Streaming Machine Learning (SML)

Alessio Bernardo

04-07-2022

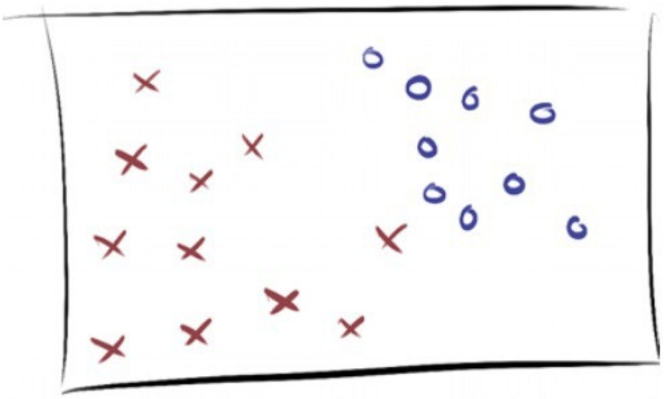
Part III

Classification

Credits

- Albert Bifet DATA STREAM MINING 2020-2021 course at Telecom Paris
- Alessio Bernardo & Emanuele Della Valle

SML Classification models



Naïve Bayes

- Based on Bayes Theorem, where c is the class and d is the instance to classify:

$$P(c|d) = \frac{P(c) * P(d|c)}{P(d)}$$

- Estimate the probability of observing attribute a and the prior probability $P(c)$:

$$P(c|d) = \frac{P(c) * \prod_{a \in d} P(a|c)}{P(d)}$$

John, G. H., & Langley, P. **Estimating continuous distributions in Bayesian classifiers**. arXiv preprint 2013.

Naïve Bayes

Mean and Variance with a batch of *n* samples

$$\hat{x} = \frac{1}{n} * \sum_{i=1}^{n} x_i \qquad \sigma^2 = \frac{1}{n-1} * \sum_{i=1}^{n} (x_i - \hat{x})^2$$

Mean and Variance with a stream $x_1, \ldots, x_i, \ldots, x_n$

$$s_i = s_{i-1} + x_i \qquad q_i = q_{n-1} + {x_i}^2$$

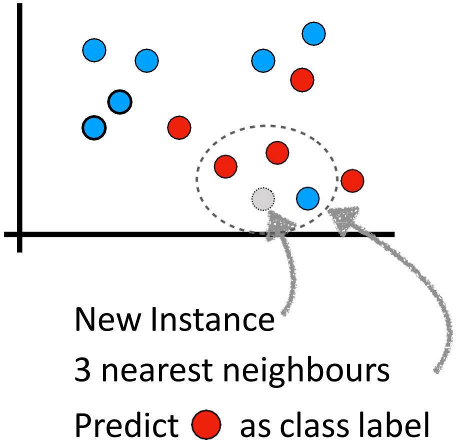
$$\hat{x}_i = \frac{s_i}{i} \qquad {\sigma_i}^2 = \frac{1}{i-1} * (q_i - \frac{{s_i}^2}{i})$$

John, G. H., & Langley, P. **Estimating continuous distributions in Bayesian classifiers**. arXiv preprint 2013.

K-Nearest Neighbours (KNN)

- The most common label of the *k* instances closer to a new instance determines its label
- The distance between instances is calculated (commonly) using the Euclidean Distance:

