Data mining for Software Engineering
-- Some Final Remarks

Jerzy Stefanowski
Poznań University of Technology
notes ver. 2010

Absence List - What We Should ...

Other interesting topics

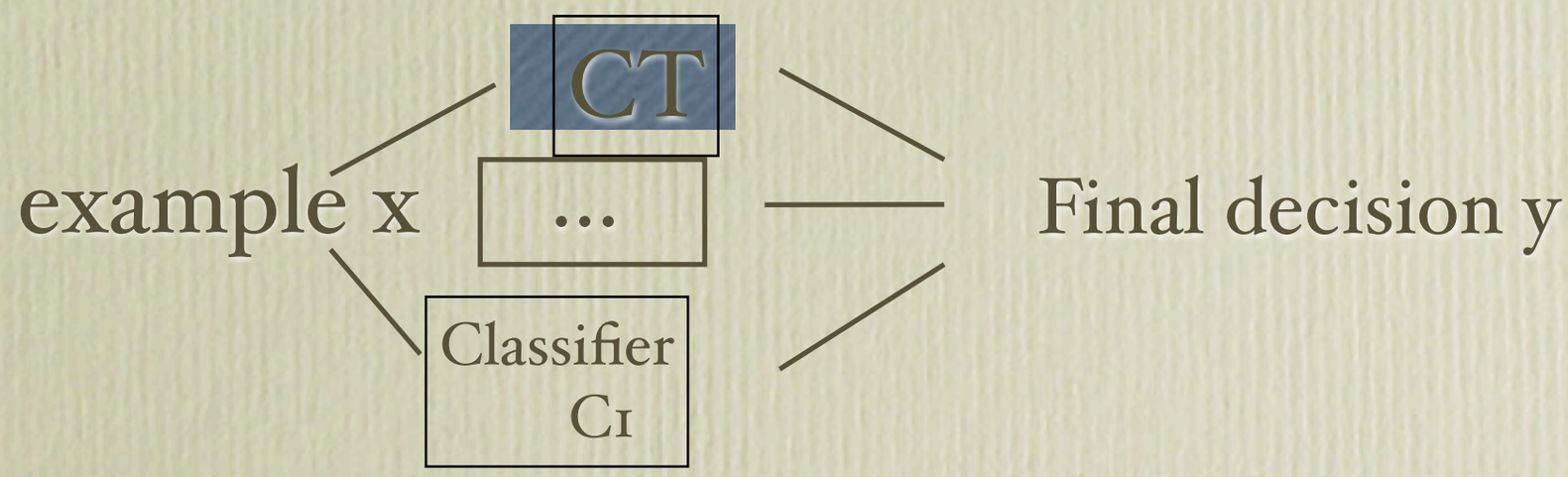
- Classifiers
  - Multiple Classifiers (Ensembles)
  - Statistical Classifiers - SVM and others
  - Class Imbalance Problems
- Visualization in Exploring Data
- Anomaly Detection
- Mining Text or Web Data

Multiple Classifiers

- Typical research → create and evaluate a single learning algorithm; compare performance of some algorithms.
- Empirical observations or applications → a given algorithm may outperform all others for a specific subset of problems
- There is no one algorithm achieving the best accuracy for all situations! [No free lunch]
- Growing research interest in combining a set of learning algorithms / classifiers into one system
- „Multiple learning systems try to exploit the local different behavior of the base learners to enhance the accuracy of the

