Labeller adjusted active learning: Accounting for "I don't know?!?" and "Definitely not this!"

A1, A2

November 2, 2013

Abstract

We discuss the active learning of a classifier when 1) a labeller returns a probability vector over the labels 2) the labeller's certainty of the labels is not constant over the item space. We show how the labeller uncertainty can be modelled using Gaussian process latent variable model, and how incorporating the uncertain labeller model to the optimal sampling scheme improves classification under uncertain labeling.

1 Introduction

The optimal design of sampling for improving model estimates, also known as active learning, is a popular method for training classifiers on large collection of items. In short, an unlabelled item database is ranked based on the usefullness of the expected label for the current state of the classifier. Then an external human labeller is asked for the correct label and the model is re-estimated. In theory, the learning rate of the classifier is optimal as the data so collected is the most useful.

The process relies on the labeller ability to produce the correct label, hence the often used name 'oracle'. The oracle is assumed all-knowing, and several approaches have been introduced to alleviate this unrealistic assumption. In general, the label is allowed to be wrong with some probability, reflecting e.g. the limited knowledge of the labeller.

The five papers studying labellers with imperfect answers [][][][][] assume the following: ...

In this paper we discuss two novel ideas. First, we want to accommodate the answers "definitely not this one" and "I don't know". Second, we want to account for the labeller's limited knowledge in the active learning process.

2 Outline of approach

As the core problem of interest we tackle the training of a classifier in a semi-supervised manner. We have a set of labeled training items, and we want to improve the classifier by querying an external labeller for some more labels for unlabeled items. The idea is that asking is expensive, so we should only query the most useful items *given the classifier and the labeller* at hand.

Formally, let $\mathcal{D} = \{i = (y_i, x_i)\}$ denote a set of items, where $y_i \in \mathcal{Y}$ with $|\mathcal{Y}| = K$ denote labels for the items and $x_i \in \mathcal{X} \subseteq \mathbb{R}^p$ denote other features. Write for any $x \in \mathcal{D} \cap \mathcal{X}$ the correct label y_x. We assume the pool-based sampling scheme, i.e. we have a pool $\mathcal{U} \subset D \cap \mathcal{X}$ of unlabeled items which we can use for querying the labeller for their labels. We assume that the cost of querying is constant.

Our proposed approach, coined *labeller adjusted uncertainty sampling*, can be outlined by the following schematic algorithm [cf. "prototypical active learning algorithm" presented by **?**]:

1: Initialize $\mathcal{L} = \mathcal{L}_0 \subset D$
2: **while** Some stopping criterion is not met **do**
3: Train classifier c with a set of correcly labeled items $\mathcal{L}$
4: Select unlabeled item $x^* \in \mathcal{U}$ based on the highest *labeller adjusted informativeness*:
5: Estimate informativeness of $x \in \mathcal{U}$ for the classifier c
6: Estimate uncertainty of $x \in \mathcal{U}$ for the labeller l
7: Combine both measures
8: Query labeller for the label $y^*_{x^*}$ of x^*; add $(y^*_{x^*}, x^*)$ to $\mathcal{L}$
9: **end while**
10: Return final classifier c

This is an extension of the well-known "uncertainty sampling strategy". However, instead of only taking the classifier's into account, we also take the knowledge of the labeller into account.

This introduces three challenges which we address in detail in the subsequent sections. First, we need a way to enable the labeller to give flexible answers, e.g., to return "I don't know" or "I am sure it is this label" (line 8 in the algorithm, discussed in Section 3). Second, we need a model to estimate the labeller's uncertainty for a possible candidate query (line 6, Section 4). And, third, we need a suitable way to combine the classifier's informativeness and the labeller's uncertainty (line 7, Section 5).

Example. Throughout the paper we use a simple two-dimensional example to illustrate each challenge and our proposed solution. For the item set D we simulated three 2D Gaussian clusters, 50 points each; Figure 1(left) shows the data set with the ground-truth labels. We use gaussian discriminant model with class specific covariance matrices as the base classifier (aka QDA).

3 A model for flexible answers

We assume that the classifier which we want to train can be written in the general form

$$y|x,\theta \quad \sim \quad p(y|x,\theta) \propto \prod_k p(k|x,\theta)^{[y=k]} \tag{1}$$

with θ being the parameters of the model. We define that for each queried item x, the labeller returns a probability vector $h_x \in \mathcal{H} = \{h \in \mathbb{R}_+^K : \sum_{k\in\mathcal{Y}} h_k = 1\}$. We furthermore assume that the labeller is biased towards the correct label, i.e. $\text{argmax}_k h_{xk} = y_x$.

