16. Online Learning
Motivation

• So far, we have handled offline learning methods, but very often data is observed in an ongoing process.

• Therefore, it would be good to adapt the learned models with newly arriving data without having to consider old data.

• This is called online learning.

• Major benefits:
  • algorithms are adaptive to new observations
  • learning is usually faster
Clarification of Notions

There are (at least) three notions:

• **incremental** learning:
  • model is updated with new samples, but old samples might be considered again

• **online** learning:
  • after considering a data sample, it can be disregarded (no need to store it)

• **learning at frame rate**:
  • learning is fast enough such that it can be performed in one perception cycle
Example: Pedestrian Detection

Offline Approach:

1. Collect and annotate a large data set from different sensor modalities, here: camera and 2D laser

[Spinello, Triebel, and Siegwart, IJRR 2010]
Example: Pedestrian Detection

Offline Approach:

1. Collect and annotate a large data set from different sensor modalities, here: **camera** and **2D laser**
2. **Train** a classifier for each sensor modality

[Spinello, Triebel, and Siegwart, IJRR 2010]
Example: Pedestrian Detection

Offline Approach:

1. Collect and annotate a large data set from different sensor modalities, here: **camera** and **2D laser**
2. **Train** a classifier for each sensor modality
3. **Apply** the classifiers and **fuse** data

[Spinello, Triebel, and Siegwart, IJRR 2010]
Example: Pedestrian Detection

Testing Phase:

Fused pedestrian detection and tracking

But: non-adaptive; large training set required

Fused detection and tracking of cars and pedestrians

[Spinello, Triebel, and Siegwart, IJRR 2010]
Major Problem of this Approach

Offline learning:

- cannot adapt to new environments
- cannot learn new classes
- cannot consider new instances of a given class
- is often very slow
- requires a large amount of manually annotated training data

Ideas:

- use **online** learning for adaptivity
- use **active** learning to reduce (human) work load
Active Learning

• To reduce the required training data, the learner can **select** the data it needs to learn from.
• This is called **active** learning
• Major advantages:
  • only those samples where classification is hard are used for re-training
  • humans only need to give ground truth labels for a smaller set of samples
• In principle, active learning can be used with offline and online learning, but online makes more sense
Active Learning

Training data → Optimization → $f(x)$

$f(x)$

training data

“Laptop”

“Cup”

$y$ $x$
Active Learning

- Training data
- New Data

Optimization $\rightarrow$ Prediction

function $f(x)$

Uncertainty, Labels

- "Laptop"
- "Cup"

Training data

Online Learning
Active Learning

Training data \rightarrow \text{Optimization} \xrightarrow{\begin{array}{c} f(x) \\ \end{array}} \text{Prediction} \rightarrow \text{New Data}

Uncertainty, Labels

\text{function} f(x)

training data

“Laptop”

“Cup”

\begin{align*}
\text{training data} & \quad \text{"Laptop"} \\
\text{"Cup"} & \quad \text{\quad } \\
\end{align*}

\begin{align*}
\gamma & \quad f(x) \\
\chi & \quad ? \\
\end{align*}
**Active Learning**

- **Optimization**: $f(x)$
- **Prediction**: Uncertainty, Labels
- **Label Query**: Supervisor

**Diagram:**
- Training data
- New Data
- Supervisor

**Function:** $f(x)$

**Images:**
- “Laptop”
- “Cup”
- “Present”

**Text:**
- “Present”
Active Learning

Training data → Optimization

Function $f(x)$ → Prediction

Uncertainty, Labels → Label Query

New training data → Extend training data

“Laptop”

“Cup”

“Present”

Training data

Optimization

Prediction

Label Query

Extend training data
Active Learning

Major Benefits:
- Adapts to new situations
- Requires less training samples
Uncertainty Estimates

• A key step in active learning is the **selection** step
• It requires a good estimate of **uncertainties**
• Examples for classification algorithms to compute uncertainties:
  • Support Vector Machine (“Platt scaling”)
  • Gaussian Process Classifier (“predictive variance”)
  • Tree-based classifiers (entropy in leaf nodes)

But: It is important how uncertainty correlates to incorrect classifications!
Uncertainty Estimation: An Example

Comparison by evaluation on an unseen class:

The GP is less overconfident than the SVM!

[Paul, Triebel, Rus, Newman, IROS 2012]
Application: Traffic Sign Classification

uncertainty histograms

Trained classes

Unseen classes

The GP is less overconfident than the SVM!

[Grimmett, Paul, Triebel, Posner ICRA 2013]
Definition 1: The overconfidence of a classifier is the mean of all confidence values that correspond to incorrect classifications.

Definition 2: The underconfidence of a classifier is the mean of all uncertainty values that correspond to correct classifications.
The 3 Criteria of Learning Algorithms

- Efficiency
  - memory
  - run time

- Accuracy
  - precision
  - recall

Run time
Memory

Precision
Recall

Camera
Laser
Fusion

Efficiency vs. Accuracy

Online Learning
The 3 Criteria of Learning Algorithms

- **Efficiency**
  - memory
  - run time

- **Accuracy**
  - precision
  - recall

- **Confidence**
  - overconfidence
  - underconfidence
Active Learning is well suited for semantic mapping because:

- it can deal with large amounts of data.
- data is not independent.
- the task is essentially an online learning problem.
Active Learning for Semantic Mapping

Concrete Approach:

- Use IVM (a sparse online GP) for classification
- Sort classified data by uncertainty
- Request labels for the most uncertain samples
- Remove uninformative samples ("forgetting")
The Informative Vector Machine

Main differences to standard GP classifier:

- it uses a **subset** ("active set") of training points
- the (inverse) posterior covariance matrix is computed **incrementally**

Decision of inclusion in the active set based on information-theoretic criterion

**Slight caveat:** Training of hyper-parameters needs to be done iteratively

From: [http://staffwww.dcs.shef.ac.uk/people/N.Lawrence/ivm/](http://staffwww.dcs.shef.ac.uk/people/N.Lawrence/ivm/)
Semantic Mapping: Results

- **GP** “overtakes” and stays better than **SVM**.
- Active learning better than passive learning.
- Random selection is not better.

