THE UNIVERSITY *of* EDINBURGH
informatics

Applied Machine Learning (AML)

Further Topics

Oisin Mac Aodha • Siddharth N.

Semi-Supervised Learning

Semi-Supervised Learning

- Supervised classifiers learn from **labelled** data.

Semi-Supervised Learning

- Supervised classifiers learn from **labelled** data.
- However, annotating data can be time consuming and expensive.

Semi-Supervised Learning

- Supervised classifiers learn from **labelled** data.
- However, annotating data can be time consuming and expensive.
- In practice we may have a mix of labelled (i.e. supervised) and unlabelled data (i.e. unsupervised) available to us.

Semi-Supervised Learning

- Supervised classifiers learn from **labelled** data.
- However, annotating data can be time consuming and expensive.
- In practice we may have a mix of labelled (i.e. supervised) and unlabelled data (i.e. unsupervised) available to us.
- The goal of semi-supervised learning is to train models with both labelled and unlabelled data.

Semi-Supervised Setting

- In semi-supervised learning we have labelled and unlabelled data.

Semi-Supervised Setting

- In semi-supervised learning we have labelled and unlabelled data.
- Labelled data: $\mathcal{D}_l = \{(\boldsymbol{x}_n, y_n)\}_{n=1}^{N_l}$

Semi-Supervised Setting

- In semi-supervised learning we have labelled and unlabelled data.
- Labelled data: $\mathcal{D}_l = \{(\boldsymbol{x}_n, y_n)\}_{n=1}^{N_l}$
- Unlabelled data: $\mathcal{D}_u = \{\boldsymbol{x}_n\}_{n=1}^{N_u}$

Semi-Supervised Setting

- In semi-supervised learning we have labelled and unlabelled data.
- Labelled data: $\mathcal{D}_l = \{(\boldsymbol{x}_n, y_n)\}_{n=1}^{N_l}$
- Unlabelled data: $\mathcal{D}_u = \{\boldsymbol{x}_n\}_{n=1}^{N_u}$
- In practice $N_u >> N_l$, i.e. we have more unlabelled data than labelled data.

Comparing the Different Problem Settings

Supervised Learning

$$\mathcal{D}_l = \{(\boldsymbol{x}_n, y_n)\}_{n=1}^{N_l}$$

Comparing the Different Problem Settings

Supervised Learning

$$\mathcal{D}_l = \{(\boldsymbol{x}_n, y_n)\}_{n=1}^{N_l}$$

Unsupervised Learning

$$\mathcal{D}_u = \{\boldsymbol{x}_n\}_{n=1}^{N_u}$$

Comparing the Different Problem Settings

Supervised Learning

$$\mathcal{D}_l = \{(\boldsymbol{x}_n, y_n)\}_{n=1}^{N_l}$$

Unsupervised Learning

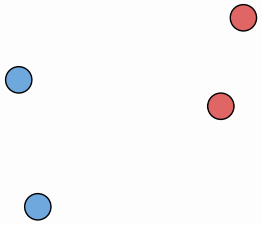
$$\mathcal{D}_u = \{\boldsymbol{x}_n\}_{n=1}^{N_u}$$

Semi-Supervised Learning

$$\mathcal{D} = \{(\boldsymbol{x}_n, y_n)\}_{n=1}^{N_l} \cup \{\boldsymbol{x}_n\}_{n=1}^{N_u}$$

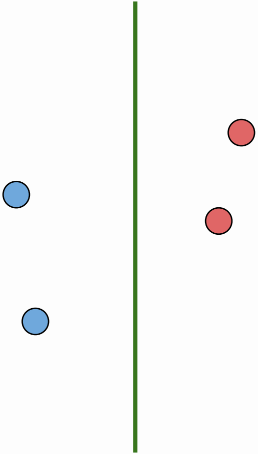
Semi-Supervised Example

- Here we have a binary classification problem with four datapoints.



