Combining Active Learning and Accuracy Estimation using Unlabeled Data

Alexander Dushku II
Carnegie Mellon University
Pittsburgh, PA 15213
adushkui@andrew.cmu.edu

Derek Liu
Carnegie Mellon University
Pittsburgh, PA 15213
hsuehtil@andrew.cmu.edu

Ishani Chatterjee
Carnegie Mellon University
Pittsburgh, PA 15213
ichatter@andrew.cmu.edu

Abstract

This project tries to answer whether we can improve active learning models through combining the results of error estimation models. An error estimation model measures the performance of classifiers which perform classification using only unlabeled data through co-training. We invented several ways to capture "uncertainty" given the performance and probabilistic outputs of classifiers. With these new measures, we designed several active learning models and compared against standard uncertainty-based active learning model using entropy. As a results, incorporating information of error estimation improves the performance of active learning model.

1 Overview

Active learning is a class of algorithm that has access to a limited number of ground truth labels to make classifications. Active learning algorithms make use of the information they have to intelligently query more labels from an expert. A good active learning model is able to improve its classification as quickly as possible by querying as few labels as possible. Efficient querying is particularly important in cases where experts are rare, labeling is expensive or time intensive, and data is very large with high dimensionality. In this paper we consider an alternative approach to traditional active learning models.

Rather than use a traditional approach that leverages a limited number of data labels to determine which need to be queried next, our approach uses multiple unsupervised learning algorithms to determine points to query. In this paper we will leverage the error estimation model presented in [Pla14] in order to supply those points. We will evaluate the possibility that the error estimation model can produce more efficient querying than a maximum uncertainty based query.

In this paper we formulate many different query strategies instead of using uncertainty-based query and present a comparison of performance of these strategies with the base-line uncertainty-based Active Learning evaluated on a subset of the NELL dataset.

2 Related Work

2.1 Uncertainty Based Active Learning (AL)

Uncertainty based active learning is a machine learning algorithm with aims of achieving greater accuracy with fewer training labels by querying the data points that will help it best learn the distinction between classes. These queried points are then labeled by an expert.

Active learning is used in three main scenarios: (1) Membership Query Synthesis: The active learner generates data by itself and queries for labels, (2) Stream-Based Selective Sampling: The active

learner obtains labeled data points sequentially, assuming getting labeled data is inexpensive, and then decides to further ask for labels or stop. (3) Pool-Based Sampling: In this scenario, the active learner can see small amounts of labeled data and a large pool of unlabeled data. Then it will pick data points from an unlabeled pool for the query. Because Pool-Based Sampling is applicable in most of the practical problems, it is the most common of the three scenarios.

There are many different strategies to perform querying, such as Uncertainty Sampling, Query-By-Committee, and so on. Because discussing different query strategies is not the main focus of this project, we refer to active learning survey paper [Set10] for more comprehensive discussion. Here we are focusing on query points based on uncertainty. An important thing to mention is that, uncertainty based querying may not be the best strategy because it tends to "throws away" information about the remaining label distribution. In other words, if active learning models fail to "see" data from other distributions, it would probably fail to query points from that distribution. There are other methods designed for solving this problem, such as margin sampling [SDW01]. But it still fails when facing large datasets.

In this project, we are considering the pool-based sampling scenario with uncertainty based query strategy. And we use Logistic Regression to classify NELL dataset [Mit10] with the expectation that the high dimensions of the data make it linearly separable.

2.2 Error Estimation (EE)

Most existing approaches to estimate performance of classifiers are supervised, which means that accuracy of classifiers comes from comparing true labels with predicted labels. However, much of existing data is unlabeled. Estimating accuracy of classifiers using only unlabeled data becomes an important issue in machine learning and many other fields.

A common scenario to build error estimation models is called "multiple approximations" setting, where you only have unlabeled data and predict results from multiple classifiers of the data. An error estimation model tries to gather this information to predict the performance of each classifier in terms of error rate.