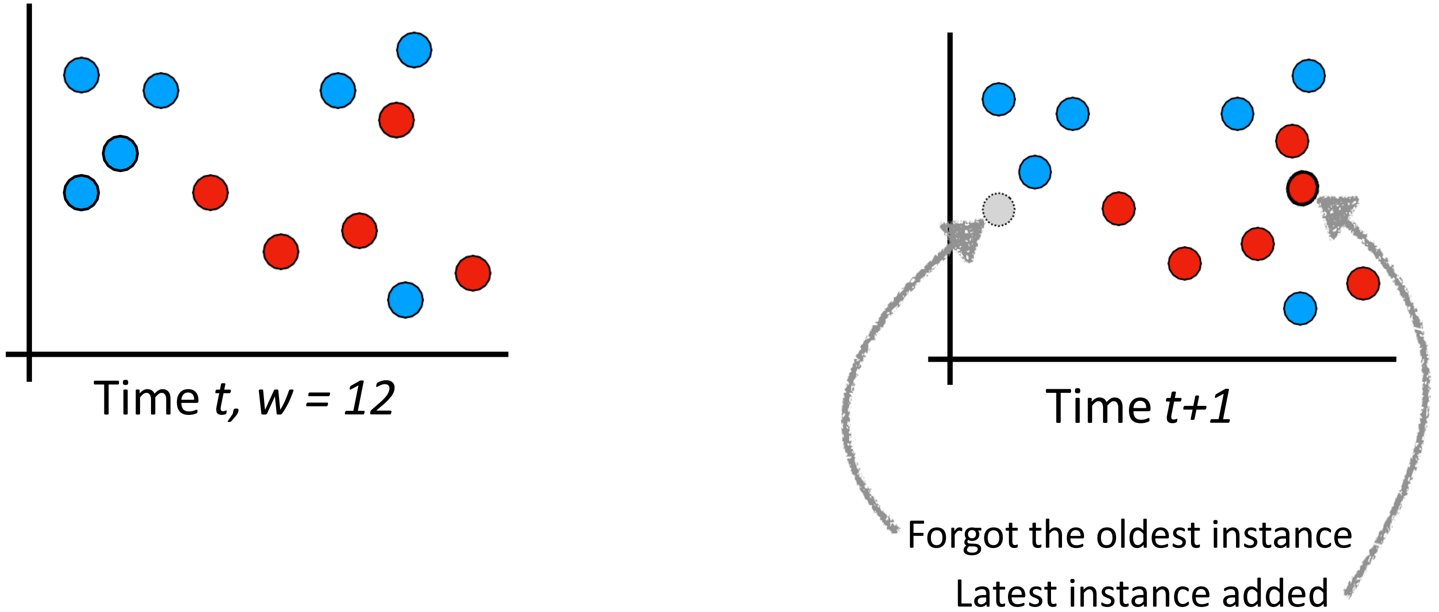
$$d(a,b) = \sqrt{\sum_{i=1}^{m} (a_i - b_i)^2}$$



Bifet, A., Pfahringer, B., Read, J., & Holmes, G. **Efficient data stream classification via probabilistic adaptive windows**. 28th ACM symposium on applied computing (2013).

Online K-Nearest Neighbours (KNN)

- Use a *fixed size sliding window* to save the instances



Bifet, A., Pfahringer, B., Read, J., & Holmes, G. **Efficient data stream classification via probabilistic adaptive windows**. 28th ACM symposium on applied computing (2013).

Online KNN with ADWIN (KNN-ADWIN)

- If a concept drift occurs, with KNN there is the risk that the instances saved into the window belong to the old concept
- Use ADWIN to *automatically* set the *size* of the *sliding window* to save the instances

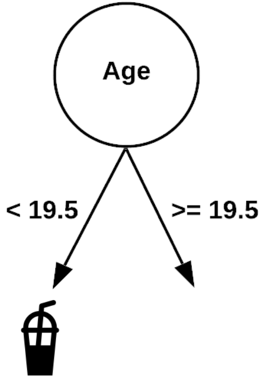
Bifet, A., Pfahringer, B., Read, J., & Holmes, G. **Efficient data stream classification via probabilistic adaptive windows**. 28th ACM symposium on applied computing (2013).