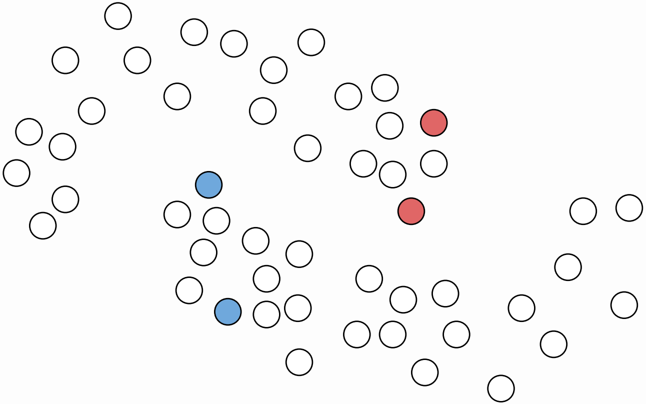
Semi-Supervised Example

- Here we have a binary classification problem with four datapoints.



Semi-Supervised Example

- The unlabelled data indicates structure that is not captured by the previous classifier.



Real World Instances of Semi-Supervised Learning

- In speech recognition is may be easy to obtain large quantities of unlabelled audio data but very time consuming to pay annotators to manually label all of it.

Real World Instances of Semi-Supervised Learning

- In speech recognition is may be easy to obtain large quantities of unlabelled audio data but very time consuming to pay annotators to manually label all of it.
- In medical settings, it may be relatively easy to collect data from patients (e.g. via x-ray, CT scan, etc.), but very challenging to get doctors to look at the data and provide their expert opinion.

Real World Instances of Semi-Supervised Learning

- In speech recognition is may be easy to obtain large quantities of unlabelled audio data but very time consuming to pay annotators to manually label all of it.
- In medical settings, it may be relatively easy to collect data from patients (e.g. via x-ray, CT scan, etc.), but very challenging to get doctors to look at the data and provide their expert opinion.
- Plus many more ...

Semi-Supervised Assumptions

Most semi-supervised approaches make at least one of the following assumptions.

Semi-Supervised Assumptions

Most semi-supervised approaches make at least one of the following assumptions.

Smoothness Assumption

Points that are close to each other are more likely to share a target value (e.g. the same class label).

Semi-Supervised Assumptions

Most semi-supervised approaches make at least one of the following assumptions.

Smoothness Assumption

Points that are close to each other are more likely to share a target value (e.g. the same class label).

Cluster Assumption

The data tend to form discrete clusters, and points in the same cluster are more likely to share a target.

Semi-Supervised Assumptions

Most semi-supervised approaches make at least one of the following assumptions.

Smoothness Assumption

Points that are close to each other are more likely to share a target value (e.g. the same class label).

Cluster Assumption

The data tend to form discrete clusters, and points in the same cluster are more likely to share a target.

Manifold Assumption

The data lie approximately on a manifold of much lower dimension than the input space.

Self-Training

- **Self-training** is one conceptually simple approach for semi-supervised learning.

Self-Training

- **Self-training** is one conceptually simple approach for semi-supervised learning.
- The central idea is to use the model $f_{\boldsymbol{\theta}}$ itself to make predictions on unlabelled data.

Self-Training

- **Self-training** is one conceptually simple approach for semi-supervised learning.
- The central idea is to use the model $f_{\boldsymbol{\theta}}$ itself to make predictions on unlabelled data.
- We then add **high confident** predictions ($f_{\boldsymbol{\theta}}(\boldsymbol{x}_u) > \tau$) to the labelled training set.

Self-Training

- **Self-training** is one conceptually simple approach for semi-supervised learning.
- The central idea is to use the model $f_{\boldsymbol{\theta}}$ itself to make predictions on unlabelled data.
- We then add **high confident** predictions ($f_{\boldsymbol{\theta}}(\boldsymbol{x}_u) > \tau$) to the labelled training set.
- We refer to the labels $\hat{y}_u$ derived from predictions as **pseudo labels**.

Self-Training

- **Self-training** is one conceptually simple approach for semi-supervised learning.
- The central idea is to use the model $f_{\boldsymbol{\theta}}$ itself to make predictions on unlabelled data.
- We then add **high confident** predictions ($f_{\boldsymbol{\theta}}(\boldsymbol{x}_u) > \tau$) to the labelled training set.
- We refer to the labels $\hat{y}_u$ derived from predictions as **pseudo labels**.

Require: labelled data $\mathcal{D}_l$, unlabelled data $\mathcal{D}_u$, number steps N, confidence threshold τ

Self-Training

- **Self-training** is one conceptually simple approach for semi-supervised learning.
- The central idea is to use the model $f_{\boldsymbol{\theta}}$ itself to make predictions on unlabelled data.
- We then add **high confident** predictions ($f_{\boldsymbol{\theta}}(\boldsymbol{x}_u) > \tau$) to the labelled training set.
- We refer to the labels $\hat{y}_u$ derived from predictions as **pseudo labels**.