The most intuitive approach to estimating error is called Majority Vote Approach which assumes that each classifier makes individual errors independently. The intuition of measuring error rate based on this assumption is that the classifier which disagrees with most of the other classifiers tends to have a higher error rate. However, this assumption may not be correct in practice because classifiers are not independent in most cases. For example, if two students are answering a difficult question, they both are more likely to answer the question incorrectly than an easier question. Therefore, we need to build an error estimation model without assuming independent errors, which is the aim of this project.

In order to have more accurate error estimation models, our work follows a recent paper in Machine Learning field [Pla14]. Its approach is using graphical models to capture dependency of making errors and it does achieve better results. Other methods for building error estimation models exist in medical research, because the data they face is mostly unlabeled. For more detail about error estimation research in this area, please refer to our cited survey paper [CH14].

3 Data Description

We evaluate our approach using Never-Ending Language Learning (NELL) dataset [Mit10]. NELL contains over 50 million candidate beliefs by reading the web. It encodes each word into a set of features, and the classifier will predict whether this word belongs to this belief based on the word features extracted by reading the web. For example, assume the input word and belief are "Carnegie Mellon" and "University" respectively , the classifier would output the possibility of "Carnegie Mellon" belonging to category "University". More details about NELL dataset can be found in [CBKSHM10].

In order to perform error estimation, we also need classification results from at least three different classifiers. The classifiers we used are: (1) CPL: A Logistic Regression classifier that uses words and phrases that appear with the noun phrase as features, (2) CPL KB: A knowledge based language

classification model, (3) SEAL: A machine learning model for natural language processing designed by Seal's ML team (4) CMC: A Logistic Regression classifier that considers orthographic features of the noun phrases.

In terms of using NELL data, we treat the classifier prediction results as the input of error estimation models. We also use the encoded word features to train Logistic Regression for active learning models using uncertainty-based query as well as our own formulated query strategies. We considered the "Country" category within the dataset.

4 Proposed Method

4.1 Error Estimation

In this project, we consider a "multiple approximations" setting mentioned above. An error estimation model outputs estimated error rates, which represent performance, for each classifier by jointly analyzing the predicted outputs from those classifiers. In order to capture dependency between making errors across models. We use the error estimation model called Coupled Error Estimation presented in [Pla14]. The model performs much better than majority vote approach, because it incorporates the graphical model to capture the dependency of making errors between classifiers and across different categories of the NELL dataset.

Coupled Error Estimation Model (CEE)
We consider the problem setting in which we have several different approximations $f_1, \cdots, f_N,\ f : X \to \{0, 1\}$ from N different classifiers, and we wish to predict accuracy of each classifier using only unlabeled data. The model can also generate the most likely single label given different approximations, but we are not going to use this to predict the most likely label because our project is to try to use error estimation to improve active learning.

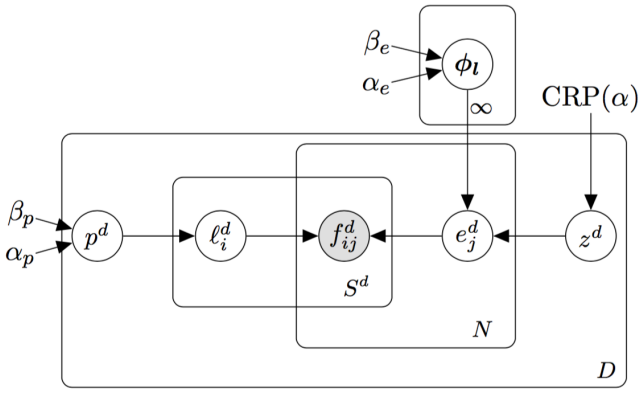

Figure 1: Probabilistic graphical model for error rate estimation using only unlabeled data.