The oracle labeller is then the special case "definitely k" with $h = \mathbf{1}\{h_k = 1\}$. The "I don't know" case is $\mathbf{1}\{h_k = K^{-1}\ \forall k\}$, and in between are a continuum of alternatives, including the "definitely not k" with $\mathbf{1}\{h_k = 0\}$ and "one of these M" with $\mathbf{1}\{h_k > 0\ \forall k \in M\}$.

The connection between h and y needs to be made in order for the extension to be useful in learning the classifier parameters θ. We assume that the following model is valid:

$$h|x,\theta \quad \sim \quad p(h|x,\theta) \propto p(h|x) \prod_k p(k|x,\theta)^{h_k} \tag{2}$$

in which the y variable is no longer necessary to be observed. The model is a continuous version of the categorical model, and as y is a categorical variable, we can replace the observations y with indicator vector $\mathbf{1}(h_y = 1)$. An interpretation for the approach is given by the multinomial distribution: the product is approximately the likelihood of a multinomial variable if h would be integers. The scaling of h is not important, the pairwise proportions of h_k are the main information. As a practical example, the labeller could be given a set of ten tokens to distribute among the label candidates.

Example. To illustrate this approach, we conducted a simulation study with our two-dimensional toy data set. The parameters $\theta = \{\mu_k, \Sigma_k : k = 1, ..., K\}$ for the classifier for a given set of observations $L = \{(h_i; x_i)\}$ are given as weighted averages and covariance matrices, i.e. each queried item x_i is 'observed' in each of the K classes with a weight h_i. Two points from each cluster were given as the initial data for the classifier.

We simulated two labeller. Both knew one of the classes, say k', and when queried with an item of this class they returned a corresponding indicator vector. When queried with an item of another classes one labeller returned 0.5 for $k \neq k'$ (and 0 for k' as the labeller knows the label is not k'). The other labeller was forced to make a decision, simulated here by a random pick.

We ran 20 different samplings sequences from the item set, each step drawing uniformly from the pool of unlabeled items. The classification rate over a test set for the forced-to-choose labeller were averaged over 10 random picking sequences per sampling, estimating the expectation over the forcing event. Figure 1(right) shows the $(0.05, 0.5, 0.95)$ quantile envelope plots of the classification rate as a function of sample size.

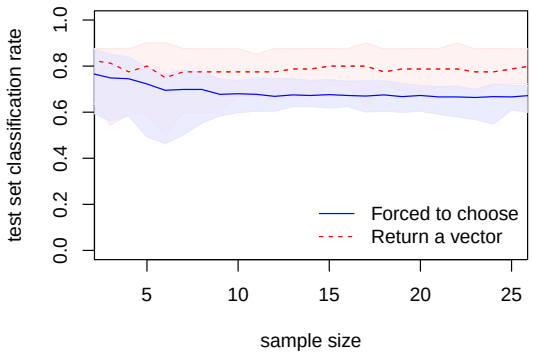

Figure 1: Classification rate example, two modes the labeller can provide labeling information: Choose one vs. a weight vector. The shaded area is the 90% range over data sampling sequences, 20 different samplings. The forced-to-choose values are further averaged over 10 random selection sequences per sampling sequence.

The illustration nicely shows that the accuracy of the classifier is improved if the labeller is allowed to express uncertainty for a query and is not forced to choose a single value. The answer h contains information not only on the label y but also on the labeller. This allows us to model the labeller's uncertainty.

4 The answer depends on the question

Obviously, the uncertainty of a labeller depends on the specific query which item x to label. Therefore, we assume that $\alpha(x) \in \mathbb{R}$ is a value that describes the uncertainty the labeller has on the label of x, and assume that α is continuous and varies smoothly in $\mathcal{X}$. The interpretation is that high value of α means high uncertainty, and vice versa.

Continuing from the Equation 2, we now introduce a prior for the h values. We model them as dependent on the smooth α, which we assume in this paper to be adequately described by a log-Gaussian process in $\mathcal{X}$. The natural model for h is the Dirichlet distribution, we simplify the problem by assuming symmetry, i.e., the Dirichlet parameter α is the same for all classes.