Require: labelled data $\mathcal{D}_l$, unlabelled data $\mathcal{D}_u$, number steps N, confidence threshold τ

$f_{\boldsymbol{\theta}} \leftarrow \texttt{train\_model}(\mathcal{D}_l)$

Self-Training

- **Self-training** is one conceptually simple approach for semi-supervised learning.
- The central idea is to use the model $f_{\boldsymbol{\theta}}$ itself to make predictions on unlabelled data.
- We then add **high confident** predictions ($f_{\boldsymbol{\theta}}(\boldsymbol{x}_u) > \tau$) to the labelled training set.
- We refer to the labels $\hat{y}_u$ derived from predictions as **pseudo labels**.

Require: labelled data $\mathcal{D}_l$, unlabelled data $\mathcal{D}_u$, number steps N, confidence threshold τ

```
f_θ ← train_model(D_l)
for n ← 1 to N do
    Sample x_u ∈ D_u
```

Self-Training

- **Self-training** is one conceptually simple approach for semi-supervised learning.
- The central idea is to use the model $f_{\boldsymbol{\theta}}$ itself to make predictions on unlabelled data.
- We then add **high confident** predictions ($f_{\boldsymbol{\theta}}(\boldsymbol{x}_u) > \tau$) to the labelled training set.
- We refer to the labels $\hat{y}_u$ derived from predictions as **pseudo labels**.

Require: labelled data $\mathcal{D}_l$, unlabelled data $\mathcal{D}_u$, number steps N, confidence threshold τ

$f_{\boldsymbol{\theta}} \leftarrow \texttt{train\_model}(\mathcal{D}_l)$
for $n \leftarrow 1$ to N **do**
  Sample $\boldsymbol{x}_u \in \mathcal{D}_u$
  if $f_{\boldsymbol{\theta}}(\boldsymbol{x}_u) > \tau$ **then**
    $\mathcal{D}_l \leftarrow \mathcal{D}_l \cup (\boldsymbol{x}_u, \hat{y}_u)$
    $\mathcal{D}_u \leftarrow \mathcal{D}_u \setminus \boldsymbol{x}_u$

Self-Training

- **Self-training** is one conceptually simple approach for semi-supervised learning.
- The central idea is to use the model $f_{\boldsymbol{\theta}}$ itself to make predictions on unlabelled data.
- We then add **high confident** predictions ($f_{\boldsymbol{\theta}}(\boldsymbol{x}_u) > \tau$) to the labelled training set.
- We refer to the labels $\hat{y}_u$ derived from predictions as **pseudo labels**.

Require: labelled data $\mathcal{D}_l$, unlabelled data $\mathcal{D}_u$, number steps N, confidence threshold τ

$f_{\boldsymbol{\theta}} \leftarrow \texttt{train\_model}(\mathcal{D}_l)$
for $n \leftarrow 1$ to N **do**
  Sample $\boldsymbol{x}_u \in \mathcal{D}_u$
  if $f_{\boldsymbol{\theta}}(\boldsymbol{x}_u) > \tau$ **then**
    $\mathcal{D}_l \leftarrow \mathcal{D}_l \cup (\boldsymbol{x}_u, \hat{y}_u)$
    $\mathcal{D}_u \leftarrow \mathcal{D}_u \setminus \boldsymbol{x}_u$
    $f_{\boldsymbol{\theta}} \leftarrow \texttt{train\_model}(\mathcal{D}_l)$
return $f_{\boldsymbol{\theta}}$

Self-Training Limitations

- One obvious flaw with self-training is that if the model generates **incorrect** predictions for unlabelled data it is retrained on these incorrect predictions.

Self-Training Limitations

- One obvious flaw with self-training is that if the model generates **incorrect** predictions for unlabelled data it is retrained on these incorrect predictions.
- If this keeps repeating, the model will become progressively worse.

Self-Training Limitations

- One obvious flaw with self-training is that if the model generates **incorrect** predictions for unlabelled data it is retrained on these incorrect predictions.
- If this keeps repeating, the model will become progressively worse.
- This problem is referred to as **confirmation bias**.