In this model, we assume there is a prior knowledge, $p^d \sim \text{Beta}(\alpha_p, \beta_p)$, of the true label being equal to 1. For each data point, there is a true, unobserved label $l_i^d \sim \text{Bernoulli}(p^d)$. We further assume there is another underlying distribution for potential error rates, $[\phi_l]j \sim \text{Beta}(\alpha_e, \beta_e)$ and error rate e_j^d is determined using $[phi_{z^d}]_j$. This model also incorporates the Chinese Restaurant Process (CRP) to assign clusters $z^d \sim CRP(\alpha)$ to each domain. Finally, we can derive output of function approximation according to $\hat{f}_{ij}^d = \{l_i^d, 1 - l_i^d\}$. In terms of training, we perform Gibbs Sampling [GG84] to sequentially sample data from those distributions. The sampling processes are showed in Figure.1

In terms of training, we perform Gibbs Sampling [GG84] to sequentially sample data from those distributions. Infinite sampling would theoretically produce the true distribution. The sampling processes are as follow:

$$P(p^d \mid \cdot) = \text{Beta}(\alpha_p + \textstyle\sum_{i=1}^{S^d} l_i^d, \beta_p + S^d - \sum_{i=1}^{S^d} l_i^d)$$

$$P(l_i^d \mid \cdot) \propto (p^d)^{l_i}(1-p^d)^{1-l_i^d} \textstyle\prod_{j=1}^{N} (e_j^d)^{\mathbf{1}_{\{\hat{f}_{ij}^d \neq l_i^d\}}} (1-e_j^d)^{\mathbf{1}_{\{\hat{f}_{ij}^d \neq l_i^d\}}}$$

$$P(z^d = k \mid \cdot) \propto Z_k^d \textstyle\prod_{j=1}^{N} (e_j^d)^{\sigma_j^d}(1-e_j^d)^{S^d-\sigma_j^d} \quad \text{if } X_k^d > 0$$

$$P(z^d = k \mid \cdot) \propto \alpha \frac{\mathcal{B}(\alpha_e+\sigma^d, \beta_e+S^d-\sigma_d)}{\mathcal{B}(\alpha_e,\beta_e)} \quad \text{otherwise}$$

$$P([\phi_k]_j \mid \cdot) = \text{Beta}(\alpha_e + \textstyle\sum_{d=1}^{D} \mathbf{1}_{\{z^d=k\}}\sigma_j^d, \beta_e + \sum_{d=1}^{D} \mathbf{1}_{\{z^d=k\}}(S^d - \sigma_j^d))$$

where:

$$Z_k^d = \textstyle\sum_{\hat{d}=1, \hat{d}\neq d}^{D} \mathbf{1}_{\{z^{\hat{d}}=k\}}$$

$$\sigma_j^d = \textstyle\sum_{i=1}^{S^d} \mathbf{1}_{\{\hat{f}_{ij}^d \neq l_i^d\}}$$

4.2 Standard Uncertainty Query Strategy

Ultimately all our querying strategies will be compared to the common strategy based on uncertainty sampling. Our goal is to design a learner using predictions made by other classifiers and their respective error estimation rates that can outperform uncertainty sampling.

4.3 Alternative Query Strategies

We define five query strategies QS1, QS2, QS3, QS4, and QS5, purely based on predictions of other unsupervised classifiers and their error estimates. We also define a combination model that samples based on uncertainty simultaneously with querying based on QS1, QS2, QS3, QS4, and QS5, each paired individually with uncertainty-guided sampling to boost its performance. Once we have defined these strategies we will evaluate their performance empirically.

QS1: Best Classifier Uncertainty (B)
In this method we query the points that are most uncertain according to the most accurate (least estimated error rate) unsupervised classifier. This is perhaps the most naive way to query based on error estimation. The point to query QP is given by:

$$QP = \text{argmin}_n \mid p^n_{error_{min}} - 0.5 \mid$$

Where $p^n_{error_{min}}$ is $P(y^n = 1 \mid x^n)$ output by the classifier with lowest error rate.