Multiple classifiers - definitions

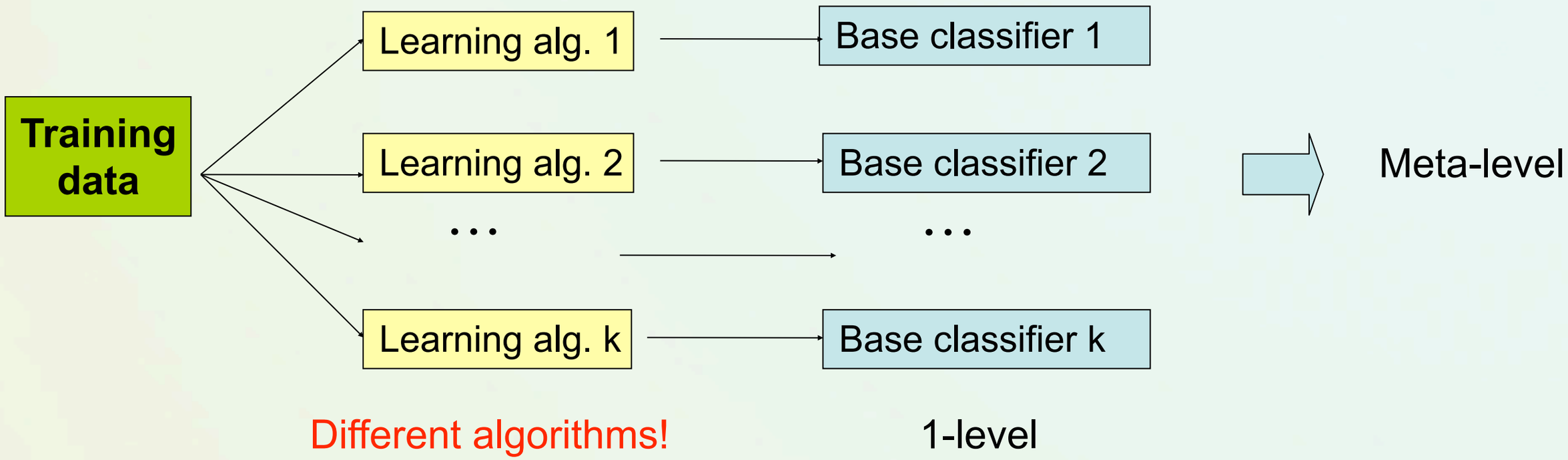
- Multiple classifier – a set of classifiers whose individual predictions are combined in some way to classify new examples.
- Various names: ensemble methods, committee, classifier fusion, combination, aggregation,...
- Integration should improve predictive accuracy!
- Diversity of component classifiers – if they make errors, then they should not correlated!



Approaches to create multiple systems

- Homogeneous classifiers – use of the same algorithm over diversified data sets
  - Bagging (Breiman)
  - Boosting (Freund, Schapire)
  - Multiple partitioned data
  - Multi-class specialized systems, (e.g. ECOC pairwise classification)
- Heterogeneous classifiers – different learning algorithms over the same data
  - Voting or rule-fixed aggregation
  - Stacked generalization or meta-learning