The prior $p(h|x,\alpha)$ for the labeller's variable output then becomes

$$h|x,\alpha \sim Dir(\alpha(x)) \quad (3)$$
$$\alpha \sim \log -GP(\mu, C) \quad (4)$$

with some hyper-prior mean μ and covariance function C. In what follows the parameter α shall be known as the *labeller's uncertainty*.

Example. We simulated a labeller with a circular "area of expertice", a disc in the feature space within which the labeller knows the labels. Outside the labeller returns "I don't know". Figure 2 depicts the Kriging surface estimate of α after 10 and 20 uniformly chosen queries.

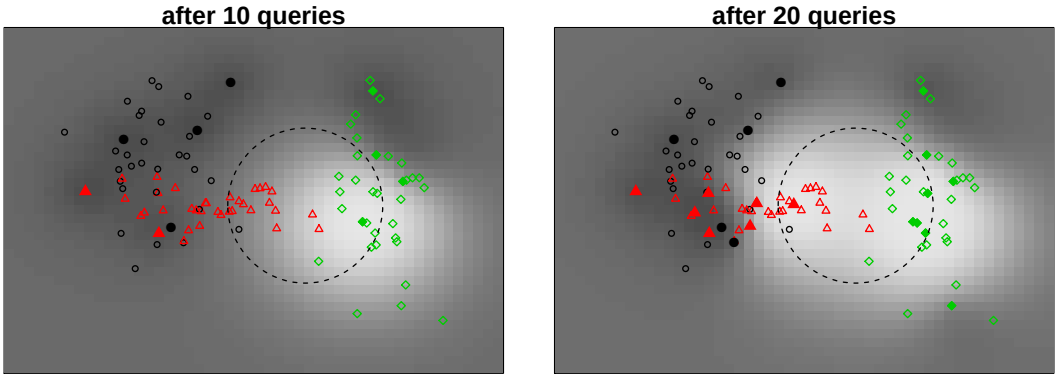

Figure 2: Labeller's uncertainty as estimated after 10 (left) and 20 (right) queries. Symbols denote the item set, filled symbols indicate items that were queried. The Dashed disc is the labeller's area of expertice where he knows the label. Dark color of the background is for low certainty areas and light color is for high certainty areas of the feature space.

Being now able to detect the area of expertice of a labeller, an immediate question arises: How can we combine the two models in order to do active learning?

5 Informativeness estimation and adjustment

Here, we focus our discussion on the case where the labeller model is independent from the classifier model, i.e., the classifier itself does not take the labeller's uncertainty into accout, but the classifier's informativeness and the labeller's uncertainty are combined in some explicit way.

As outlined in Section 2, in every iteration of the algorithm the model parameter estimates are used for computing an informativeness function $I(x) = I(x;\hat{\theta})$ (line 5). This provides a measure of usefulness of gaining the label of an item $x \in \mathcal{U}$. There exists several formulations of $I(x)$, and for illustration we will focus on the entropy, defined as

$$I_c(x) := -E_y[\log p(y|x,\hat{\theta})] \quad (5)$$
$$= -\sum_k p(k|x,\hat{\theta}) \log p(k|x,\hat{\theta}) \quad (6)$$

where the probabilities are the posterior predictive probabilities at the current iteration.

The standard uncertainty sampling strategy queries the item which maximizes this informativeness I_c. Because of the knowledge of the labeller who does not always know the label, the expected gains are reduced. In a straightforward manner, the informativeness given by the labeller could also be expressed using the entropy,

$$I_e(x) \quad := \quad -E_h \log p(h|x, \mathcal{L}) \tag{7}$$

where $\mathcal{L}$ is all so far collected data. The entropy of a Dirichlet distributed h depends on the α, leading to the formula

$$\begin{aligned} I_e(x) \quad = \quad & E\left\{\log B(\alpha(x)) + \right. \\ & \left. K(\alpha(x) - 1)[\psi(K\alpha(x)) - \psi(\alpha(x))]\right\} \end{aligned} \tag{8}$$

where B is the beta-function, ψ is the digamma function, and the expectation is over $\alpha(x)$. Unfortunately this function is not monotonous in α: The entropy curve has a mode at $\alpha = 1$ and as α increases the h will converge to a constant vector, the uniform distribution over the classes. Uniform h should be regarded as the least informative answer from the labeller. We therefore suggest the use of variance which is a monotonically decreasing function of α,

$$I_l(x) := Var[h|x, \alpha] = \frac{K - 1}{K^2(K\alpha(x) + 1)}.$$

The two $I(\cdot)$. functions do not share units, and to overcome this we use the geometric mean to arrive at the *labeller adjusted informativeness*

$$I(x) := (I_l(x) I_c(x))^{1/2},$$

which builds the the basis of our proposed *labeller adjusted uncertainty sampling* strategy for active learning.

