Efficient labeling with Active Learning

François HU
ENSAE - Société Générale Insurance

Joint work with **Caroline HILLAIRET** (CREST-ENSAE), **Marc JUILLARD** (Société Générale Insurance) and **Romuald ELIE** (CREST-ENSAE)

Online International Conference in Actuarial Science, Data Science and Finance (OICA), 2020

April 28, 2020

Summary

Motivating application

- Insurance organisations store **voluminous textual data** on a daily basis :
  - free text areas used by call center agents,
  - e-mails,
  - customer reviews,...
- These textual data are **valuable** and can be used in many use cases ...
  - optimize business processes,
  - analyze customer expectations and opinions,
  - control compliance (GDPR type) and fight against fraud, ...
- ... however
  - it is impossible for human experts to analyse all these quantities,
  - and the data usually comes **unlabelled**

Solution : exploit this large pool of unlabelled data with Active Learning

Motivating application

- Insurance organisations store **voluminous textual data** on a daily basis :
  - free text areas used by call center agents,
  - e-mails,
  - customer reviews,...
- These textual data are **valuable** and can be used in many use cases ...
  - optimize business processes,
  - analyze customer expectations and opinions,
  - control compliance (GDPR type) and fight against fraud, ...
- ... however
  - it is impossible for human experts to analyse all these quantities,
  - and the data usually comes **unlabelled**

Solution : exploit this large pool of unlabelled data with Active Learning

Motivating application

- Insurance organisations store **voluminous textual data** on a daily basis :
  - free text areas used by call center agents,
  - e-mails,
  - customer reviews,...
- These textual data are **valuable** and can be used in many use cases ...
  - optimize business processes,
  - analyze customer expectations and opinions,
  - control compliance (GDPR type) and fight against fraud, ...
- ... however
  - it is impossible for human experts to analyse all these quantities,
  - and the data usually comes **unlabelled**

Solution : exploit this large pool of unlabelled data with Active Learning

Motivating application

- Insurance organisations store **voluminous textual data** on a daily basis :
  - free text areas used by call center agents,
  - e-mails,
  - customer reviews,...
- These textual data are **valuable** and can be used in many use cases ...
  - optimize business processes,
  - analyze customer expectations and opinions,
  - control compliance (GDPR type) and fight against fraud, ...
- ... however
  - it is impossible for human experts to analyse all these quantities,
  - and the data usually comes **unlabelled**

Solution : exploit this large pool of unlabelled data with Active Learning

Notation and goal

Notations :

- let $\mathcal{X}$ be the instance space, $\mathcal{Y}$ the label space and $\mathcal{H} : \mathcal{X} \to \mathcal{Y}$ a class of hypotheses with finite VC dimension d
- let $\mathcal{P}$ be the distribution over $\mathcal{X} \times \mathcal{Y}$ and $\mathcal{P}_{\mathcal{X}}$ the marginal of $\mathcal{P}$ over $\mathcal{X}$. **In practice** instead of $\mathcal{P}_{\mathcal{X}}$ we have a pool of unlabeled data $\mathcal{U} = (x_i^{(pool)})_{i=1}^{U}$

Goal : label a sub-sample of $\mathcal{U}$ in order to construct an optimal training set $\mathcal{L} = \{(x_i^{(train)}, y_i^{(train)})\}_{i=1}^{L}$ for our learning algorithm $\mathcal{A}$ (which give us $\hat{h} \in \mathcal{H}$)

For any $h \in \mathcal{H}$, define :

- **Risk** : $R(h) = \mathbb{P}(h(x) \neq y)$
- **Empirical risk** : $\hat{R}_{\{(x_i, y_i)\}_{i=1}^{n}}(h) = \frac{1}{n} \sum_{i=1}^{n} \mathbb{1}\left(h(x_i) \neq y_i\right)$

Given a holdout set (or test set) $\mathcal{T} = \{(x_i^{(test)}, y_i^{(test)})\}_{i=1}^{T}$, our aim is to produce a highly-accurate classifier (i.e. minimize $\hat{R}_{\mathcal{T}}(\hat{h})$) using as few labels as possible.

