that $S_0 \sim A_0$ is actually $|A_0|!$. Considering relation between δ and δ' explained in Formula (7), we also need to enumerate all unlabeled data instances to calculate the OCS change of $S_0 \sim A_0$. Therefore, the time complexity of EMC for a given unlabeled data instance $\mathbf{x}_0$ is $O(U|A_0|!)$, where U is the number of unlabeled data instances. Clearly, it is intractable to calculate the conditional probability $P(S_0|A_0, \mathbf{x}_0)$ for all the OCS $S_0 \sim A_0$. To reduce the number of OCS to enumerate, a straightforward method is to do random sub-sampling over all the OCS $S_0 \sim A_0$. However, such a method introduces another problem: is such a sub-sample a "good" approximation of all OCSs $S_0 \sim A_0$? That is, is the EMC obtained using this sub-sample similar to the EMC obtained by considering all OCSs $S_0 \sim A_0$? To solve this problem, we propose the following sub-sampling scheme: first, we create two random sub-samples T_0' and T_0'' over all the OCS $S_0 \sim A_0$ such that: (1) $S_0 \in T_0' \Rightarrow S_0 \sim A_0$ and $S_0 \in T_0'' \Rightarrow S_0 \sim A_0$. In other words, both T_0' and T_0'' only contains the OCS whose corresponding unordered class set is A_0. (2) $T_0' \cap T_0'' = \emptyset$, and (3) $|T_0'| = |T_0''| = m$ where m is a small number. Then, we define an instance-wise EMC $\kappa(\mathbf{f}_L, \mathbf{x}_0, \mathbf{x}_u, T_0')$ of the unlabeled instance $\mathbf{x}_0$ on an arbitrary unlabeled instance $\mathbf{x}_u$ and a sub-sample set T_0' where $S_0 \in T_0' \Rightarrow S_0 \sim A_0$ as follows:

$$\kappa(\mathbf{f}_L, \mathbf{x}_0, \mathbf{x}_u, T_0') = \frac{1}{Z} \sum_{S_0 \in T_0'} P(S_0|A_0, \mathbf{x}_0)\delta'(\mathbf{f}_L, \mathbf{f}_{L\cup\langle\mathbf{x}_0, S_0\rangle}, \mathbf{x}_u) \tag{6}$$

where $u \notin L$ and T_0' only contains the OCS whose corresponding unordered class set is A_0. $Z = \sum_{S_0 \in T_0'} P(S_0|A_0, \mathbf{x}_0)$ is the partition function. δ' reflects the OCS change observed on a specific unlabeled example $\mathbf{x}_u$ and its output OCS. The relation between δ and δ' is explained in Formula (7).

Clearly, the instance-wise EMC is similar to the EMC in Formula (3) while considering only one unlabeled instance $\mathbf{x}_u$ and a certain unordered class set A_0. If both T_0' and T_0'' are "good" approximations for all OCSs $S_0 \sim A_0$, then the instance-wise EMC $\kappa(\mathbf{f}_L, \mathbf{x}_0, \mathbf{x}_u, T_0')$ and $\kappa(\mathbf{f}_L, \mathbf{x}_0, \mathbf{x}_u, T_0'')$ on both sub-samples should be approximately equal to each unlabeled instance $\mathbf{x}_u$. In other words, the quantity $\kappa(\mathbf{f}_L, \mathbf{x}_0, \mathbf{x}_u, T_0') - \kappa(\mathbf{f}_L, \mathbf{x}_0, \mathbf{x}_u, T_0'') \approx 0$ for all $u \notin L$, which can be validated using a t-test with a hypothesis that the mean of this quantity is 0. If the t-test does not reject the hypothesis, we may consider both T_0' and T_0'' as "good" approximations over all the OCS $S_0 \sim A_0$, and take $T_0' \cup T_0''$ as the sub-sample over all the OCS $S_0 \sim A_0$ and only considers the OCS $S_0 \in T_0' \cup T_0''$. The conditional probability

$P(S_0|A_0, \mathbf{x}_0)$ of the OCS $S_0 \in T_0' \cup T_0''$ should also be normalized to exclude the OCS not in the sub-sample $T_0' \cup T_0''$; otherwise, we increase m and repeat the scheme until the t-test does not reject the hypothesis. By applying the subsampling technique above, the time complexity of EMC for a given unlabeled data instance $\mathbf{x}_0$ is reduced to $O(Um)$.