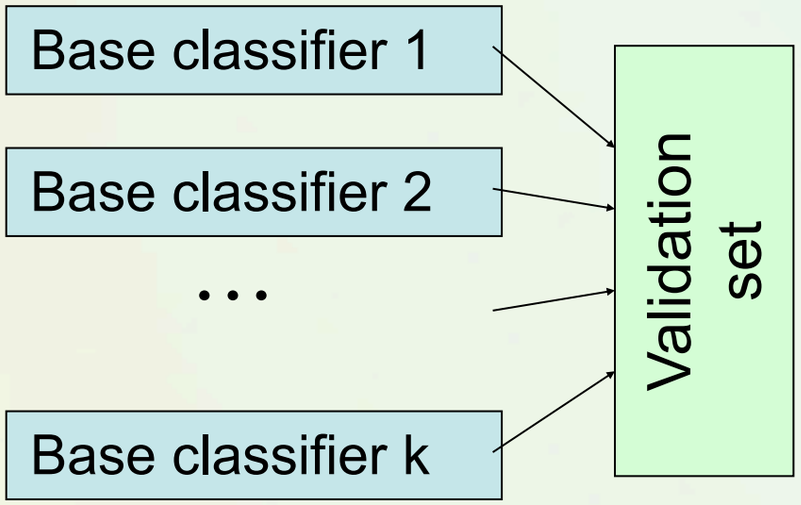
The Combiner Classifier - 1

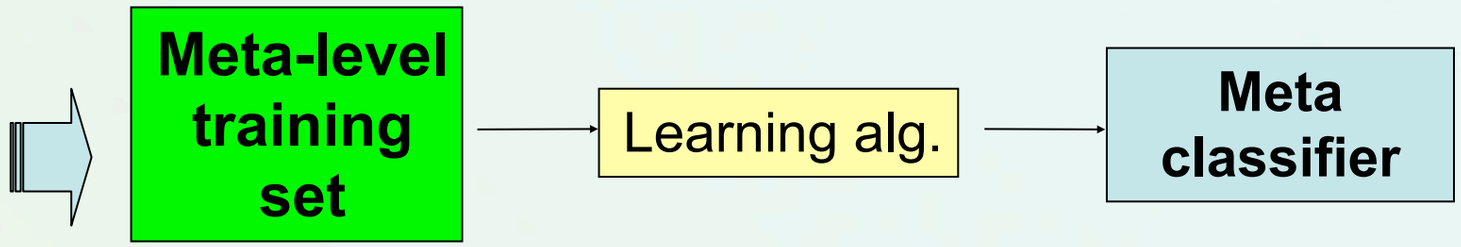


Chan & Stolfo : *Meta-learning*.

- Two-layered architecture:
  - 1-level – base classifiers.
  - 2-level – meta-classifier.
- Base classifiers created by applying the different learning algorithms to the same data.