Entropy Minimisation

- Self-training has the implicit effect of encouraging the model to output low entropy (i.e. high-confidence) predictions.

Figures taken from Probabilistic Machine Learning by Kevin Murphy.

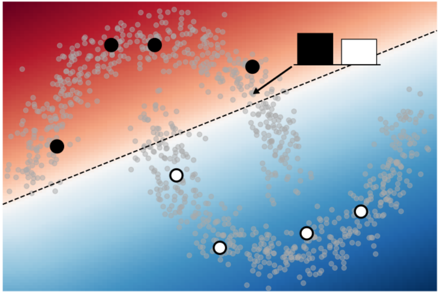
Entropy Minimisation

- Self-training has the implicit effect of encouraging the model to output low entropy (i.e. high-confidence) predictions.
- Alternatively, we could add an additional loss for the unlabelled data, e.g. directly encourage low entropy $\mathcal{L}_u = -f_\theta(\mathbf{x}_u)\log(f_\theta(\mathbf{x}_u)) - (1 - f_\theta(\mathbf{x}_u))\log(1 - f_\theta(\mathbf{x}_u))$.

Figures taken from Probabilistic Machine Learning by Kevin Murphy.

Entropy Minimisation

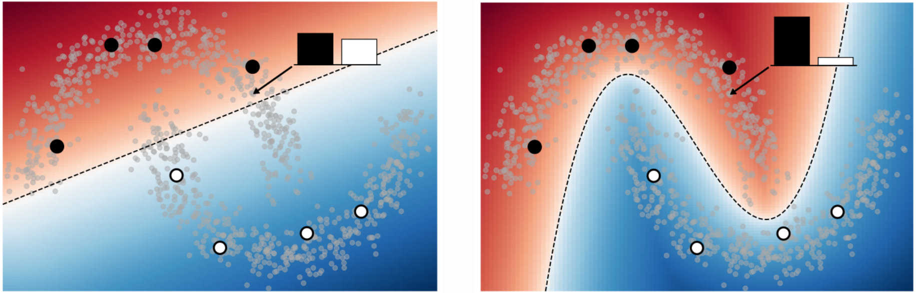
- Self-training has the implicit effect of encouraging the model to output low entropy (i.e. high-confidence) predictions.
- Alternatively, we could add an additional loss for the unlabelled data, e.g. directly encourage low entropy $\mathcal{L}_u = -f_\theta(\mathbf{x}_u)\log(f_\theta(\mathbf{x}_u)) - (1 - f_\theta(\mathbf{x}_u))\log(1 - f_\theta(\mathbf{x}_u))$.



Figures taken from Probabilistic Machine Learning by Kevin Murphy.

Entropy Minimisation

- Self-training has the implicit effect of encouraging the model to output low entropy (i.e. high-confidence) predictions.
- Alternatively, we could add an additional loss for the unlabelled data, e.g. directly encourage low entropy $\mathcal{L}_u = -f_\theta(\mathbf{x}_u)\log(f_\theta(\mathbf{x}_u)) - (1 - f_\theta(\mathbf{x}_u))\log(1 - f_\theta(\mathbf{x}_u))$.



Figures taken from Probabilistic Machine Learning by Kevin Murphy.

Label Propagation

- Label Propagation is a semi-supervised approach that exploits the smoothness assumption to assign labels to unlabelled data.

Label Propagation

- Label Propagation is a semi-supervised approach that exploits the smoothness assumption to assign labels to unlabelled data.
- It constructs a graph, where the datapoints are nodes, and the edges between them represent their similarity.

Label Propagation

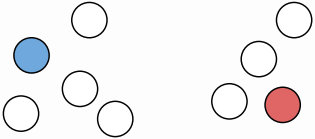
- Label Propagation is a semi-supervised approach that exploits the smoothness assumption to assign labels to unlabelled data.
- It constructs a graph, where the datapoints are nodes, and the edges between them represent their similarity.
- Known labels are ‘propagated‘ across the edges of the graph from labelled nodes to unlabelled ones.

Label Propagation

- Label Propagation is a semi-supervised approach that exploits the smoothness assumption to assign labels to unlabelled data.
- It constructs a graph, where the datapoints are nodes, and the edges between them represent their similarity.
- Known labels are ‘propagated‘ across the edges of the graph from labelled nodes to unlabelled ones.
- When complete, each unlabelled datapoint has an estimated label which can be then be used for training any supervised learning method.