QS2: Accuracy Weighted Uncertainty (WSP)
This method combines all the classifiers' output weighted by their accuracies. For every point n, its score Score^n is computed:

$$\text{Score}^n = \tfrac{1}{Z} \textstyle\sum_i (1-e_i) \times p_i^n \qquad (1)$$

Where Z is normalization factor, e_i are the error rates of classifier i, p_i^n is the $P(y^n = 1 \mid x^n)$ predicted by classifier i for point n. You then query a point QP according to the most uncertain points in this combined measure:

$$QP = \text{argmin}_n \mid \text{Score}^n - 0.5 \mid$$

This takes advantage of the independence of the errors of all classifiers to identify the points that are most difficult to classify for our model.

QS3: Accuracy Weighted Mean Variance (WMV)
Instead of just looking at the accuracy weighted average of the classes' output we can also query based on the variance of their predictions. For example, given four unsupervised classifiers, two could be very confident about classifying an example as 1 and the remaining two could be very confident classifying as 0. In this case, a high variance would be observed in prediction probabilities among classifiers. We computed a score by weighting the measures for mean and variance by λ

and tested a range of λs in order to determine the optimal relative weighting for each. In QS3, the mean and variance are the accuracy-weighted mean and accuracy-weighted variance of prediction probabilities. To be precise, the score Score$_n$ for each point is computed as:

$$\text{Score}_n = (1 - \lambda) \times (\text{mean}(P^n) - 0.5) - \lambda \times \text{var}(P^n)$$

Where the weighted meanP^n is computed the same was as Score$_n$ in eq.(1). And the weighted variance of nth data is

$$\text{var}P^n = \frac{1}{Z} \sum_i (1 - e_i)(p_i^n - \text{mean}(P^n))^2$$

Then the query point becomes the point with the lowest Score$_n$.

QS4: Unweighted Mean Variance (MV)
This method is almost identical to the one described above except that rather than calculating the accuracy weighted mean and variance of each classifier we use the unweighted mean and subsequently use this unweighted average to determine the unweighted variance. Again we determine the ideal weights for λ experimentally.

To be precise, the score Score$_n$ for each point is computed as:

$$\text{Score}_n = (1 - \lambda) \times (\text{mean}(P^n) - 0.5) - \lambda \times \text{var}(P^n)$$

Where mean(P^n) and var(P^n) are the unweighted mean and variances of the predicted probabilities of classifier. Then the query point becomes the point with the lowest Score$_n$.

QS5: Accuracy Weighted Distance from Uncertainty Threshold (WSD)
In this method we attempt to capture the degree to which each classifier has given a score close to 0.5. The accuracy weighted average described above would give the same value for classifier output of 0.5, 0.5, 0.5, 0.5 and for 1, 0, 1, 0. This strategy attempts to capture the difference between those two examples. For each classifier we take the absolute value of the probability output minus 0.5 and then take the accuracy weighted average of them all. This will give a very low value for 0.5, 0.5, 0.5, 0.5 and a very high value for 1, 0, 1, 0. Then we query the lowest values first for this strategy.

Combined Uncertainty and Error Estimation Querying
It stands to reason that there are relative strengths and weaknesses for an EE and more traditional active learning approach. Because of this some mixture of querying strategies could be ideal. The idea behind combining error estimation is that it can provide us a measure of uncertainty for each data point independent of the logistic regression decision boundary. Therefore, we expect it to solve the problem of not detecting other distributions. Also, error estimation may be able to enhance query quality compared to traditional uncertainty measures within the first few iterations before the true boundary is learned.

To combine the uncertainty and error estimation querying we allow our model to query data points in equal measure based on the standard uncertainty based active learning technique and each of the error estimation techniques in turn. We will determine which of these combination models will provide the best learner.

5 Results and Discussion

5.1 Error Estimation v.s. Standard Uncertainty Querying

Here we compare the results for error estimation based queries with standard uncertainty based querying. All models begin with the same seed of 300 randomly selected points to be labeled. From there the EE models queried 10 points with their respected query strategies mentioned above. The uncertainty model likewise queried 10 points according to the points closest to its decision boundary. We chose a step size of 10 because it was the smallest grain that we had the computational resources to handle.