Learning the meta-classifier

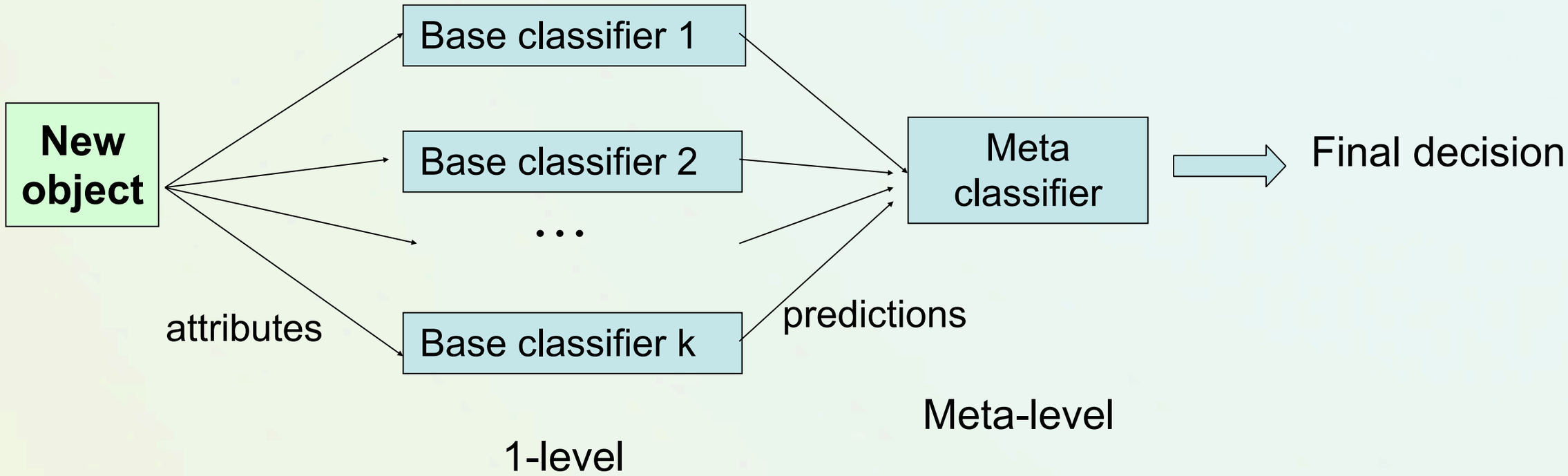




Predictions				Dec. class
Cl.1	Cl.2	…	Cl.K	
A	A	…	B	**A**
A	B	…	C	**B**

- Predictions of base classifiers on an extra validation set (not directly training set – apply „internal" cross validation) with correct class decisions → a meta-level training set.
- An extra learning algorithm is used to construct a meta-classifiers.
- The idea → a meta-classifier attempts to learn relationships between predictions and the final decision;
  It may correct some mistakes of the base classifiers.

The Combiner - 2



Classification of a new instance by the combiner

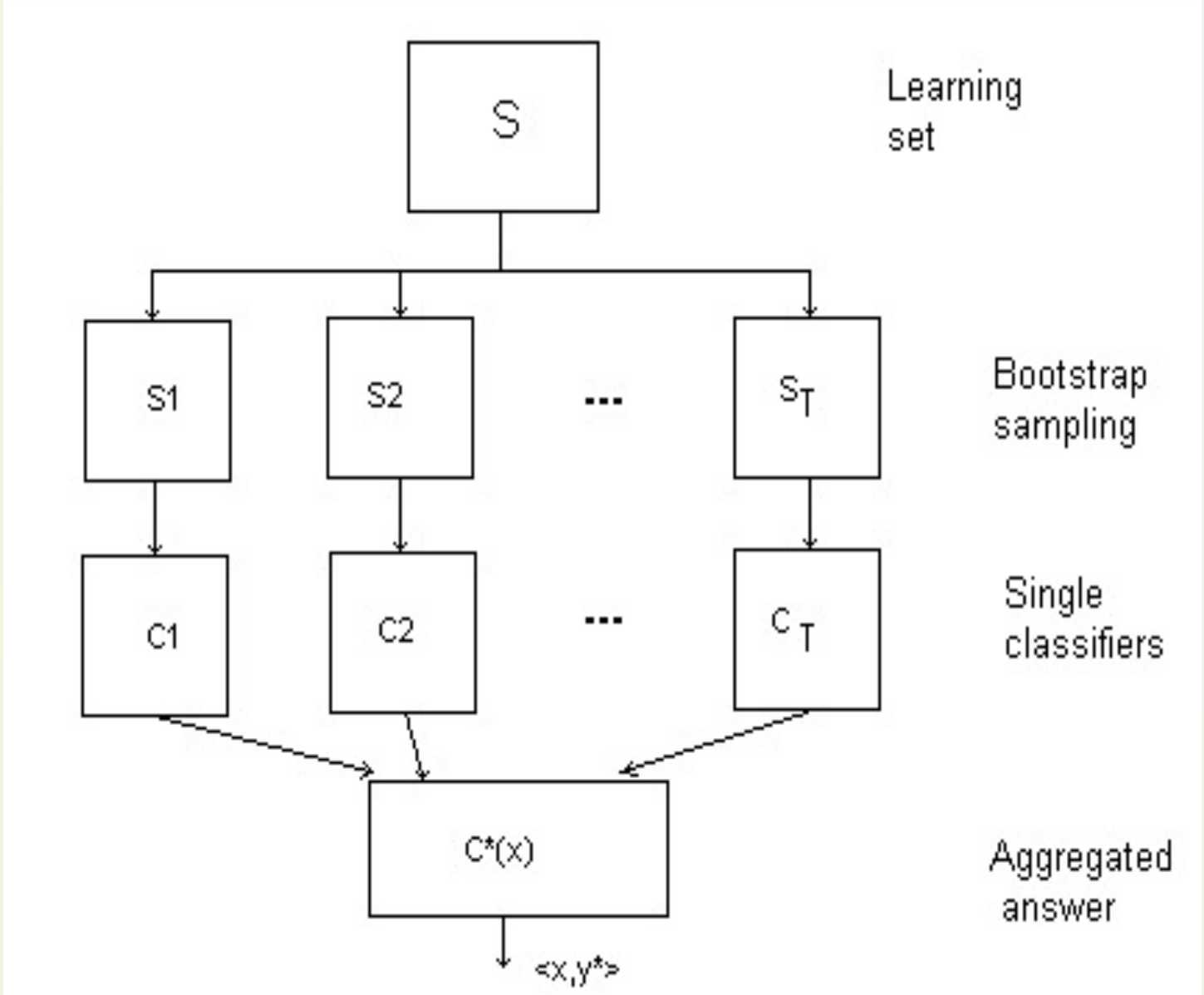
- Chan & Stolfo [95/97] : experiments that their combiner ({CART,ID3,K-NN}→NBayes) is better than equal voting.

