SOMACHINE

Machine Learning, Big Data, and Deep Learning in Astronomy

Singular Problems in ML

Salvador García

Andalusian Research Institute of Data Science and Computational Intelligence (DaSCI)

Dpto. Ciencias de la Computación e I.A.

Universidad de Granada

salvagl@decsai.ugr.es

http://sci2s.ugr.es

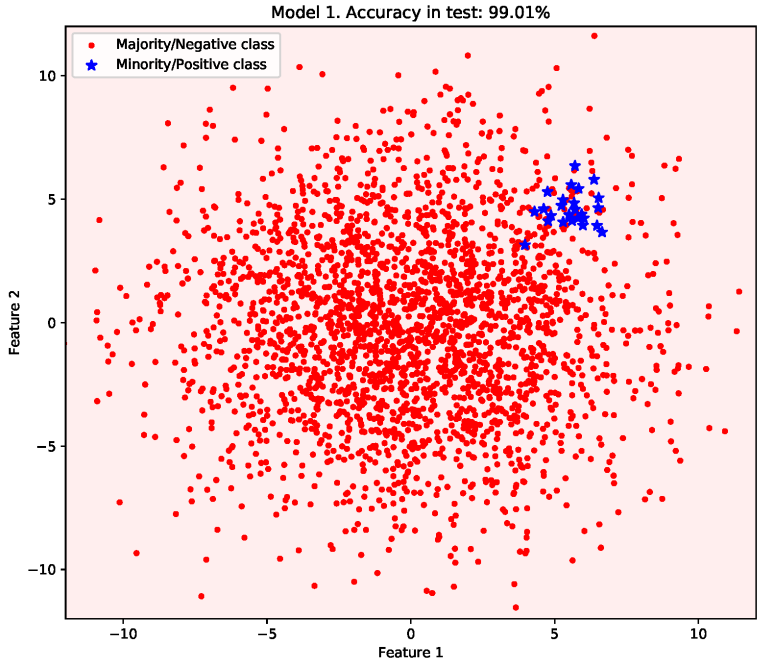
Summary

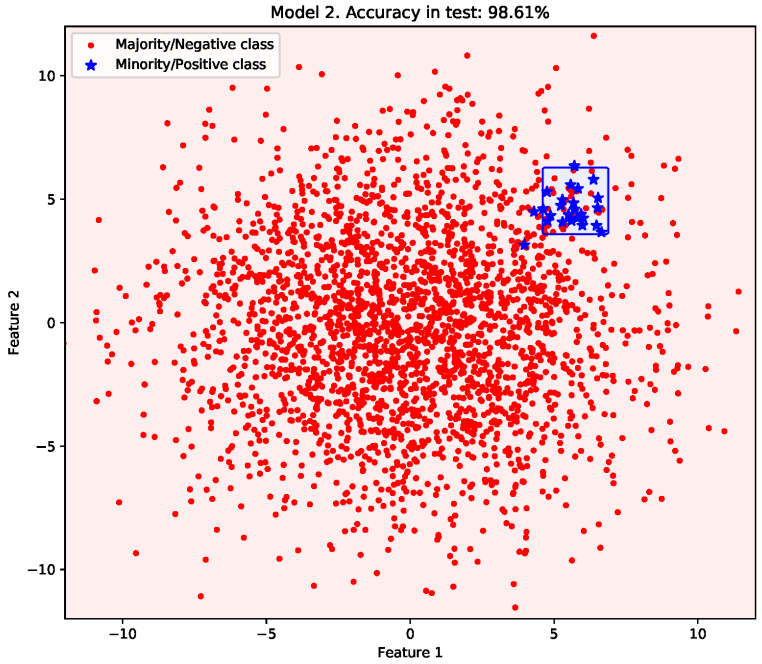
- Imbalanced Learning.
- Time Series Forecasting.
- Anomaly Detection.
- More Singular Problems:
  - Multi-Instance Learning
  - Multi-Label Learning
  - Multi-View Learning
  - Label Distribution Learning
  - Ordinal Regression and Monotonic Classification
  - Semi-Supervised Learning
  - Few-Shot Learning

Imbalanced Learning

Problem Definition

- Real application areas in engineering characterised by having a **very different distribution** of examples among their classes.
- Intrinsic to the problem or due to limitations during the data collection process.
- Positive class often represents the concept of the highest interest for the problem, whereas the negative class represents counter-examples.
- **Problem of imbalanced data-sets**: it sets a handicap for the correct identification of the different concepts to be learnt.

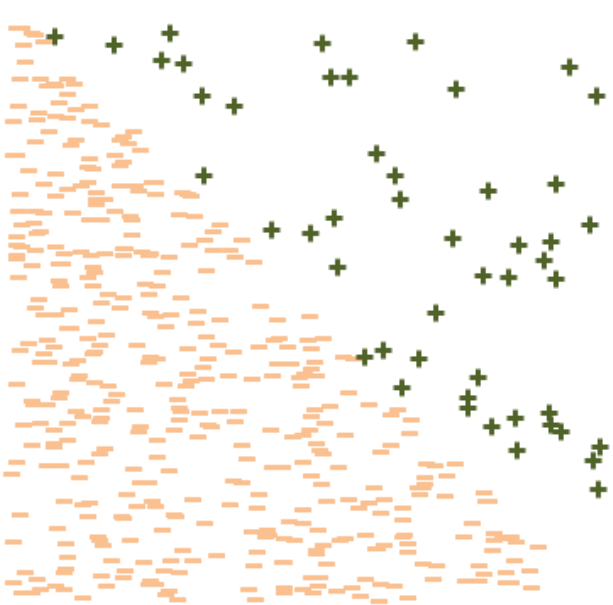




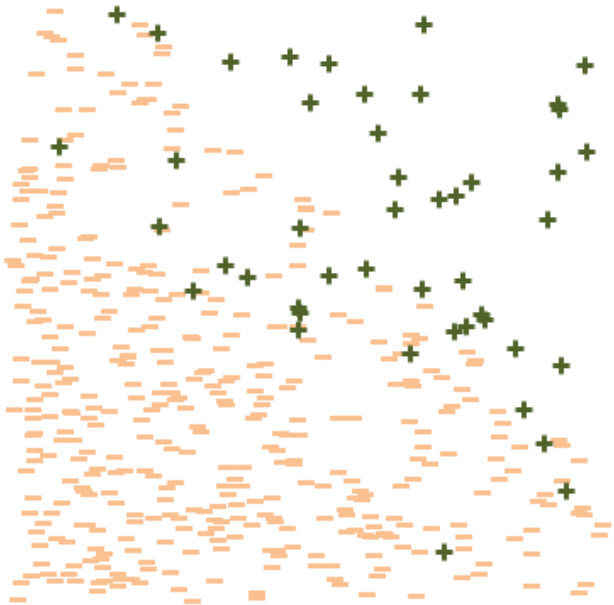
Imbalanced Learning

Properties and Difficulty

- **Intrinsic Data Characteristics**
  - Not only imbalance hinders classification performance
  - IR $\approx$ 9



Easy problem



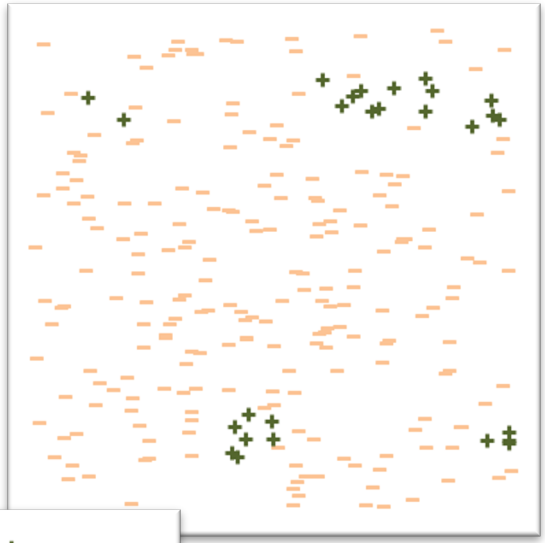
Difficult problem

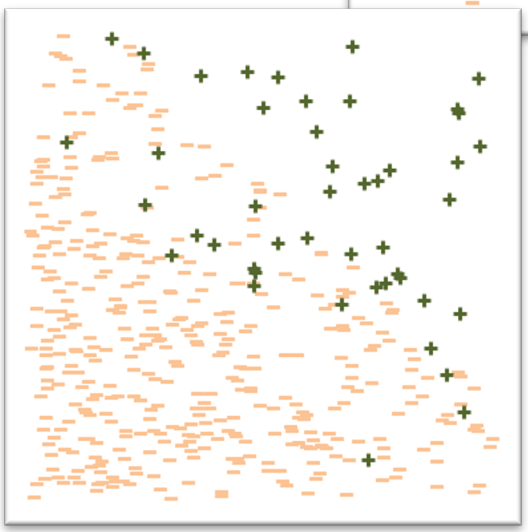
V. López, A. Fernandez, S. García, V. Palade, F. Herrera, An Insight into Classification with Imbalanced Data: Empirical Results and Current Trends on Using Data Intrinsic Characteristics. Information Sciences 250 (2013) 113-141

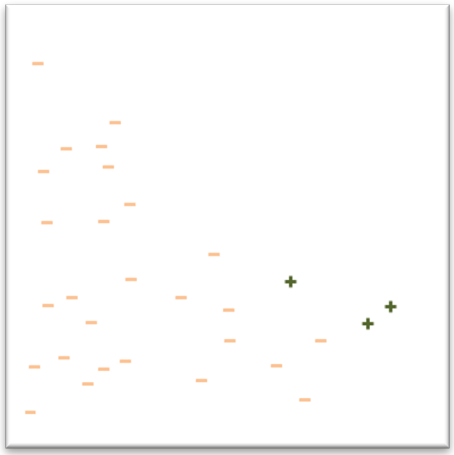
Imbalanced Learning

Properties and Difficulty

- **Intrinsic Data Characteristics**: sources of difficulties.
  - *Overlapping*,
  - Small disjuncts,
  - Lack of data,
  - …
- Majority classes overlaps the minority class:
  - Ambiguous boundary between classes
  - Influence of noisy examples
  - Difficult borderline areas, …







Imbalanced Learning

Evaluation

- How can we evaluate an algorithm in imbalanced domains?
  - **Confusion matrix for a two-class problem**

	Positive Prediction	Negative Prediction
Positive Class	True Positive (TP)	False Negative (FN)
Negative Class	False Positive (FP)	True Negative (TN)

It doesn't take into account the *"Individual Rates"*, which are very important in imbalanced problems

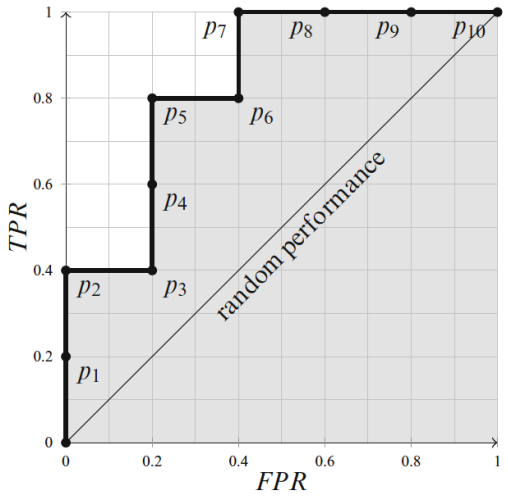
Classical evaluation:

Error Rate: (FP + FN)/N
Accuracy Rate: (TP + TN) /N

Imbalanced Learning

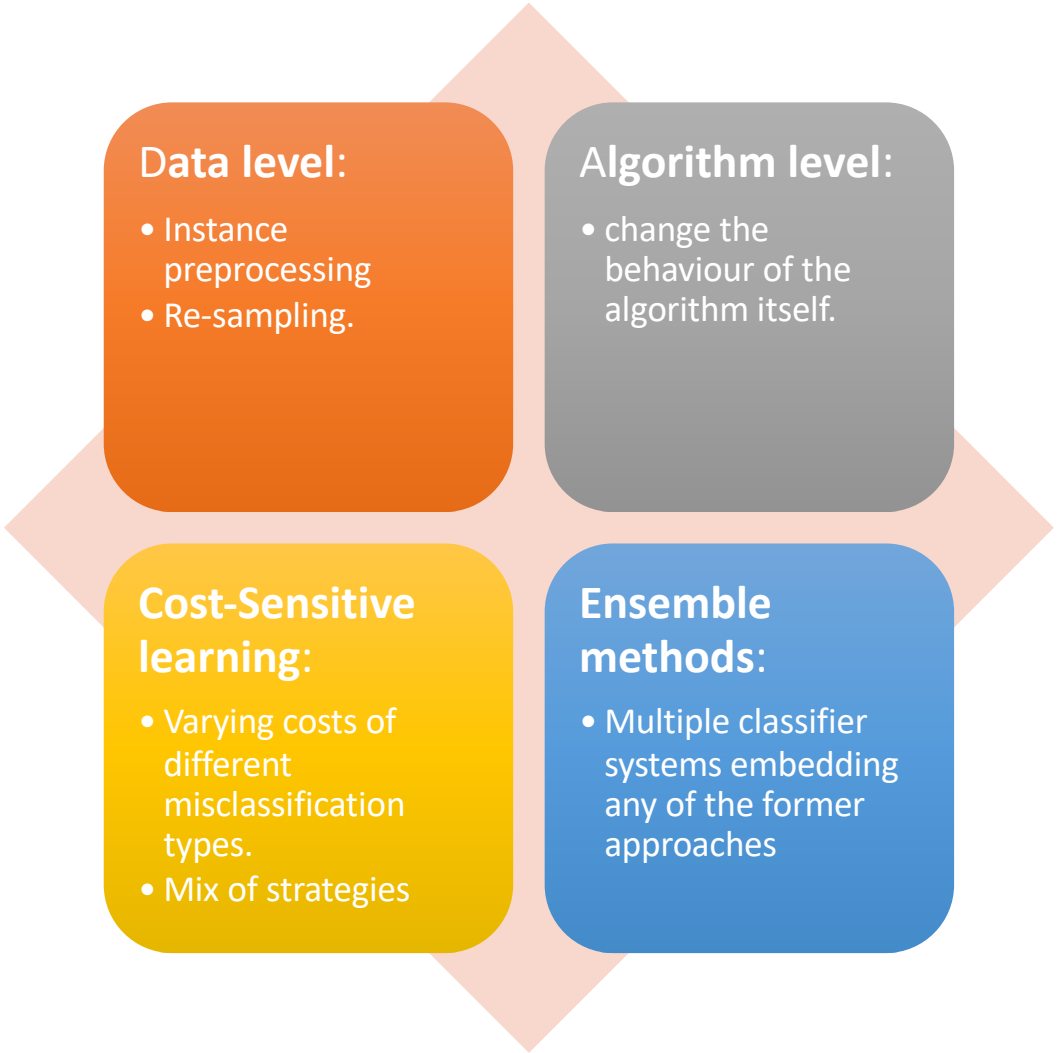
Evaluation

- Imbalanced evaluation based on the geometric mean:
  - Positive true ratio (sensitivity): $a^+ = \frac{TP}{(TP + FN)}$
  - Negative true ratio (specificity): $a^- = \frac{TN}{(TN + FP)}$
  - Evaluation function: True ratio $GM = \sqrt{a^+ \cdot a^-}$
  - F-measure $F_\beta = (1 + \beta^2) \cdot \frac{precision \cdot recall}{(\beta^2 \cdot precision) + recall}$
  - ROC Curve (AUC)



Imbalanced Learning

Strategies to adress imbalanced datasets



Imbalanced Learning

Resampling the original dataset

- **Resampling** is the process of manipulating the distribution of the training examples in an effort to improve the performance of classifiers.
- There is no guarantee that the training examples occur in their optimal distribution in practical problems.
- The idea of resampling is "to **add or remove** examples with the hope of reaching the optimal distribution of the training examples" and thus, enhancing the potential ability of classifiers.

Imbalanced Learning

Resampling the original dataset

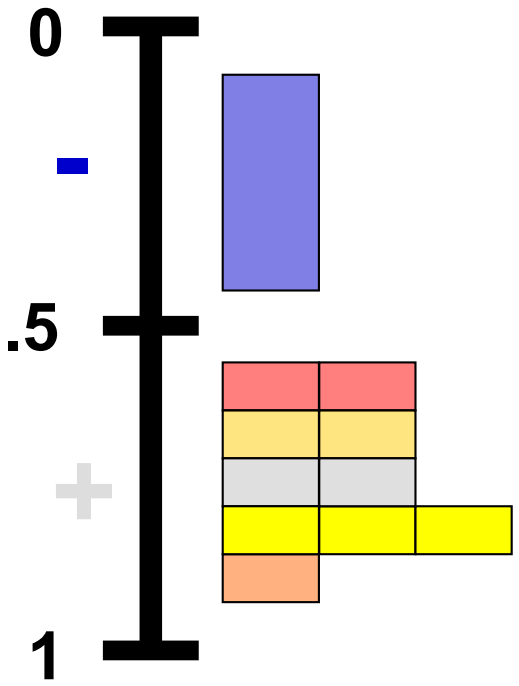
Over Sampling

- Random
- Focused

Under Sampling

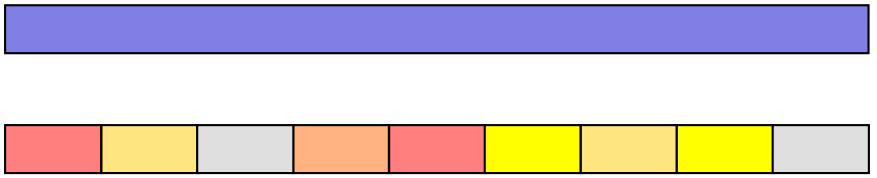
- Random
- Focused

Cost Modifying



 # examples of -

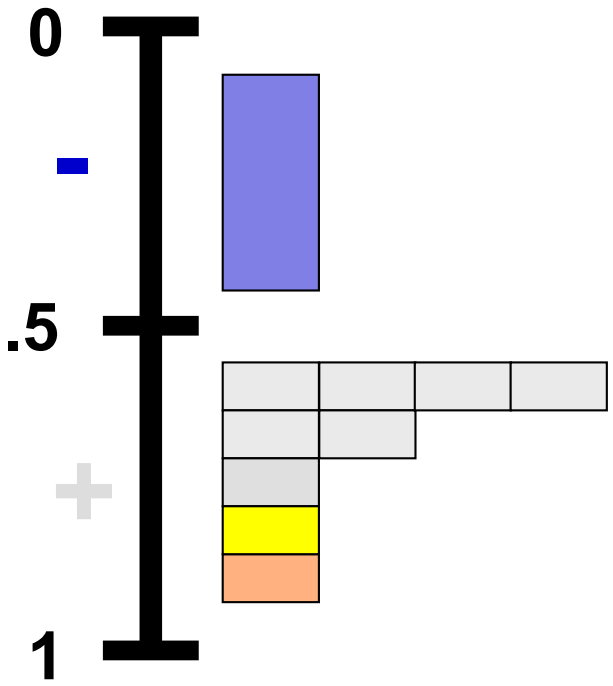
 # examples of +

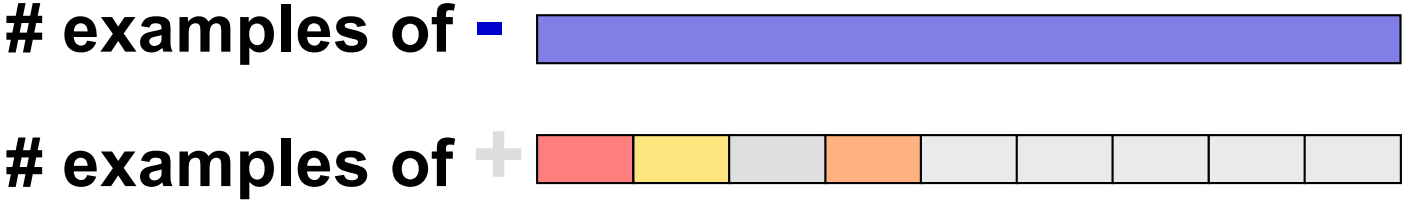


Imbalanced Learning

Resampling the original dataset

- Over Sampling
  - Random
  - Focused
- Under Sampling
  - Random
  - Focused
- Cost Modifying





Imbalanced Learning

Resampling the original dataset

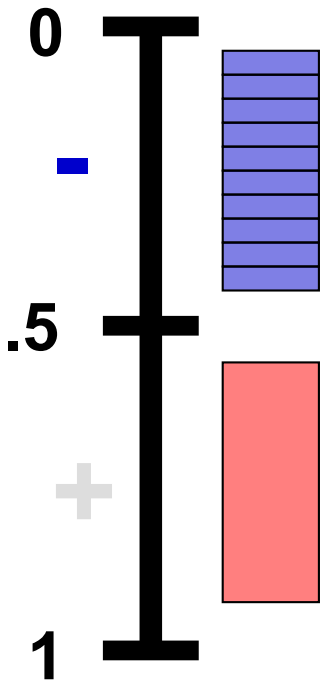
Over Sampling

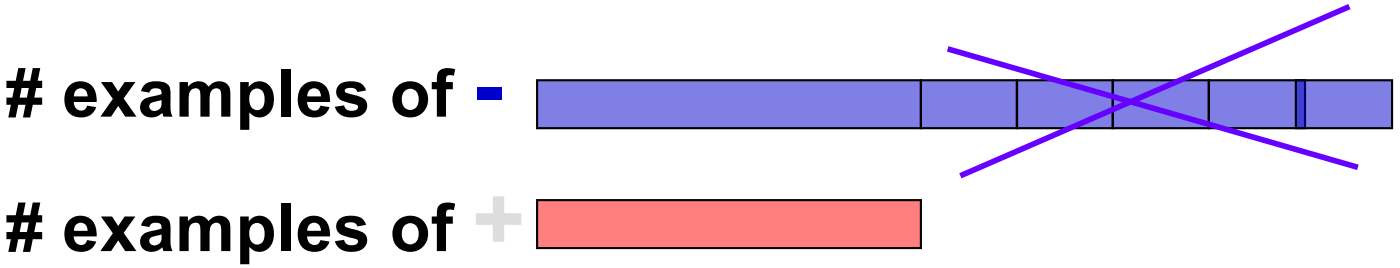
- Random
- Focused

Under Sampling

- Random
- Focused

Cost Modifying

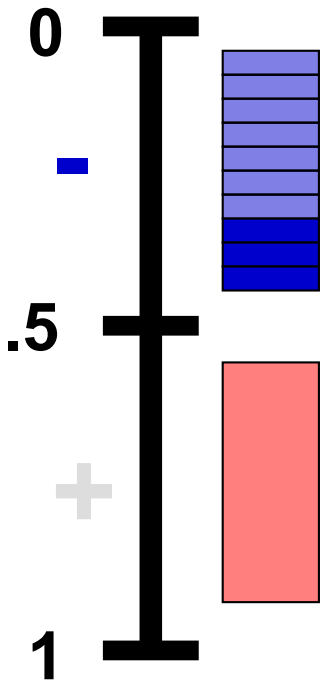


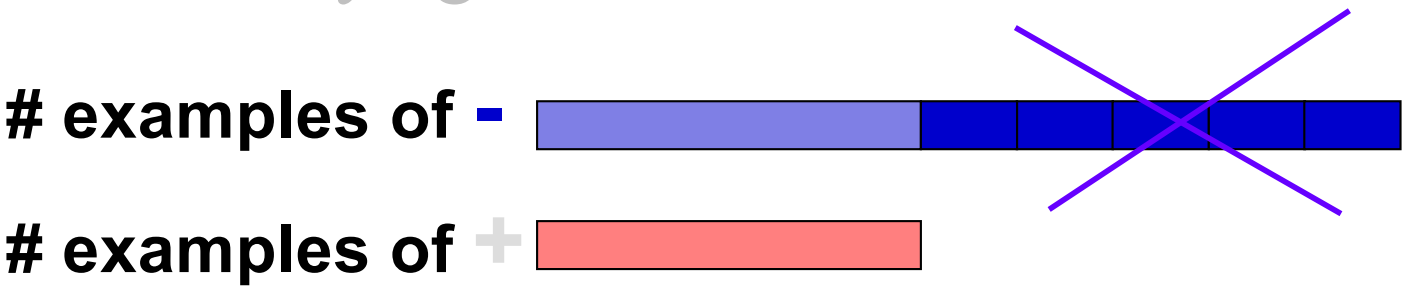


Imbalanced Learning

Resampling the original dataset

- Over Sampling
  - Random
  - Focused
- Under Sampling
  - Random
  - Focused
- Cost Modifying