The results of this empirical test are clear: while the MV and WMV initially outstripped the standard active learning model, it very quickly caught up and surpassed all the EE models. This makes

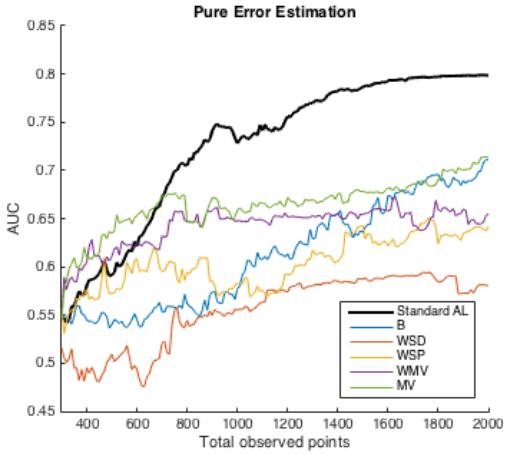

Figure 2: Comparing querying only based on error estimation and standard uncertainty querying

intuitive sense because while the points sampled by EE may be difficult to classify, they may not be representative of the actual decision surface and certainly aren't representative of the general distribution of the data.

5.2 Combining EE with Standard Uncertainty Querying

Next let's consider a similar experiment, but instead of querying only according to EE strategies, we combine EE sampling with standard uncertainty sampling. We sample 5 data points based on an EE strategy and 5 based on the AL uncertainty strategy. Figure.3 summarizes standard uncertainty-based query combined with all EE strategies except MV and WMV. In Figure.4 we see the results of

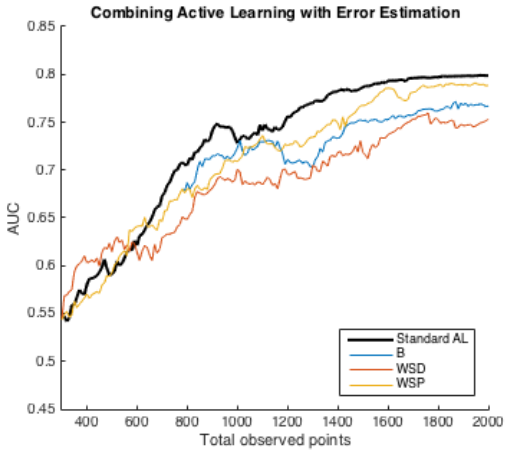

Figure 3: Combining EE and standard uncertainty querying

combining MV and AL sampling for different values of λ. We observe that combining the standard AL with all other EE strategies does not improve performance but combining with MV approach provides provides some improvement, especially in the early stages of active learning. The MV strategy outperforms other EE strategies, indicating that it is the one that, when combined with active learning, queries points that are near to the true boundary of uncertainty of ground truth data. Because the MV method captures relative disagreement between the classifiers and not just how much different they predict from an absolute threshold, i.e., 0.5, this measure of taking relative disagreement into account captures the true uncertainty of points.

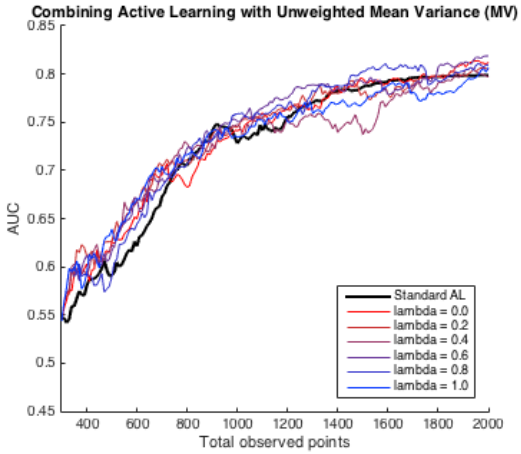

Figure 4: Combining MV and standard uncertainty querying

With WMV the results are even clearer, Figure.5. A combined approach significantly improves active learning in the early stages of the algorithm and while the relative difference diminishes as both algorithms gain access to more data, the WMV mixed approach continues to have a slight edge. The results vary a little for different lambda values and some additional work could help us determine with greater confidence what the relationship is between λ values and the performance of the algorithm.