[Graph showing performance over epochs for different detection methods: SVM active, SVM passive, SVM random, IVM active, IVM passive, IVM random. The graph demonstrates improved performance with active learning compared to passive and random selection.]
Which Classifier Should We Choose?

![Diagram showing a scatter plot with GPC, SVM, and Boosting points.]

- **Efficiency**
  - High
  - Low

- **Accuracy**
  - High
  - Low

Classifier options:
- GPC
- SVM
- Boosting
Which Classifier Should We Choose?

- SVMs are accurate, but (more) overconfident
- GPCs are less overconfident, but very inefficient
- Boosting is very efficient, but also overconfident
Can We Reduce Over-(Under-)Confidence?

- Start with a very efficient multi-class boosting classifier
- Modify it so that it reduces overconfidence
- Use also the confidence to weight training samples

Result ("Confidence Boosting"):

- wrong, certain classified samples receive higher weight
- overconfidence reduced in every learning round
- overall classifier is less overconfident
  ⇒ better for Active Learning

Note: This is the alternative approach to making a non-overconfident classifier efficient!
Online Gradient Boosting

**Online**: outer loop over data points, inner loop over weak classifiers

- For each point, all weak classifiers are updated

- **Agreement** between prediction and ground truth
- Sample receives low weight if agreement high

**Note**: Agreement does not use the confidence!

---

**Algorithm 1**: Online Multi-class Gradient Boost [Saffari et al. 2010]

1. **Data**: training data \((X, y)\) with \(C\) classes
2. **Input**: number of weak learners \(M\), loss function \(\ell\), agreement function \(a\)
3. **Output**: weak learners \(f_1, \ldots, f_M\)
4. Initialize \((f_1, \ldots, f_M)\)
5. for \(n = 1, \ldots, N\) do
6.   for \(m = 1, \ldots, M\) do
7.     \(f_m \leftarrow \text{UpdateWeakLearner}(f_m, x_n, y_n, w_n)\)
8.     \(p_{nm} \leftarrow f_m(x_n)\)
9.     \(\alpha_{nm} \leftarrow a(p_{nm}, y_n)\)
10.    \(g_n \leftarrow g_n + \alpha_{nm}\)
11.    \(w_n \leftarrow -\nabla \ell(g_n)\)
12. end
13. end
Confidence Boosting

Main idea:
Use also the confidence to compute the agreement

Intuitively:

- if classification is correct ⇒ use the positive confidence
- if classification is wrong ⇒ use negative confidence

Result:

- wrong, certain classified samples receive higher weight
- overconfidence reduced in every learning round
- overall classifier is less overconfident ⇒ better for Active Learning
Example: AL for 3D Object Recognition

Qualitative Results:

True Label = lime

True Label = binder

True Label = bowl
Example: AL for 3D Object Recognition

Confidence Boosting
- classifies better than GradientBoost.
- generates fewer label queries.
- is less overconfident.
- is orders of magnitude faster than GPC.
Problem: Most often, data occurs in streams (online) and not in pools (offline)

But: Online Random Forests
- require a balanced class dist.
- depend on the order of the data

Idea: use Mondrian Forests [1]

Main differences:
- MFs also store the range of the data in each dimension
- MFs are independent on the class labels

Mondrian Forests

- start with two points
- insert a split
Mondrian Forests

- start with two points
- insert a split
- add a third point
Mondrian Forests

- start with two points
- insert a split
- add a third point
- sample a “time” from an exponential distr.
- insert a node above the given node
- insert a random split
Mondrian Forests

- start with two points
- insert a split
- add a third point
- sample a “time” from an exponential distr.
- insert a node **above** the given node
- insert a random split
- add a new node inside the outer box
Mondrian Forests

- start with two points
- insert a split
- add a third point
- sample a “time” from an exponential distr.
- insert a node above the given node
- insert a random split
- add a new node inside the outer box
- sample again and insert new node at the end
- this requires extending a split
Why the Name “Mondrian” Forests?
Why the Name “Mondrian” Forests?

Piet Mondrian: “Composition with Red, Yellow, Blue, and Black” 1926
Gemeentemuseum, Den Haag
Application to Real Data

KiTTI benchmark data set:
- 3D point clouds
- cars, pedestrians, bikes, trucks, etc.
- segmentation given ("tracklets")

Goal:
- online classification
- use active learning

Major problems:
- un-balanced classes
- order of appearance
The Data Set

The real data

The data uniformly resampled
Results (no Active Learning)

The real data

The data uniformly resampled

• Mondrian Forests require only 5% of the data to reach 90% accuracy
• Standard Random Forests can not reach that

Summary

• Online learning has important advantages over offline learning: **efficiency** and **adaptivity**

• In active learning the algorithm **selects** the data to learn from (human in the loop)

• This requires good **confidence** estimates

• The GP classifiers tends to be less **overconfident**, but inefficient

• A faster alternative method is **Confidence Boosting**, it is very efficient and not much overconfident

• For stream-based Active Learning **Mondrian Forests** are better suited