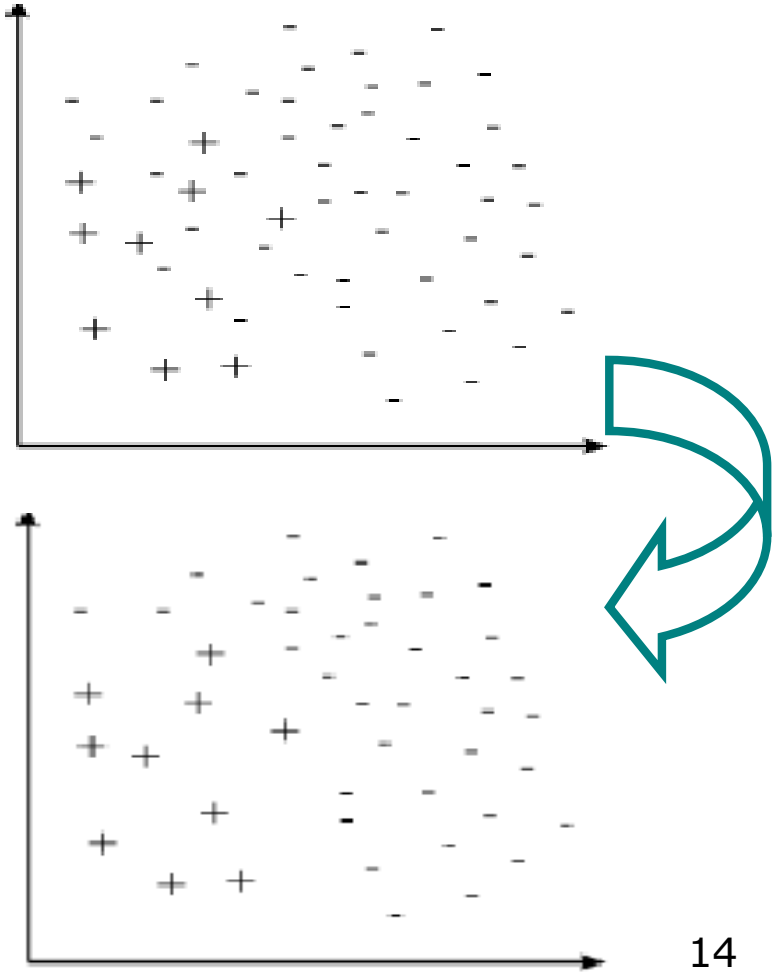


Imbalanced Learning

Classical Undersampling Algorithms

Tomek Links

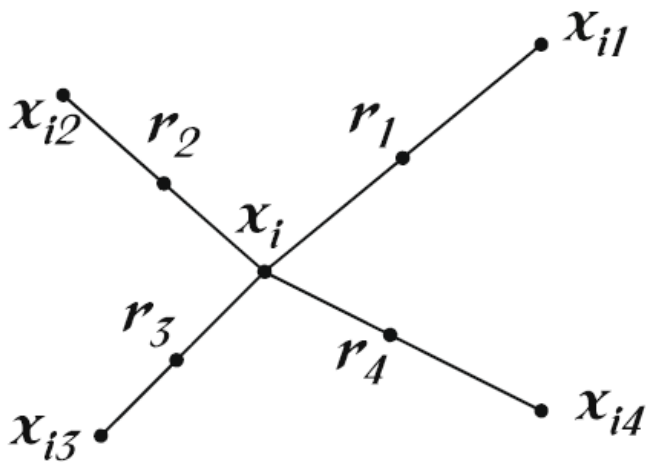
- To remove both noise and borderline examples of the majority class
- Tomek link
  - Ei, Ej belong to different classes, d (Ei, Ej) is the distance between them.
  - A (Ei, Ej) pair is called a Tomek link if there is no example El, such that d(Ei, El) < d(Ei, Ej) or d(Ej , El) < d(Ei, Ej).



Imbalanced Learning

Classical Oversampling Algorithms: SMOTE

- Synthetic Minority Over-sampling TEchnique and SMOTE related approaches:
  - **Generation** of new minority class examples
  - Interpolation among several minority class instances that lie together



For each minority sample

- **Find its k-nearest minority neighbours**
- **Randomly select j neighbours**
- **Randomly generate synthetic samples along the lines joining the minority sample selected and its j neighbours**

(j depends on the amount of oversampling desired)

N.V. Chawla, K.W. Bowyer, L.O. Hall, W.P. Kegelmeyer. SMOTE: synthetic minority over-sampling technique. Journal of Artificial Intelligence Research 16 (2002) 321-357

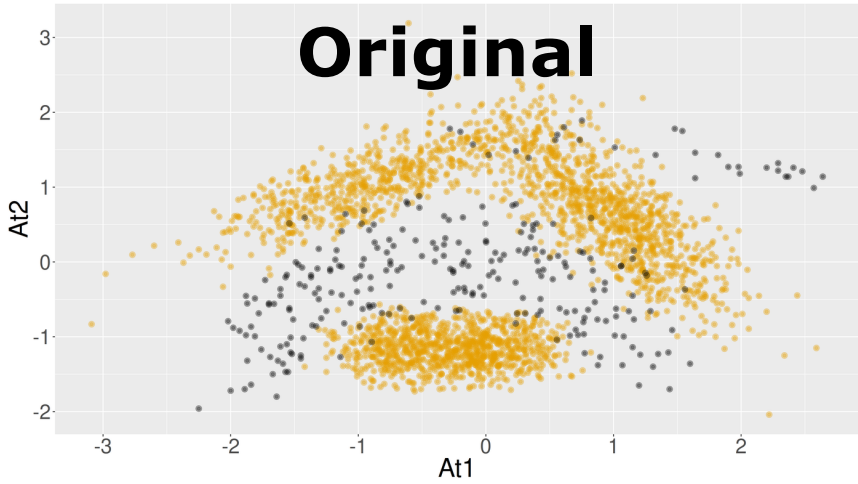
Imbalanced Learning

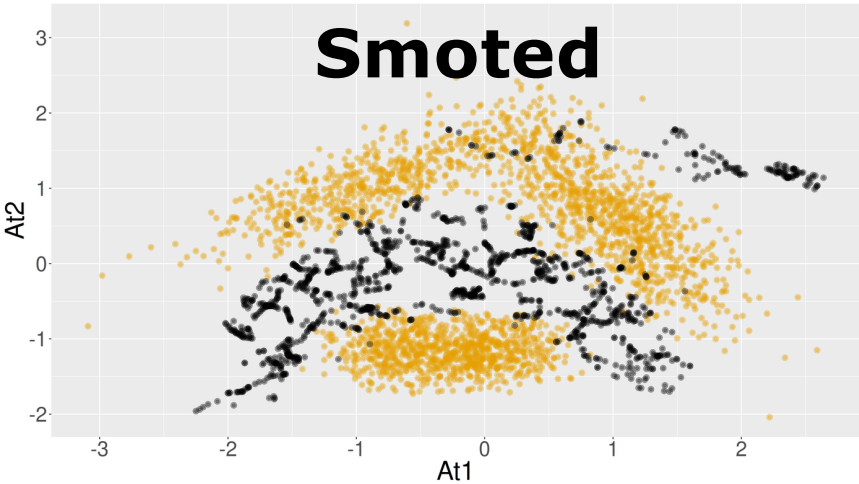
Classical Oversampling Algorithms: SMOTE

Synthetic samples are generated in the following way:

1. Take the difference between the feature vector (sample) under consideration and its nearest neighbor.
2. Multiply this difference by a random number between 0 and 1
3. Add it to the feature vector under consideration.

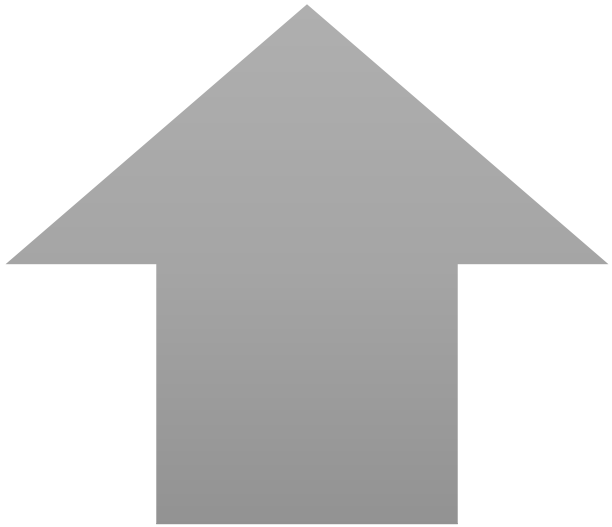
Consider a sample (6,4) and let (4,3) be its nearest neighbor.
(6,4) is the sample for which k-nearest neighbors are being identified
(4,3) is one of its k-nearest neighbors.
Let:
f1_1 = 6 f2_1 = 4 f2_1 - f1_1 = -2
f1_2 = 4 f2_2 = 3 f2_2 - f1_2 = -1
The new samples will be generated as
(f1',f2') = (6,4) + rand(0-1) * (-2,-1)
rand(0-1) generates a random number between 0 and 1.





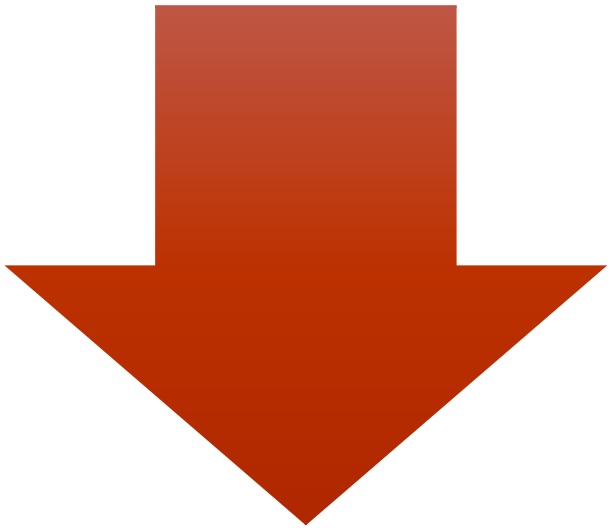
Imbalanced Learning

SMOTE Shortcomings



Overgeneralization

- SMOTE's is inherently dangerous since it **blindly generalizes** the minority area disregard majority class.
- This strategy is particularly problematic in the case of highly skewed class distributions:
  - Minority class is very sparse w.r.t. the majority class.
  - Results in a greater chance of class mixture.

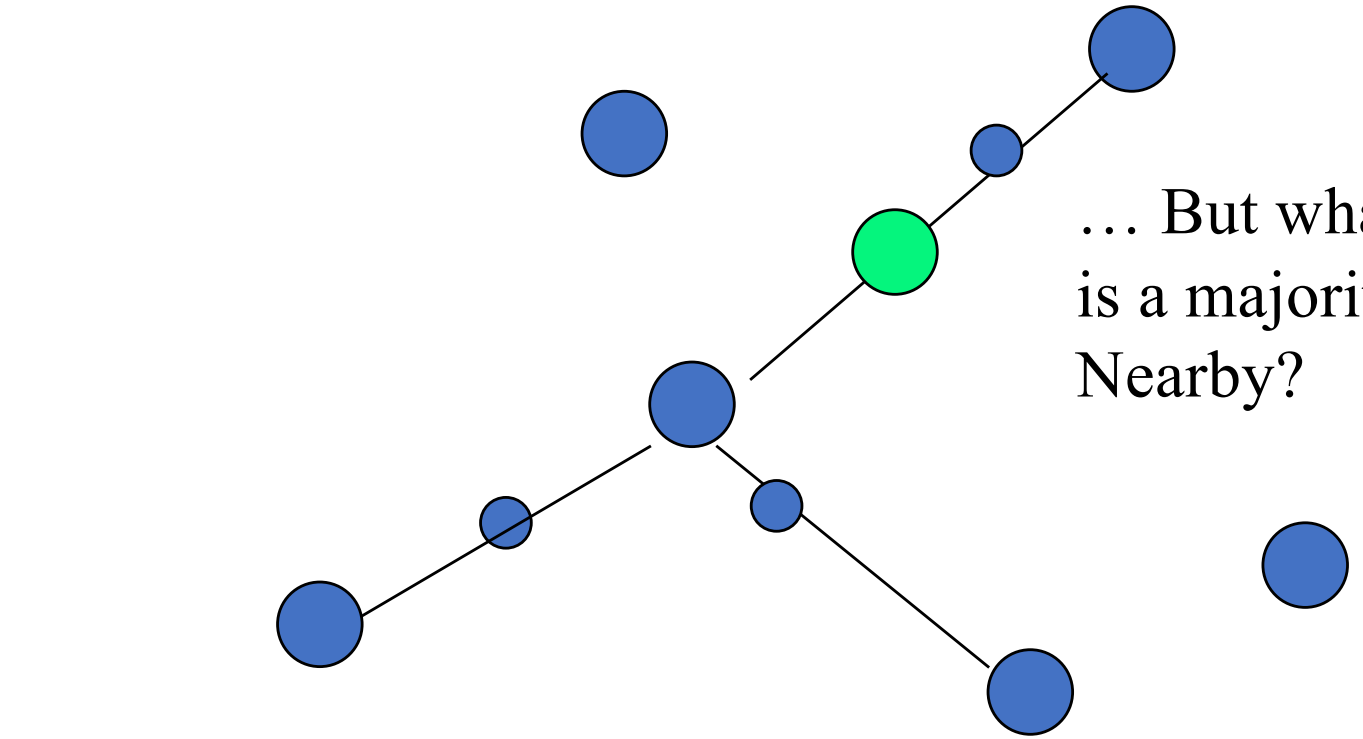


Lack of Flexibility

- The number of synthetic samples generated by SMOTE is fixed in advance,
- This does not allow for any flexibility in the re-balancing rate.

Imbalanced Learning

SMOTE Shortcomings



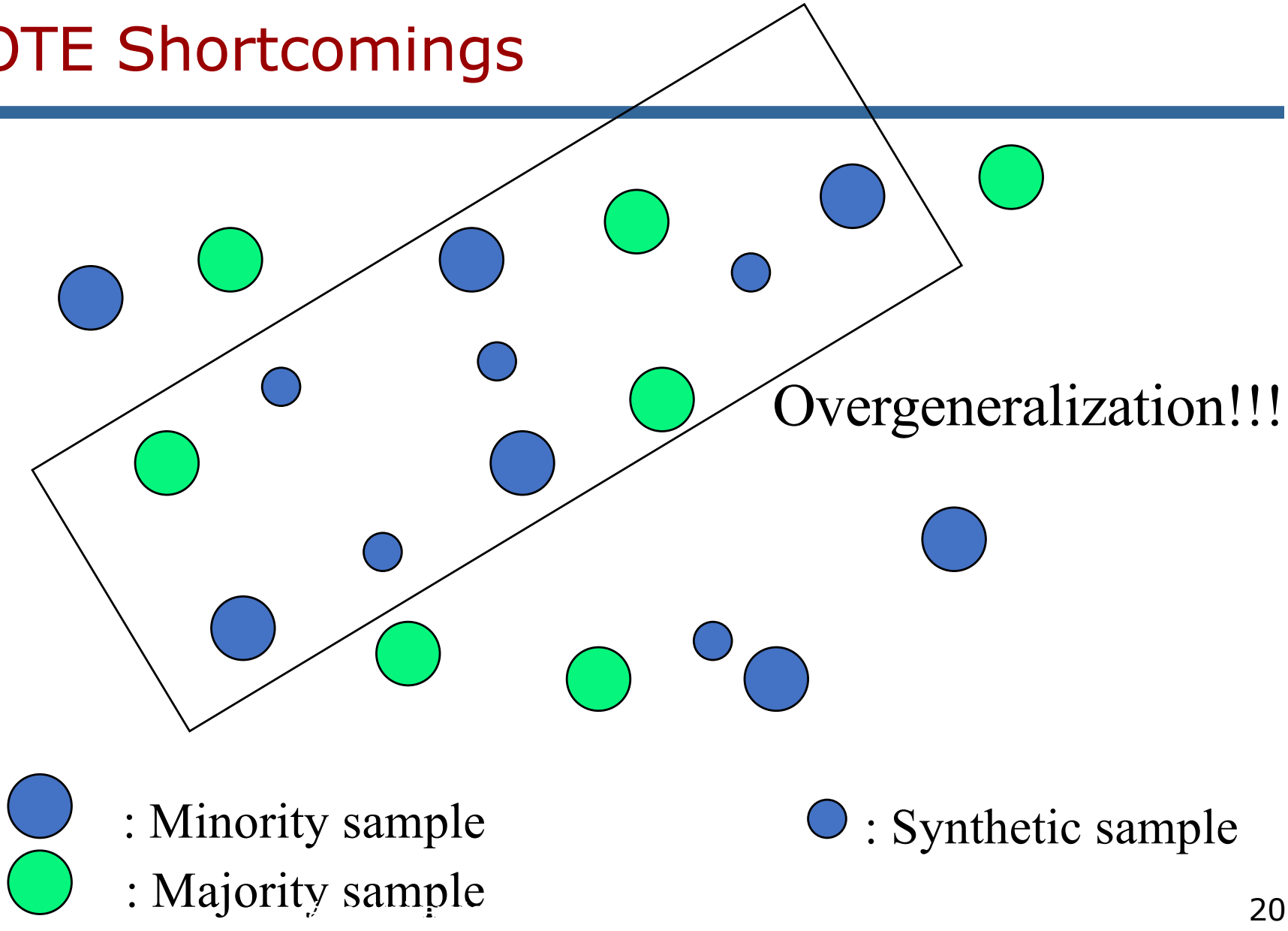
: Minority sample

: Synthetic sample

: Majority sample

Imbalanced Learning

SMOTE Shortcomings



Imbalanced Learning

Some SMOTE Extensions

Safe_Level_SMOTE:	•C. Bunkhumpornpat, K. Sinapiromsaran, C. Lursinsap. Safe-level-SMOTE: Safe-level-synthetic minority over-sampling technique for handling the class imbalanced problem. Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD-09). LNAI 5476, Springer-Verlag 2005, Bangkok (Thailand, 2009) 475-482
Borderline_SMOTE:	•H. Han, W.Y. Wang, B.H. Mao. Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning. International Conference on Intelligent Computing (ICIC'05). Lecture Notes in Computer Science 3644, Springer-Verlag 2005, Hefei (China, 2005) 878-887
SMOTE-RSB:	•E. Ramentol, Y. Caballero, R. Bello, F. Herrera, SMOTE-RSB*: A Hybrid Preprocessing Approach based on Oversampling and Undersampling for High Imbalanced Data-Sets using SMOTE and Rough Sets Theory. *Knowledge and Information Systems 33:2 (2012) 245-265.*
SMOTE-IPF:	•Sáez, J.A., Luengo, J., Stefanowski, J., Herrera, F.: SMOTE-IPF: addressing the noisy and borderline examples problem in imbalanced classification by a re-sampling method with filtering. *Information Sciences* 291, 184–203 (2015)

Imbalanced Learning

Cost-sensitive learning

- Weighting errors made on minority class examples higher than those of the majority class in computing training error:
  - C(+, -) > C(-, +)
  - C(+, +) = C(-, -) = 0
- Needs a cost matrix, which encodes misclassification penalty.
- Consider the cost-matrix throughout the building of the model for achieving the lowest cost.
- However, the cost matrix is often unavalaible

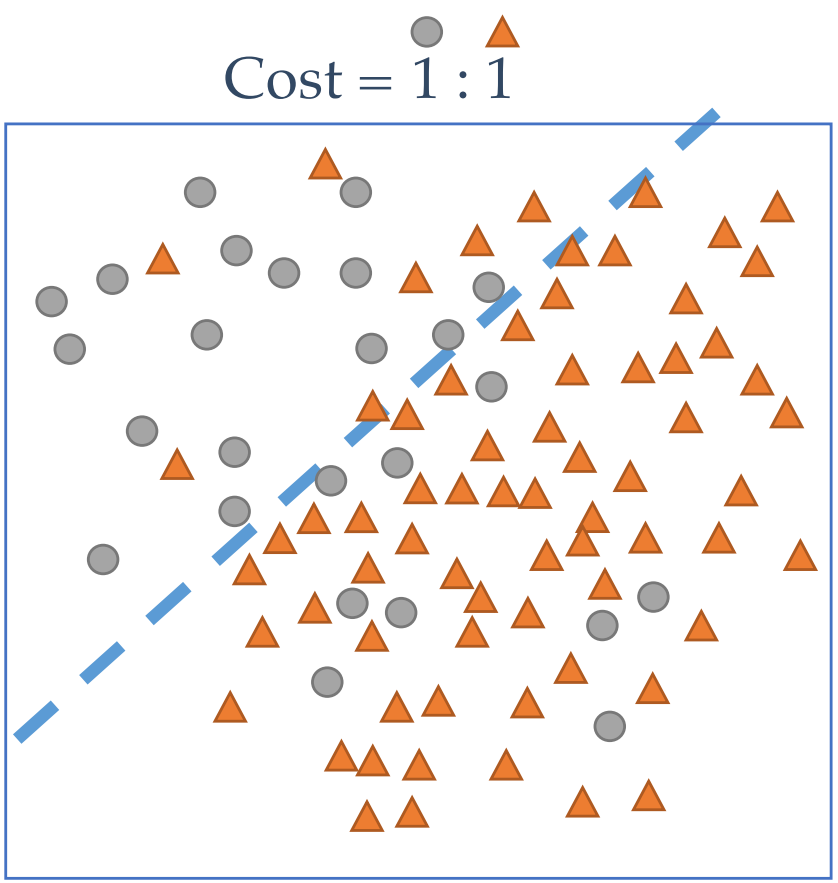
	actual negative	actual positive
predict negative	$C(0,0) = c_{00}$	$C(0,1) = c_{01}$
predict positive	$C(1,0) = c_{10}$	$C(1,1) = c_{11}$

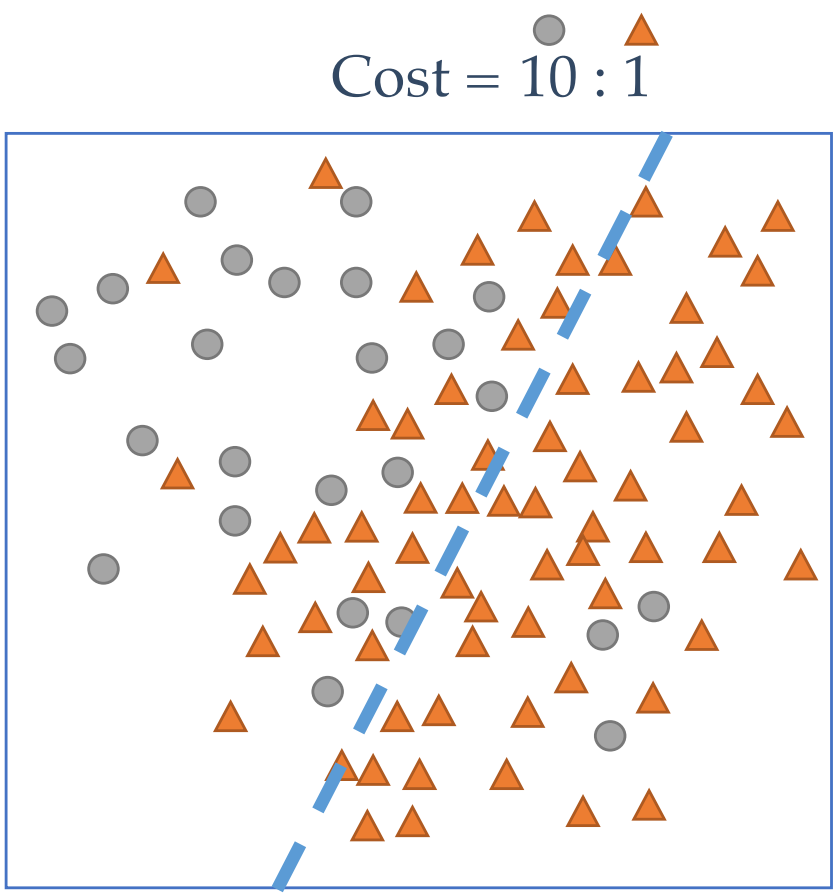
	fraudulent	legitimate
refuse	\$20	−\$20
approve	$-x$	$0.02x$

C. Elkan, The foundations of cost-sensitive learning, in: Proceedings of the 17th International Joint Conference on Artificial Intelligence, 2001, pp. 973–978

Imbalanced Learning

Cost-sensitive learning





Imbalanced Learning

How to obtain cost-matrix

- **Provided by an expert.**
  - Supplied data is accompanied by the cost matrix that comes directly from the nature of a problem.
  - This usually requires an access to a domain expert that can assess the most realistic cost values, i.e. credit card fraud detection
- **Estimated using training data**.
  - No a priori information on cost matrix is available during classifier training.
  - This requires either heuristic setting of cost values or learning them from training data:
    - IR for cost estimation
    - Thresholding via validation set

Imbalanced Learning

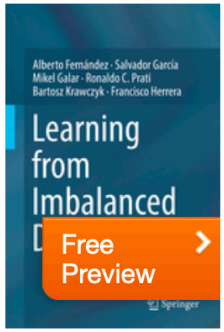
Python libraries: imbalanced-learn

- In spite of the number of different libraries developed for Python, it was not until 2017 when the first solution for the task of imbalanced classification was released.