Notation and goal

Notations :

- let $\mathcal{X}$ be the instance space, $\mathcal{Y}$ the label space and $\mathcal{H} : \mathcal{X} \to \mathcal{Y}$ a class of hypotheses with finite VC dimension d
- let $\mathcal{P}$ be the distribution over $\mathcal{X} \times \mathcal{Y}$ and $\mathcal{P}_{\mathcal{X}}$ the marginal of $\mathcal{P}$ over $\mathcal{X}$. **In practice** instead of $\mathcal{P}_{\mathcal{X}}$ we have a pool of unlabeled data $\mathcal{U} = (x_i^{(pool)})_{i=1}^{U}$

Goal : label a sub-sample of $\mathcal{U}$ in order to construct an optimal training set $\mathcal{L} = \{(x_i^{(train)}, y_i^{(train)})\}_{i=1}^{L}$ for our learning algorithm $\mathcal{A}$ (which give us $\hat{h} \in \mathcal{H}$)

For any $h \in \mathcal{H}$, define :

- **Risk** : $R(h) = \mathbb{P}(h(x) \neq y)$
- **Empirical risk** : $\hat{R}_{\{(x_i,y_i)\}_{i=1}^{n}}(h) = \frac{1}{n} \sum_{i=1}^{n} \mathbb{1}\left(h(x_i) \neq y_i\right)$

Given a holdout set (or test set) $\mathcal{T} = \{(x_i^{(test)}, y_i^{(test)})\}_{i=1}^{T}$, our aim is to produce a highly-accurate classifier (i.e. minimize $\hat{R}_{\mathcal{T}}(\hat{h})$) using as few labels as possible.

Notation and goal

Notations :

- let $\mathcal{X}$ be the instance space, $\mathcal{Y}$ the label space and $\mathcal{H} : \mathcal{X} \rightarrow \mathcal{Y}$ a class of hypotheses with finite VC dimension d
- let $\mathcal{P}$ be the distribution over $\mathcal{X} \times \mathcal{Y}$ and $\mathcal{P}_{\mathcal{X}}$ the marginal of $\mathcal{P}$ over $\mathcal{X}$. **In practice** instead of $\mathcal{P}_{\mathcal{X}}$ we have a pool of unlabeled data $\mathcal{U} = (x_i^{(pool)})_{i=1}^{U}$

Goal : label a sub-sample of $\mathcal{U}$ in order to construct an optimal training set $\mathcal{L} = \{(x_i^{(train)}, y_i^{(train)})\}_{i=1}^{L}$ for our learning algorithm $\mathcal{A}$ (which give us $\hat{h} \in \mathcal{H}$)

For any $h \in \mathcal{H}$, define :

- **Risk** : $R(h) = \mathbb{P}(h(x) \neq y)$
- **Empirical risk** : $\hat{R}_{\{(x_i,y_i)\}_{i=1}^{n}}(h) = \frac{1}{n} \sum_{i=1}^{n} \mathbb{1}\left(h(x_i) \neq y_i\right)$

Given a holdout set (or test set) $\mathcal{T} = \{(x_i^{(test)}, y_i^{(test)})\}_{i=1}^{T}$, our aim is to produce a highly-accurate classifier (i.e. minimize $\hat{R}_{\mathcal{T}}(\hat{h})$) using as few labels as possible.

Notation and goal

Notations :

- let $\mathcal{X}$ be the instance space, $\mathcal{Y}$ the label space and $\mathcal{H} : \mathcal{X} \rightarrow \mathcal{Y}$ a class of hypotheses with finite VC dimension d
- let $\mathcal{P}$ be the distribution over $\mathcal{X} \times \mathcal{Y}$ and $\mathcal{P}_{\mathcal{X}}$ the marginal of $\mathcal{P}$ over $\mathcal{X}$. **In practice** instead of $\mathcal{P}_{\mathcal{X}}$ we have a pool of unlabeled data $\mathcal{U} = (x_i^{(pool)})_{i=1}^{U}$

Goal : label a sub-sample of $\mathcal{U}$ in order to construct an optimal training set $\mathcal{L} = \{(x_i^{(train)}, y_i^{(train)})\}_{i=1}^{L}$ for our learning algorithm $\mathcal{A}$ (which give us $\hat{h} \in \mathcal{H}$)

For any $h \in \mathcal{H}$, define :

- **Risk** : $R(h) = \mathbb{P}(h(x) \neq y)$
- **Empirical risk** : $\hat{R}_{\{(x_i, y_i)\}_{i=1}^{n}}(h) = \frac{1}{n} \sum_{i=1}^{n} \mathbb{1}\left(h(x_i) \neq y_i\right)$

Given a holdout set (or test set) $\mathcal{T} = \{(x_i^{(test)}, y_i^{(test)})\}_{i=1}^{T}$, our aim is to produce a highly-accurate classifier (i.e. minimize $\hat{R}_{\mathcal{T}}(\hat{h})$) using as few labels as possible.