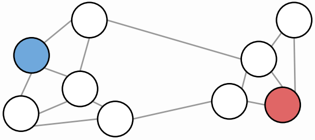
Label Propagation Example

1. As input we have labelled (here blue or red) and unlabelled (here white) data.



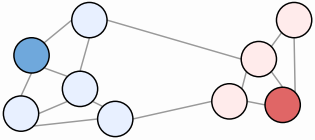
Label Propagation Example

1. As input we have labelled (here blue or red) and unlabelled (here white) data.
2. We define a similarity measure between pairs of datapoints. Here datapoints that are closer in feature space are determined to be more similar.



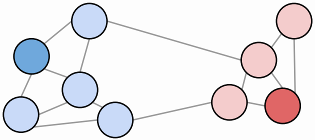
Label Propagation Example

1. As input we have labelled (here blue or red) and unlabelled (here white) data.
2. We define a similarity measure between pairs of datapoints. Here datapoints that are closer in feature space are determined to be more similar.
3. Finally we iteratively propagate labels from the labelled to the unlabelled nodes.



Label Propagation Example

1. As input we have labelled (here blue or red) and unlabelled (here white) data.
2. We define a similarity measure between pairs of datapoints. Here datapoints that are closer in feature space are determined to be more similar.
3. Finally we iteratively propagate labels from the labelled to the unlabelled nodes.



Summary

- Semi-supervised learning is a training paradigm that allows us to make use of both labelled and unlabelled data.

Summary

- Semi-supervised learning is a training paradigm that allows us to make use of both labelled and unlabelled data.
- We have to make some assumptions about the underlying data distribution e.g. smoothness.

Summary

- Semi-supervised learning is a training paradigm that allows us to make use of both labelled and unlabelled data.
- We have to make some assumptions about the underlying data distribution e.g. smoothness.
- There are many different techniques in the literature. Some are general purpose, others are specific to specific types of models.

Active Learning

Active Learning

- In the case of semi-supervised learning we relied on algorithmic approaches to either infer missing labels or to exploit the data structure to learn more effective models.

Active Learning

- In the case of semi-supervised learning we relied on algorithmic approaches to either infer missing labels or to exploit the data structure to learn more effective models.
- In contrast, in active learning we interactively query an annotator (i.e. oracle) who provides information about unlabelled data.

Active Learning

- In the case of semi-supervised learning we relied on algorithmic approaches to either infer missing labels or to exploit the data structure to learn more effective models.
- In contrast, in active learning we interactively query an annotator (i.e. oracle) who provides information about unlabelled data.

Goal

Learn a model that generalises well with the smallest number of queries to the annotator.

Active Learning

- In the case of semi-supervised learning we relied on algorithmic approaches to either infer missing labels or to exploit the data structure to learn more effective models.
- In contrast, in active learning we interactively query an annotator (i.e. oracle) who provides information about unlabelled data.

Goal

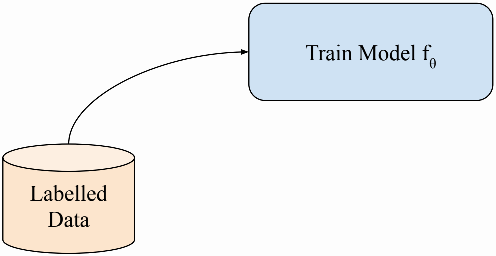
Learn a model that generalises well with the smallest number of queries to the annotator.

Assumption

Not all datapoints are equally informative, i.e. some are more useful than others.

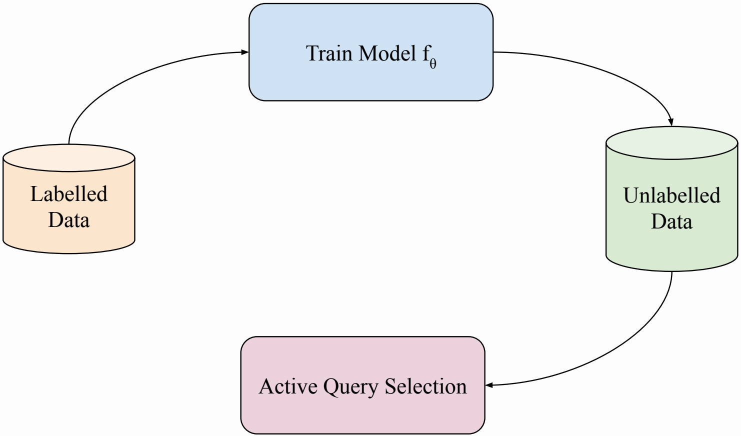
Active Learning Loop

- In active learning we iteratively query the oracle labeller to get labels for unlabelled data.



