Empolis – Workshop

– Machine Learning –

Machine Learning 102

Prof. Dr. Adrian Ulges
Fachbereich DCSM
Hochschule RheinMain

21. März 2016

Outline

1. Active Learning

2. Anomaly Detection
   - Local Outlier Factor
   - One-class SVMs

Active Learning

Active Learning

- Remember: Acquiring labels is expensive
- Idea: Acquire labels only for the **'interesting' samples**
- Iterate:
  1. Machine selects a sample x
  2. Human expert assigns a label y to sample x
  3. Machine updates its model with (x, y)

```
model = init()
labeled_samples = {}
while ...:
      x := select_sample(model, samples)
      y := expert_label(x)
      labeled_samples.add( (x,y) )
      model := model.train(labeled_samples)
```

Active Learning

Remarks

- In general, this works with any **base classifier**.
- Only requirement: the base classifier can compute the **posterior** $P(Y = y|\mathbf{x})$
- Often, active learning targets two **different goals** *("exploration vs. exploitation")*
  - Detecting positive Samples **(→ satisfy the user)**
  - Exploration of feature space **(→ improve the classifier)**
- Challenge: Labels are usually **extremely scarse** (i.e., benchmarking our classifier is usually not possible)

Active Learning: Approaches

- Key Problem: Which are the **interesting samples** to select?
- There are different strategies towards **sample selection** (or **querying**)

Approaches

- **Uncertainty sampling**: Select the sample $\mathbf{x}$ which the base classifier is most uncertain about:

$$\mathbf{x}^* = \arg\min_{\mathbf{x}} \left| P(Y = 1|\mathbf{x}) - 0.5 \right|$$

- **Relevance sampling**: Select the sample $\mathbf{x}$ most likely to be positive *(sometimes used in cases where positive samples are extremely rare / hard to come by)*

$$\mathbf{x}^* = \arg\max_{\mathbf{x}} P(Y = 1|\mathbf{x})$$

Active Learning: Approaches

Approaches (cont'd)

- **Query-by-Commitee**: use an ensemble of classifiers (e.g., a random forest). Select the sample that most classifiers (e.g., trees in the forest) disagree on.
- **Model change**: Select the sample that is expected to lead to the biggest change in the model.
  - Example: Logistic Regression (weight vector $\mathbf{w}$)
  - For each candidate sample $\mathbf{x}$, retrain the classifier once with $y = 1$ and once with $y = 0$, obtaining two new weight vectors $\mathbf{w}_1$ and $\mathbf{w}_0$.
  - Pick the sample that leads to the largest difference:

$$\mathbf{x}^* = \arg\max_{\mathbf{x}} \; P(Y = 1|\mathbf{x}) \cdot |\mathbf{w}_1 - \mathbf{w}| + P(Y = 0|\mathbf{x}) \cdot |\mathbf{w}_0 - \mathbf{w}|$$

  - Very very expensive (just for candidate refinement)

Active Learning: Approaches

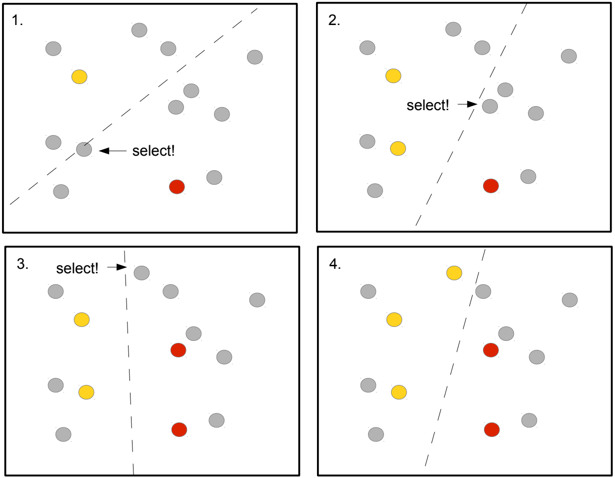
Approaches (cont'd)

- **Density-based methods** use the above quality measures Q and enforce an exploration of **new** areas
  - **Alternative 1**: Cluster the most highly-ranked candidate samples (according to Q) and pick representatives form the different clusters
  - **Alternative 2**: Downgrade candidate samples that are close to already labeled samples *(density-weighted repulsion)*

$$\mathbf{x}^* = \arg\max_{\mathbf{x}} \; Q(\mathbf{x}) \cdot (p^+(\mathbf{x}) + \epsilon)^{-\gamma}$$

Active Learning: Illustration in Feature Space

An Example (using uncertainty sampling)

