A model for the labeler in active learning

A1, A2

October 17, 2013

Abstract

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

1 Basis model for active learning

Using the language of the 2010 survey by Settles.

Task: Train a classifier in a semi-supervised manner. We have a set $\mathcal{L}$ of labeled training items, and we want to improve the classifier by asking an labeler for some more labels.

The scenario: Pool-based sampling. We have a pool $\mathcal{U}$ of unlabeled items which we can use for querying the oracle for their label.

Query strategy: uncertainty sampling using entropy as query informativeness.

1.1 Uncertain labeling

For a given unlabeled item x, the labeler gives a label-probability vector instead of the true label (oracle) or a randomly sampled label (noisy oracle).

The idea: labeler is modelled as being uncertain of the label, and she can express this.

Model: Let $\mathcal{D} = \{(y, x)\}$ denote a set of items, where $y \in C$ with $|C| = m$ denote labels for the items, and $x \in \mathcal{X} \subseteq \mathbb{R}^p$ denote other features (covariates). Write for any x the correct label y_x.

Start with a training set of correcly labeled items $\mathcal{L} = \{(y_i; x_i)\}$. Denote $\mathcal{U} = \mathcal{D} \setminus \mathcal{L}$. The classifier model

$$\begin{aligned} y|x, \theta &\sim p(y|x, \theta) && (1) \\ \theta &\sim p(\theta) && (2) \end{aligned}$$

learns the θ from the $\mathcal{L}$. Active learning is about increasing the model fitness by querying most informative x to get y_x so as to optimally increase upon $\mathcal{L}$ with minimum number of queries. The basic setting assumes an *oracle*, an external labeler which provides the correct label for any query $x^\star \in \mathcal{U}$ and which we then use to enhance the estimate of θ.

Now define a *multivariate oracle*: when queried for a label for $x^\star \in \mathcal{U}$ the multivariate oracle's answer is a distribution over C instead of the correct label:

$$\begin{aligned} h^\star &\sim p(h^\star|x^\star,\alpha), \quad h^\star \in \mathcal{H} = \{h \in \mathbb{R}_+ : 0 < \sum_{c\in C} h_c < \infty\} && (3) \\ p(y^\star = c'|x^\star) &= h_{c'} / \sum h_c && (4) \\ \alpha &\sim p(\alpha) && (5) \end{aligned}$$

We get the oracle from this by setting

$$p(h|x^\star,\alpha) = 1(h_{y_{x^\star}} > 0, h_c \neq 0 \;\forall\; c \neq y_{x^\star}).$$

In other words, the active learning maps queries with functions

$$\begin{aligned} \text{oracle} \quad f_o(x^\star) &= y_{x^\star} && (6) \\ \text{multivariate oracle} \quad f_{co}(x^\star) &= h_{x^\star} && (7) \end{aligned}$$

How do we use the multivariate oracle for active learning? The oracle is used in the following algorithmic manner:

1. Compute the informativeness $I(x) := I(x|\theta,\mathcal{L})$ of each $x \in \mathcal{U}$
2. Augment $\mathcal{L} := \mathcal{L} \cup \{(f_o(x^\star), x^\star)\}$ where $x^\star = argmax\ I(x)$.
3. Check some stopping rule.

The multivariate oracle case suggest two approaches to the algorithm, of which both lead to the same probabilistic model. First is the 'heuristic' realisation approach, where we sample the label from the distribution provided by the oracle, viz.

2'. Augment $\mathcal{L} := \mathcal{L} \cup \{(\tilde{y}_{x^\star}, x^\star)\}$ where $x^\star = argmax\ I(x)$ and $\tilde{y}_{x^\star} \sim h_{x^\star}$.

This leads to a new source of noise in $\mathcal{L}$, which then is handled by the augmenting the classifier model.

The other option is to change the structure of the data. Let again $\mathcal{L}_t$ be the queried set by now consisting of pairs (h_x, x). Then

2". Augment $\mathcal{L} := \mathcal{L} \cup \{(h_{x^\star}, x^\star)\}$ where $x^\star = argmax\ I(x)$.

If $\mathcal{L}_0$ is the (clean) training set, and $\mathcal{L}_t$ is the queried set after t iterations, we add to the model (1)

$$\tilde{p}(h_x|x,\theta) = E_{\tilde{y}|h_x} p(\tilde{y}|x,\theta) \qquad (h_x, x) \in \mathcal{L}_t \qquad (8)$$

This is then used for inferring θ, and for deriving new values for $I(x)$.

2 Model for an uncertain labeler

Imagine an uncertain labeler that has a fixed area of knowledge, outside of which he knowns nothing. For illustration, let's imagine a sphere in the feature space inside of which the multivariate oracle knows the label for any point, and outside the sphere he simply replies by uniform weighting of the labels.

Now consider the case where an item outside the sphere is most informative. The item is queried, and the multivariate oracle returns an uniform distribution. How does the answer affect our system? For the Naive Bayes classifier it is easy to show that the equal allocation of evidence to different classes shifts the class-wise parameters towards their global counterparts. This is likelily to reduce entropy very little, in which case the same item is very likely to be picked for querying also in the next round.

2.1 Modeling labeler uncertainty

In order to avoid querying with 'useless' points we could try to model

$$h \sim p(h|x, \alpha).$$

For illustration we pick the Dirichlet distribution with some parameter field $\alpha(x) > 0 \; \forall x \in \mathcal{X}$. In the sphere example the idea would then be that inside the sphere $\alpha(x)$ is low, thus producing 'certain' responses, and outside the spehere $\alpha(x)$ is large, producing flat distributions. If we can learn the difference, we can guide our sampling to more generally useful queries.

The $\alpha(x) > 0$ is positive valued function of the features. Let's assume the features live in a continuous metric space so that we can do smoothing in it. Set

$$\alpha(x) \quad \sim \quad \log GP(x, \mu, \rho) \tag{9}$$

where ρ is a stationary covariance function. The observation likelihood for n samples, without the hyperparameters, becomes

$$\begin{aligned} p(h_1, ..., h_n | x_1, ..., x_n, \alpha) \quad &= \quad \prod p(h_i|\alpha(x_i))p(\alpha(\mathbf{x})) & (10) \\ & & (11) \end{aligned}$$

for which we can write $\alpha_i = \alpha(x_i)$ and get

$$\begin{aligned} l(\alpha; \mathbf{h}, \mathbf{x}) \quad \propto \quad & \sum_i [-\log B(\alpha_i) + \alpha_i \sum_k \log h_{ik}] - & (12) \\ & \sum_i [\log \alpha_i] - \frac{1}{2}(\log \alpha - \mu)^T Q(\log \alpha - \mu) & (13) \end{aligned}$$

where Q is the inverse covariance matrix, taken as known, and $B(\alpha) = \Gamma^K(\alpha)/\Gamma(K\alpha)$ is the Beta-function. Direct optimization with respect to α is straightforward as the function is convex.

2.2 Informativeness of the labeler

Imagine the labeler giving $h = h(x)$ vector as a response to a query x. The informativeness tag for an item x based on the entropy of h is then

$$I_o(x) := -E \sum_k h_k(x) \log h_k(x)$$

where $h(x)$ is a Dirichlet random variable with parameter $\alpha(x)$. Taking the expectation w.r.t. h leads to

$$I_o(x) = \log B(\alpha(x)) + K(\alpha(x) - 1)[\psi(K\alpha) - \psi(\alpha(x))]$$

where ψ is the digamma function.