Preprocessing	*Technique*
Under-Sampling	Random majority under-sampling with replacement Extraction of majority-minority Tomek links Under-sampling with Cluster Centroids NearMiss-(1 & 2 & 3) Condensend Nearest Neighbour One-Sided Selection Neighboorhood Cleaning Rule Edited Nearest Neighbours Instance Hardness Threshold Repeated Edited Nearest Neighbours AllKNN
Over-Sampling	Random majority over-sampling with replacement SMOTE - Synthetic Minority Over-sampling Technique bSMOTE(1 & 2) - Borderline SMOTE of types 1 and 2 SVM SMOTE - Support Vectors SMOTE ADASYN - Adaptive synthetic sampling approach for imbalanced learning
Hybrid sampling	SMOTE + TomekLinks SMOTE + ENN
Ensemble sampling	EasyEnsemble BalanceCascade

Imbalanced Learning

The Imbalanced Learning Book



© 2018

Learning from Imbalanced Data Sets

Authors: **Fernández Hilario**, A., **García López**, S., **Galar**, M., **Prati**, R.C., **Krawczyk**, B., **Herrera**, F.

Offers a comprehensive review of imbalanced learning widely used worldwide in many real applications, such as fraud detection, disease diagnosis, etc

» see more benefits

About this book

This book provides a general and comprehensible overview of imbalanced learning. It contains a formal description of a problem, and focuses on its main features, and the most relevant proposed solutions. Additionally, it considers the different scenarios in Data Science for which the imbalanced classification can create a real challenge.

This book stresses the gap with standard classification tasks by reviewing the case studies and ad-

» Show all

Buy this book

eBook 96,29 €

price for Spain (gross)

Buy eBook

- ISBN 978-3-319-98074-4
- Digitally watermarked, DRM-free
- Included format: PDF, EPUB
- Immediate eBook download after purchase and usable on all devices
- Bulk discounts available

Hardcover 124,79 €

Softcover 124,79 €

VISA MasterCard AMERICAN EXPRESS PayPal INVOICE

» FAQ » Policy

Services for this Book

» Download Product Flyer

» Download High-Resolution Cover

Time Series Forecasting

Forecasting

- **Forecasting**: Predicting the future as accurately as possible, given all the information available including historical data and knowledge of any future events that might impact the forecasts
- It is usually, an integral part of decision-making.
- Example: Forecast of electricity demand

- **Can be applied when:**
  - **Numerical data about the past is available**
  - **It is reasonable to assume that some aspects of the past patterns will continue into the future**

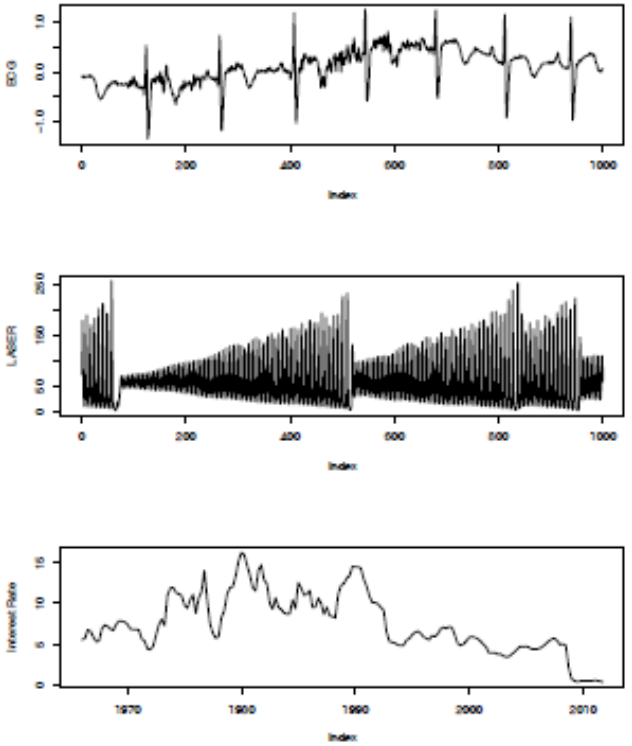
Time Series Forecasting

Time series

- Anything that is observed over time is a time series
- Time series observed at regular intervals of time (every minute, hourly, daily, weekly, …)

$$\{X_{t_1}, X_{t_2}, X_{t_3}, \ldots . X_{t_n}\}$$

- Time series forecasting intends to estimate how the sequence of observations will continue in the future

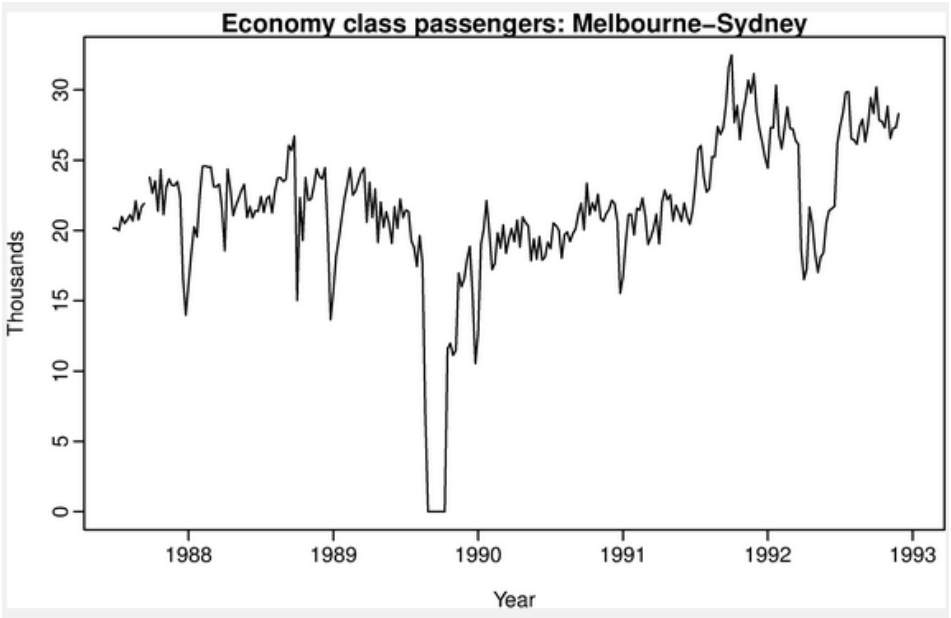


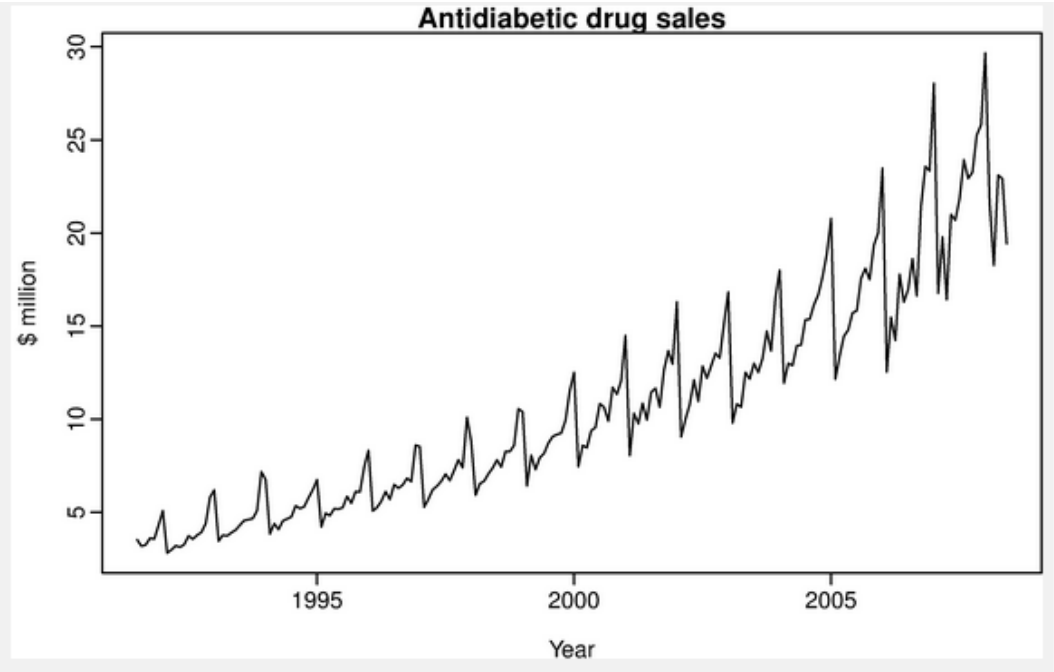
Time Frame (How far can we predict?)

- **short-term (1 - 2 periods)**
- **medium-term (5 - 10 periods)**
- **long-term (12+ periods)**

Time Series Forecasting

Time plots & References



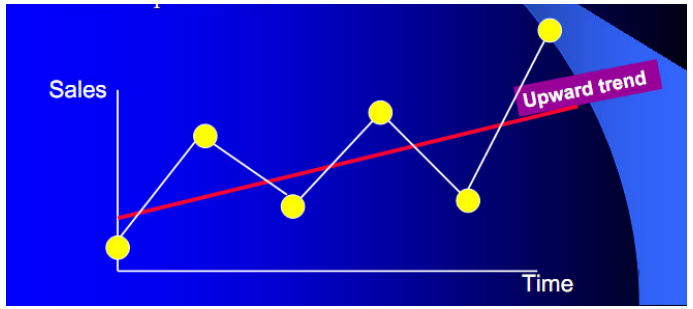


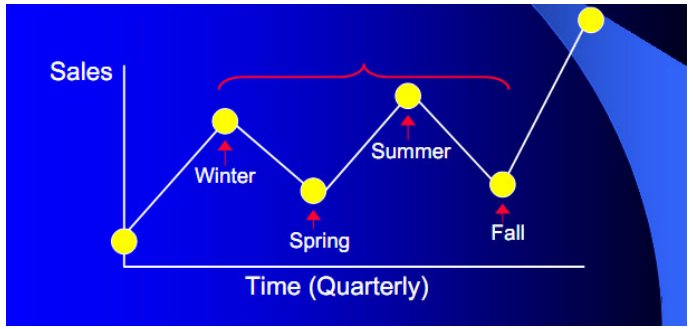
- C. Chatfield, «The analysis of time series: An Introduction», Chapman & Hall/CRC, 2003
- J.D. Hamilton, «Time Series Analysis», Princeton University Press, 1994
- R. Hyndman, G. Athanasopoulus, «Forecasting and time series» 2013

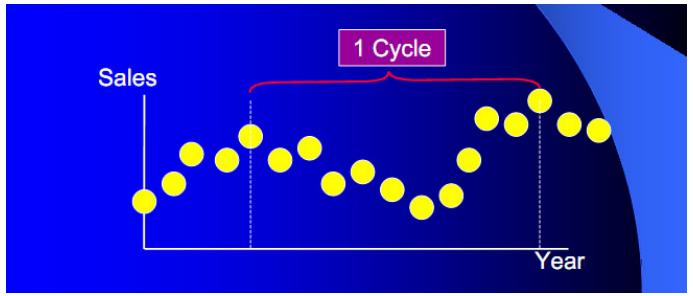
Time Series Forecasting

Time Series patterns

- *Trend*: long-term increase or decrease in the data
- *Seasonal pattern*: data affected by seasonal factors such as time of the year or day of the week
- *Cycle*: data exhibits rises and falls that are not of a fixed period; variable and unknown length







Time Series Forecasting

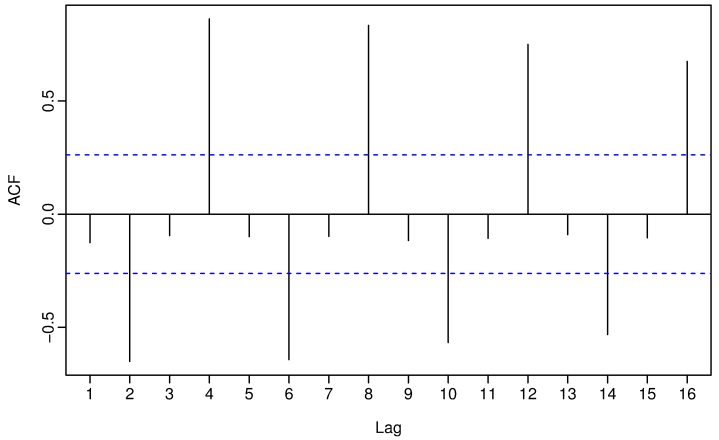
Autocorrelation

Relationship between lagged values of a time series

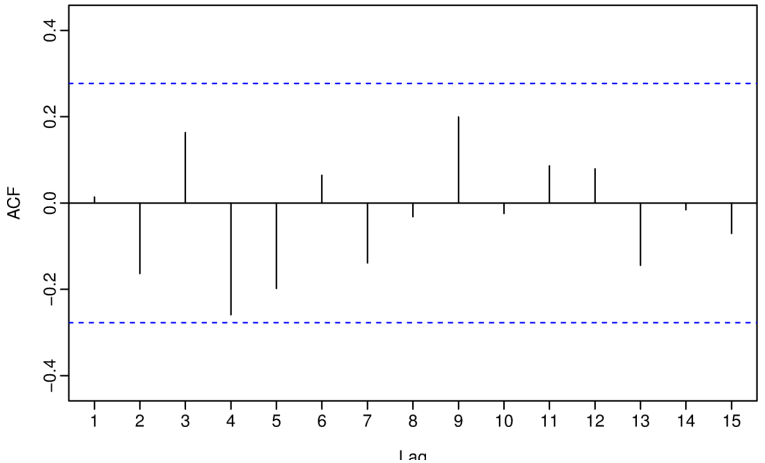
$$r_k = \frac{\sum_{t=k+1}^{T} (y_t - \bar{y})(y_{t-k} - \bar{y})}{\sum_{t=1}^{T} (y_t - \bar{y})^2},$$

Autocorrelation Function (ACF)

r_1	r_2	r_3	r_4	r_5	r_6	r_7	r_8	r_9
-0.126	-0.650	-0.094	0.863	-0.099	-0.642	-0.098	0.834	-0.116



Series showing NO autocorrelation



Time Series Forecasting

Evaluating forecast accuracy

- Forecast error: $e_i = y_i - \hat{y}_i$
- Scale-dependent errors

$$\text{MAE} = \text{mean}(|e_i|), \qquad \text{RMSE} = \sqrt{\text{mean}(e_i^2)}.$$

- Percentage error: $p_i = 100 e_i / y_i$
- Scaled errors

$$\text{MAPE} = \text{mean}(|p_i|). \qquad \text{sMAPE} = \text{mean}\left(200|y_i - \hat{y}_i|/(y_i + \hat{y}_i)\right)$$

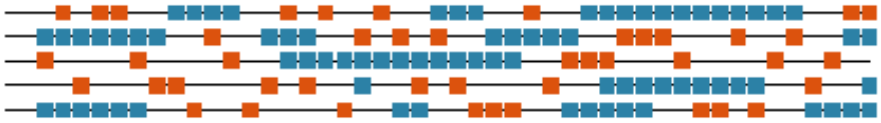
$$q_j = \frac{e_j}{\frac{1}{T-1}\sum_{t=2}^{T}|y_t - y_{t-1}|}. \qquad \text{MASE} = \text{mean}(|q_j|).$$

Time Series Forecasting

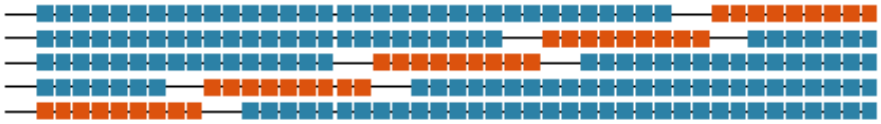
Validation

- As in any other modeling task it is essentical to conduct a right evaluation
- Data should be split into training and test parts
- Improved through Cross-validation
- Even further improved through Blocked Cross-Validation

cross-validation



non-dep. cross-validation



blocked cross-validation

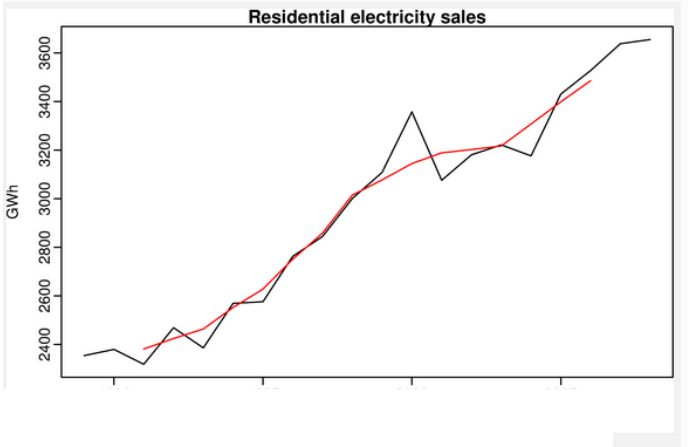
last block

Time Series Forecasting

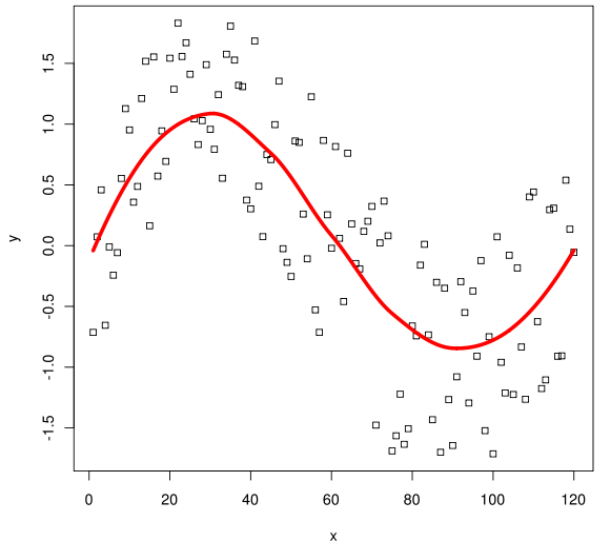
Time Series Decomposition

- Moving averages

$$\hat{T}_t = \frac{1}{m} \sum_{j=-k}^{k} y_{t+j}$$



- STL (Seasonal and Trend decomposition using Locally weighted smoothing) is a robust and versatil decomposition method



- To forecast a decomposed time series, we forecast individual components, and then compute the predicted value

Time Series Forecasting

Algorithms

- ARIMA models

$$\hat{Y}_{t+1} = f\left(Y_t, Y_{t-1}, Y_{t-2}, \ldots\right)$$

  - Stationarity
  - Differencing
  - Random walk model: A time series built by adding the error term to each new value
  - Autoregressive models
  - Moving average models
- Non-seasonal ARIMA(p,d,q)

$$y'_t = c + \phi_1 y'_{t-1} + \cdots + \phi_p y'_{t-p} + \theta_1 e_{t-1} + \cdots + \theta_q e_{t-q} + e_t$$

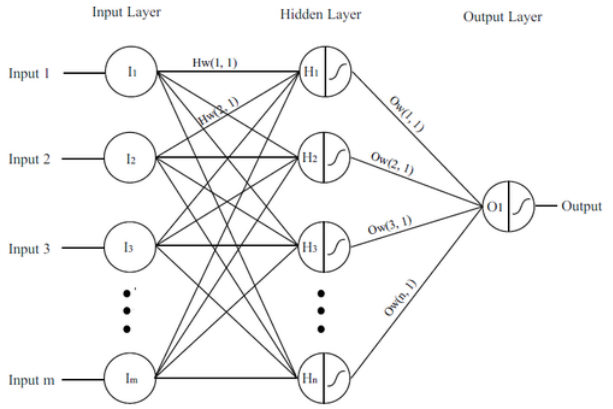
  - p: order of the autoregressive part
  - d: degree of the first differencing part
  - q: order of the moving average part

White noise	ARIMA(0,0,0)
Random walk	ARIMA(0,1,0) with no constant
Random walk with drift	ARIMA(0,1,0) with a constant
Autoregression	ARIMA(p,0,0)
Moving average	ARIMA(0,0,q)

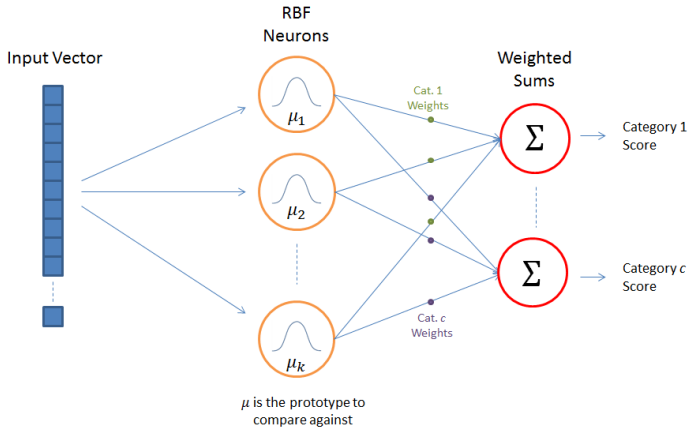
Time Series Forecasting

Neural Networks

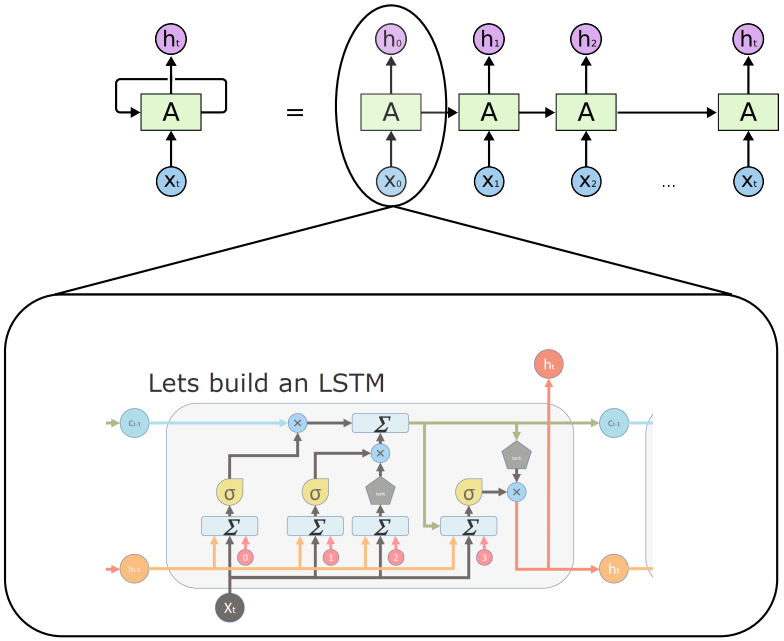
- Multilayered perceptrons



- RBF



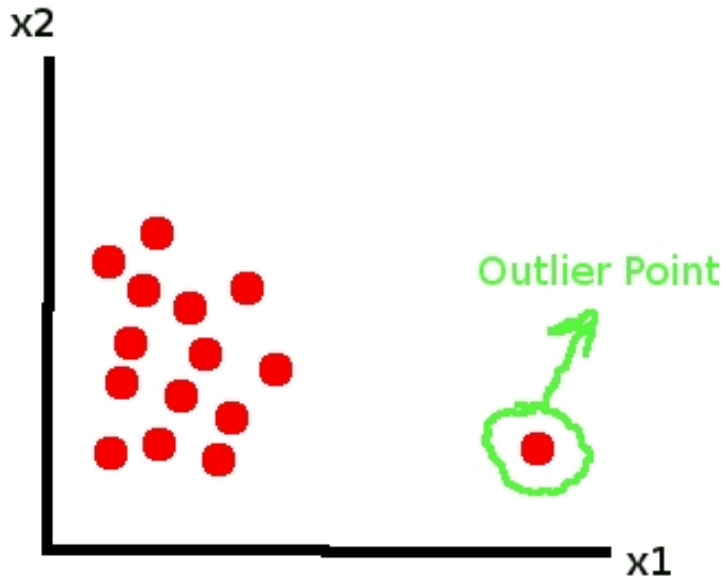
- Recurrent neural networks (LSTMs)



Anomaly Detection

What are anomalies?