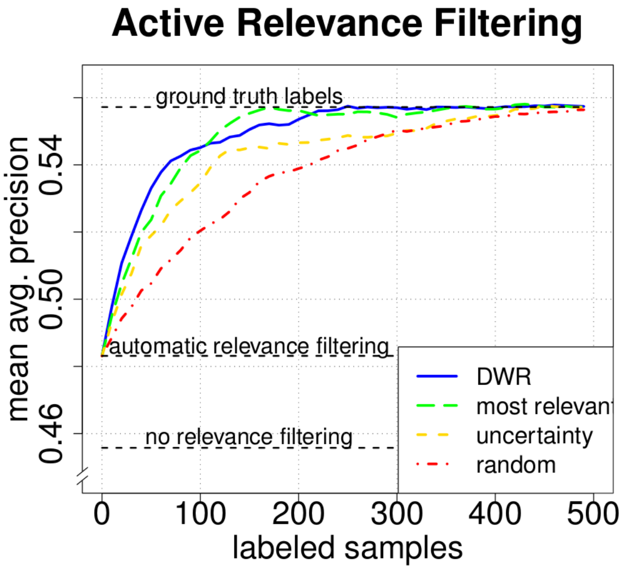


Active Learning: Example[1]

The TRECVID Collaborative Annotation Effort

[1]image source: Ayache, Quenot: *Video Corpus Annotation using Active Learning, 2008*

Active Learning: Example



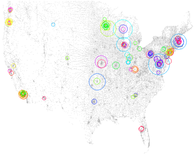
Outline

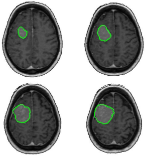
Anomaly Detection: Mission Statement

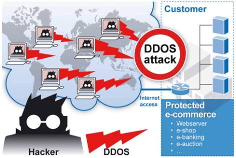
Goal: identify samples that do not conform to an expected pattern, or to other samples in the dataset[2]

Applications

- credit card fraud detection
- detecting tumours in imagery
- detecting technical component failure
- finding errors in text
- network intrusion detection



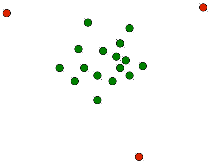




[2] Chandola, Banerjee, Kumar: Anomaly Detection: A Survey. ACM Computing Surveys, 2009.

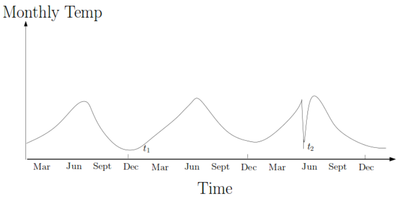
Anomaly Detection: Types of Anomalies

1. **point anomalies**: individuals that are unusual (e.g., in high distance) to the flock



2. **contextual anomalies**: use contextual features (e.g., location) and behavioral features (e.g., the temperature). An anomal occurs if the behavioral features are unusual *given the contextual ones*.
   Note: can be reduced to point anomalies using context-specific models
3. **collective anomalies**: a **combination** of samples that is unusual (whilst the individual samples are not necessarily).
   Example: … `http-web smtp-mail buffer-overflow ssh ftp` …

Our focus here will be on **point anomalies**.

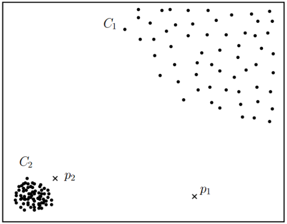


Anomaly Detection: Characteristics

Learning Setups

- Usually, there are two labels: *normal* vs. *abnormal*
- Labeled training data can be really difficult to find!
- **supervised** techniques: Training data from both classes given (but often highly imbalanced)
- **semi-supervised** techniques: Training data only for the *normal* class
- **unsupervised** techniques: training data without labels (there may be anomalies, but we do not know where)

Can we use Absolute Distance as an Anomaly Criterion?



Anomaly Detection: Methods

There is a plethora of anomaly detection methods[3]

- ... some using regular classifiers
- ... some using density-based modeling
- ... some using rule mining
- ...

We will look at two of the most prominent ones:

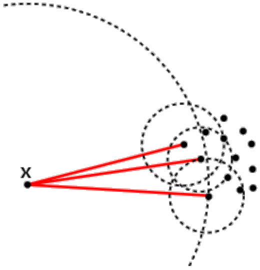
- a density-based method ("local outlier factor")
- a classification-based method ("one-class SVMs")

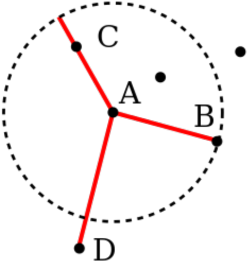
[3] Chandola, Banerjee, Kumar: Anomaly Detection: A Survey. ACM Computing Surveys, 2009.

Outline

Local Outlier Factor (LOF)[4]

- **Idea**: A sample **x** is a point anomaly if the point density in its surrounding is lower than in its nearest neighbor's surroundings.
- **Anomaly Measure**: Measure the distance to **x**'s neighbors, measure the same distance for each neighbor, and compare.





Derivation

[4] Breunig et al.: LOF: Identifying Density-based Local Outliers. Proc. ACM SIGMOD, 2000.