Bagging [L.Breiman, 1996]

- Bagging = Bootstrap aggregation
- Generates individual classifiers on bootstrap samples of the training set
- As a result of the sampling-with-replacement procedure, each classifier is trained on the average of 63.2% of the training examples.
- For a dataset with N examples, each example has a probability of 1-(1-1/N)N of being selected at least once in the N samples. For N→∞, this number converges to (1-1/e) or 0.632 [Bauer and Kohavi, 1999]
- Bagging traditionally uses component classifiers of the same type (e.g., decision trees), and combines prediction by a simple majority voting across.

More about „Bagging”

- Bootstrap aggregating – L.Breiman [1996]



input S – learning set, T – no. of bootstrap samples, LA – learning algorithm

output C* - multiple classifier

for i=1 **to** T **do**

begin

S_i:=bootstrap sample from S;

C_i:=$LA(S_i)$;

end;

$$C^*(x) = \text{argmax}_y \sum_{i=1}^{T} (C_i(x) = y)$$

Boosting [Freund & Schapire]

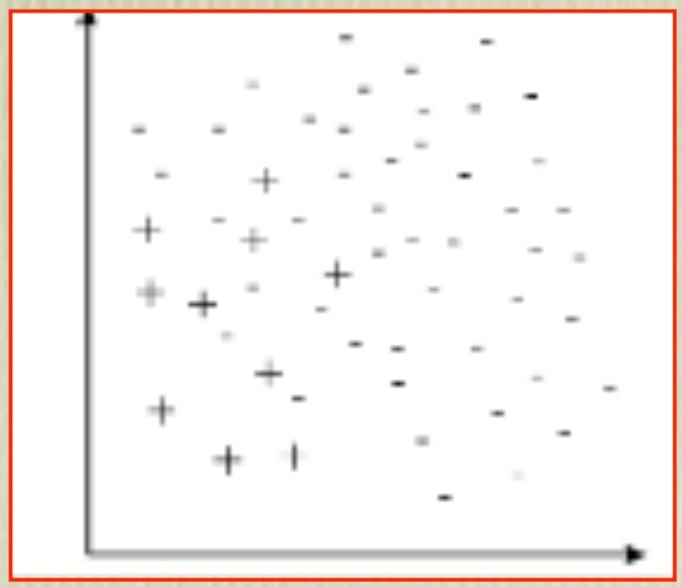
- In general takes a different weighting schema of resampling than bagging.
- Iterative procedure:
  - The component classifiers are built sequentially, and examples that are misclassified by previous components are chosen more often than those that are correctly classified!
  - So, new classifiers are influenced by performance of previously built ones. New classifier is encouraged to become expert for instances classified incorrectly by earlier classifier.
- There are several variants of this algorithm – AdaBoost the most popular (see also arcing).

Random forests [Breiman]

- Feature selection within bagging framework.
- At every level, choose a random subset of the attributes (not examples) and choose the best split among those attributes.
- Combined with selecting examples like basic bagging.
- Doesn't overfit.