Decision Trees

Recommending drinks

Gender	Age	Drink
F	13	
M	13	
F	23	
M	32	
F	42	
M	16	

- Which feature best determines the drink?
  - **Age**



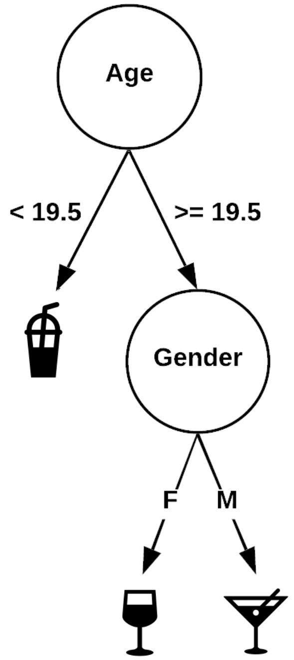
https://en.wikipedia.org/wiki/Decision_tree_learning

Decision Trees

Recommending drinks

Gender	Age	Drink
~~F~~	~~13~~	
~~M~~	~~13~~	
F	23	
M	32	
F	42	
~~M~~	~~16~~	

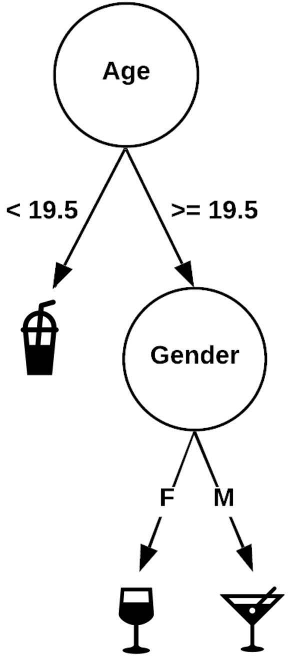
- Which feature best determines the drink?
  - ➢ **Age**



https://en.wikipedia.org/wiki/Decision_tree_learning

Decision Trees

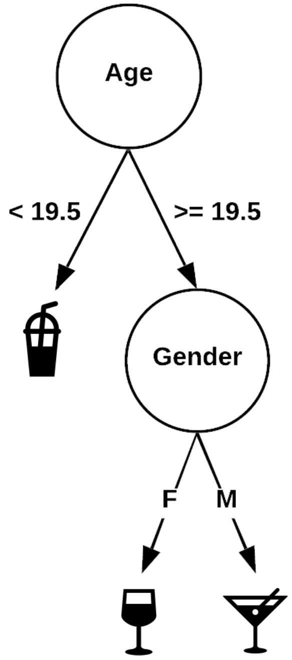
- Each node tests a features
- Each branch represents a value
- Each leaf assigns a class
- Greedy recursive induction:
  - Sort all examples through tree
  - X_i = most discriminative attribute using the Gini index or Information Gain (H)
  - New node for X_i, new branch for each value, leaf assigns majority class
  - Stop if no error or limit on #instances



https://en.wikipedia.org/wiki/Decision_tree_learning

Hoeffding Trees (VFDT)

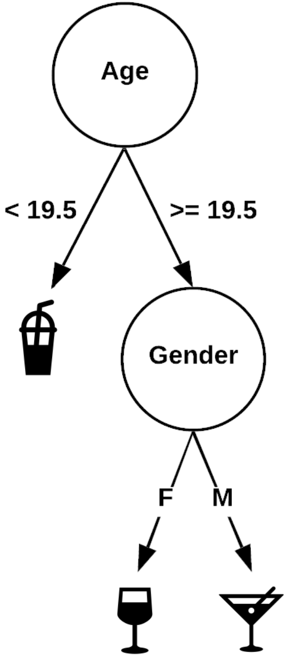
- Build the decision tree incrementally
- The final tree must be identical (with high probability) to a tree built using a batch decision tree algorithm
- With theoretical guarantees on the error rate



Pedro Domingos and Geoff Hulten**. Mining high-speed data streams**. 2000

Hoeffding Trees (VFDT)

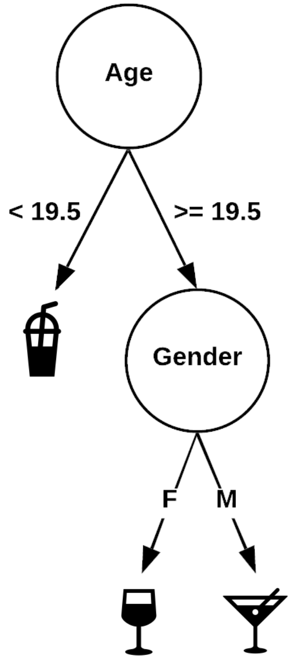
- Which attribute to choose at each splitting node?
- A small sample can often be enough to choose the optimal splitting attribute
  - Collect sufficient statistics from a small set of examples
  - Estimate the merit of each attribute
- How large should be the sample?
  - **Fixed size**: defined *apriori* without looking for the data