- Anomaly is a pattern in the data that does not conform to the expected behavior
- Also referred to as outliers, exceptions, peculiarities, surprise, etc.
- Anomalies translate to significant (often critical) real life entities
  - Cyber intrusions
  - Credit card fraud
  - Faults in a System



Anomaly Detection

What are anomalies?

Key Challenges

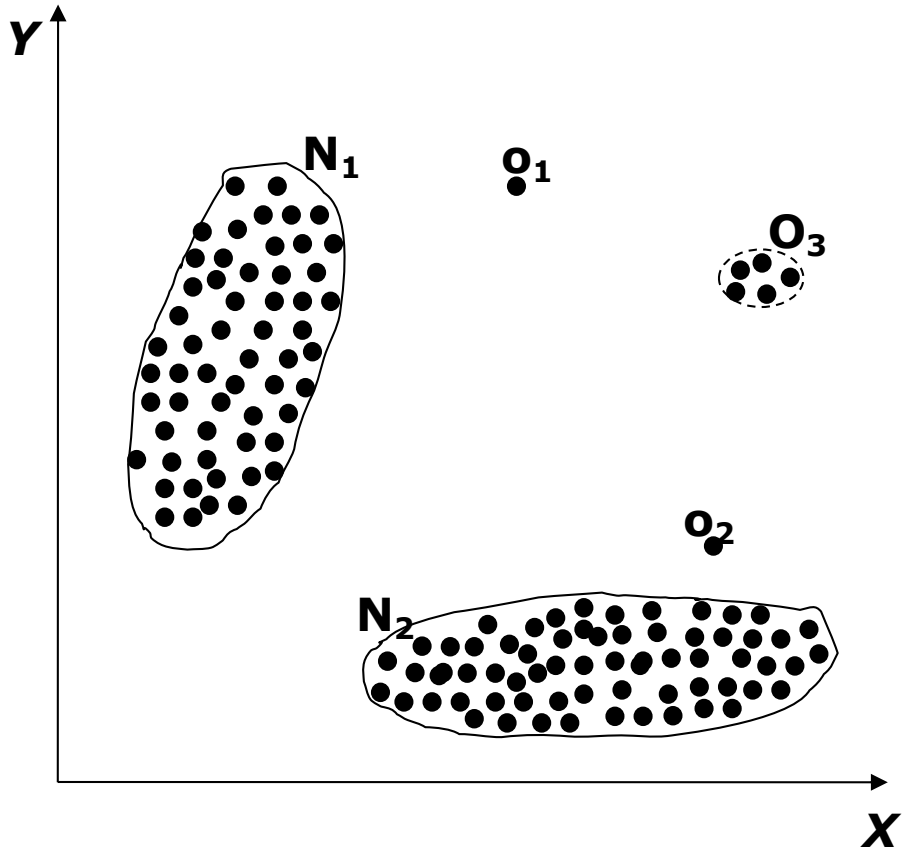
- Defining a representative normal region is challenging
- The boundary between normal and outlying behavior is often not precise
- The exact notion of an outlier is different for different application domains
- Availability of labeled data for training/validation
- Malicious adversaries
- Data might contain noise
- Normal behavior keeps evolving

Anomaly Detection

What are anomalies?

Point Anomalies

- An individual data instance is anomalous w.r.t. the data

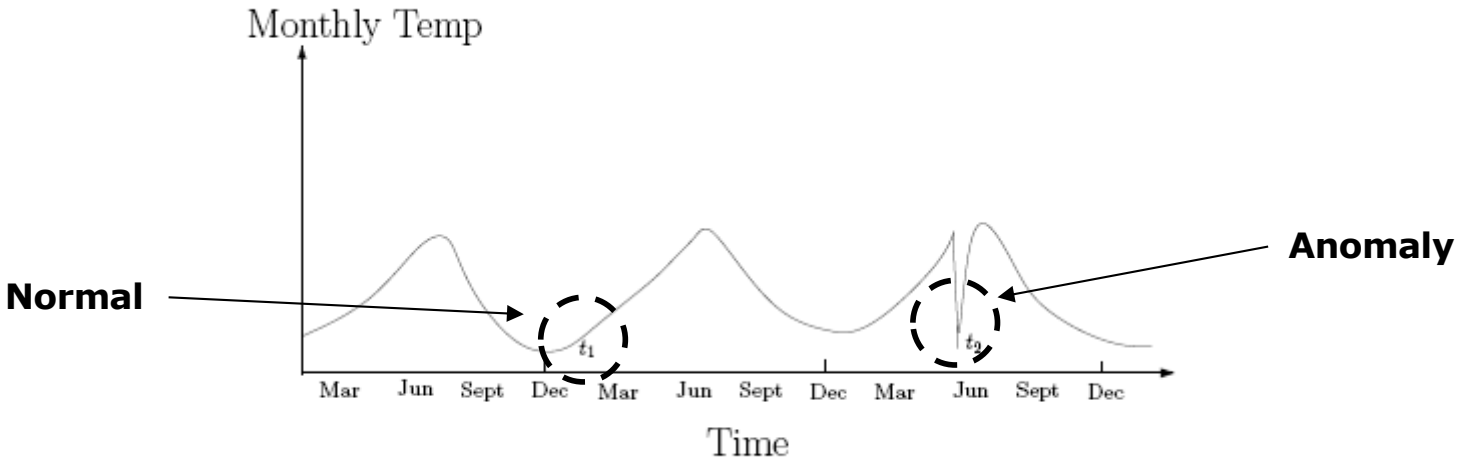


Anomaly Detection

What are anomalies?

Contextual Anomalies

- An individual data instance is anomalous within a context
- Requires a notion of context
- Also referred to as conditional anomalies*



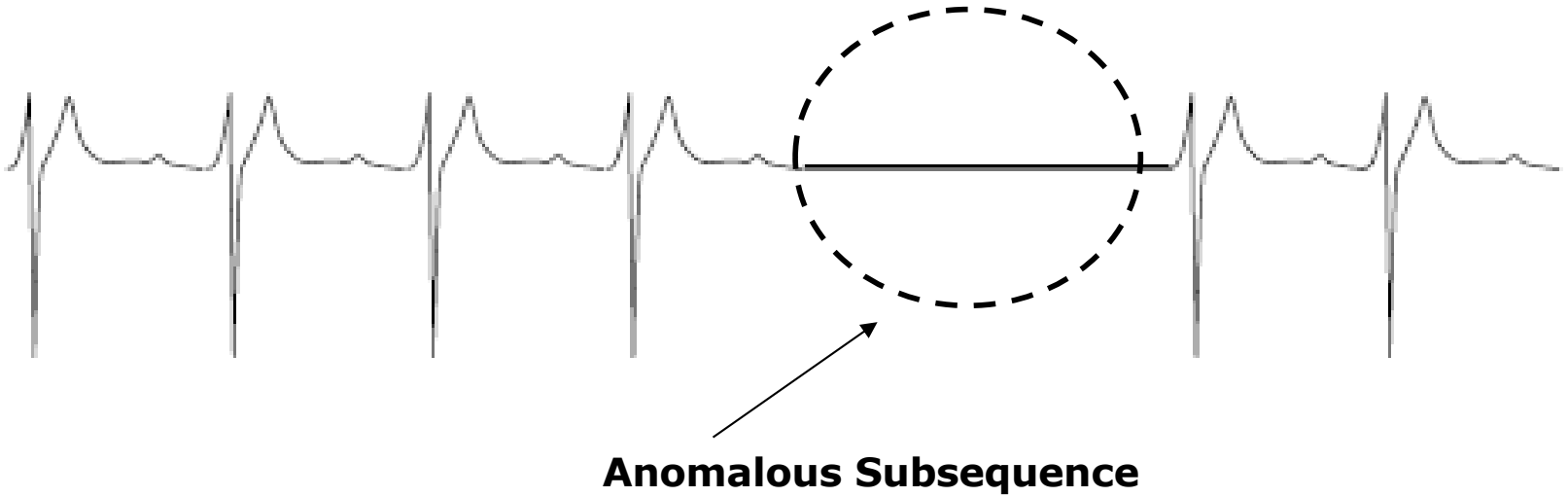
* Xiuyao Song, Mingxi Wu, Christopher Jermaine, Sanjay Ranka, Conditional Anomaly Detection, IEEE Transactions on Data and Knowledge Engineering, 2006.

Anomaly Detection

What are anomalies?

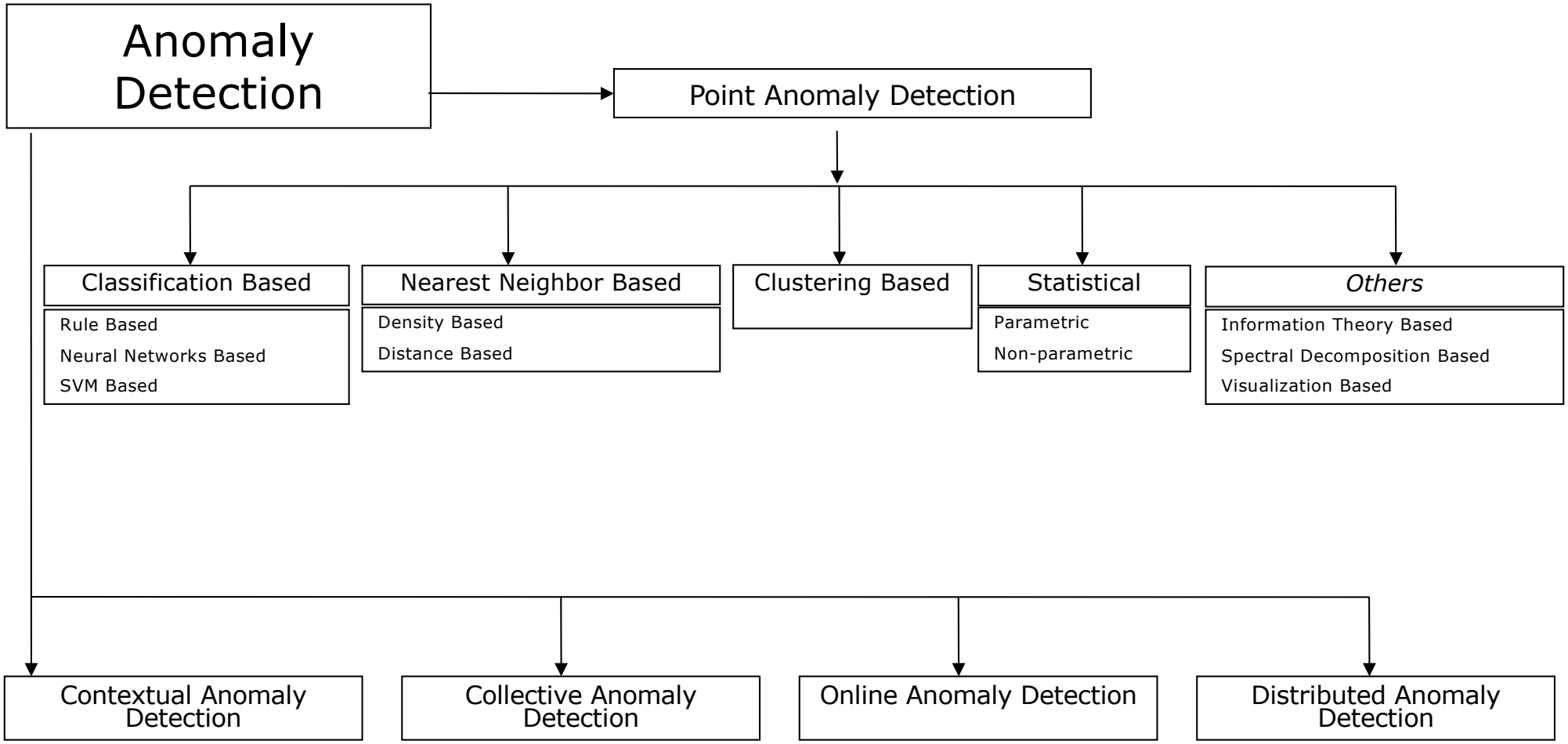
Collective Anomalies

- A collection of related data instances is anomalous
- Requires a relationship among data instances
  - Sequential Data
  - Spatial Data
  - Graph Data
- The individual instances within a collective anomaly are not anomalous by themselves



Anomaly Detection

Taxonomy



* Anomaly Detection – A Survey, Varun Chandola, Arindam Banerjee, and Vipin Kumar, ACM Computing Surveys, Vol. 41, No. 3, Article 15, Publication date: July 2009.

Anomaly Detection

Classification based Techniques

Classification Based Techniques

- **Main idea:** build a classification model for normal (and anomalous (rare)) events based on labeled training data, and use it to classify each new unseen event
- Classification models must be able to handle skewed (imbalanced) class distributions
- Categories:
  - *Supervised classification techniques*
    - Require knowledge of both ***normal*** and ***anomaly*** class
    - Build classifier to distinguish between normal and known anomalies
  - *Semi-supervised classification techniques*
    - Require knowledge of ***normal*** class only!
    - Use modified classification model to learn the normal behavior and then detect any deviations from normal behavior as anomalous

Anomaly Detection

Classification based Techniques

- **Advantages:**
  - ***Supervised classification techniques***
    - Models that can be easily understood
    - High accuracy in detecting many kinds of known anomalies
  - ***Semi-supervised classification techniques (One-class)***
    - Models that can be easily understood
    - Normal behavior can be accurately learned
- **Drawbacks:**
  - ***Supervised classification techniques***
    - Require both labels from both normal and anomaly class
    - Cannot detect unknown and emerging anomalies
  - ***Semi-supervised classification techniques (One-class)***
    - Require labels from normal class
    - Possible high false alarm rate - previously unseen (yet legitimate) data records may be recognized as anomalies

Anomaly Detection

Nearest Neighbour based Techniques

- ***Key assumption*: normal points have close neighbors while anomalies are located far from other points**
- General two-step approach
  1. Compute neighborhood for each data record
  2. Analyze the neighborhood to determine whether data record is anomaly or not
- **Categories:**
  - Distance based methods
    - Anomalies are data points most distant from other points
  - Density based methods
    - Anomalies are data points in low density regions

Anomaly Detection

Nearest Neighbour based Techniques

Distance based Outlier Detection

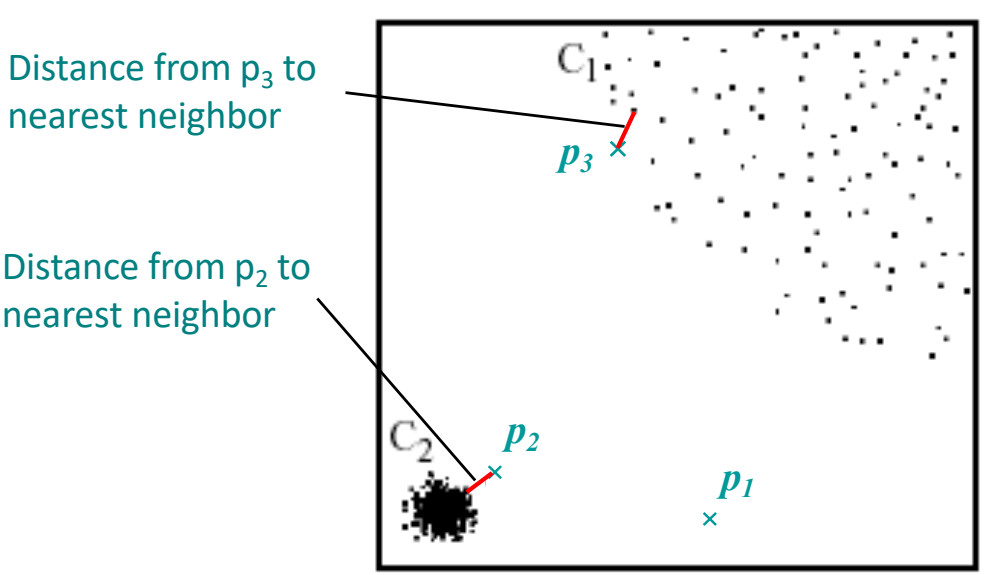
- *Nearest Neighbor (NN) approach*
  - For each data point *d* compute the distance to the *k-th* nearest neighbor d_k
  - Sort all data points according to the distance d_k
  - Outliers are points that have the largest distance d_k and therefore are located in the more sparse neighborhoods
  - Usually data points that have top *n*% distance d_k are identified as outliers
    - *n* – user parameter
  - Not suitable for datasets that have modes with varying density

Anomaly Detection

Nearest Neighbour based Techniques

Advantages of Density based Techniques

- *Local Outlier Factor (LOF) approach*
  - Example:



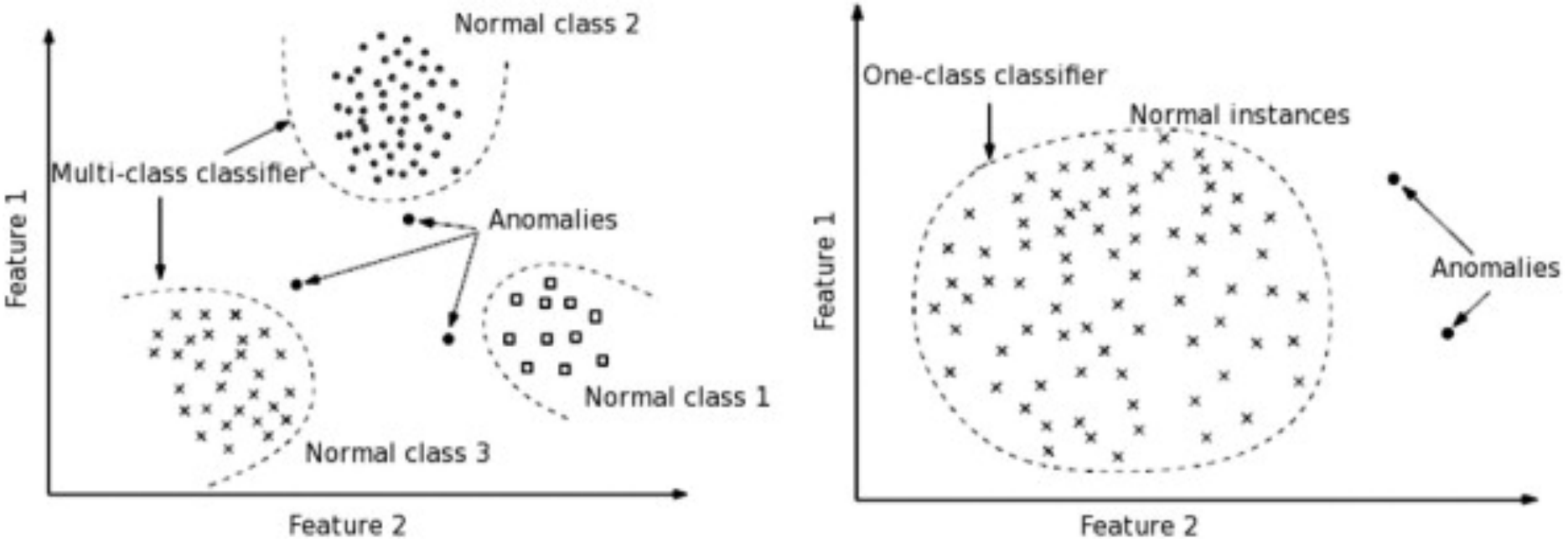
In the *NN* approach, p_2 is not considered as outlier, while the *LOF* approach find both p_1 and p_2 as outliers

NN approach may consider p_3 as outlier, but LOF approach does not

Anomaly Detection

One-class Classification

Several classes vs One-class classification

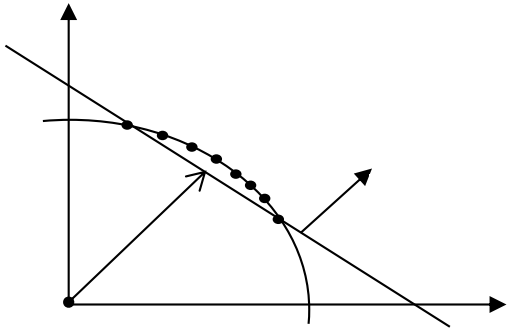


One-class 1-NN is an semi-supervised algorithm that learns a decision function for novelty detection: classifying new data as similar or different to the training set.

