Purchase of data labels by batches: study of the impact on the planning of two active learning strategies

V. Lemaire, A. Bondu, and F. Clérot

France Télécom R&D Lannion, TECH/EASY/TSI
http://perso.rd.francetelecom.fr/lemaire
vincent.lemaire@orange-ftgroup.com

Abstract. Active machine learning algorithms are used when large numbers of unlabelled examples are available and getting labels for them is costly (e.g. requires a human expert). Active learning methods select examples to build a training set for a predictive model and aim at the most informative examples. The number of examples to be labelled at each iteration of the active strategy is, most often, randomly chosen or fixed to one. However in practical situations, this number is a parameter which influences the performance of the active strategy. This paper studies the influence of this parameter on two active learning strategies.

1 Introduction

Machine learning consists of methods and algorithms which learn behavior to a predictive model, using training examples. Passive learning strategies use examples which are randomly chosen. Active learning strategies allow the predictive model to constructs its training set in interaction with a human expert. The learning starts with few labelled examples. Then, the model selects the examples with no label which it considers the most informative and asks their desired associated outputs to the human expert. The model learns faster using active learning strategies, reaching the best performances using less data. Active learning is more specifically attractive for applications for which data is expensive to obtain or to label.

Active learning strategies are also useful on “new problem”, for instance classification problem where informative examples or informative data are unknown. The question is how to obtain the information required to solve this new problem? An operational planning of an active algorithm applied on a “new classification problem” could be defined as the addition on individual cost, individual step, which allow to catch information to solve this “new problem”:

- (I) an initialisation : which, how and how many labels have to be buy at the beginning (before the first learning)).
- (PP) a pre-partition [1];
- at each step of the active strategy:

  - (PS) a pre-selection [2];
  - (D) a diversification [3];
  - (B) the purchase of N example(s) (customarily $N = 1$)
  - (E) the iteration evaluation [4];
- (M) the model used.

Planning the purchase of new examples (per packages) is a compromise (C) between these different steps which include the dilemma between exploration [5] and exploitation [6], such that:

$$C = EW(\alpha_1 I + \alpha_2 PP + \alpha_3 PS + \alpha_4 D + \alpha_5 B + \alpha_5 E + \alpha_6 M)$$

where EW is the evaluation of the overall procedure. The quality of an active strategy is usually represented by a curve assessing the performance of the model versus the number of training examples labelled .

For the conception of an automatic shunting system (for phone servers) which takes into account emotions in speech [7] (our "new problem") this approach can be used. In this case, data is composed by turn of speech which are exchanged between users and the machine. Each piece of data has to be listened by a human expert to be labelled as containing (or not) negative emotions. The purpose of active strategies which are considered in this article is to select the most informative unlabelled examples. These approaches minimize the labelling cost inducted by the training of a predictive model. For the conception of automatic shunting system (for phone servers), which takes into account emotions in speech, our corpus contains more than 100000 turns of speech. Therefore the operational planning is very important.

Two main active learning strategies are used in the literature (see section 2). We suspect that such active learners are good for "exploitation" (labelling examples near the boundary to refine it), but they do not conduct "exploration" (searching for large areas in the instance space that they would incorrectly classify); even worse than the random sampling when labels are bought by packet. One way to examine the "exploration" behavior of these two main strategies is to buy more than one label at every iteration (the "weight" of α_5 above), this is the purpose of this paper.

2 Active Learning

2.1 Notations

$\mathcal{M} \in \mathbb{M}$ is the predictive model which is trained using an algorithm $\mathcal{L}$. $\mathbb{X} \subseteq \mathbb{R}^n$ represents all the possible input examples of the model and $x \in \mathbb{X}$ is a particular example. $\mathbb{Y}$ is the set of the possible outputs of the model; $y \in \mathbb{Y}$ a class label related to $x \in \mathbb{X}$.

During its training, the model observes only one part $\Phi \subseteq \mathbb{X}$ of the universe. The set of examples is limited and the associated labels are not necessarily known. The set of examples for which the labels are known (at a step of the training algorithm) is called L_x and the set of examples for which the labels are unknown is called U_x with $\Phi = U_x \cup L_x$ and $U_x \cap L_x \equiv \emptyset$.