Approximating the OCS change of an instance The third important quantity to be estimated is the OCS-related model change $\delta(\mathbf{f}_L, \mathbf{f}_{L\cup\langle\mathbf{x}_0,S_0\rangle})$. We calculate the model change by observing and assessing changes in the ordered class sets (OCSs) assigned for each unlabeled example $\mathbf{x}_u, i$ by models $\mathbf{f}_L$ and $\mathbf{f}_{L\cup\langle\mathbf{x}_0,S_0\rangle}$. More formally, we express the model change as:

$$\delta(\mathbf{f}_L, \mathbf{f}_{L\cup\langle\mathbf{x}_0,S_0\rangle}) = \sum_{\mathbf{x}_u} \delta'(\mathbf{f}_L, \mathbf{f}_{L\cup\langle\mathbf{x}_0,S_0\rangle,\mathbf{x}_u}) \tag{7}$$

where δ' reflects the OCS change observed on a specific unlabeled example $\mathbf{x}_u$ and its output OCS.

The OCS change can be easily measured as the absolute ranking change on all k classes of $\mathbf{x}_u$. Formally, we define a function $\text{rank}(\mathbf{f}, \mathbf{x}, c)$ which returns the ranking of class c in the output $\mathbf{f}(\mathbf{x}) = \{f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_k(\mathbf{x})\}$. Therefore, the OCS change $\delta'(\mathbf{f}_L, \mathbf{f}_{L\cup\langle\mathbf{x}_0,S_0\rangle}, \mathbf{x}_u)$ can be calculated as $\sum_{i=1}^{k} ||\text{rank}(\mathbf{f}_L, \mathbf{x}_u, i) - \text{rank}(\mathbf{f}_{L\cup\langle\mathbf{x}_0,S_0\rangle}, \mathbf{x}_u, i)||$. However, such estimation is not perfect: it assumes all the k classes contribute equally to the change. This is however inconsistent with the fact: the changes of classes on higher rankings should be emphasized. For example, if the class on the first ranking changes, the predicted class label will also change. To address this problem, we introduce Discounted Cumulative Gain (DCG) (Järvelin and Kekäläinen 2002). Briefly, DCG discounts the change of a class over a log expression of its ranking, which understates the changes of classes on lower rankings. Formally, the OCS change $\delta'(\mathbf{f}_L, \mathbf{f}_{L\cup\langle\mathbf{x}_0,S_0\rangle}, \mathbf{x}_u)$ with DCG can be calculated as:

$$\delta'(\mathbf{f}_L, \mathbf{f}_{L\cup\langle\mathbf{x}_0,S_0\rangle}, \mathbf{x}_u) = \sum_{i=1}^{k} \frac{||\text{rank}(\mathbf{f}_L, \mathbf{x}_u, i) - \text{rank}(\mathbf{f}_{L\cup\langle\mathbf{x}_0,S_0\rangle}, \mathbf{x}_u, i)||}{\log_2[1 + \text{rank}(\mathbf{f}_L, \mathbf{x}_u, i)]} \tag{8}$$

Experiments and results

We test our approach on synthetic and real-world data. The two synthetic datasets are built from two UCI multi-class classification datasets where the OCSs are simulated; the three real-world datasets contain OCS that are assessed by human annotators and are extracted directly.

Experimental settings

The two synthetic OCS datasets are generated from UCI Vehicle Silhouettes and Optical Digits datasets. We do this by taking $\frac{1}{3}$ of data instances in these datasets to train an AMSVM with class labels only. After training, we apply the trained AMSVM to every instance in the remaining $\frac{2}{3}$ of the dataset, and calculate the probabilitity distribution of all its classes via soft-max function. We generate the OCS for the instance by ordering the classes in terms of their probability and by excluding those classes that fall below probability 0.05. In the OCS experiments we use only the $\frac{2}{3}$ of data that consists of the original feature vectors and the correspoding (calculated) OCS.

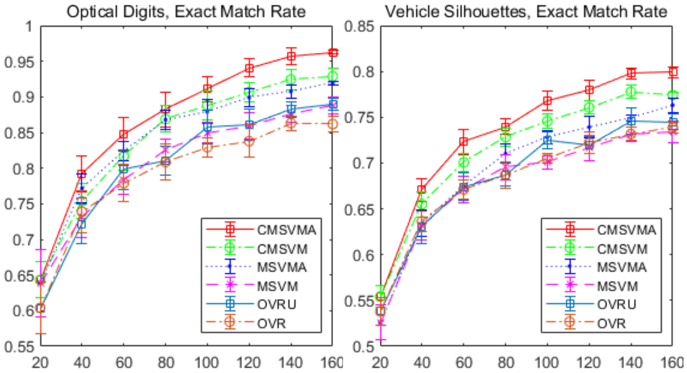

Figure 1: Performance (exact match rate) on two synthetic datasets in experiments.