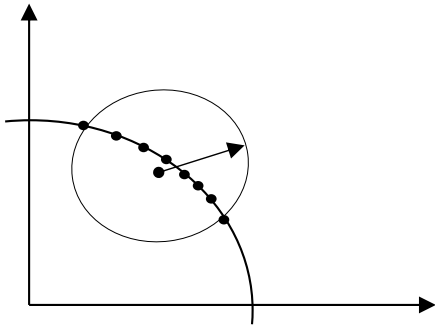
Anomaly Detection

One-class Classification

- One Class SVM
  - Find the optimal ***hyperplane*** to separate the target class from the origin with maximum margin



- Support Vector Data Description
  - Use the minimum hyperspere to enclose the target class

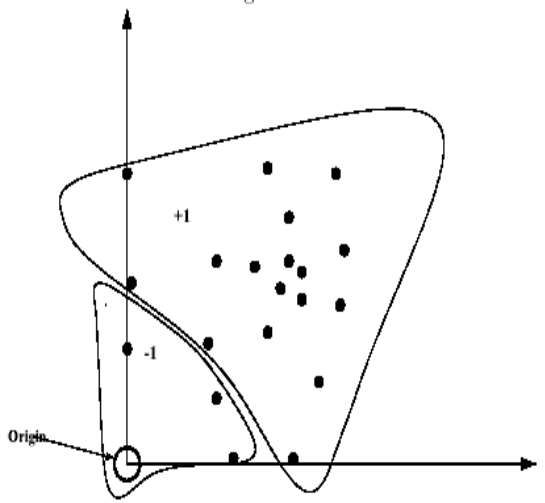


OCSVM Solves optimization problem to find rule f with maximal margin

$f(x)=\langle w,x\rangle+b$

If $f(x)<0$, label x as anomalous

Figure 1: One-class SVM

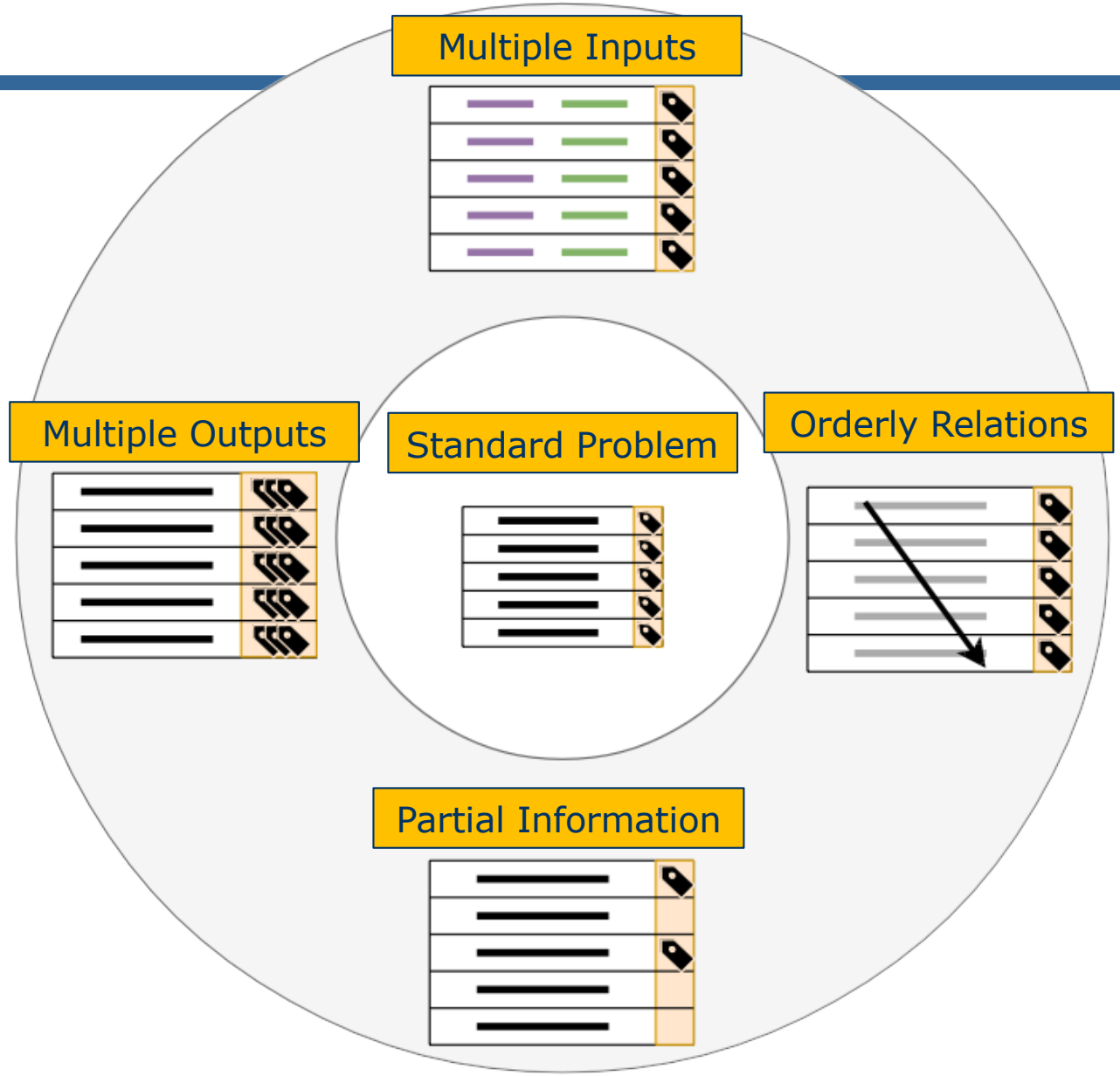


One-Class SVM Classifier. The origin is the only original member of the second class.

More singular Problems

Summary

They arise from variations in the input and output structures that do not fit the standard problem.



Multi-Instance Learning

Origin: Predicting Drug Activity

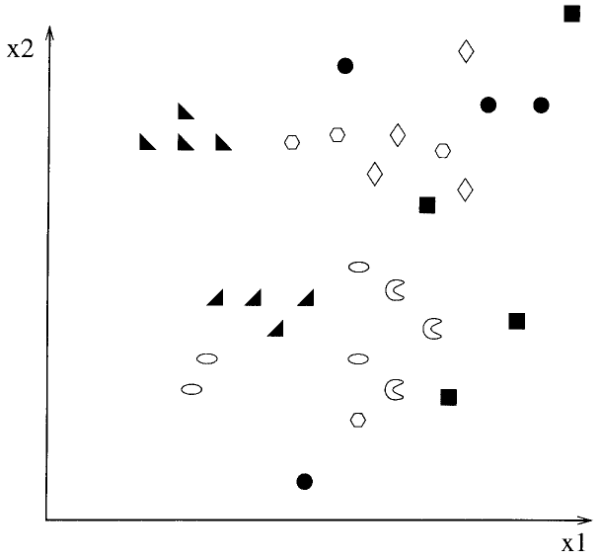
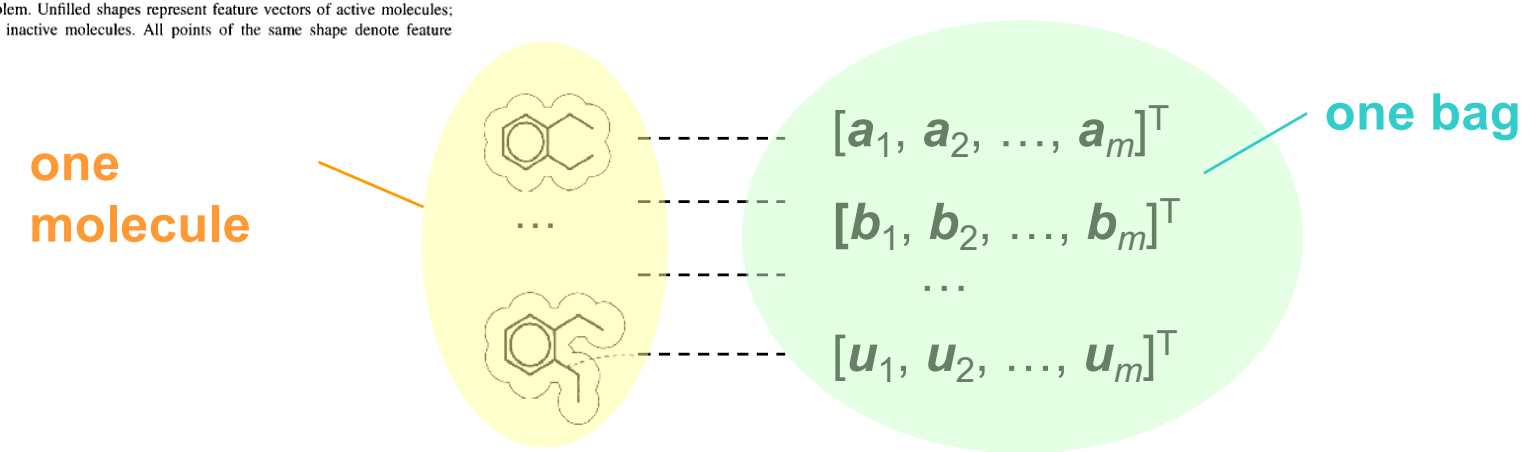


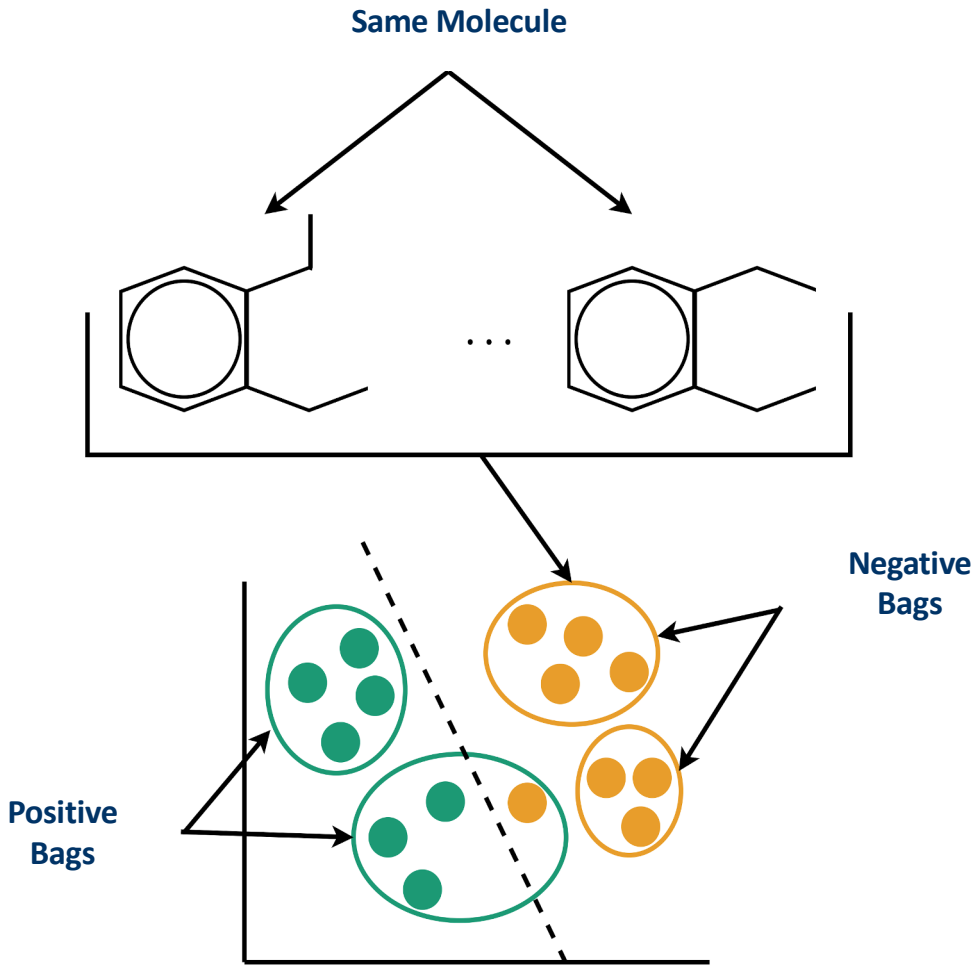
Fig. 14. A multiple instance learning problem. Unfilled shapes represent feature vectors of active molecules; filled shapes represent feature vectors of inactive molecules. All points of the same shape denote feature vectors of the same molecule.

Dietterich, T. G., Lathrop, R. H., & Lozano-Pérez, T. (1997). Solving the multiple instance problem with axis-parallel rectangles. Artificial intelligence, 89(1-2), 31-71.



Multi-Instance Learning

Definition



- Each example consists of a bag of instances
- The labels on the bags are known
- The labels of the instances are unknown
- A bag is positively labeled if at least one of its instances is positively labeled, otherwise it is negative

Multi-Instance Learning

Definition

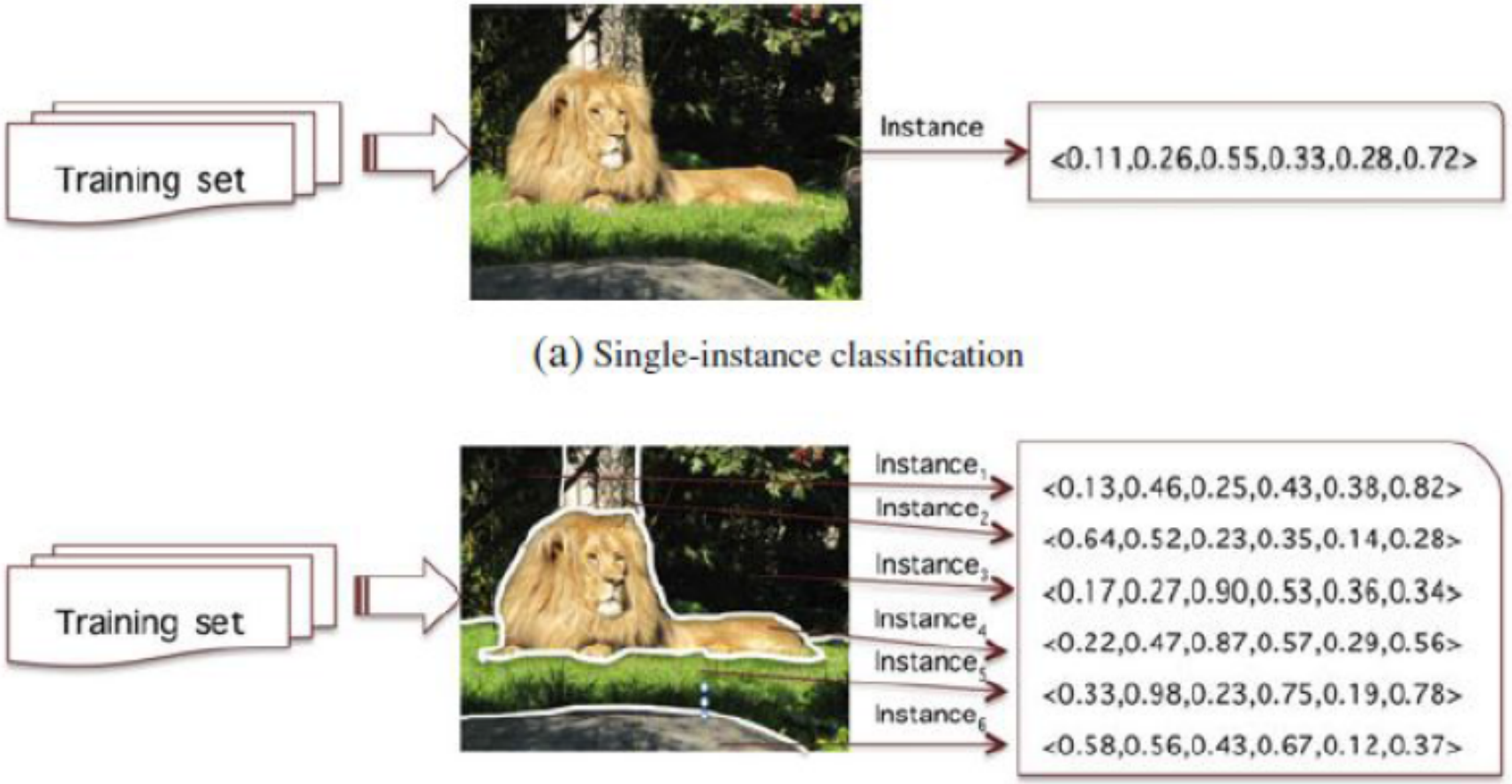


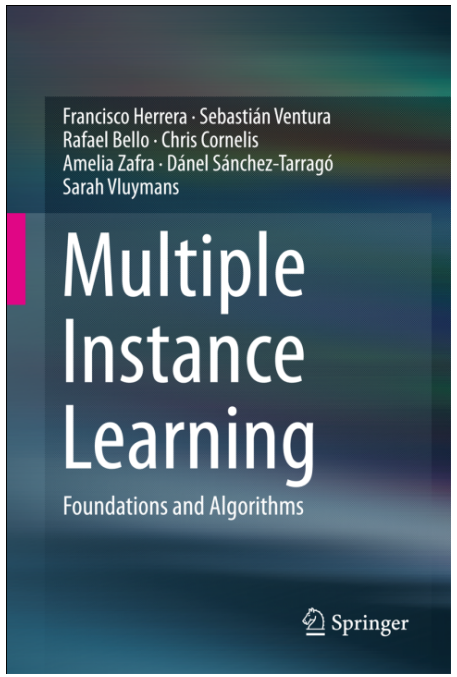
Fig. 3.2 Training data set for classification task

Multi-Instance Learning

Definition

- ✓ **Citation kNN**
- ✓ **Support Vector Machine for multi-instance learning**
- ✓ **Multiple-decision tree**
- ✓ … …

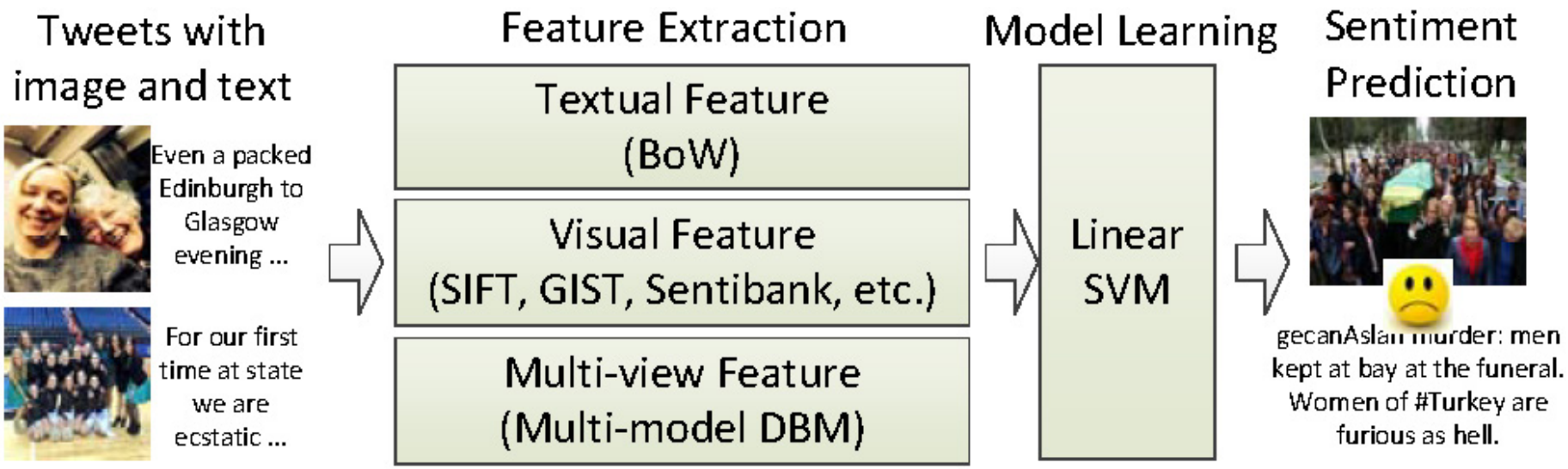
See: **http://link.springer.com/book/10.1007%2F978-3-319-47759-6**



- Drug activity prediction
- Classification of images by segments
- Bankruptcy prediction