Passive Learning : a naive solution

Passive Learning : sample $x_i^{(train)}, \ldots, x_L^{(train)}$ $i.i.d \sim \mathcal{P}_{\mathcal{X}}$ then request their label

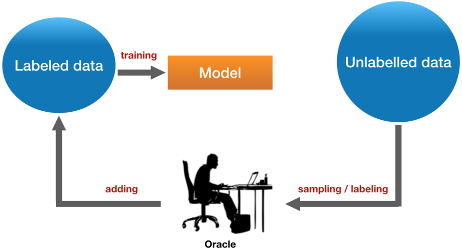


Figure: Conventional passive learning

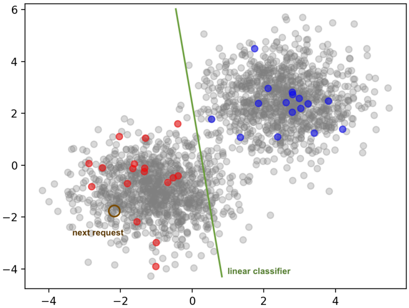


Figure: an illustration of passive learning

Active Learning : a better solution

Active Learning : Let a learning algorithm sequentially requests the labels of $\mathcal{U}$

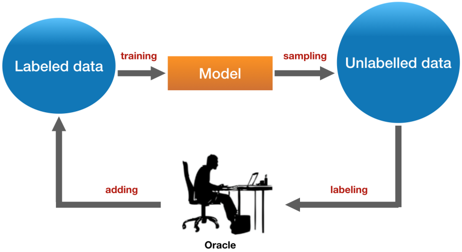


Figure: Conventional active learning

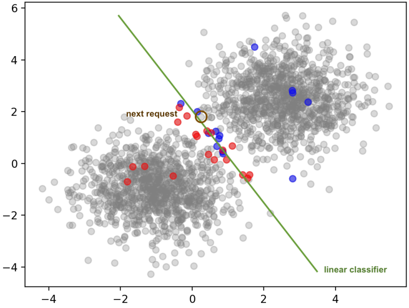
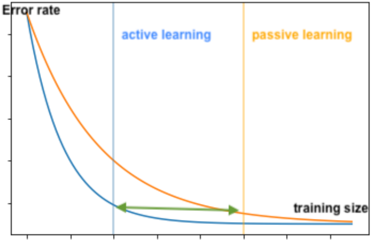


Figure: an illustration of active learning

Active Learning : a better solution



[Hanneke, 14] : Let h^* the optimal Bayes classifer and

$$\epsilon = R(\hat{h}) - R(h^*)$$

Then under a given hypothesis (bounded noise) and active learning algorithm (A^2)

- passive learning :

$$\epsilon \sim \frac{d}{n}$$

- active learning :

$$\epsilon \sim exp\left(-constant \cdot \frac{n}{d}\right)$$

Active learning

Uncertainty Sampling

Uncertainty Sampling : label the instances for which the current model is least certain as to what the correct output should be.

Example : for binary classification, label the instances whose posterior probability of being positive is nearest 0.5 :

$$x^{(train)} = \arg\min_{x \in \mathcal{U}} \{|P(y = 1|x) - 0.5|\}$$

Entropy-based active learning [Lewis and Gale, 94] :

$$x_H^{(train)} = \arg\max_{x \in \mathcal{U}} \left\{ -\sum_{y \in Y} P(y|x) \log P(y|x) \right\}$$

Uncertainty Sampling

$$x_H^{(train)} = \arg\max_{x \in \mathcal{U}} \left\{ - \sum_{y \in Y} P(y|x) \log P(y|x) \right\}$$

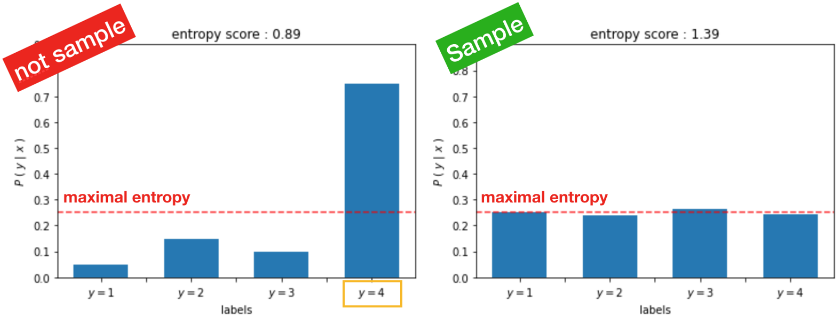


Figure: Entropy score for two instances : x (left) and x' (right)