The real-world datasets consists of two Million Song datasets (CD1 and CD2) (Bertin-Mahieux et al. 2011) and one Face Sentiment dataset (Mozafari et al. 2012). Each Million Song dataset consists of a collection of songs. In each dataset, the feature vector of each instance contains the timbre information of this song, the OCS of each instance contains one or two classes indicating the genre that this song likely belongs to. Please notice that each song can only belong to one genre, and the OCS of this song just indicates the competing choices of genres. In Face Sentiment data, the feature of each instance is a 128×120 gray-scale image of a facial expression, where we extract 256 features using a convolutional neural network. The class label of each instance indicates the sentiment of facial expression. However, each image is annotated by 9 human annotators. Therefore, we may sort the classes according to their vote numbers in the descending order, and take the ordered set of classes as the OCS for each instance. The basic properties of two synthetic datasets and three real-world datasets are summarized in Table 1.

To demonstrate the benefits of our multi-class classifier trained with ordered class set (OCS) and our expected model change (EMC) active learning strategy, we compare it with a number of existing multi-class classifiers with and without an active learning strategy. These include: (1) one-vs-rest classifier trained only on class labels, (2) one-vs-rest classifier trained only on class labels with uncertainty sampling active learning strategy, (3) approximate multi-class SVM (AMSVM) trained only on class labels, (4) AMSVM trained only on class labels with EMC active learning strategy (EMC can be also applied to multi-class classifier with class labels only by taking the class label as an OCS of size 1), and (5) our multi-class classifier trained with ordered class sets (OCS) but without active learning (examples are picked randomly). The details of all methods in the experiments are as follows:

OVR: k one-vs-rest binary classifiers trained *independently*,

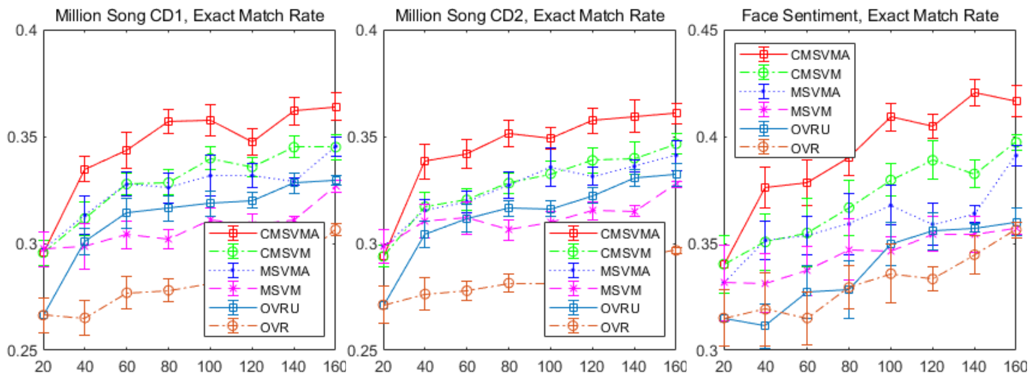

Figure 2: Performance (exact match rate) on three real-world datasets in experiments.

Dataset Name	# of Instances	# of Features	# of Classes	Size of OCS
Vehicle Silhouettes	946	18	4	Simulated
Optical Digits	5620	64	10	Simulated
Million Song CD1	35409	90	13	1∼2
Million Song CD2	89073	90	15	1∼2
Face Sentiment	584	256	4	1∼4

Table 1: Properties of all datasets in experiments.

one for each class; The instances to be labeled next are selected randomly; (k is the number of classes in the dataset, sic passim.)

OVRU: k one-vs-rest binary classifiers trained *independently*, one for each class; The instances to be labeled next are selected using least confident uncertainty sampling (Settles, Craven, and Friedland 2008) active learning strategy;

MSVM: Approximate multi-class SVM (AMSVM) where k one-vs-rest binary classifiers are trained *jointly*, one for each class; The instances to be labeled next are selected randomly;

MSVMA: Approximate multi-class SVM (AMSVM) where k one-vs-rest binary classifiers are trained *jointly*, one for each class; The instances to be labeled next are selected using our EMC active learning strategy;

CMSVM: Our multi-class classifier incorporated with OCS where k one-vs-rest binary classifiers are trained *jointly*, one for each class; The instances to be labeled next are selected randomly;

CMSVMA: Our multi-class classifier incorporated with OCS where k one-vs-rest binary classifiers are trained *jointly*, one for each class; The instances to be labeled next are selected using our EMC active learning strategy.