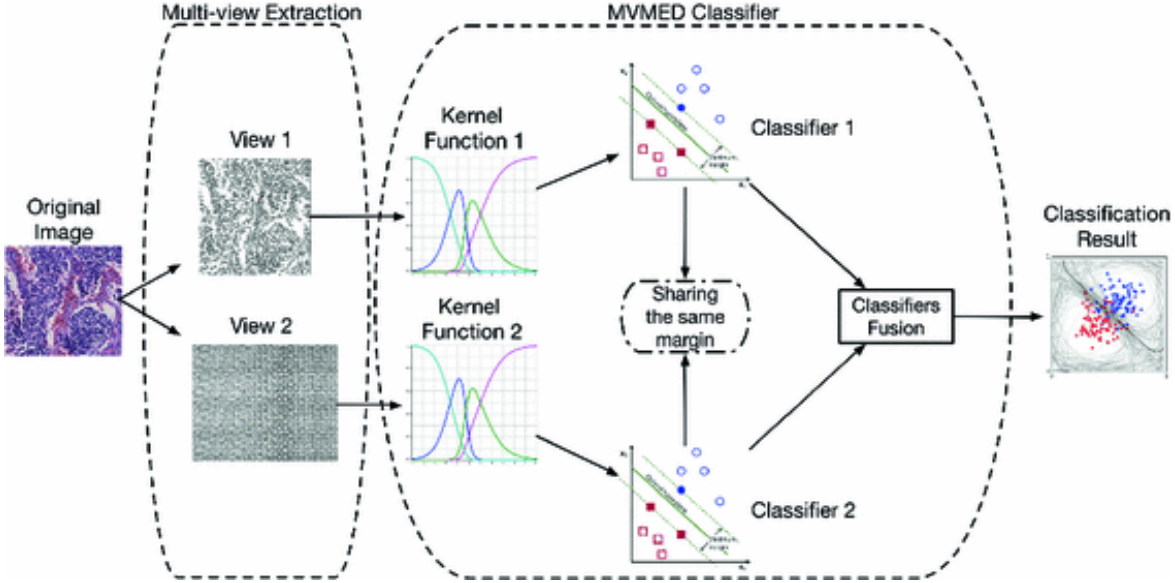
Multi-View Learning

Example: Sentiment Analysis



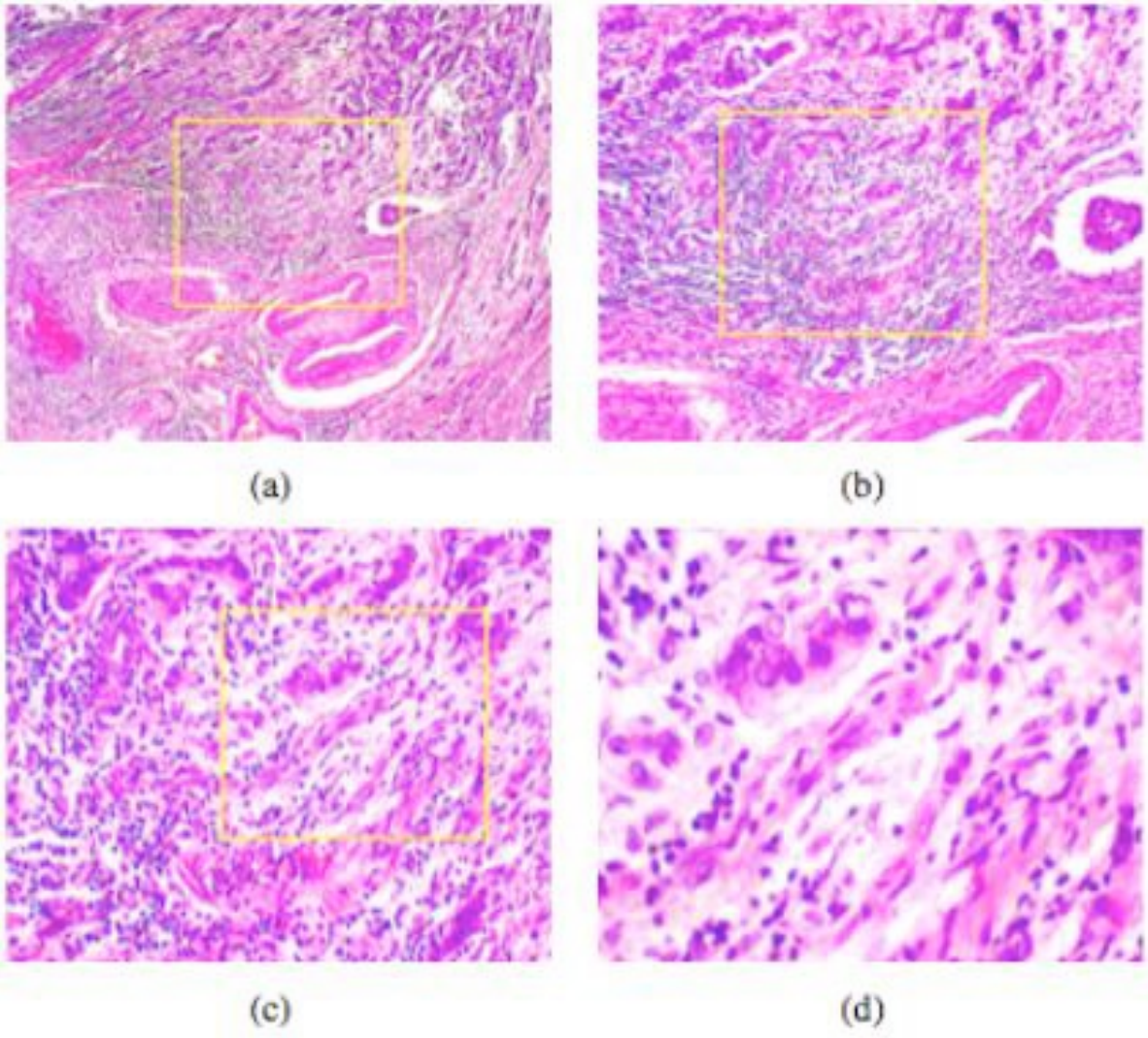
Multi-View Learning

Example: Extraction on an image



Multi-View Learning

Example: Breast Cancer Identification

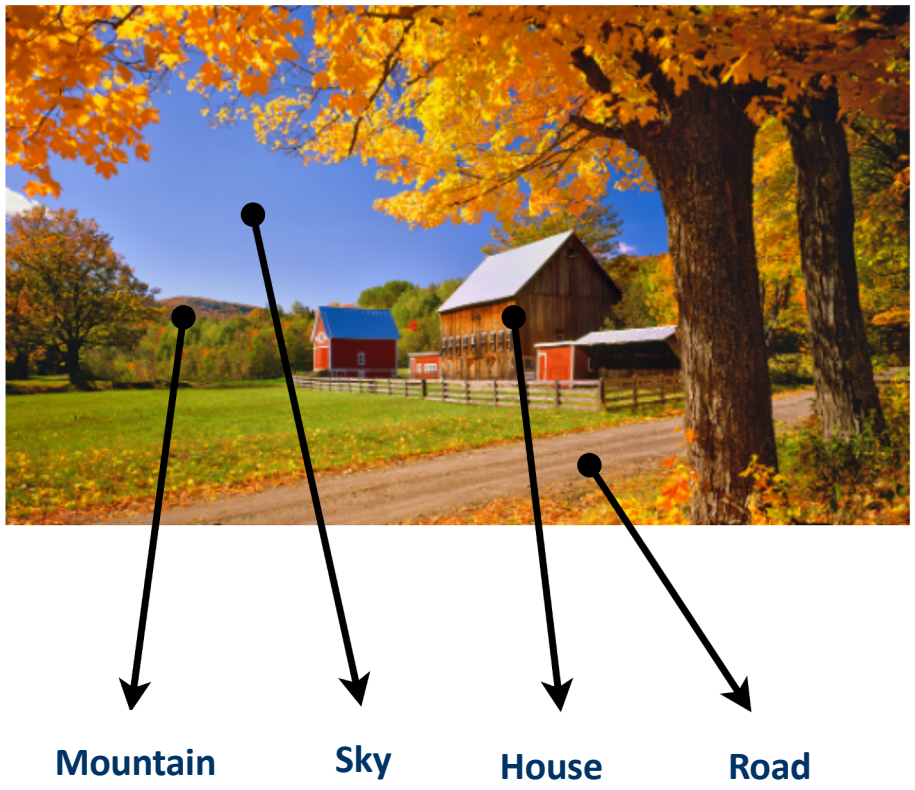


Tissue samples corresponding to malignant breast cancer

a) Magnification 40x
b) Magnification 100x
c) Magnification 200x
d) Magnification 400x

Multiple Outputs Learning

Definition and Formulations



- Each instance is associated with several labels
- Different instances can be associated with a different number of labels
- There are several formulations for this problem:
  - Multi-Label Classification / Multi-Domain Classification
  - Label Distribution Learning
  - Multi-Target Regression
  - Label Ranking

Multi-Label Classification

Techniques

Problem Transformation Methods

- Transforms the multi-label problem into single-label problem(s)
- Use any off-the-shelf single-label classifier to suit requirements
- i.e., **Adapt your data to the algorithm**

Algorithm Adaptation Methods

- Adapt a single-label algorithm to produce multi-label outputs
- Benefit from specific classifier advantages (e.g., efficiency)
- i.e., **Adapt your algorithm to the data**

Many methods involve a mix of both approaches.

Multi-Label Classification

Techniques

For example,

- Binary Relevance: L binary problems (one vs. all)
- Label Powerset: one multi-class problem of 2^L class-values
- Pairwise: $\frac{L(L-1)}{2}$ binary problems (all vs. all)
- Copy-Weight: one multi-class problem of L class values

At training time, with $\mathcal{D}$:

1. Transform the multi-label training data to single-label data
2. Learn from the single-label transformed data

At testing time, for $\tilde{\mathbf{x}}$:

1. Make single-label predictions
2. Translate these into multi-label predictions

Multi-Label Classification

Binary Relevance

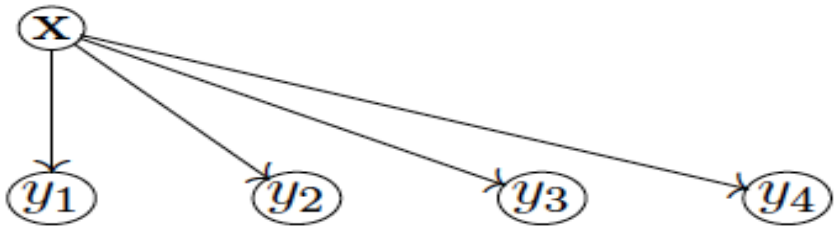
$\mathbf{X}$	Y_1
$\mathbf{x}^{(1)}$	0
$\mathbf{x}^{(2)}$	1
$\mathbf{x}^{(3)}$	0
$\mathbf{x}^{(4)}$	1
$\mathbf{x}^{(5)}$	0

$\mathbf{X}$	Y_2
$\mathbf{x}^{(1)}$	1
$\mathbf{x}^{(2)}$	0
$\mathbf{x}^{(3)}$	1
$\mathbf{x}^{(4)}$	0
$\mathbf{x}^{(5)}$	0

$\mathbf{X}$	Y_3
$\mathbf{x}^{(1)}$	1
$\mathbf{x}^{(2)}$	0
$\mathbf{x}^{(3)}$	0
$\mathbf{x}^{(4)}$	0
$\mathbf{x}^{(5)}$	0

$\mathbf{X}$	Y_4
$\mathbf{x}^{(1)}$	0
$\mathbf{x}^{(2)}$	0
$\mathbf{x}^{(3)}$	0
$\mathbf{x}^{(4)}$	1
$\mathbf{x}^{(5)}$	1

Prediction: $\hat{\mathbf{y}} = [h_1(\tilde{\mathbf{x}}), \ldots, h_L(\tilde{\mathbf{x}})]$



Disadvantages:

- Does not model label dependency, {adult, family} possible
- Class imbalance, e.g., $P(\neg\texttt{family}) \gg P(\texttt{family})$

Multi-Label Classification

Label Powerset

$\mathbf{X}$	$Y \in 2^L$
$\mathbf{x}^{(1)}$	0110
$\mathbf{x}^{(2)}$	1000
$\mathbf{x}^{(3)}$	0110
$\mathbf{x}^{(4)}$	1001
$\mathbf{x}^{(5)}$	0001

- complexity: many class labels
- imbalance: not many examples per class label
- overfitting: how to predict new value?

Multi-Label Classification

Ensemble-based Voting

Ensemble methods (e.g., RAkEL, EPS) make prediction via a voting scheme. For some test instance $\tilde{\mathbf{x}}$:

	$\hat{y}_1$	$\hat{y}_2$	$\hat{y}_3$	$\hat{y}_4$
$\mathbf{h}^1(\tilde{\mathbf{x}})$	1	0	1	
$\mathbf{h}^2(\tilde{\mathbf{x}})$		1	1	0
$\mathbf{h}^3(\tilde{\mathbf{x}})$	1		1	0
$\mathbf{h}^4(\tilde{\mathbf{x}})$	1	0		0
$\mathbf{h}(\tilde{\mathbf{x}})$	3	1	3	0
$\hat{\mathbf{y}}$	1	0	1	0

(majority vote; can also use weighted vote, *threshold*)

- more predictive power (ensemble effect)
- can predict new label combinations

Multi-Label Classification

Pairwise Binary

$\mathbf{X}$	Y_{1v2}
$\mathbf{x}^{(1)}$	0
$\mathbf{x}^{(2)}$	1
$\mathbf{x}^{(3)}$	0
$\mathbf{x}^{(4)}$	1

$\mathbf{X}$	Y_{1v3}
$\mathbf{x}^{(1)}$	0
$\mathbf{x}^{(2)}$	1
$\mathbf{x}^{(4)}$	1

$\mathbf{X}$	Y_{1v4}
$\mathbf{x}^{(2)}$	1
$\mathbf{x}^{(5)}$	0

$\mathbf{X}$	Y_{2v3}
$\mathbf{x}^{(3)}$	1

$\mathbf{X}$	Y_{2v4}
$\mathbf{x}^{(1)}$	1
$\mathbf{x}^{(3)}$	1
$\mathbf{x}^{(4)}$	0
$\mathbf{x}^{(5)}$	0

$\mathbf{X}$	Y_{3v4}
$\mathbf{x}^{(1)}$	1
$\mathbf{x}^{(4)}$	0
$\mathbf{x}^{(5)}$	0

Predict $y_{j,k} = \mathbf{h}_{j,k}(\tilde{\mathbf{x}})$ *for all* $1 \leq j < k \leq L$

$$y_{j,k} = \begin{cases} 0, & \lambda_j \succ \lambda_k \\ 1, & \lambda_k \succ \lambda_j \end{cases}$$

Issues:

- this produces pairwise rankings, how to get a labelset?
- how much sense does it make to find a decision boundary between overlapping labels?
- can be expensive in terms of numbers of classifiers ($\frac{L(L-1)}{2}$)

Multi-Label Classification

Copy-Weight

$\mathbf{X}$	Y_1	Y_2	Y_3	Y_4
$\mathbf{x}^{(1)}$	0	1	1	0
$\mathbf{x}^{(2)}$	1	0	0	0
$\mathbf{x}^{(3)}$	0	1	0	0
$\mathbf{x}^{(4)}$	1	0	0	1
$\mathbf{x}^{(5)}$	0	0	0	1

. . . make a single multi-class problem with L possible class values:

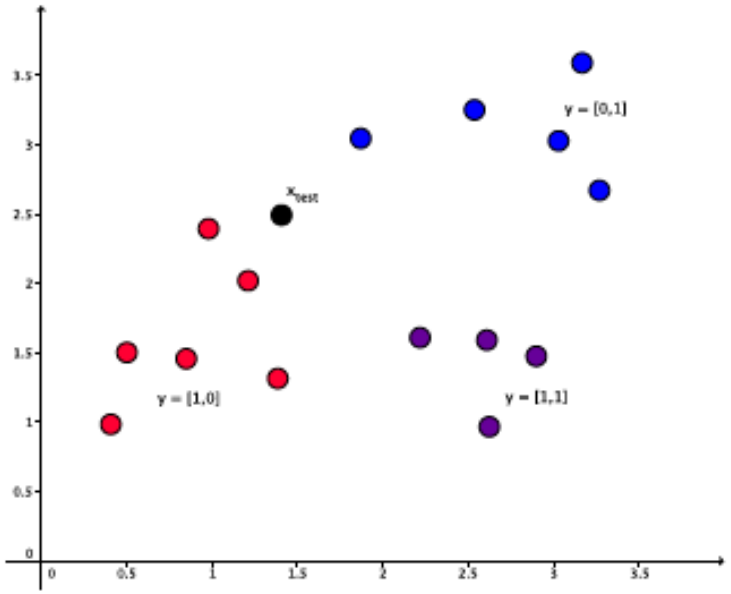
$\mathbf{X}$	$Y \in \{1, \ldots, L\}$	w
$\mathbf{x}^{(1)}$	2	0.5
$\mathbf{x}^{(1)}$	3	0.5
$\mathbf{x}^{(2)}$	1	1.0
$\mathbf{x}^{(3)}$	2	1.0
$\mathbf{x}^{(4)}$	1	0.5
$\mathbf{x}^{(4)}$	4	0.5
$\mathbf{x}^{(5)}$	4	1.0

each example duplicated $|Y^{(i)}|$ times, weighted as $\frac{1}{|Y^{(i)}|}$.

Multi-Label Classification

MLk-NN

- kNN assigns to $\tilde{\mathbf{x}}$ the majority class of the k 'nearest neighbours'
- MLkNN [Zhang and Zhou, 2007] assigns to $\tilde{\mathbf{x}}$ the most common *labels* of the k nearest neighbours neighbours

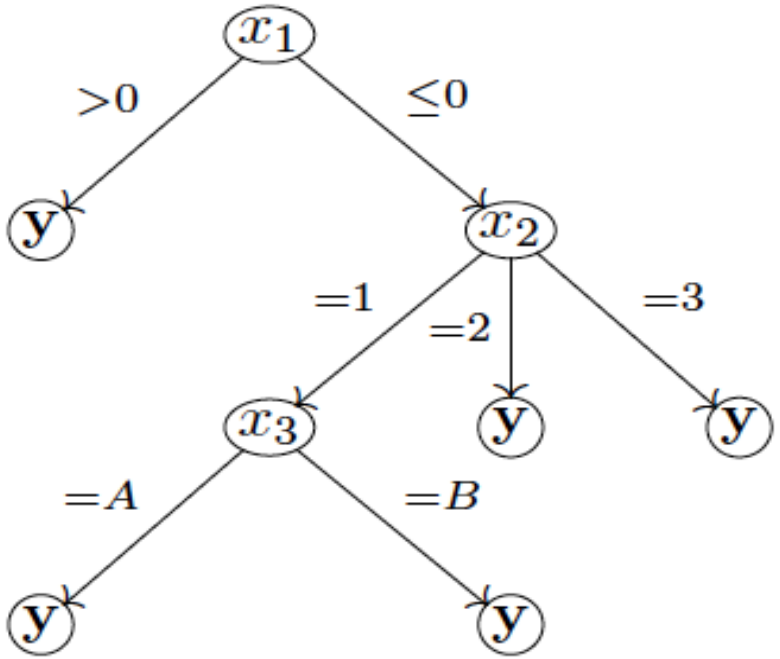


- . . . combined with Bayesian inference (MAP principle):

Multi-Label Classification

- Multi-label C4.5 [Clare and King, 2001]: Extension of the popular C4.5 decision tree algorithm; with multi-label entropy:

$$H_{\mathrm{ML}}(S) = \sum_{j=1}^{L} P(y_j)\log(P(y_j)) + (1 - P(y_j))\log(1 - P(y_j))$$



- constructed just like C4.5
- allows multiple labels at the leaves

Multi-Label Classification

RankSVM

RankSVM, a Maximum Margin approach [Elisseeff and Weston, 2002]:

- one classifier for each label

$$h_j(\mathbf{x}) = \mathbf{w}_j^\top \mathbf{x} + b_j$$

- use kernel trick for non-linearity
- define multi-label margin, for each $(\mathbf{x}^{(i)}, \mathbf{y}^{(i)})$ in training set $\mathcal{D}$:

$$\min_{(\mathbf{x}^{(i)},\mathbf{y}^{(i)})\in\mathcal{D}} \min_{j,k} \frac{\mathbf{w}_j^\top \mathbf{x} + b_j - \mathbf{w}_k^\top \mathbf{x} - b_k}{\|\mathbf{w}_j - \mathbf{w}_k\|}$$

- solve with quadratic programming
- improved performance over BR with SVMs

Multi-Label Classification

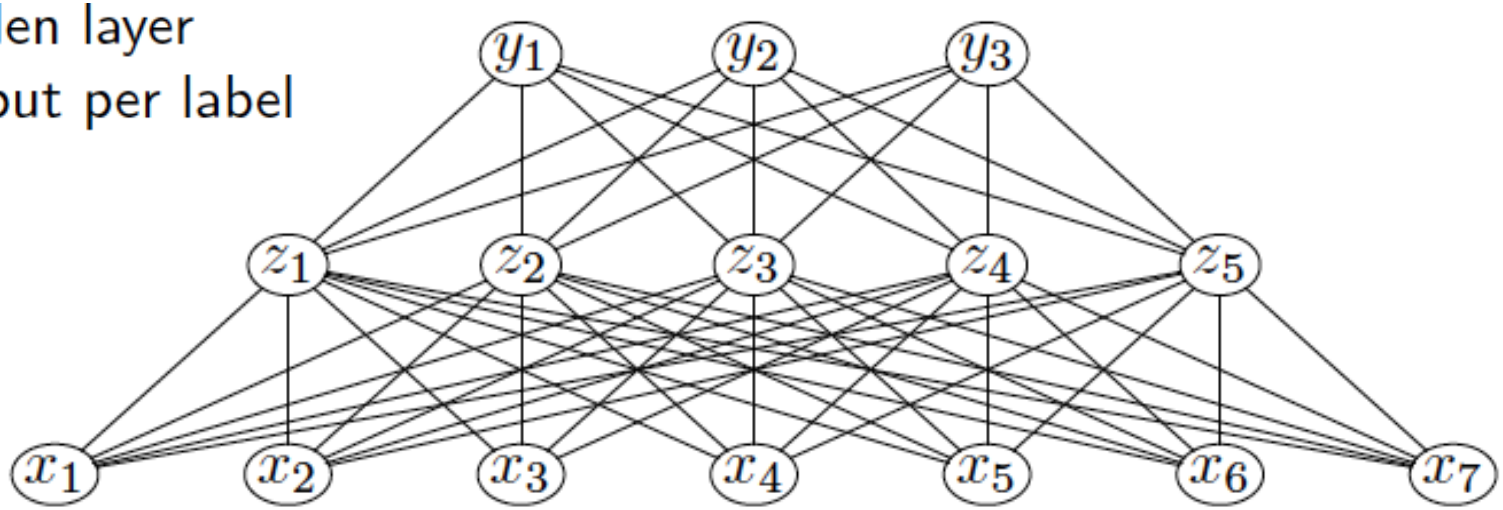
Multi-Layer Perception

BPMLL [Zhang and Zhou, 2006] is

- a regular back-prop. neural network with multiple outputs
- trained with gradient descent + error back-propagation
- with an error function based on ranking (relevant labels should be ranked higher than non-relevant labels)

$$E = \sum_{i=1}^{N} E_i = \sum_{i=1}^{N} \frac{1}{|Y_i||\bar{Y}_i|} \sum_{(j,k)\in Y_i \times \bar{Y}_i} \exp(-(y_k^{(i)} - y_j^{(i)}))$$

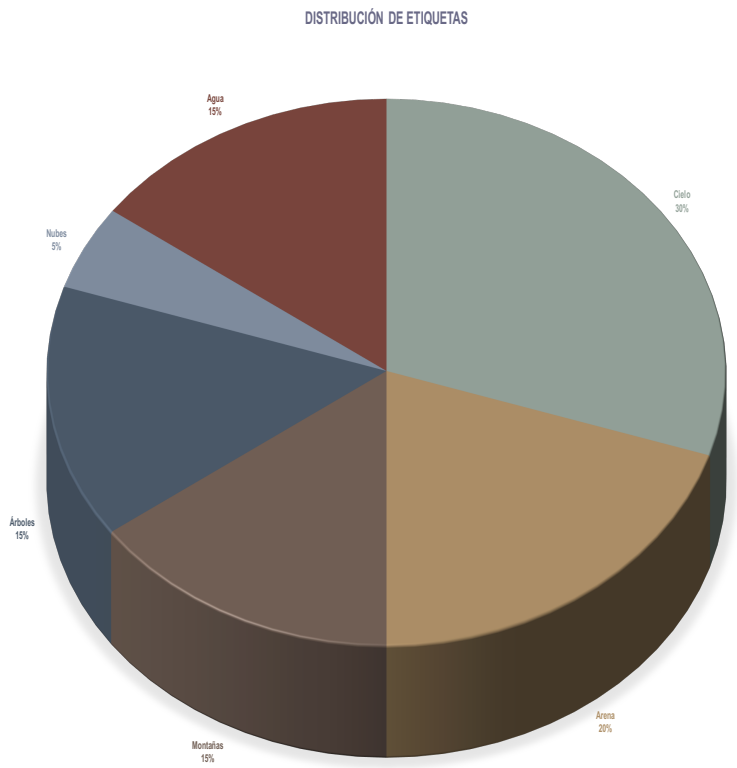
- one hidden layer
- one output per label



Label Distribution Learning

Multi-Label vs. Label Distribution

A generalisation of the classification problem



Single Label Learning (c outputs)

- Beach

Multi-Label Learning (2^c-1 outputs)

- Sand
- Sky
- Water
- Mountains
- Trees
- Clouds

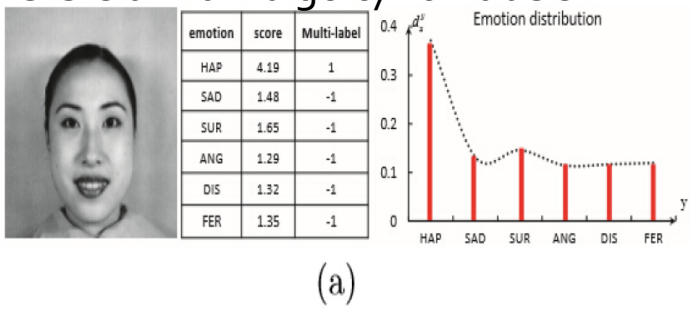
Label Distribution Learning (∞ outputs)