LOF: Derivation

We define $N_k(x)$ to be the set of K nearest neighbors to x, and $dist_k(x)$ as the distance of x to its kth nearest neighbor. Then, we define a distance between x and y:

$$d_{reach}(x, y) := max\{dist_k(y), d(x, y)\}$$

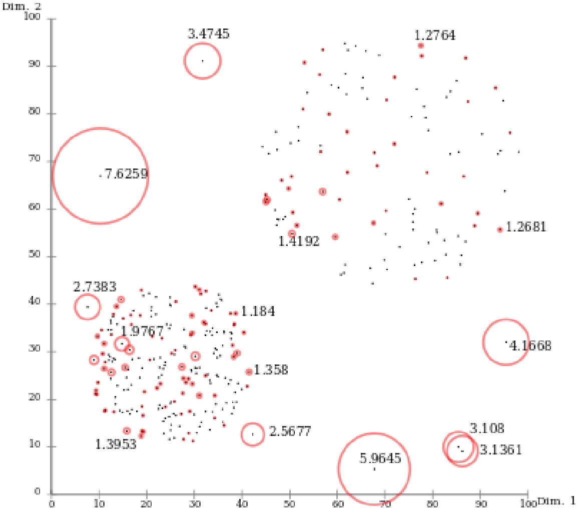
(basically, this is $d(x, y)$, with a little tweak to achieve more stable results). Then we define the local reachability **density** as:

$$lrd_k(x) := \left(\frac{1}{\#N_k(x)} \cdot \sum_{y \in N_k(x)} d_{reach}(x, y) \right)^{-1}$$

We compute the local outlier factor by comparing x' density with the one of its neighbors:

$$LOF_k(x) := \frac{1}{\#N_k(x)} \cdot \sum_{y \in N_k(x)} \frac{lrd_k(y)}{lrd_k(x)}$$

LOF: Example[5]



[5] Source: de.wikipedia.org

LOF: Discussion

- rather expensive: $O(n^2)$
  (speed-up by approximate NN search possible)
- Speed-up using **sampling techniques**: determine LOF based on subsample, then when candidates have been identified, on larger sample
- **Benefit**: no prior assumptions regarding distribution of data

Outline

1. Active Learning

2. Anomaly Detection
   - Local Outlier Factor
   - One-class SVMs

One-Class SVMs [6]

Remember SVMs?

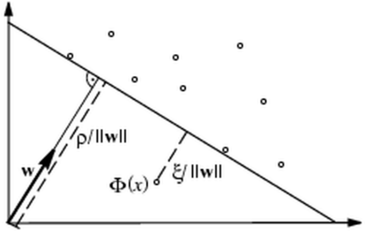
- a binary classifier that separates classes by a maximum margin hyperplane $\mathbf{w}, b$
- slack variables $\xi_1, ..., \xi_n$ that allow some training errors
- kernel trick to introduce non-linearity: map samples x_i to high-dimensional space $\phi(x_i)$

One-class SVMs

- same idea: find a hyperplane that separates "normal" samples from "abnormal" ones
- an unsupervised method: Our samples $x_1, ..., x_n$ may contain anomalies. We introduce slack variables to take them into account.

[6] Arcolano, Rudoy: One-Class Support Vector Machines: Methods and Applications. Project Presentation, Harvard University, 2008.

One-Class SVMs



Tradeoff between two Goals

- shift hyperplane as far away from the origin as possible
- at the same time, minimize number of "errors"

Quadratic Program

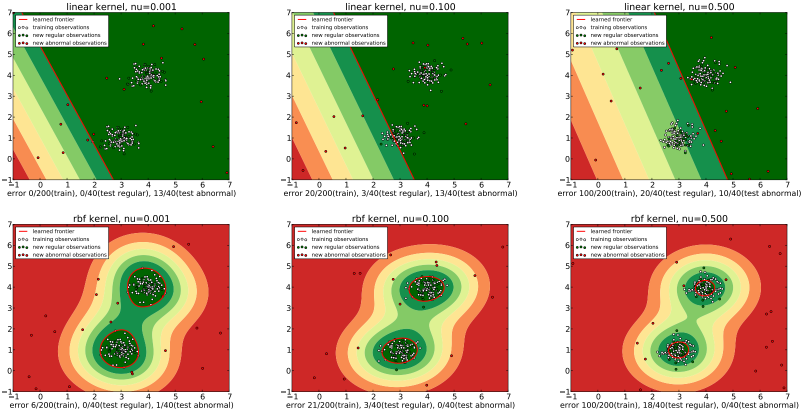
$$min_{\mathbf{w},\xi,\rho} \ \frac{1}{2} \cdot ||\mathbf{w}||^2 + \frac{1}{\nu \cdot m} \cdot \sum_i (\xi_i - \rho)$$

subject to

$$\xi_i \geq 0 \quad \text{and} \quad < r, \phi(\mathbf{x}_i) > \ \geq \ \rho - \xi_i \quad \forall i$$

- The parameter ν represents an upper bound on the fraction of training samples misclassified.
- For $\nu \to 0$, we obtain a "hard margin" problem.

One-Class SVMs: Example



One-Class SVMs: Remarks

- Applicable in high-dimensional settings where density estimation becomes tricky
- Key step: Kernel engineering (often: ad-hoc choice)
- Scalability to very large datasets tricky (usually, < 5000 samples for non-linear kernels)