Query By Committee

Query By Committee (QBC): construct a committee of models $C = \{\hat{h}_1, \ldots, \hat{h}_N\}$ trained on the current labeled data $\mathcal{L}$. Here, the most informative query is the instance about which the "committee" disagrees the most

- **The committee** $C = \{\hat{h}_1, \ldots, \hat{h}_N\}$:

  Query by bagging (Qbag) [Abe and Mamitsuka, 98] : Bootstrap N times $\mathcal{L}$ then train a learning algorithm on each bootstrapped data

- **Measure of disagreement** :

  Average Kullback Leibler divergence [MacCallum and Nigam, 98] :

$$x_{KL}^{(train)} = \arg\max_{x \in \mathcal{U}} \left\{ \frac{1}{N} \sum_{i=1}^{N} D(P_{\hat{h}_i} || P_{committee}) \right\}$$

  where

  - $D(P_{\hat{h}_i} || P_{committee}) = \sum_{y \in Y} P_{\hat{h}_i}(y|x) \log \left\{ \frac{P_{\hat{h}_i}(y|x)}{P_{committee}(y|x)} \right\}$ and
  - $P_{committtee}(y|x) = \frac{1}{N} \sum_i P_{\hat{h}_i}(y|x)$

Query By Committee

$$x_{KL}^{(train)} = \arg\max_{x \in \mathcal{U}} \left\{ \frac{1}{N} \sum_{i=1}^{N} D(P_{\hat{h}_i} || P_{committee}) \right\}$$

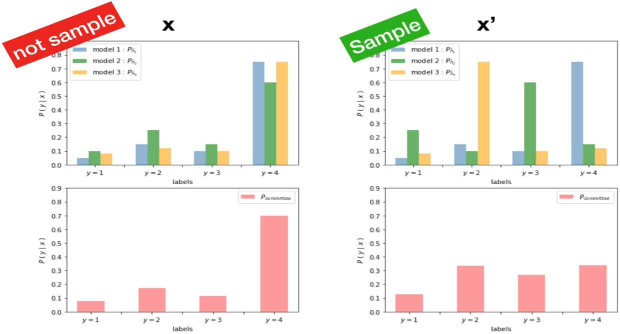


Figure: Distribution of the committee for two instances : x (left) and x' (right)

Other types of Active Learning

- Another disagreement-based active learning : **Agnostic Active Learning (A^2)** [Hanneke, 14]
- **Expected Model Change** : Sample instances that would impact the greatest change to the current model if we knew its label (example : "expected gradient length" (EGL) [Settles et al, 08])
- **Expected Error Reduction** : Sample instances that would make its generalization error likely to be reduced
- **Density Weighted Sampling** [Settles and Craven, 08] :

$$\boxed{x^{(train)}_{density} = \arg\max_{x \in \mathcal{U}} \left\{ \phi_A(x) \times \left(\frac{1}{U} \sum_{x' \in \mathcal{U}} sim(x, x') \right)^{\beta} \right\}}$$

with ϕ_A a measure of informativeness of x according to some sampling strategy A (uncertainty sampling, QBC, ...)

Experimentations

Text data : Net Promoter Score (NPS)

About the data : Net Promoter Score

1. **Score** : the client's score of an insurance product (score between 0 and 10)
2. **Verbatim** : explanation (in french) of the score by the client

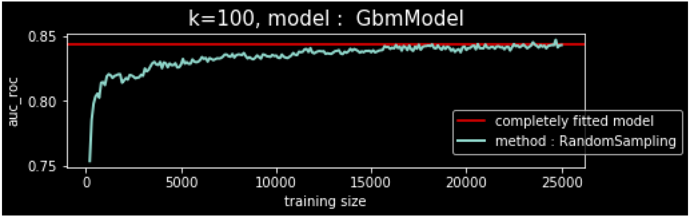
Encoding the text data : word2vec [Mikolov 2013]