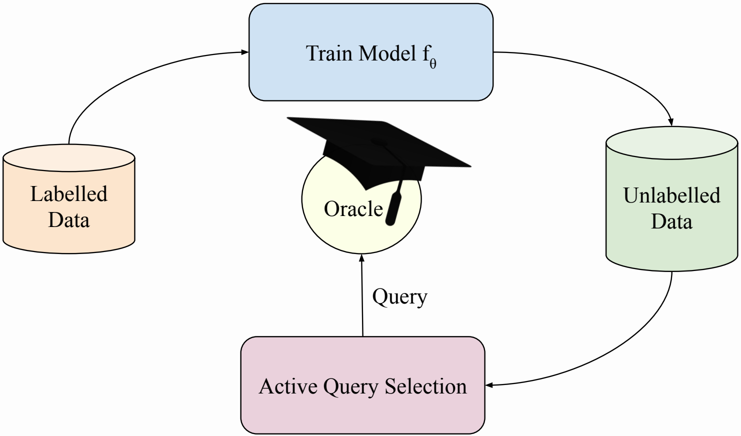
Active Learning Loop

- In active learning we iteratively query the oracle labeller to get labels for unlabelled data.



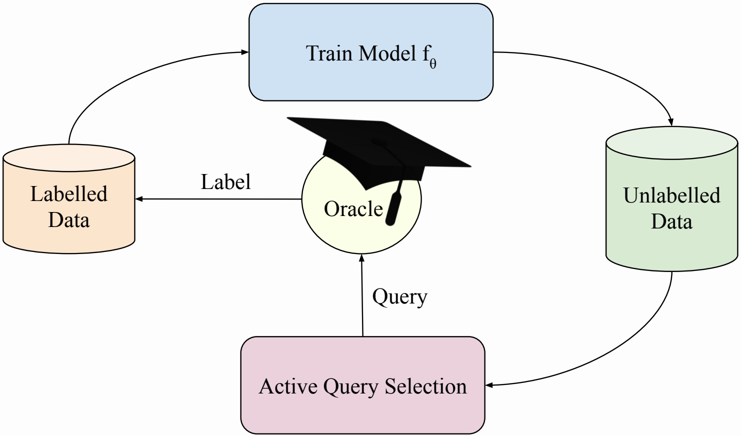
Active Learning Loop

- In active learning we iteratively query the oracle labeller to get labels for unlabelled data.



Active Learning Loop

- In active learning we iteratively query the oracle labeller to get labels for unlabelled data.



Query Selection

Multiple different heuristic query selection strategies have been proposed in the literature.

Query Selection

Multiple different heuristic query selection strategies have been proposed in the literature.

- **Random**
  - Trivial baseline where we just randomly select queries from the unlabelled set without replacement.

Query Selection

Multiple different heuristic query selection strategies have been proposed in the literature.

- Random
  - Trivial baseline where we just randomly select queries from the unlabelled set without replacement.
- Uncertainty sampling
  - Choose the query that the model is most uncertain about, e.g. close to a decision boundary.

Query Selection

Multiple different heuristic query selection strategies have been proposed in the literature.

- **Random**
  - Trivial baseline where we just randomly select queries from the unlabelled set without replacement.
- **Uncertainty sampling**
  - Choose the query that the model is most uncertain about, e.g. close to a decision boundary.
- **Query by committee**
  - Train an ensemble of models and choose the query that has most disagreement from the the models in the ensemble.