Pedro Domingos and Geoff Hulten**. Mining high-speed data streams**. 2000

Hoeffding Trees (VFDT)

- Which attribute to choose at each splitting node?
- A small sample can often be enough to choose the optimal splitting attribute
  - ➢ Collect sufficient statistics from a small set of examples
  - ➢ Estimate the merit of each attribute
- How large should be the sample?
  - ❌ ➢ **Fixed size**: defined *apriori* without looking for the data
  - ✅ ➢ **Moving size**: Choose the sample size that allow to differentiate between the alternatives.

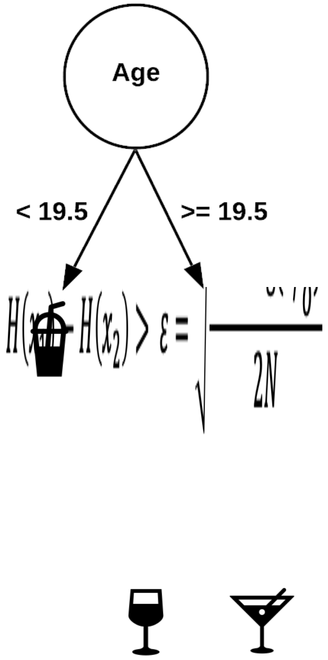


Pedro Domingos and Geoff Hulten. **Mining high-speed data streams**. 2000

Hoeffding Trees (VFDT)

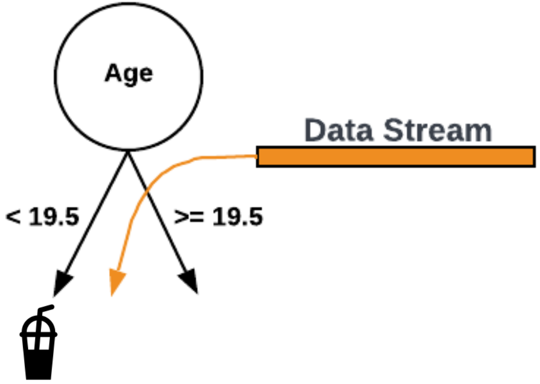
- **Moving size:** Use Hoeffding bound to guarantee that the best attribute is really the *best*:
  - ➢ Let X_1 and X_2 be, respectively, the two most informative attribute
  - ➢ Split if: $H(x_1) - H(x_2) > \ \varepsilon = \sqrt{\frac{R^2 * \log(^1/_\delta)}{2N}}$

    where *R* is the *G* range, δ is the confidence bound and *N* is the number of instances seen by that node



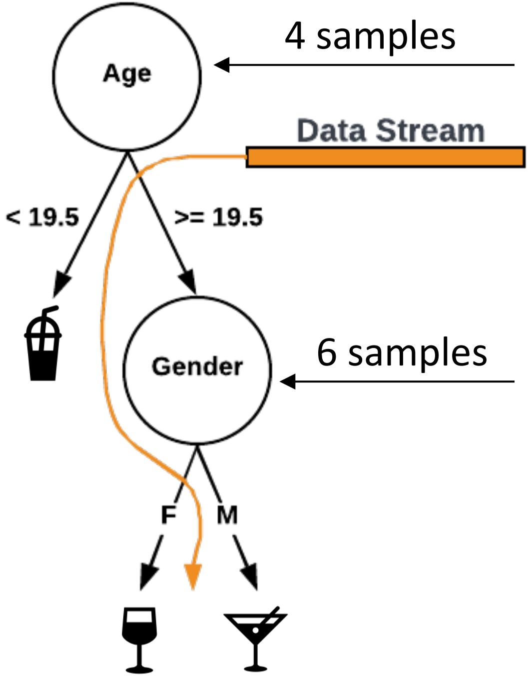
Pedro Domingos and Geoff Hulten**. Mining high-speed data streams**. 2000