The concept which is learned can be seen as a function, $f : \mathbb{X} \rightarrow \mathbb{Y}$, with $f(x_1)$ is the desired answer of the model for the example x_1 and $\widehat{f} : \mathbb{X} \rightarrow \mathbb{Y}$ the answer obtained of the model; an estimation of the concept. The elements of L_x and the associated labels constitute a training set T. The training examples are pairs of input vectors and desired labels such as $(x, f(x)) : \forall x \in L_x, \exists (x, f(x)) \in T$.

2.2 Active Learning Methods

Introduction The point of view of selective sampling is adopted [8] in this article. The model observes only one restricted part of the universe $\Phi \subseteq \mathbb{X}$ which is materialized by training examples without label. The image of a *"bag"* containing instances for which the model can ask associated labels is usually used to describe this approach.

```
Considering:
    • $\mathcal{M}$ a predictive model provided with a training algorithm $\mathcal{L}$
    • $U_x$ and $L_x$ the sets of examples respectively not labelled and labelled
    • $n$ the desired number of training examples
    • $T$ the training set with $\|T\| < n$
    • $\mathcal{U} : \mathbb{X} \times \mathbb{M} \rightarrow \Re$ the function which estimates the utility of an example for
      the training of the model

Repeat
    (A) Train the model $\mathcal{M}$ using $\mathcal{L}$ and $T$ (and possibly $U_x$).
    (B) Find the example such that $q = argmax_{u \in U_x} \mathcal{U}(u, \mathcal{M})$
    (C) Withdraw $q$ from $U_x$ and ask the label $f(q)$ from the expert.
    (D) Add $q$ to $L_x$ and add $(q, f(q))$ to $T$
until $\|T\| < n$
```

Algorithm 1: Selective sampling, Muslea 2002

The problem of selective sampling was posed formally by Muslea [9] (see Algorithm 1). It uses a utility function, $\mathcal{U}tility(u, \mathcal{M})$, which estimates the utility of an example u for the training of the model $\mathcal{M}$. Using this function, the model selects examples for which it hopes the greatest improvement of its performances, and shows these examples to the expert.

The Algorithm 1 is generic insofar as only the function $\mathcal{U}tility(u, \mathcal{M})$ must be modified to express a particular active learning strategy. How to measure the interest of an example will be discussed now.

Uncertainty sampling is an active learning strategy [10] which is based on the confidence that the model has in its predictions. The model must be able to produce an output and to estimate the relevance of its answers. The model estimates the probability of observing each class, given an instance $x \in \mathbb{X}$. This estimate is done selecting the class which maximizes $\hat{P}(y_j|x)$ (with $y_j \in \mathbb{Y}$) among all possible classes. The weaker the probability to observe the predicted class, the more prediction is considered uncertain. This strategy of active learning selects unlabelled examples which maximize the uncertainty of the model. The uncertainty can be expressed as follow :

$$\mathcal{U}ncertain(x) = \frac{1}{argmax_{y_j \in \mathbb{Y}} \hat{P}(y_j|x)} \qquad x \in \mathbb{X}$$

Sampling by risk reduction The purpose of this approach is to reduce the generalization error, $E(\mathcal{M})$, of the model [11]. It chooses examples to be labelled so as to minimize this error. In practice this error cannot be calculated because the distribution of instances in $\mathbb{X}$ is unknown. Nicholas Roy [11] shows how to bring this strategy into play since all the elements of $\mathbb{X}$ are not known. He uses a uniform prior for $P(x)$ which gives :

$$\widehat{E}(\mathcal{M}^t) = \frac{1}{|L|} \sum_{i=1}^{|L|} \mathcal{L}oss(\mathcal{M}^t, x_i)$$