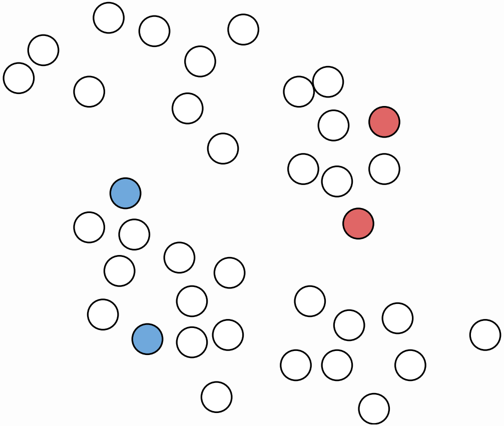
Query Selection

Multiple different heuristic query selection strategies have been proposed in the literature.

- **Random**
  - Trivial baseline where we just randomly select queries from the unlabelled set without replacement.
- **Uncertainty sampling**
  - Choose the query that the model is most uncertain about, e.g. close to a decision boundary.
- **Query by committee**
  - Train an ensemble of models and choose the query that has most disagreement from the the models in the ensemble.
- **Expected model change**
  - Choose the query that would most change the current model if added to the training set. Expensive to compute.

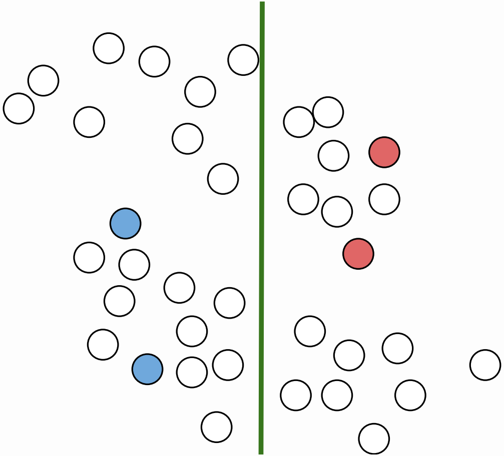
Uncertainty Sampling Example

- On the right we see labelled and unlabelled data for a binary classification task.



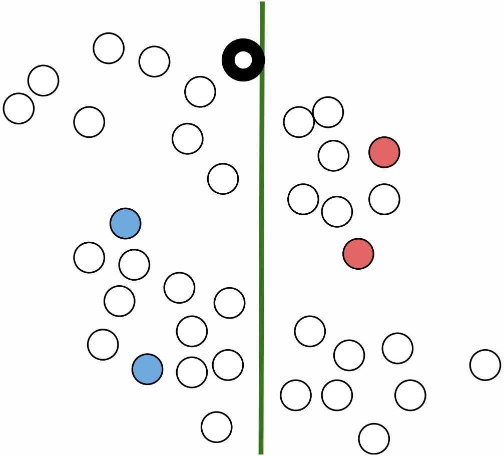
Uncertainty Sampling Example

- On the right we see labelled and unlabelled data for a binary classification task.
- We first fit our model (here a linear classifier) to the labelled data.



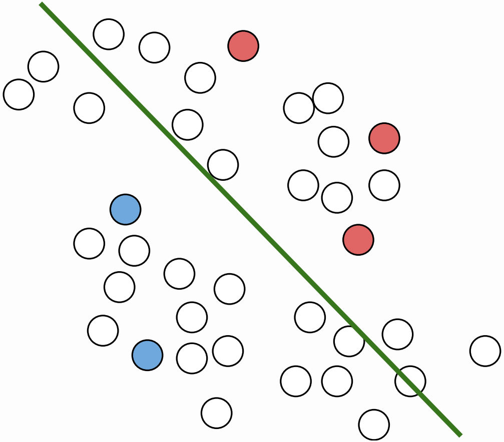
Uncertainty Sampling Example

- On the right we see labelled and unlabelled data for a binary classification task.
- We first fit our model (here a linear classifier) to the labelled data.
- We choose the query to be labelled that the model is most uncertain about. For a logistic regression classifier it would the datapoint closest to the decision boundary, i.e. $P(y_u|\boldsymbol{x}_u) \approx 0.5$.



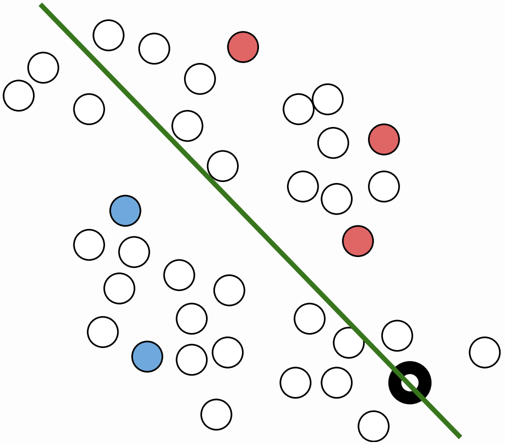
Uncertainty Sampling Example

- We add the new datapoint to the labelled set and retrain the classifier.