Class imbalance

- Data set is said to present a class imbalance if it contains many more examples of one class than the other.
  - There exist many domains that do not have a balanced data set.
  - There are a lot of problems where the most important knowledge usually resides in the minority class.

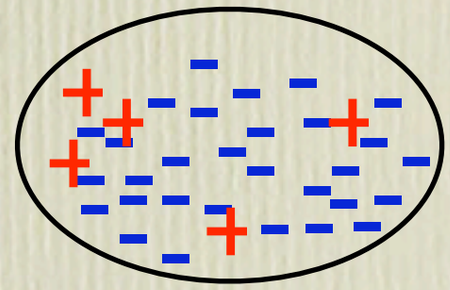


- Some real-problems: Fraudulent credit card transactions, Learning word pronunciation, Prediction of telecommunications equipment failures, Detection oil spills from satellite images, Medical diagnosis, Intrusion detection, Insurance risk modeling, Hardware fault detection

Imbalance → Difficulties

- Standard approach to learn classifiers such as decision tree induction are designed under assumption of partly balanced classes and to optimize overall accuracy without taking into account the relative distribution of each class.

- As a result, these classifiers tend to ignore small classes while concentrating on classifying the large ones accurately



Introduction to Imbalanced Data Sets

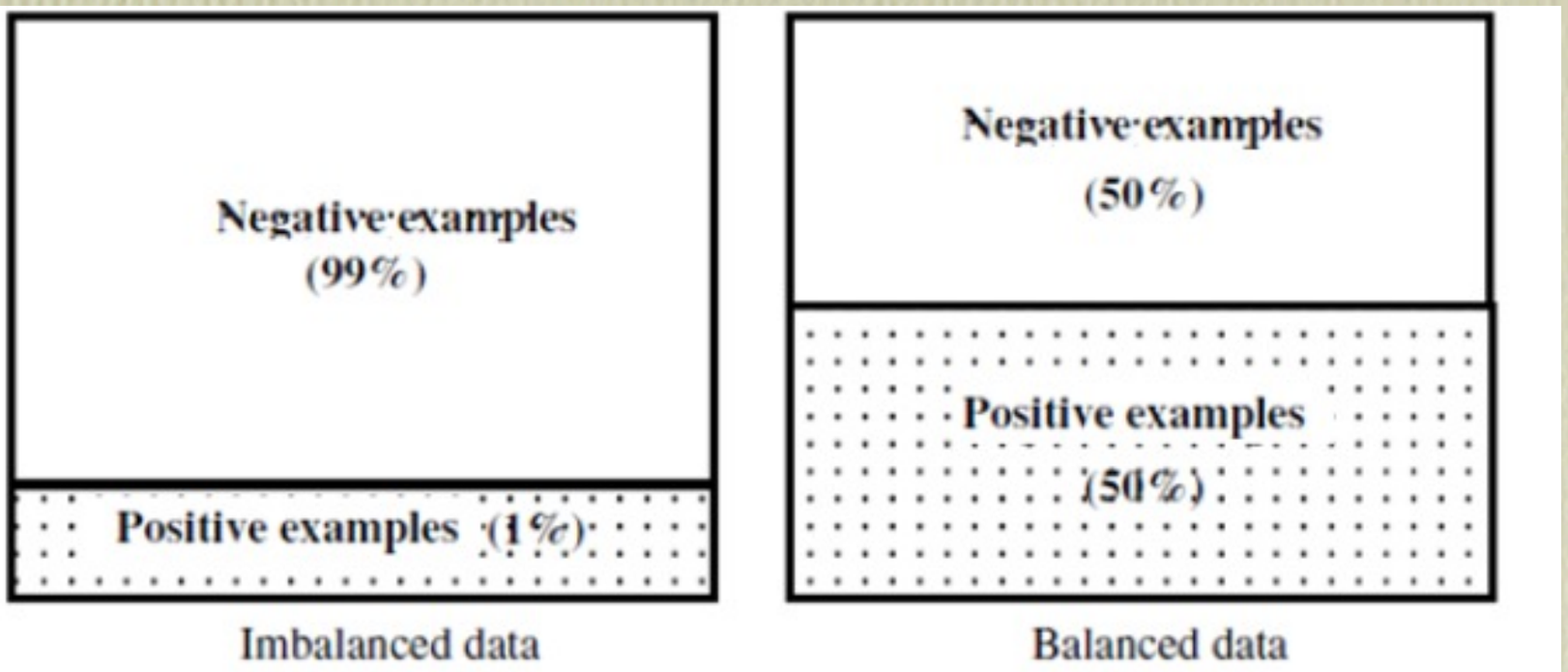


Fig. 1. Imbalanced and balanced data sets.

biased towards the majority class

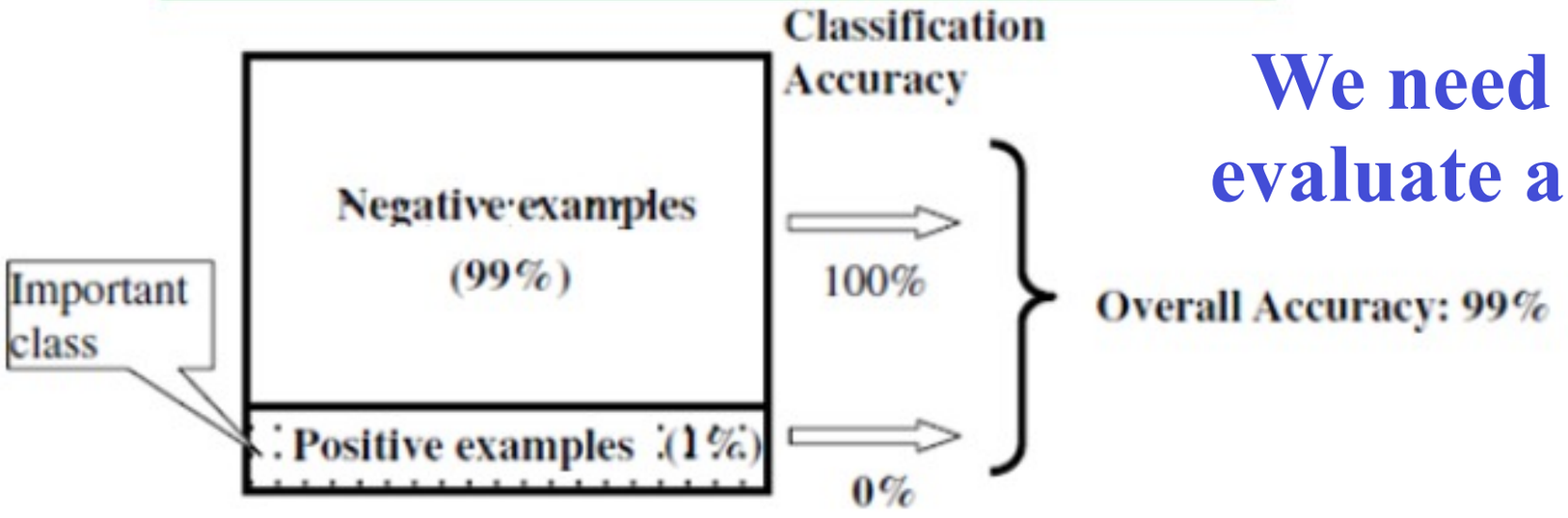


Fig. 2. The illustration of class imbalance problems.

We need to change the way to evaluate a model performance!

After Francisco Herrera lecture

Data level vs Algorithm Level

Strategies to deal with imbalanced data sets

Motivation

Over-Sampling
- **Random**
- **Focused**

→ **Retain influyent examples**
Balance the training set

Under-Sampling
- **Random**
- **Focused**

Text

→ **Remove noisy instances in the decision boundaries**
Reduce the training set

Cost Modifying (cost-sensitive)

Algorithm-level approaches: A commont strategy to deal with the class imbalance is to choose an appropriate inductive bias.