All data sets are split (before the training) into the training and test set (using $\frac{2}{3}$ and $\frac{1}{3}$ of all data instances). We evaluate the performance of the different methods by calculating the exact match rates (EMR) all the classifiers achieve on the test data. Exact match rate calculates the ratio of data instances for which the prediction is identical to its class label, over all data instances. The learning considers the training data only, and the EMR is always calculated on the test set. We also repeat the splitting and learning steps 24 times. The average EMR (Y-axis) of different classifiers on two synthetic datasets and three real-world datasets regarding increasing sizes (X-axis) of the training sets is reported in Figure 1 and Figure 2 respectively.

Experimental results

Figure 1 and Figure 2 show the benefit of our multi-class classifier trained with OCS and our EMC active learning strategy on two synthetic datasets and three real-world datasets both individually and jointly:

Effect of incorporating OCS: On all five datasets, *CMSVM* outperforms *MSVM* and *OVR*; *CMSVMA* outperforms *MSVMA* and *OVRU*. These two groups of comparisons show that our multi-class classifier trained with the OCS can improve the learning performance when compared with models that use only class label information at the same training size.

Effect of EMC strategy: Also, on all five datasets, *CMSVMA* outperforms *CMSVM*; *MSVMA* outperforms *MSVM*. These two groups of comparisons show that our EMC active learning strategy can improve the performance of the multi-class models compared with models that select next exmples randomly. Once again the results are matched at the same training data size.

Effect of combining OCS and EMC: Overall, on all the five datasets, the model *CMSVMA*, which is the combination of our multi-class classifier incorporated with OCS and our EMC active learning strategy, achieved the highest performance. This supports and confirms the effectiveness of our multi-class classifier trained with with OCS and our EMC active learning strategy.

Conclusion

Ordered class set (OCS) is a special auxiliary information arising in multi-class classification settings that can be easily obtained from human annotators at an insignificant cost and can help us to reduce the annotation efforts. In this work, we proposed a new framework for learning multi-class classification models from human feedback that utilizes OCS and a novel active learning strategy: expected model change (EMC) that matches the OCS labels. Our results show that our learning framework (1) is able to learn more efficiently and from a smaller number of labeled instances than existing methods (2) is better than models that rely on OCS or active learning individually.

Acknowledgements

The work presented in this paper was supported by NIH grants R01GM088224 and R01LM010019. The content of the paper is solely the responsibility of the authors and does not necessarily represent the official views of NIH.

References

Bertin-Mahieux, T.; Ellis, D. P.; Whitman, B.; and Lamere, P. 2011. The million song dataset. In *Proceedings of the 12th International Conference on Music Information Retrieval (ISMIR 2011)*.

Breiman, L. 1996. Bagging predictors. *Machine Learning* 24(2):123–140.

Chu, W., and Keerthi, S. S. 2005. New approaches to support vector ordinal regression. In *Proceedings of the 22nd international conference on Machine learning*, ICML '05, 145–152. New York, NY, USA: ACM.

Griffin, D., and Tversky, A. 1992. The weighing of evidence and the determinants of confidence. *Cognitive Psychology* 24(3):411 – 435.

Hauskrecht, M.; Batal, I.; Valko, M.; Visweswaran, S.; Cooper, G. F.; and Clermont, G. 2013. Outlier detection for patient monitoring and alerting. *Journal of Biomedical Informatics* 46(1):47 – 55.

Hauskrecht, M.; Batal, I.; Hong, C.; Nguyen, Q.; Cooper, G. F.; Visweswaran, S.; and Clermont, G. 2016. Outlier-based detection of unusual patient-management actions: An icu study. *Journal of Biomedical Informatics* 64:211 – 221.

He, X.; Wang, Z.; Jin, C.; Zheng, Y.; and Xue, X. 2012. A simplified multi-class support vector machine with reduced dual optimization. *Pattern Recognition Letters* 33(1):71 – 82.

Järvelin, K., and Kekäläinen, J. 2002. Cumulated gain-based evaluation of ir techniques. *ACM Trans. Inf. Syst.* 20(4):422–446.