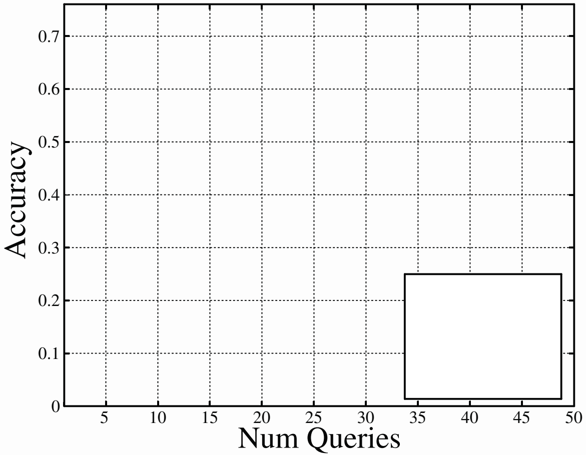
Uncertainty Sampling Example

- We add the new datapoint to the labelled set and retrain the classifier.
- We then repeat the process by selecting the next query to be labelled.



Active Learning Result

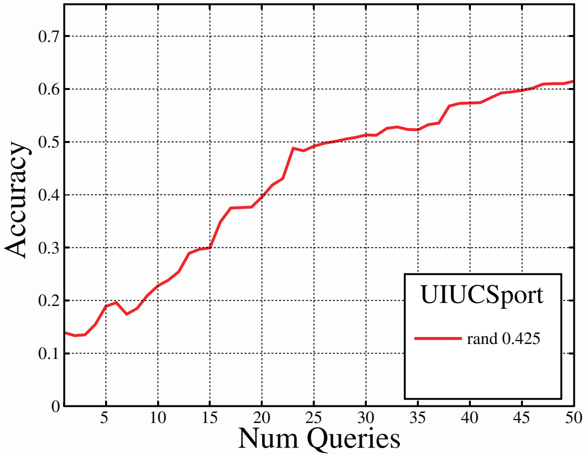
- Here we show an example of active learning applied to multiclass classification.



Figures adapted from Mac Aodha et al. CVPR 2014.

Active Learning Result

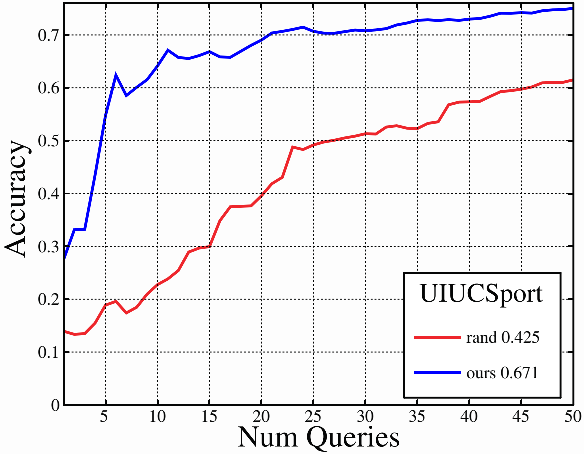
- Here we show an example of active learning applied to multiclass classification.



Figures adapted from Mac Aodha et al. CVPR 2014.

Active Learning Result

- Here we show an example of active learning applied to multiclass classification.



Figures adapted from Mac Aodha et al. CVPR 2014.

Summary

- In active learning we interactively query the annotator(s) during training.

Summary

- In active learning we interactively query the annotator(s) during training.
- The aim is to obtain ‘good’ performance with a minimal number of training examples.

Summary

- In active learning we interactively query the annotator(s) during training.
- The aim is to obtain ‘good’ performance with a minimal number of training examples.
- There are several different families of query selection strategies available. The choice of which to use will depend on the specific use case.

Summary

- In active learning we interactively query the annotator(s) during training.
- The aim is to obtain 'good' performance with a minimal number of training examples.
- There are several different families of query selection strategies available. The choice of which to use will depend on the specific use case.
- Active learning pipelines are often deployed in practical applications as data annotation can be expensive.