Boosting approaches: ensemble learning, AdaBoost, …

After Francisco Herrera lecture

Data level vs Algorithm Level

Strategies to deal with imbalanced data sets

Motivation

Over-Sampling

Random

Focused

Under-Sampling

Random

Focused

Text

Retain influyent examples
Balance the training set

Remove noisy instances in the decision boundaries

Reduce the training set

Cost Modifying (cost-sensitive)

Algorithm-level approaches: A commont strategy to deal with the class imbalance is to choose an appropriate inductive bias.

Boosting approaches: ensemble learning, AdaBoost, …

After Francisco Herrera lecture

Data level vs Algorithm Level

Strategies to deal with imbalanced data sets

Motivation

Over-Sampling
- **Random**
- **Focused**

Under-Sampling
- **Random**
- **Focused**

Text

Retain influyent examples
Balance the training set

Remove noisy instances in the decision boundaries
Reduce the training set

Cost Modifying (cost-sensitive)

Algorithm-level approaches: A commont strategy to deal with the class imbalance is to choose an appropriate inductive bias.

Boosting approaches: ensemble learning, AdaBoost, …

After Francisco Herrera lecture

Data level vs Algorithm Level

Strategies to deal with imbalanced data sets

Motivation

Over-Sampling
- **Random**
- **Focused**

Under-Sampling
- **Random**
- **Focused**

Text

Retain influyent examples
Balance the training set

Remove noisy instances in the decision boundaries
Reduce the training set

Cost Modifying (cost-sensitive)

Algorithm-level approaches: A commont strategy to deal with the class imbalance is to choose an appropriate inductive bias.

Boosting approaches: ensemble learning, AdaBoost, …

After Francisco Herrera lecture

Data level vs Algorithm Level

Strategies to deal with imbalanced data sets

Motivation

Over-Sampling
- **Random**
- **Focused**

Under-Sampling
- **Random**
- **Focused**

Text

Retain influyent examples
Balance the training set

Remove noisy instances in the decision boundaries
Reduce the training set

Cost Modifying (cost-sensitive)

Algorithm-level approaches: A commont strategy to deal with the class imbalance is to choose an appropriate inductive bias.

Boosting approaches: ensemble learning, AdaBoost, …

After Francisco Herrera lecture

Data level vs Algorithm Level

Strategies to deal with imbalanced data sets

Motivation

Over-Sampling
- **Random**
- **Focused**

Under-Sampling
- **Random**
- **Focused**

Text

Retain influyent examples
Balance the training set

Remove noisy instances in the decision boundaries
Reduce the training set

Cost Modifying (cost-sensitive)

Algorithm-level approaches: A commont strategy to deal with the class imbalance is to choose an appropriate inductive bias.

Boosting approaches: ensemble learning, AdaBoost, …

Some Final Remarks

- More on the above topics → Consult my lecture at ALGODEC Tutorial pages
- Look into my Web pages for the last lecture summary of KDD process, software review and relation to Business Intelligence Information systems.

Industries/fields where you currently apply data mining [KDD Pool - 216 votes total]

Banking (29) 13%

Bioinformatics/Biotech (18) 8%

Direct Marketing/Fundraising (19) 9%

eCommerce/Web (12) 6%

Entertainment/News (1) 0%

Fraud Detection (19) 9%

Insurance (15) 7%

Investment/Stocks (9) 4%

Manufacturing (9) 4%

Medical/Pharma (15) 7%

Retail (9) 4%

Scientific data (20) 9%

Security (8) 4%

Telecommunications (12) 6%

Travel (2) 1%

Other (19) 9%

Any questions, remarks?

- Thank you for your attendance in this course on data mining!
- Read additional materials