In this article, one estimates the generalization error ($E(\mathcal{M})$) using the empirical risk [12] given by:

$$\hat{E}(\mathcal{M}) = R(\mathcal{M}) = \sum_{i=1}^{|L|} \sum_{y_j \in \mathbb{Y}} \mathbb{1}_{\{f(x_i) \neq y_j\}} \, P(y_j|x_i) P(x_i)$$

where f is the model which estimates the probability that an example belong to a class, $P(y_i|x_i)$ the real probability to observe the class y_i for the example $x_i \in L$, $\mathbb{1}$ the indicating function equal to 1 if $f(x_i) \neq y_i$ and equal to 0 else. Therefore $R(\mathcal{M})$ is the sum of the probabilities that the model makes a bad decision on the training set (L).Using a uniform prior to estimate $P(x_i)$, one can write :

$$\hat{R}(\mathcal{M}) = \frac{1}{|L|} \sum_{i=1}^{|L|} \sum_{y_j \in \mathbb{Y}} \quad \mathbb{1}_{\{f(x_i) \neq y_j\}} \, \hat{P}(y_j|x_i)$$

In order to select examples, the model is re-trained several times considering one more “fictive” example. Each instance $x \in U$ and each label $y_j \in \mathbb{Y}$ can be associated to constitute this supplementary example. The expected cost for any single example $x \in U$ which is added to the training set is then:

$$\hat{R}(\mathcal{M}^{+x}) = \sum_{y_j \in \mathbb{Y}} \hat{P}(y_j|x) \hat{R}(\mathcal{M}^{+(x,y_j)}) \quad with\, x \in U$$

Note - The two strategies described above are not the only ones which exist. The reader can see a third main strategy which is based on Query by Committee [13] and a fourth one where authors focus on a model approach to active learning in a version-space of concepts [14, 15].

3 Number of labelled examples at every iteration

In practice, the number of labelled examples at every iteration (noted n) is chosen in an arbitrary way. Nevertheless, this parameter influences the implementation of an active learning strategy. To understand the stakes of this problem, let us consider both extreme situations. On the one hand the computation time necessary for the examples selection "explodes", labelling a single example at each iteration. In this case, the application of active learning strategies to large data bases becomes problematic. The waiting time to present an example to the human expert is too long and becomes unreasonable. On the other hand the contribution of an active learning strategy decreases, labelling a large number of examples at every iteration. The regulation of the parameter n can be seen in an intuitive way as the research for a compromise between the computation time and the efficiency of an active learning strategy.

Since the purpose here is to measure the influence of the value on n. The experiments were carried out on several classification problems, using the same model and the two strategies defined in previous section.

3.1 Evaluation criteria

The criterion which is used to estimate model performances is the area under ROC curve [16] (AUC). ROC curves are usually built considering a single class. Consequently, one handles as many ROC curves there are classes. To build ROC curves in a m classes problem, one considers a meta-class $Y_1 = y_i$ (which is the target), others classes constitute the second meta-class $Y_2 = \bigcup_{j=1\ ,\ j \neq i}^{m} y_j$. AUC is calculated for each ROC curve, and the global performance of the model is estimated by the mathematical expected value of AUC, over all classes :

$$AUC_{global} = \sum_{i=1}^{|\mathbb{Y}|} P(y_i).AUC(y_i) \quad (1)$$

AUC can be seen as a proportion of the space in which ROC curves are defined. This area is equal to 1 if the model is perfect and is equal to $\frac{1}{2}$ for random models. AUC has interesting statistical properties. It corresponds to the probability that the model attributes a more important score, to an instance belonging to the good class, than an instance of another class [16].

3.2 Protocol

Beforehand, data is normalized using mean and variance. At the beginning of experiments, the training set contains only two labelled examples which are randomly chosen among available data. At each iteration, n examples are drawn in the data set to be labelled and added to the training set. The first series of experimentation adds 1 example at each iteration using an active strategy. Then four other series of experimentation are repeated by increasing, every time, the number of added examples; the quantity of information bring to the model (n=1, 4, 8, 16).

The classifier is a Parzen window which uses a Gaussian kernel (σ, the parameter of the kernel is adjusted using a cross validation as in [17]). Each experiment has been done ten times in order to obtain an average and a variance, for every point of the result curves.

3.3 Used model

The large range of models which are able to solve classification problems and sometimes the great number of parameters useful to use them, may represent difficulties to measure the contribution of a learning strategy.