- Mostly sky
- Quite a bit of water
- Quite a lot of sand
- Some trees

Label Distribution Learning

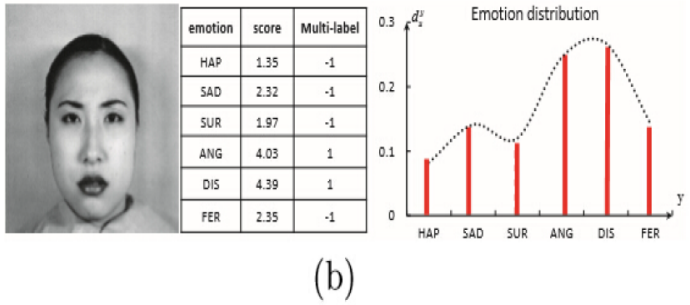
Applications

LDL presents a paradigm that naturally fits the real complexity of the problems. In these cases there is an "ambiguity" of labels



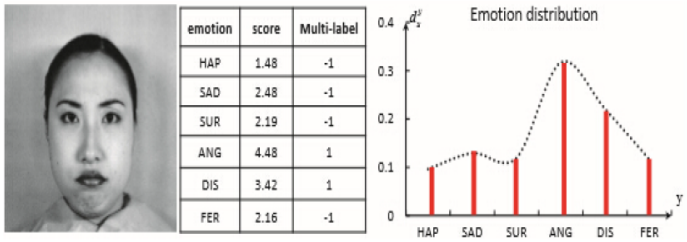
emotion	score	Multi-label
HAP	4.19	1
SAD	1.48	-1
SUR	1.65	-1
ANG	1.29	-1
DIS	1.32	-1
FER	1.35	-1

(a)



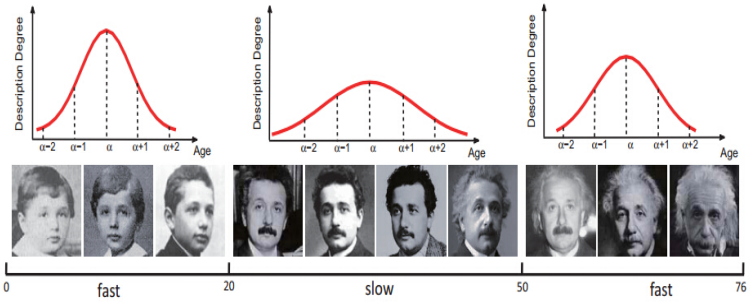
emotion	score	Multi-label
HAP	1.35	-1
SAD	2.32	-1
SUR	1.97	-1
ANG	4.03	1
DIS	4.39	1
FER	2.35	-1

(b)

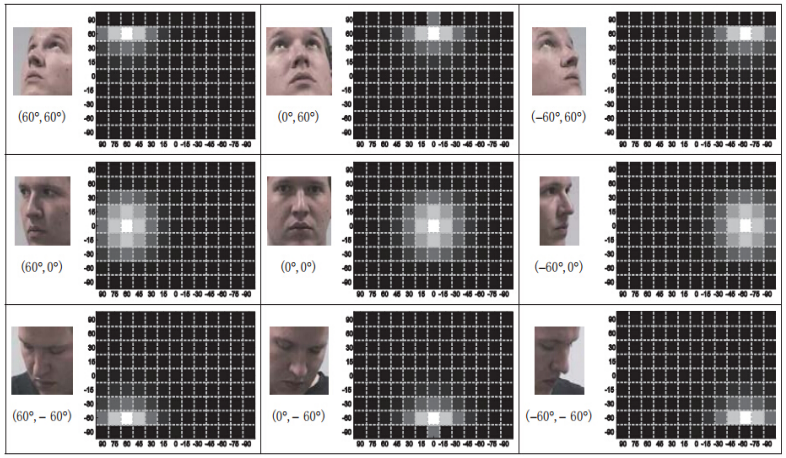


emotion	score	Multi-label
HAP	1.48	-1
SAD	2.48	-1
SUR	2.19	-1
ANG	4.48	1
DIS	3.42	1
FER	2.16	-1

Detection of emotions in facial expression



Facial age estimation

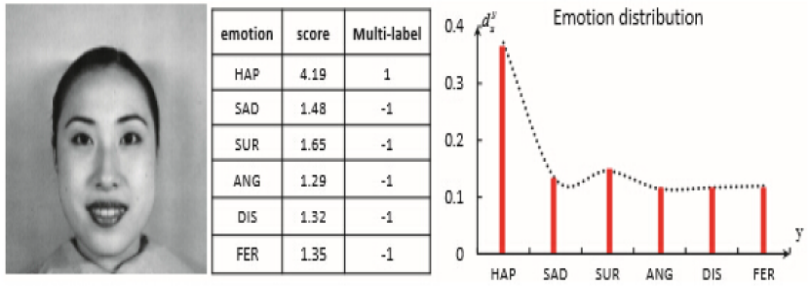


Face orientation detection

Label Distribution Learning

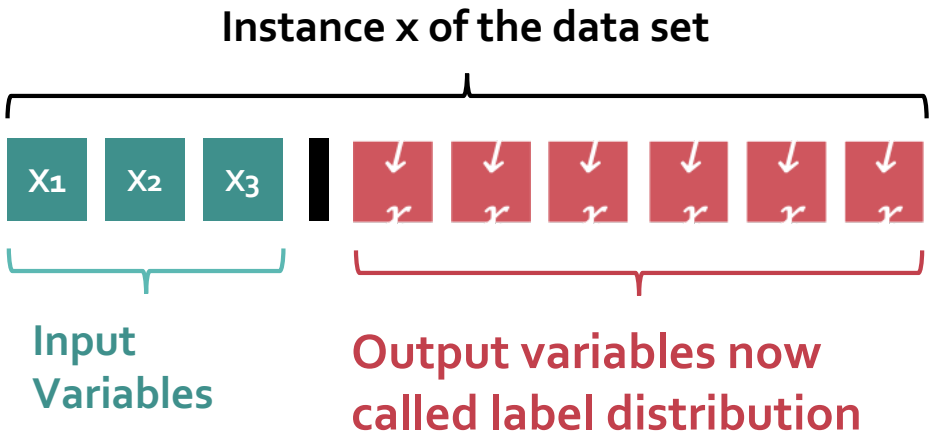
Fundamentals

LDL is a learning paradigm capable of naturally addressing multiple output problems and exploiting the relationship between those outputs. The question is: HOW much does each label describe the example vs. WHICH label(s) it describes. The output is not a confidence or probability value but a 'proportion' of the full description.



emotion	score	Multi-label
HAP	4.19	1
SAD	1.48	-1
SUR	1.65	-1
ANG	1.29	-1
DIS	1.32	-1
FER	1.35	-1

Example of Label Distribution



All Multi-Target Learning problem is LDL if two important concepts are fulfilled:

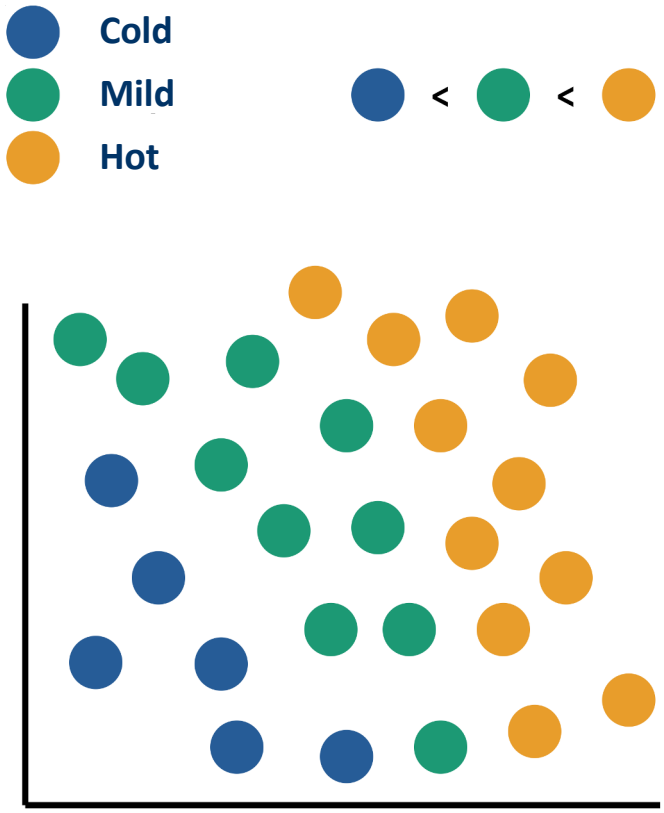
- **Description degree**
- **Label Distribution**

LDL properties

1. **Each description degree must have a real value entre [0, 1]**
2. **The sum of the label distribution must be the unit.**

Ordinal Regression/Classification

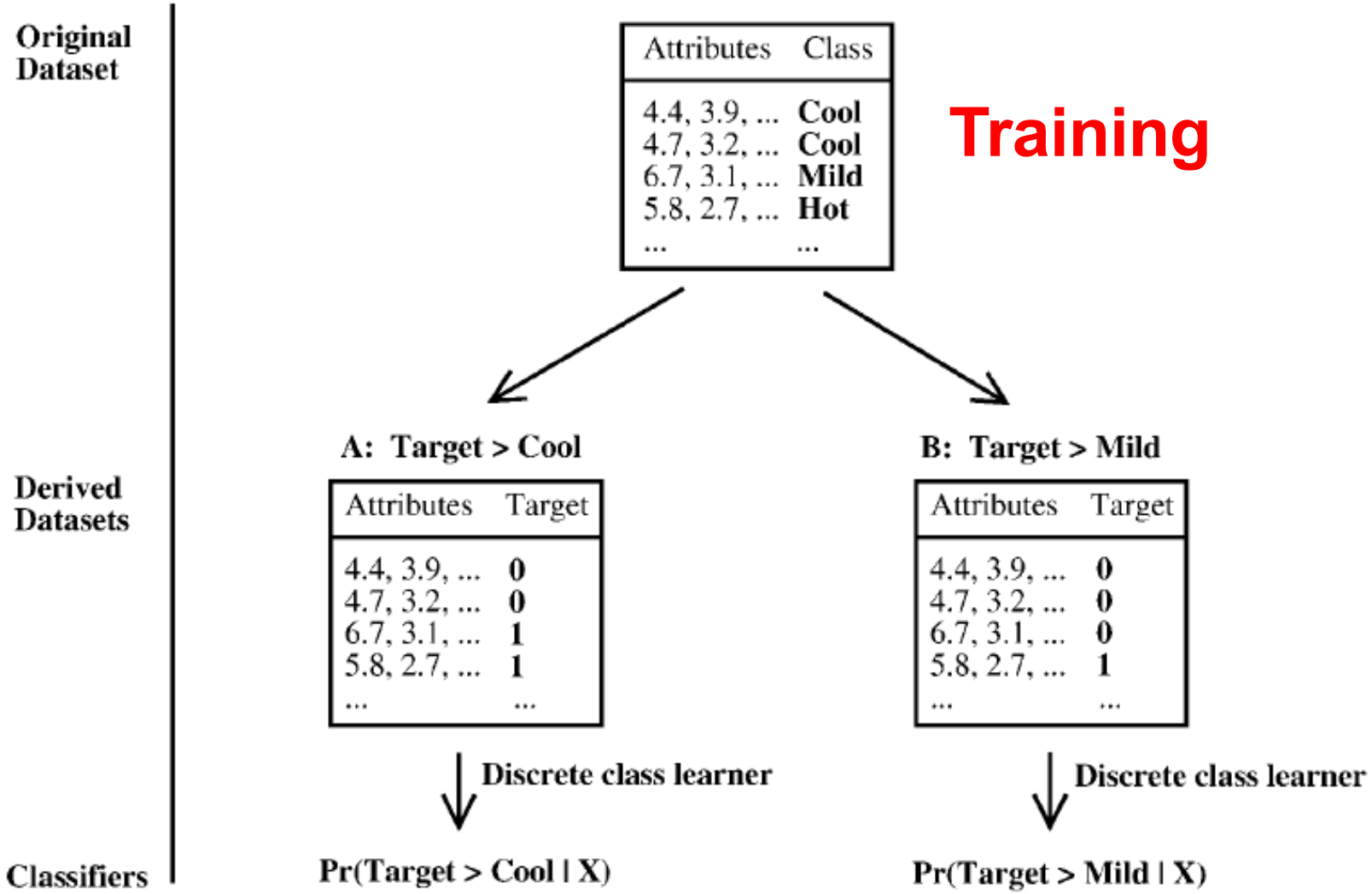
Definition



- There is an orderly relationship between the classes
- The objective is to minimize errors that consider order between classes
- The costs of misclassification are not the same for different classes
- Distances among clases are unknown
- Non-balanced classes are common

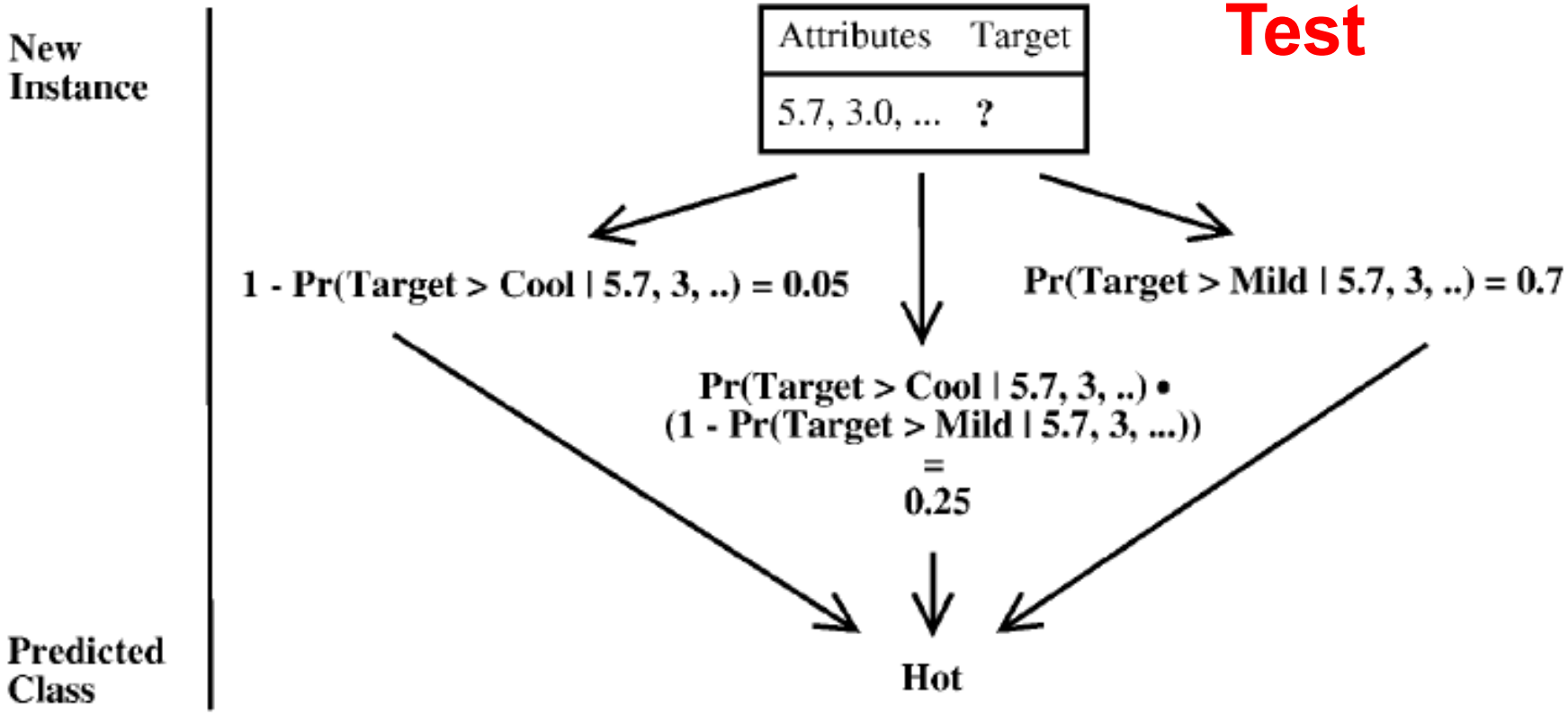
Ordinal Regression/Classification

A simple Algorithm Adaptation



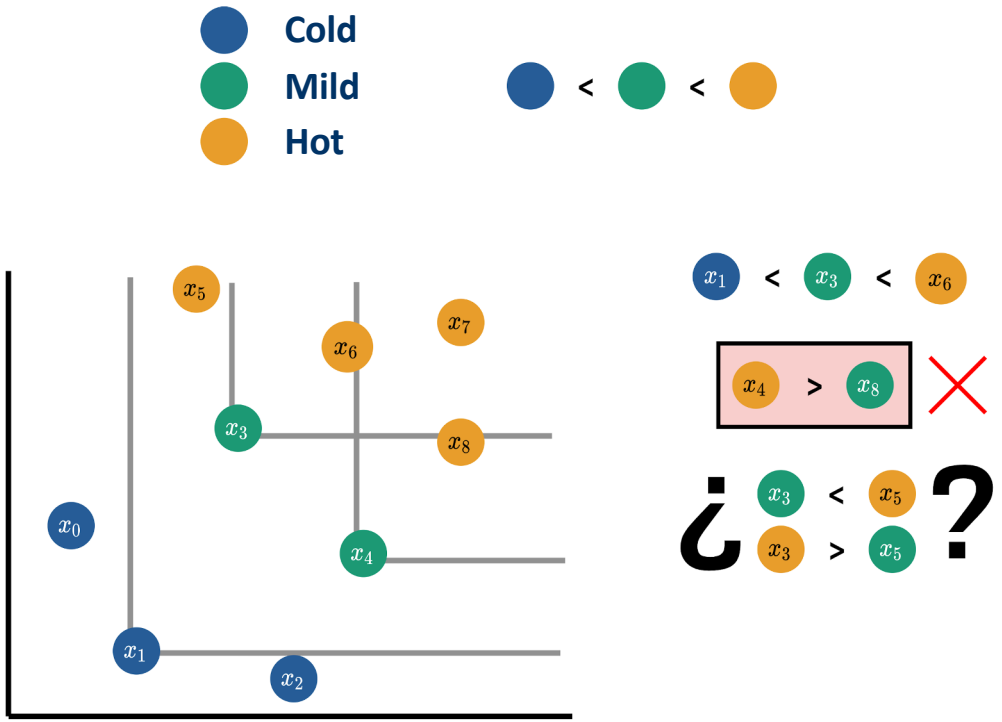
Ordinal Regression/Classification

A simple Algorithm Adaptation



Monotonic Classification

Definition



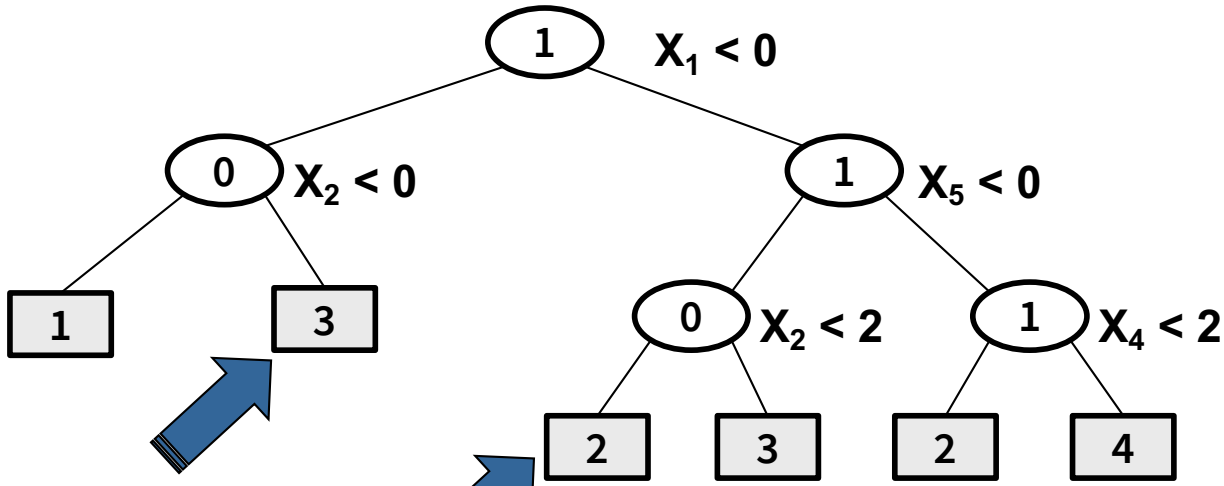
- There is an orderly relationship between the classes
- There is an orderly relationship between the examples
- If two instances are comparable, the sense of this comparison should be the same in your class assignment
- There are partial order variations, with monotone and non-monotone attributes

Monotonic Classification

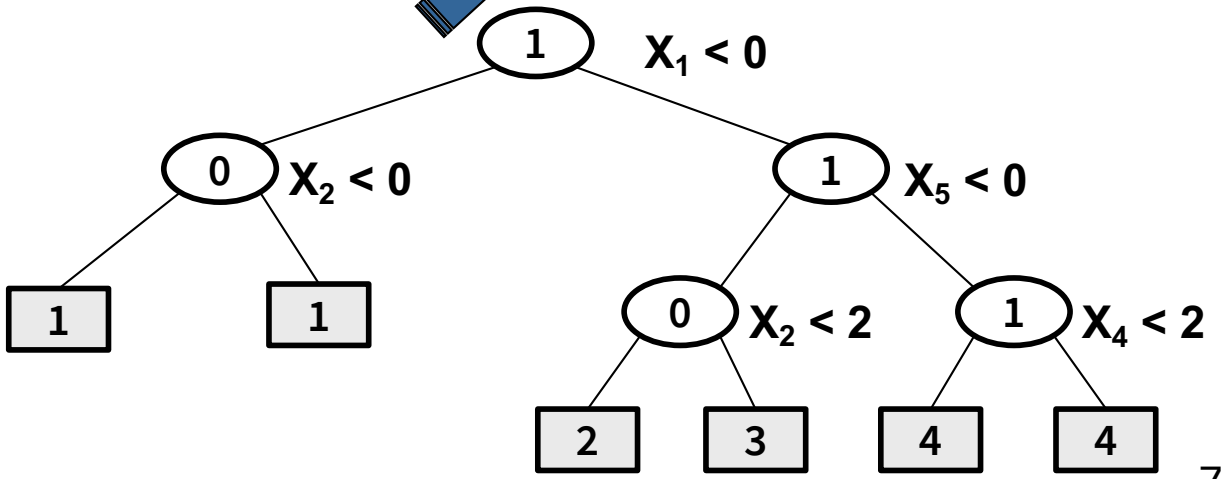
Example of Decision Tree

NON-MONOTONE DECISION TREE

(-1,0,0,0,-2) < (1,1,1,1,-1)



MONOTONE DECISION TREE



Isotonic Regression

Example

Attribute	Type	Sign
mpg	continuous	target
cylinders	multi-valued discrete	−
displacement	continuous	−
horsepower	continuous	−
weight	continuous	−
acceleration	continuous	+
model year	multi-valued discrete	+
origin	multi-valued discrete	+

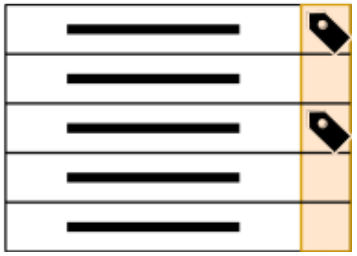
The relationship of monotony between each predictor variable and the target variable has a sign (positive = direct relationship, negative = inverse relationship)