Hoeffding Trees (VFDT)



$$H(Gender) - H(Age) > \epsilon$$

$$\epsilon = \sqrt{\frac{R^2 * \log(1/\delta)}{2N}}$$



Attributes:

- **Age**
- **Gender**

Label: **Drink**

Concept Adapting VFDT (CVFDT)

- What happens when a **concept drift** occurs?
  - ➢ The nodes are no longer representative of the current concept
- CVFDT keeps its model consistent with a sliding window of *w* samples
- It constructs “alternative branches” as preparation for changes
- If the alternative branch becomes more accurate, switch of tree branches

Cons:

- ➢ No theoretical guarantees on the error rate of CVFDT
- ➢ *W* is fixed

G. Hulten, L. Spencer, and P. Domingos. **Mining time-changing data streams.** 2001

Hoeffding Adaptive Tree (HAT)

- Replace frequency statistics counters by estimators
  - ➢ Don’t need a window to store examples, due to the fact that we maintain the statistics data needed with estimators

- Change the way of checking the substitution of alternate subtrees, using a change detector with theoretical guarantees (ADWIN)
  - ➢ Keeps sliding window consistent with the *no-change hypothesis*

Pro:

➢ Theoretical guarantees

➢ No Parameters

A. Bifet, R. Gavald`a. **Adaptive Parameter-free Learning from Evolving Data Streams.** IDA, 2009

CASH problem and AutoML

CASH problem: Combined Algorithm Selection and Hyperparameter.

AutoML aims to automate the data mining pipeline:

- Data cleaning.
- Feature engineering.
- Algorithm selection.
- Hyperparameters tuning.

Different implementations with different search spaces and hyperparameter optimizations:

- Auto Weka 2.0
- Autosklearn
- TPOT
- GAMA
- H2O

CASH problem with SML

CASH solution does not consider the adaptation of parameters in an evolving data stream.

Actual applications to a streaming scenario:

- Train AutoML only the first portion of the data stream.
- Retrain AutoML from scratch after a concept drift:
- Computational expensive.
- Large number of parallel trainings.
- Only consider algorithm selection.

EvoAutoML

- It naturally adapts the population of algorithms and configurations.
- It avoids expensive retraining.
- It addresses the Online CASH problem by finding the joint algorithm combination and hyperparameter setting that minimizes a predefined loss over a stream of data.

It considers:

- Pipeline structure
- Algorithms
- Configuration space.
- It makes predictions by majority voting.

C. Kulbach, J. Montiel, M. Bahri, M. Heyden, & A. Bifet. **Evolution-Based Online Automated Machine Learning.** PAKDD, 2022

QUIZ

1. How does the SML version of KNN work?
    a. By saving all the instances and finding the neighbors among them
    b. By saving the last *w* instances in a sliding window and finding the neighbors among them
    c. Without saving any instances
2. Why is Hoeffding Bound used in SML to build a decision tree?
    a. To find the best attribute for which make a split
    b. To make a prediction
    c. To check any statistical evidences that the number of samples seen by a node are enough to make a split

QUIZ

1. How does the SML version of KNN work?
    a. By saving all the instances and finding the neighbors among them
    b. By saving the last *w* instances in a sliding window and finding the neighbors among them
    c. Without saving any instances
2. Why is Hoeffding Bound used in SML to build a decision tree?
    a. To find the best attribute for which make a split
    b. To make a prediction
    c. To check any statistical evidences that the number of samples seen by a node are enough to make a split

QUIZ

1. How does the SML version of KNN work?
   a. By saving all the instances and finding the neighbors among them
   b. By saving the last *w* instances in a sliding window and finding the neighbors among them
   c. Without saving any instances
2. Why is Hoeffding Bound used in SML to build a decision tree?
   a. To find the best attribute for which make a split
   b. To make a prediction
   c. To check any statistical evidences that the number of samples seen by a node are enough to make a split

EXERCISE 3: Stream Classification
LAB 3: Final Challenge A