A Parzen window, with a Gaussian kernel [17], is used in experiments below since this predictive model uses a single parameter and is able to work with few examples. The "output" of this model is an estimate of the probability to observe the label y_j conditionally to the instance u:

$$\hat{P}(y_j|u) = \frac{\sum_{n=1}^{N} \mathbb{1}_{\{f(l_n)=y_j\}} K(u,l_n)}{\sum_{n=1}^{N} K(u,l_n)} \qquad \textit{with } l_n, \in L_x \textit{ and } u \in U_x \cup L_x \quad (2)$$

where

$$K(u,l_n) = e^{\frac{||u-l_n||^2}{2\sigma^2}}$$

First, a Parzen window has been realized using all training example to estimate if this model is able to solve the problem. For the three databases the answer has been positive (a good value on the AUC has been obtained). Consequently, Parzen windows are considered satisfying and valid for the following active learning procedures with regards the influence of n.

The optimal value of the kernel parameter was found using a cross-validation on the average quadratic error, using all available training data [17]. Thereafter, this value is used to fix the Parzen window parameter. Since the single parameter of the Parzen window is fixed, the training stage is reduced to count instances (within the support of the Gaussian kernel). The strategies of examples selection are thus comparable, without being influenced by the training of the model.

4 Experimentations

4.1 Database

Three public data sets which come from the *"UCI repository"* (`http://www.ics.uci.edu/~mlearn/MLRepository.html`) are used :

- **Glass Identification Database:** Classification of 6 types of glass defined in terms of their oxide content (i.e. Na, Fe, K, etc). All attributes are numeric-valued. This data set includes 214 instances (Train: 146, Test: 68) characterized by 9 attributes which are continuously valued. The 6 classes are the type of glass. The parzen window classifies an example to one of these 6 classes.
- **Iris Plant Database:** The data set contains 3 classes of 50 instances each, where each class refers to a type of iris plant. This data set includes 150 flowers (Train: 90, Test: 60) described by 4 attributes which are continuously valued. The parzen window classifies an example to one of these 3 classes.
- **Image segmentation Database:** The instances were drawn randomly from a database of 7 outdoor images. The images were hand segmented to create a classification for every pixel. This database includes 2310 images characterized by 9 pixels (Train: 310, Test: 2000).The parzen window classifies an example to one of these 7 classes

For the three data sets, which contain more than two classes, the performances are evaluated using equation 1.

4.2 Results

Figures 1, 2, 3 show obtained results on the three data sets. On every figure: (i) from up to down and left to right: 1, 4, 8 or 16 examples added at each iteration of the active algorithm; (ii) on each sub-figure horizontal and vertical axis represent respectively the number of examples labelled used and the AUC (see section 3.1). On each curve test results using sampling based on uncertainty, sampling based on risk reduction and random sampling are plotted versus the number of examples labelled in the training set. The natches represent the variance of the results ($\pm 2\sigma$). Results on AUC show that, on these three data sets it is difficult to point to a strategy. If we consider that adding:

- one example at every iteration: the uncertain strategy wins on Glass but the risk strategy wins on the others data sets.
- four examples at every iteration: the uncertain strategy wins on Glass but the risk strategy and the random strategy share the success on the others data sets.
- eight or sixteen examples at every iteration: the random strategy wins on the three data sets.

By increasing the number of examples labelled at each iteration, the active strategies are less and less competitive compared to the random strategy. We notice each time that: (i) the results do not look so different for different batch sizes (but active strategies allow to obtain the optimal AUC with a smaller number of examples) (ii) the random strategy becomes more powerful than the two active strategies when n becomes large, particularly for n $\geq$ 8.