Juslin, P.; Olsson, H.; and Winman, A. 1998. The calibration issue: Theoretical comments on suantak, bolger, and ferrell. *Organizational Behavior and Human Decision Processes* 73(1):3–26.

Lewis, D. D., and Gale, W. A. 1994. A sequential algorithm for training text classifiers. In *Proceedings of the 17th ACM International Conference on Research and Development in Information Retrieval*, 3–12. Dublin, IE: Springer Verlag, Heidelberg, DE.

Luo, Z., and Hauskrecht, M. 2018a. Hierarchical active learning with group proportion feedback. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, 2532–2538. International Joint Conferences on Artificial Intelligence Organization.

Luo, Z., and Hauskrecht, M. 2018b. Interactive structure learning with structural query-by-committee. In *Proceedings of the 2018 European Conference on Machine Learning (ECML 2018)*.

Mozafari, B.; Sarkar, P.; Franklin, M. J.; Jordan, M. I.; and Madden, S. 2012. Active learning for crowd-sourced databases. *CoRR* abs/1209.3686.

Nguyen, Q.; Valizadegan, H.; and Hauskrecht, M. 2011a. Learning classification with auxiliary probabilistic information. In *IEEE International Conference on Data Mining*, 477–486.

Nguyen, Q.; Valizadegan, H.; and Hauskrecht, M. 2011b. Sample-efficient learning with auxiliary class-label information. In *Proceedings of the Annual American Medical Informatics Association Symposium*, 1004–1012.

Nguyen, Q.; Valizadegan, H.; and Hauskrecht, M. 2013. Learning classification models with soft-label information. *Journal of American Medical Informatics Association*.

O'Hagan, A.; Buck, C.; Daneshkhan, A.; Eiser, R.; Garthwaite, P.; Jenkinson, D.; Oakley, J.; and Rakow, T., eds. 2007. *Uncertainty judgements Eliciting experts' probabilities*. John Wiley and Sons.

Peng, P., and Wong, R. C.-W. 2014. Selective sampling on probabilistic data. In *SIAM International Conference on Data Mining*, 28–36.

Peng, P.; Wong, R. C.-W.; and Yu, P. S. 2014. Learning on probabilistic labels. In *SIAM International Conference on Data Mining*, 307–315.

Roy, N., and McCallum, A. 2001. Toward optimal active learning through sampling estimation of error reduction. In Brodley, C. E., and Danyluk, A. P., eds., *Proceedings of the 18th International Conferenceon on Machine Learning*, 441–448. Williams College,Williamstown, MA, USA: Morgan Kaufmann.

Settles, B.; Craven, M.; and Friedland, L. 2008. Active learning with real annotation costs. In *Proceedings of the NIPS workshop on cost-sensitive learning*, 1–10.

Settles, B.; Craven, M.; and Ray, S. 2008. Multiple-instance active learning. In *Advances in Neural Information Processing Systems*, 1289–1296. MIT Press.

Settles, B. 2010. Active learning literature survey. Technical report.

Seung, H. S.; Opper, M.; and Sompolinsky, H. 1992. Query by committee. In *Proceedings of the Fifth Annual ACM Workshop on Computational Learning Theory*, 287–294. Pittsburgh, Pennsylvania: ACM Press.

Thiel, C. 2008. *Classification on Soft Labels Is Robust against Label Noise*. Berlin, Heidelberg: Springer Berlin Heidelberg. 65–73.

Tong, S., and Koller, D. 2000. Active learning for parameter estimation in bayesian networks. In *Advances in Neural Information Processing Systems*, 647–653. MIT Press.

Tosh, C., and Dasgupta, S. 2018. Label enhancement for label distribution learning. In *Proceedings of the 32nd Conference on Neural Information Processing Systems (NIPS 2018)*.

Vapnik, V. N. 1998. *Statistical Learning Theory*. New-York: Wiley.

Weston, J.; Gammerman, A.; Stitson, M.; Vapnik, V.; Vovk, V.; and Watkins, C. 1999. Support vector density estimation.

Xu, N.; Tao, A.; and Geng, X. 2018. Hierarchical active learning with proportion feedback on regions. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI 2018)*.

Xue, Y., and Hauskrecht, M. 2017a. Active learning of classification models with likert-scale feedback. In *SIAM International Conference on Data Mining*, 28–35.

Xue, Y., and Hauskrecht, M. 2017b. Efficient learning of classification models from soft-label information by binning and ranking. *Proceedings of the 30th International Florida AI Research Society Conference. Florida AI Research Symposium* 164–169.