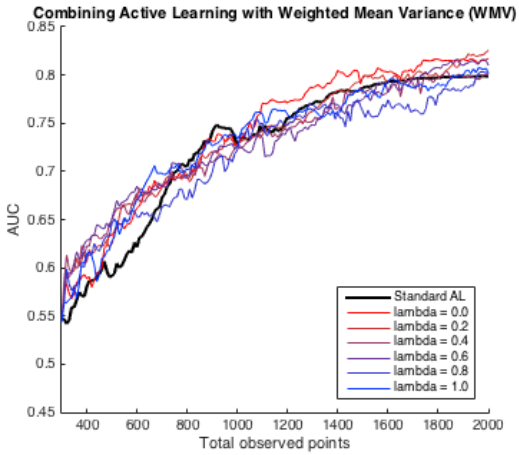

Figure 5: Combining WMV and standard uncertainty querying

6 Future Work

Determine Ideal Values For λ
Due to computational constraints we were unable to run the kinds of comprehensive tests on the values for λ in the WMV and MV querying strategies. Intuitively λ represents the tradeoff between querying points that the classifiers all struggle to classify (mean) and the points they disagree on (variance). Both would appear to be valuable points to query but each may be more important at different stages of active learning. It may, for example, turn out that the mean is more important for querying in the early stages of active learning but that later in the variance is more useful.

Generalize Results to Different NELL Datasets
There is the risk that our results are not generalizable to different datasets within NELL. By running the exact same tests on many datasets we could begin to see how well the mixed strategy generalizes for language learning applications.

Alternative Querying Strategies
We evaluated a 50/50 combination of EE and AL querying. However, we have no reason to believe that this is the optimal ratio between these two strategies. Experimentation could find that optimal combination. Additionally we could explore other established active learning strategies to build alternative ensemble querying strategies.

7 Conclusion

The AL querying strategy performs well and despite our initial optimism, we were unable to outperform that strategy with a pure EE strategy. However, by combining the two strategies we were able to improve the learning rate significantly especially in the early stages. This querying strategy provides an additional tool for active learning that can help reduce the cost of training models on very large datasets by reducing the number of points an expert must label.

References

[BM73] Blackwell, D., & MacQueen, J. B. (1973). Ferguson distributions via Plya urn schemes. The annals of statistics, 353-355.

[CBKSHM10] Carlson, A., Betteridge, J., Kisiel, B., Settles, B., Hruschka Jr, E. R., & Mitchell, T. M. (2010). Toward an Architecture for Never-Ending Language Learning

[CH14] Collins, J., & Huynh, M. (2014). Estimation of diagnostic test accuracy without full verification: a review of latent class methods. Statistics in Medicine, 33(24), 4141-4169.

[GG84] Geman, S., & Geman, D. (1984). Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. Pattern Analysis and Machine Intelligence, IEEE Transactions on, (6), 721-741.

[Mit10] Mitchell, T. (2010). Never-ending learning. Carnegie Mellon University Pittsburgh PA.

[Pla14] Platanios, E. A., Blum, A., & Mitchell, T. (2014). Estimating accuracy from unlabeled data.

[SDW01] Scheffer, T., Decomain, C., & Wrobel, S. (2001). Active hidden markov models for information extraction. In Advances in Intelligent Data Analysis (pp. 309-318). Springer Berlin Heidelberg.

[Set10] Settles, B. (2010). Active learning literature survey. University of Wisconsin, Madison, 52(55-66), 11.