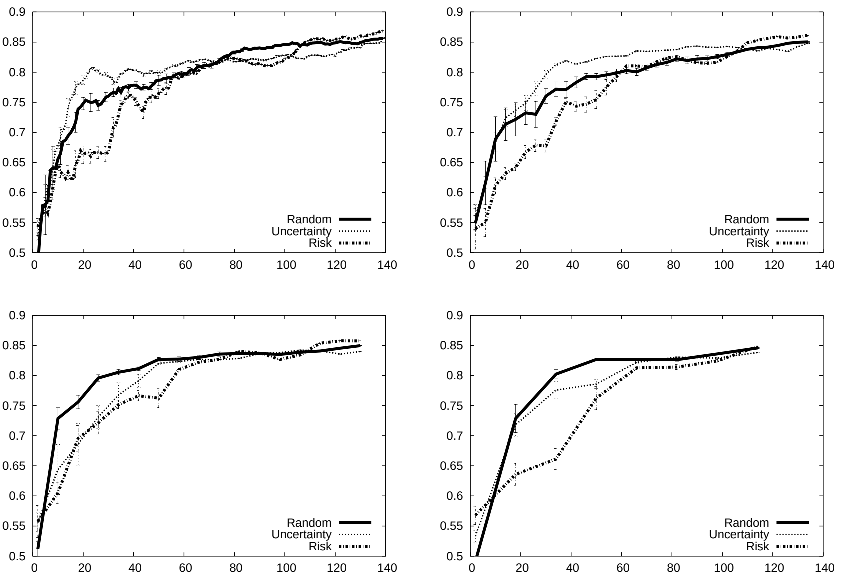

Fig. 1. Results on the data set Glass

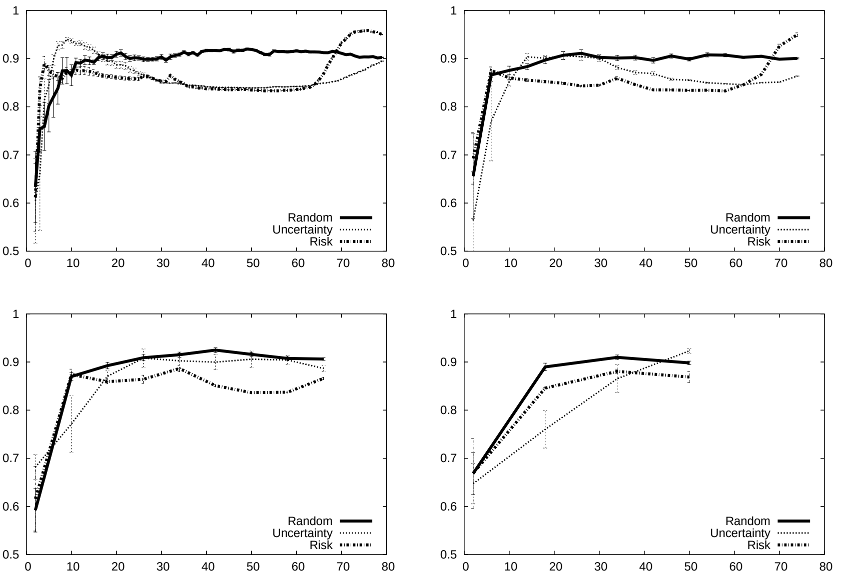

Fig. 2. Results on the data set Iris

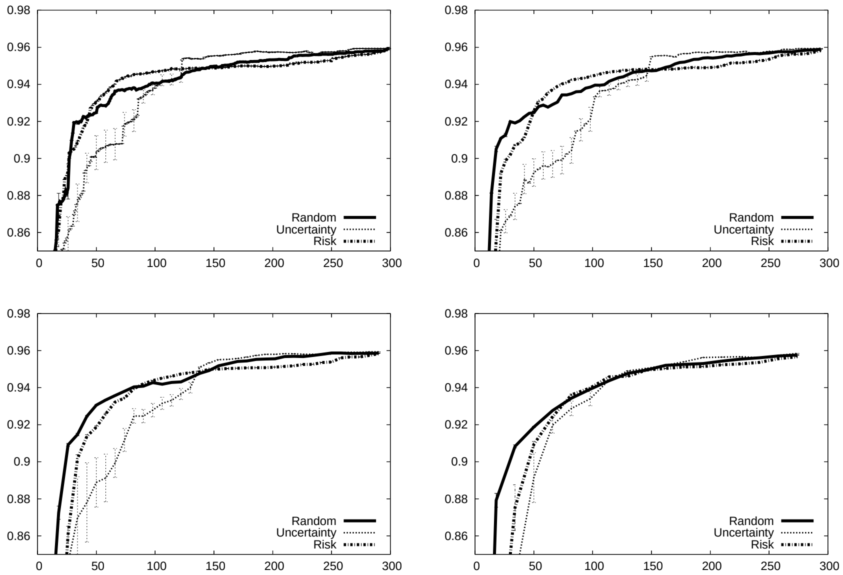

Fig. 3. Results on the data set Segment

5 Conclusion and future works

The obtained results show that the number of labelled examples at each iteration of a procedure of active learning influences the quality of the involved model. The experiments which were carried out confirm the intuition that the contribution of an active strategy (relatively to the random strategy) decreases when one increases the number of labelled examples. Methods which allow to buy at each iteration the same number (n>1) number of labels exist [18, 3] to try to incorporate a part of exploration. But to our knowledge none optimizes the value of N into each iteration (which choose a variable number of examples and/or which select "packages" of examples in an optimal way) . We are currently interested on this subject: for example the concept of "trajectory" of a model in the space of the decisions it has to take during its training.