Sentiment analysis : $Y = \{0, 1\} = \{\text{score} \leq 6, \text{score} > 6\}$

Mini-batch sampling algorithm : until we reach a stopping criterion,

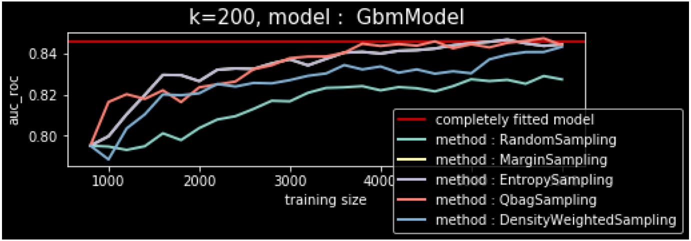
1. train our model $\hat{h}$ on the training set $\mathcal{L}$
2. select the k most informative samples $x_1^{(train)}, \ldots, x_k^{(train)}$ from the pool set $\mathcal{U}$
3. $\mathcal{U} \leftarrow \mathcal{U} - \{x_1^{(train)}, \ldots, x_k^{(train)}\}$ and $\mathcal{L} \leftarrow \mathcal{L} \cup \{x_1^{(train)}, \ldots, x_k^{(train)}\}$

Passive Learning



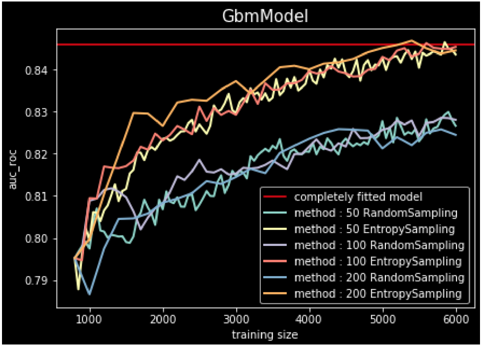
- **Learning model** : XGBoost
- **Sampling strategy** : random sampling
- **Initial training size / mini batch size** : 200 / 100
- **Stopping criterion** : 25 000

Active Learning



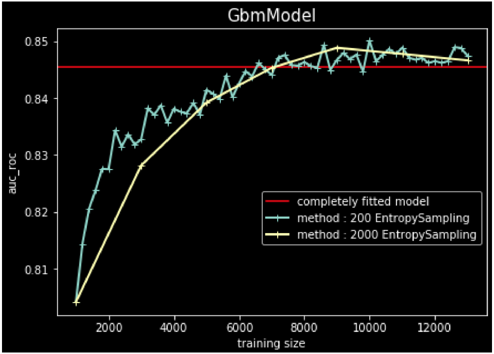
- **Learning model** : XGBoost
- **Sampling strategy** : random sampling
  - uncertainty-based sampling (Entropy, Margin)
  - disagreement-based sampling (Qbag)
  - density-based sampling (DensityWeighted)
- **Initial training size / mini batch size** : 800 / 200
- **Stopping criterion** : 6 000

Mini-batch active learning



- **Learning model** : XGBoost
- **Sampling strategy** :
  - random sampling
  - entropy sampling
- **Initial training size / mini batch size** : 800 / (50, 100, 200)
- **Stopping criterion** : 6 000

Mini-batch active learning



- **Learning model** : XGBoost
- **Sampling strategy** : entropy sampling
- **Initial training size / mini batch size** : 1000 / (200, 2000)
- **Stopping criterion** : 13 000

Conclusion

For real text database :

- Construct a good classifier if the labeled data is available ;
  $\Rightarrow$ Power many use cases
- Compared to passive learning, active sampling can construct a more highly-accurate classifier ;
  $\Rightarrow$ Reduce the cost of annotation (here at least 4 times)
- In this context : the mini-batch size can vary between 1 and 2000.
  $\Rightarrow$ Speed up the annotation process

References

1. Steve Hanneke, "Theory of Active Learning", 2014
2. Naoki Abe and Hiroshi Mamitsuka "Query Learning Strategies using Boosting and Bagging", 1998
3. D. Lewis and W. Gale, "A sequential algorithm for training text classifiers", ACM SIGIR Conference on Research and Development in Information Retrieval, 1994.
4. McCallum and K. Nigam. "Employing EM in pool-based active learning for text classification". ICML Confenrence, 1998
5. Settles and M. Craven. "An analysis of active learning strategies for sequence labeling tasks". EMNLP Conference, 2008.
6. Mikolov et al., "Distributed Representations ofWords and Phrases and their Compositionality", Neural information processing systems, 2013
7. Burr Settles, "Survey of Active Learning", 2012