Example. We applied the adjusted uncertainty sampling strategy to the example of circular area of expertice. The lower left plot of Figure 3 depicts the learned labeller's uncertainty after 25 queries, along with the posterior 95% probability ellipses of the clusters. The top left and top right plots show what happends when only I_l and I_c, respectively, are used. If only those items are queried of which the labeller is certain, it will result in greedy sampling of items close to known good quality points, regardless of their utility to the classifier. If only the classifier is concerned, the lack of information of items outside labeller's area of expertise is not realized. The use of both provides a compromise, and can lead to substantial improvement as depicted in the test set classification rates. For the joint strategy the rate reaches almost the rate achieved when total knowledge of the labels is available, marked with the horizontal line in the bottom right plot.

6 Application example

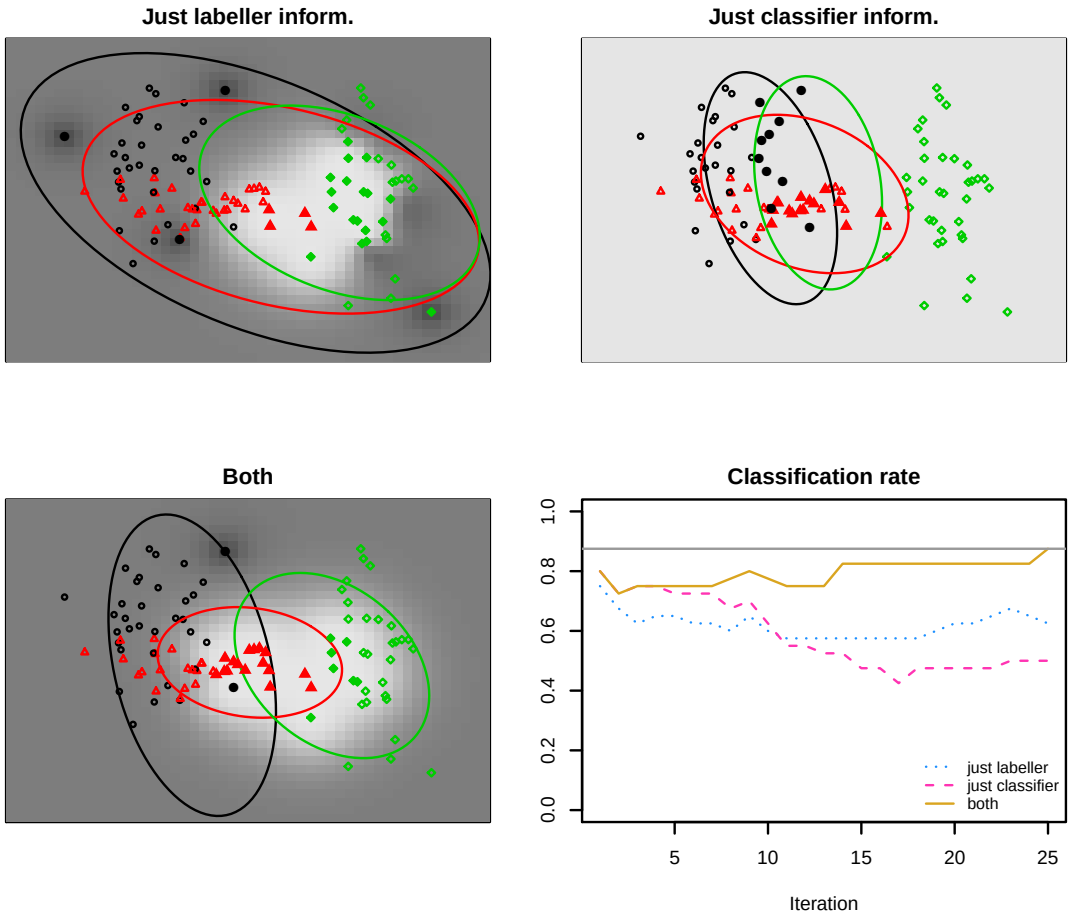

Figure 3: Three results after 25 steps depending on which query strategy was used, and their classification rates on a test set as a function of steps. The horizontal line is for full training data with an oracle.