The elaboration of a criterion (EW) the evaluation (which measures the contribution of a strategy compared to the random strategy on the whole data set) should be interesting: the performance criterion used can take several different ways according to the problem. This type of curve allows only comparisons between strategies in a punctual way, i.e. for a point on the curve (a given number of training examples). If two curves pass each other, it is very difficult to determine if a strategy is better than another (on the total set of training examples). This point will be discussed in a future paper.

Finally we note that the maximal number of examples to labelled, or an estimation of the progress of the model, have to be used to stop the algorithm.

This is very linked to the use of a test set or the model employed. The elaboration of a good criterion should be independent of the model and of a test set and it is another way of future works.

References

1. Nguyen, H.T., Smeulders, A.: Active learning using pre-clustering. In: International Conference on Machine Learning (ICML). (2003)
2. Gosselin, P.H., Cord, M.: Active learning techniques for user interactive systems : application to image retrieval. In: International Workshop on Machine Learning for MultiMedia (In conjonction with ICML). (2005)
3. Brinker, K.: Incorporating diversity in active learning with support vector machines. In: International Conference on Machine Learning (ICML). (2003) 59–66
4. Culver, M., Kun, D., Scott, S.: Active learning to maximize area under the roc curve. In: International Conference on Data Mining (ICDM). (2006)
5. Thrun, S.: Exploration in active learning. In: to appear in: Handbook of Brain Science and Neural Networks. Michael Arbib (2007)
6. Osugi, T., Kun, D., Scott, S.: Balancing exploration and exploitation: A new algorithm for active machine learning. In: International Conference on Data Mining (ICDM). (2005)
7. Bondu, A., Lemaire, V., Poulain, B.: Active learning strategies: a case study for detection of emotions in speech. In: Industrial Conference of Data Mining (ICDM), Leipzig (2007)
8. Castro, R., Willett, R., Nowak, R.: Faster rate in regression via active learning. In: Neural Information Processing Systems (NIPS), Vancouver (2005)
9. Muslea, I.: Active Learning With Multiple View. Phd thesis, University of southern california (2002)
10. Thrun, S.B., Möller, K.: Active exploration in dynamic environments. In Moody, J.E., Hanson, S.J., Lippmann, R.P., eds.: Advances in Neural Information Processing Systems. Volume 4., Morgan Kaufmann Publishers, Inc. (1992) 531–538
11. Roy, N., McCallum, A.: Toward optimal active learning through sampling estimation of error reduction. In: International Conference on Machine Learning (ICML), Morgan Kaufmann, San Francisco, CA (2001) 441–448
12. Zhu, X., Lafferty, J., Ghahramani, Z.: Combining active learning and semi-supervised learning using gaussian fields and harmonic functions. In: International Conference on Machine Learning (ICML), Washington (2003)
13. Freund, Y., Seung, H.S., Shamir, E., Tishby, N.: Selective sampling using the query by committee algorithm. Machine Learning **28**(2-3) (1997) 133–168
14. Dasgupta, S.: Analysis of greedy active learning strategy. In: Neural Information Processing Systems (NIPS), San Diego (2005)
15. Cohn, D.A., Atlas, L., Ladner, R.E.: Improving generalization with active learning. Machine Learning **15**(2) (1994) 201–221
16. Fawcett, T.: Roc graphs: Notes and practical considerations for data mining researchers. Technical Report HPL-2003-4, HP Labs (2003)
17. Chappelle, O.: Active learning for parzen windows classifier. In: AI & Statistics, Barbados (2005) 49–56
18. Lindenbaun, M., Markovitch, S., Rusakov, D.: Selective sampling for nearest neighbor classifiers. Machine Learning (2004)