Partial Information

Problem Taxonomy

Common variation types:

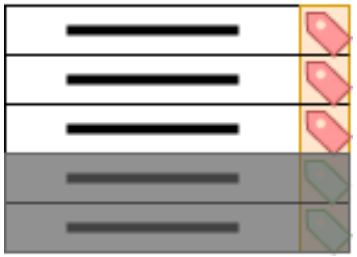
Partial information: in the training information there are unlabeled



Positive-unlabeled learning

Semi-supervised learning

or missing instances of some classes



Zero-shot learning

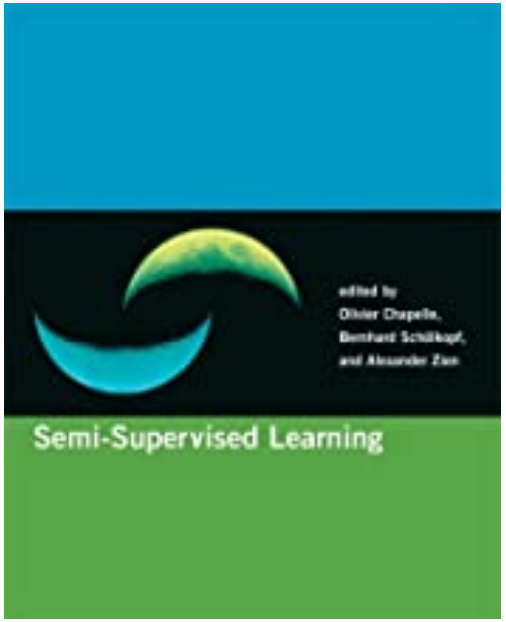
One-class classification

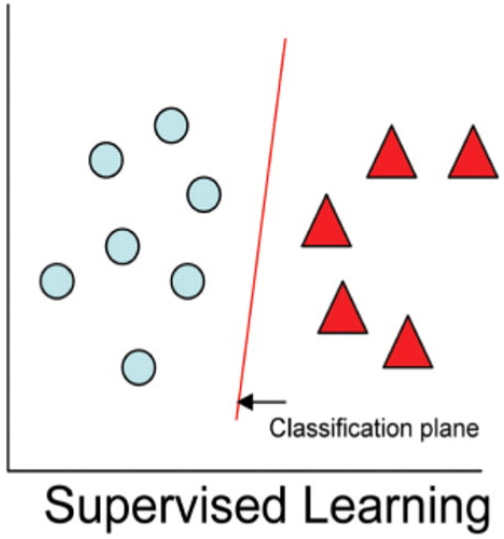
Few-shot learning

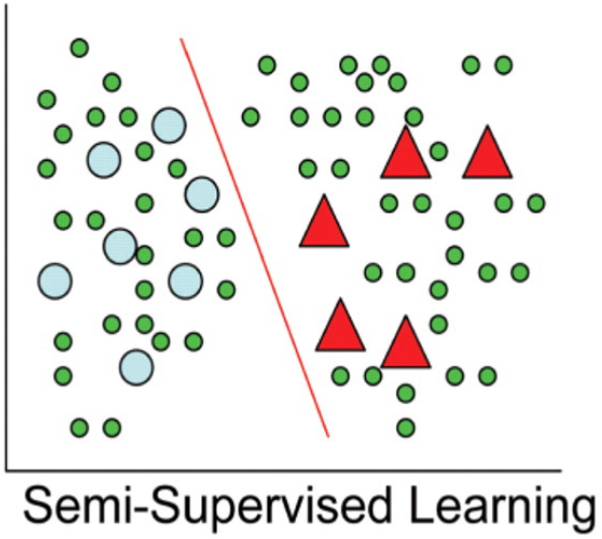
Semi-Supervised Learning

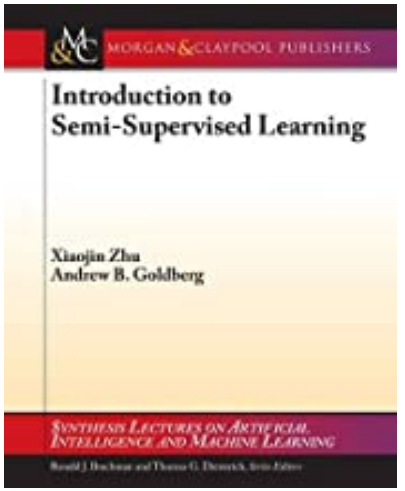
Applications

Applications where labelling data is expensive:
Classification of web pages
Speech recognition
Protein sequences
...



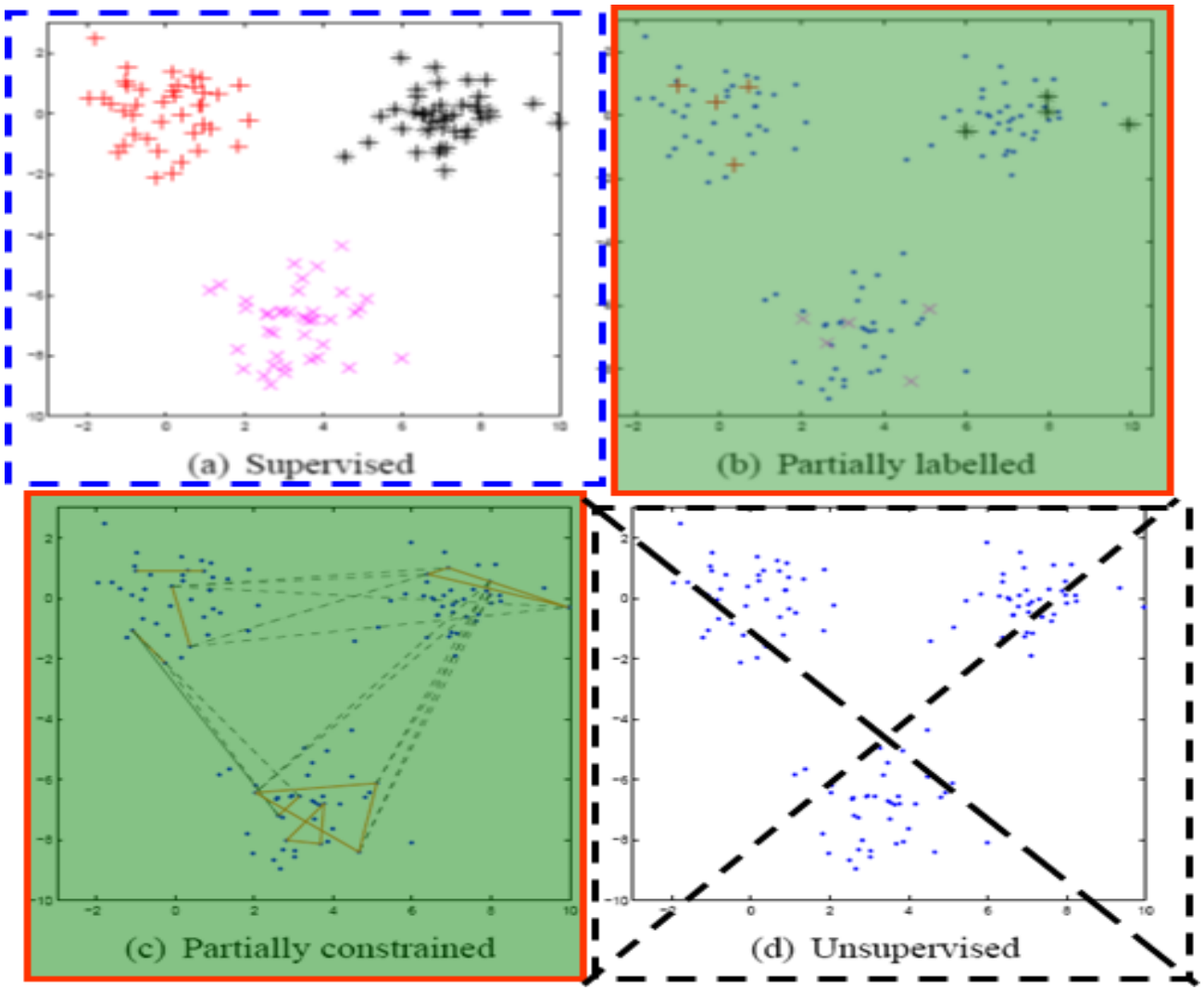






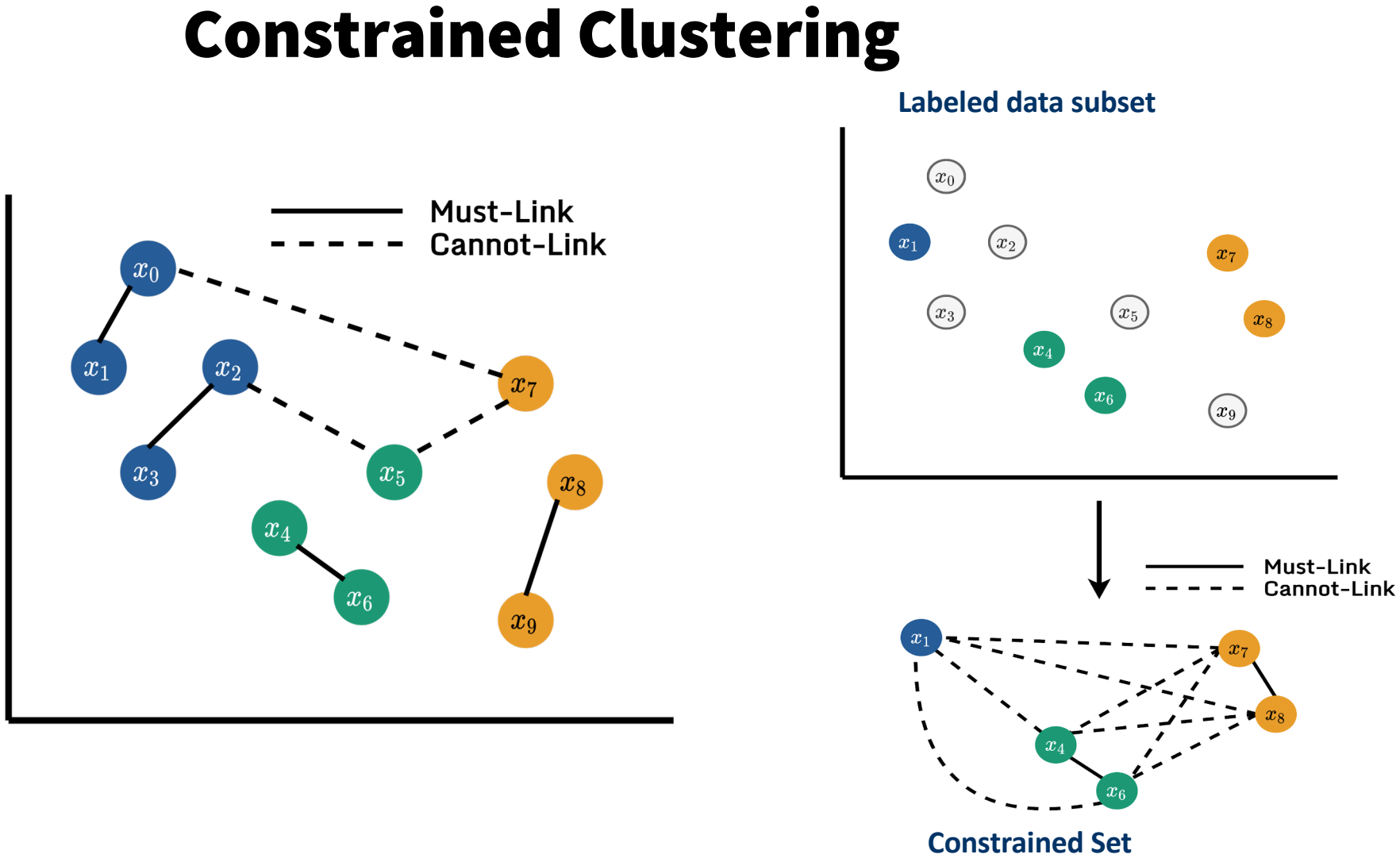
Semi-Supervised Learning

Types



Semi-Supervised Learning

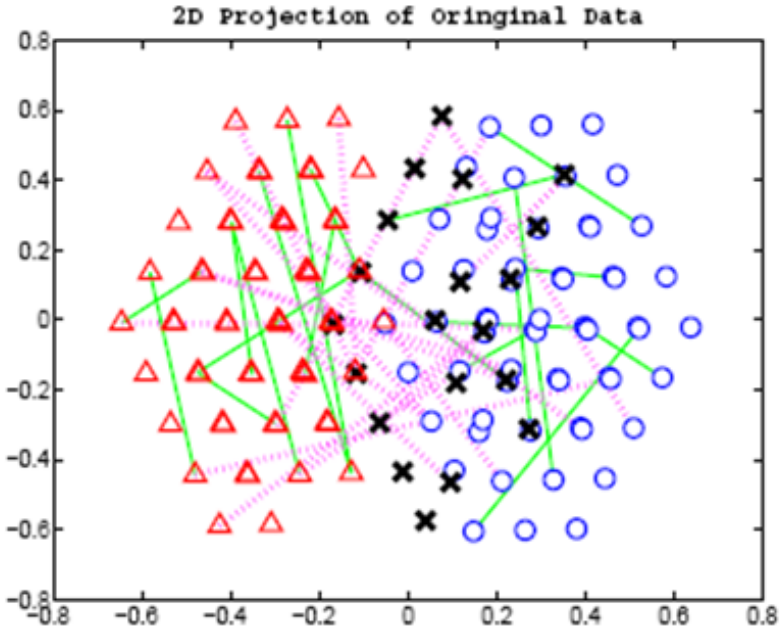
Constrained Clustering



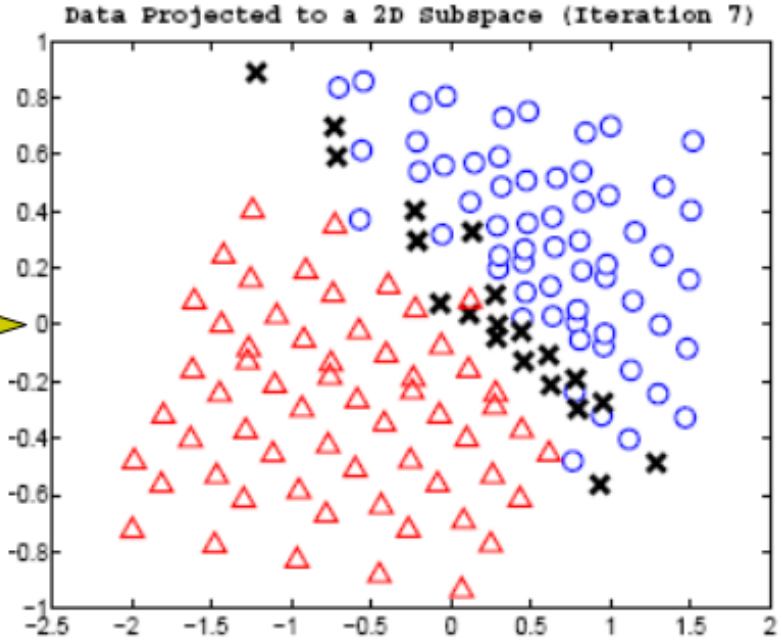
Semi-Supervised Learning

Distance Metric Learning

2D data projection using Euclidean distance metric



2D data projection using learned distance metric



Solid lines: must links

dotted lines: cannot links

Semi-Supervised Learning

Distance Metric Learning

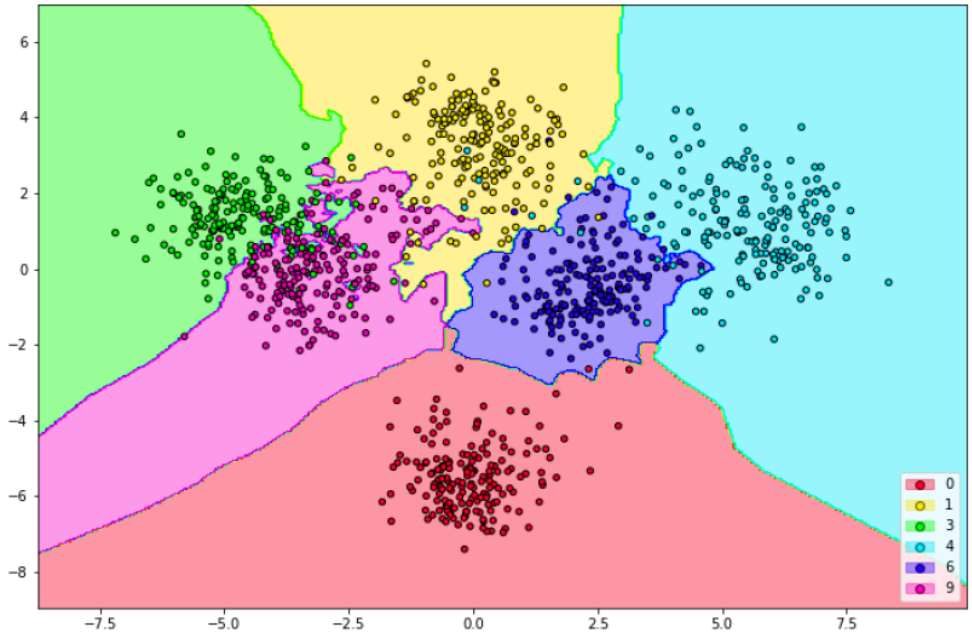


Figure 2: 'Digits' dataset consists of 1797 examples. Each of them consists of a vector with 64 attributes, representing intensity values on an 8x8 image. The examples belong to 10 different classes, each of them representing the numbers from 0 to 9. By learning an appropiate transformation we are able to project most classes on the plane, so that we perceive clearly differentiated regions associated with each of the classes.

Semi-Supervised Learning

Zero-Shot Learning

Z-SL based recognition is based on the existence of a set of labeled view classes and the knowledge of how each unseen class is semantically related to the view classes.

The reason why humans can perform Z-SL is because of its existing language knowledge base, which provides a high-level description of a new or unseen class (zebra) and makes a connection between it and the classes and visual concepts seen (horse, stripes). Inspired by this ability of humans, there is a growing interest in the ZSL machine to extend visual recognition.

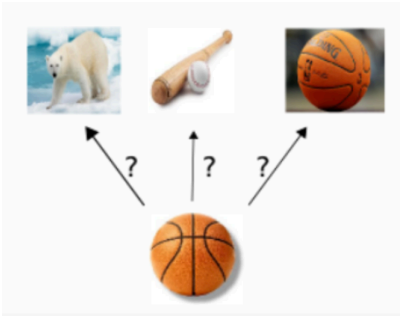
The Z-SL approaches are designed to learn the semantic middle layer, its attributes, and apply them at the time of inference to predict a new kind of data.

Semi-Supervised Learning

Few-Shot Learning

F-SL is a classification task in which one or a few examples are used to classify many new examples in the future.
This characterises tasks that are seen in the field of facial recognition, such as face identification and verification, where people must be correctly classified with different facial expressions, lighting conditions, accessories and hairstyles by giving them one or a few template photos.
Siamese Network: Deep CNNs are first trained to discriminate between examples in each class. The idea is that the models learn feature vectors that are effective in extracting abstract features from the input images.

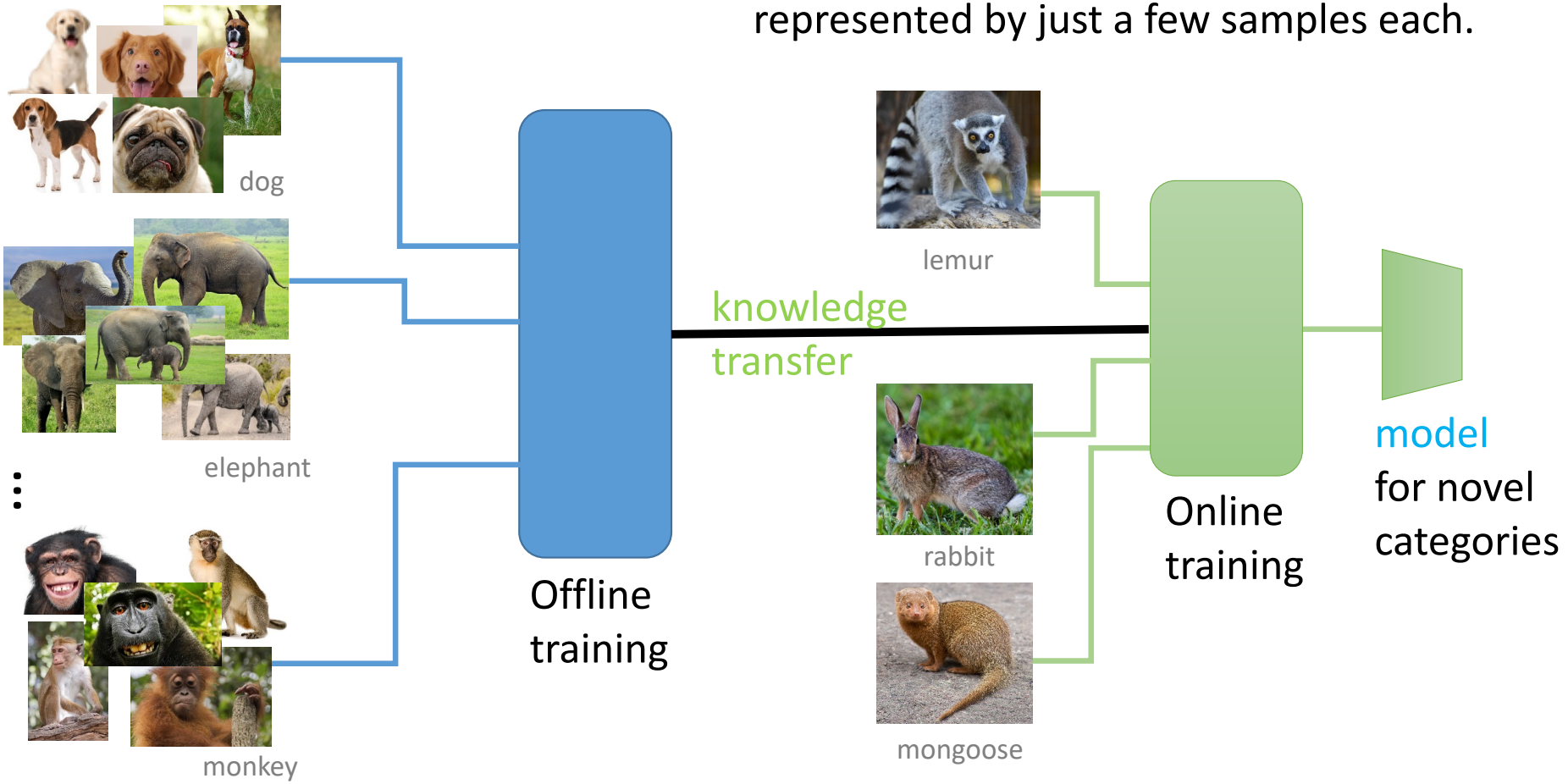
The models are re-proposed for verification in order to predict whether the new examples match a template for each class.



Few-Shot Learning

Problem Statement

Using a large annotated offline dataset, perform given task for novel categories, represented by just a few samples each.



Few-Shot Learning

Technologies

Meta-learning
Learn a learning strategy to adjust well to a new few-shot learning task

Metric learning
Learn a `semantic` embedding space using a distance loss function

Few-shot learning

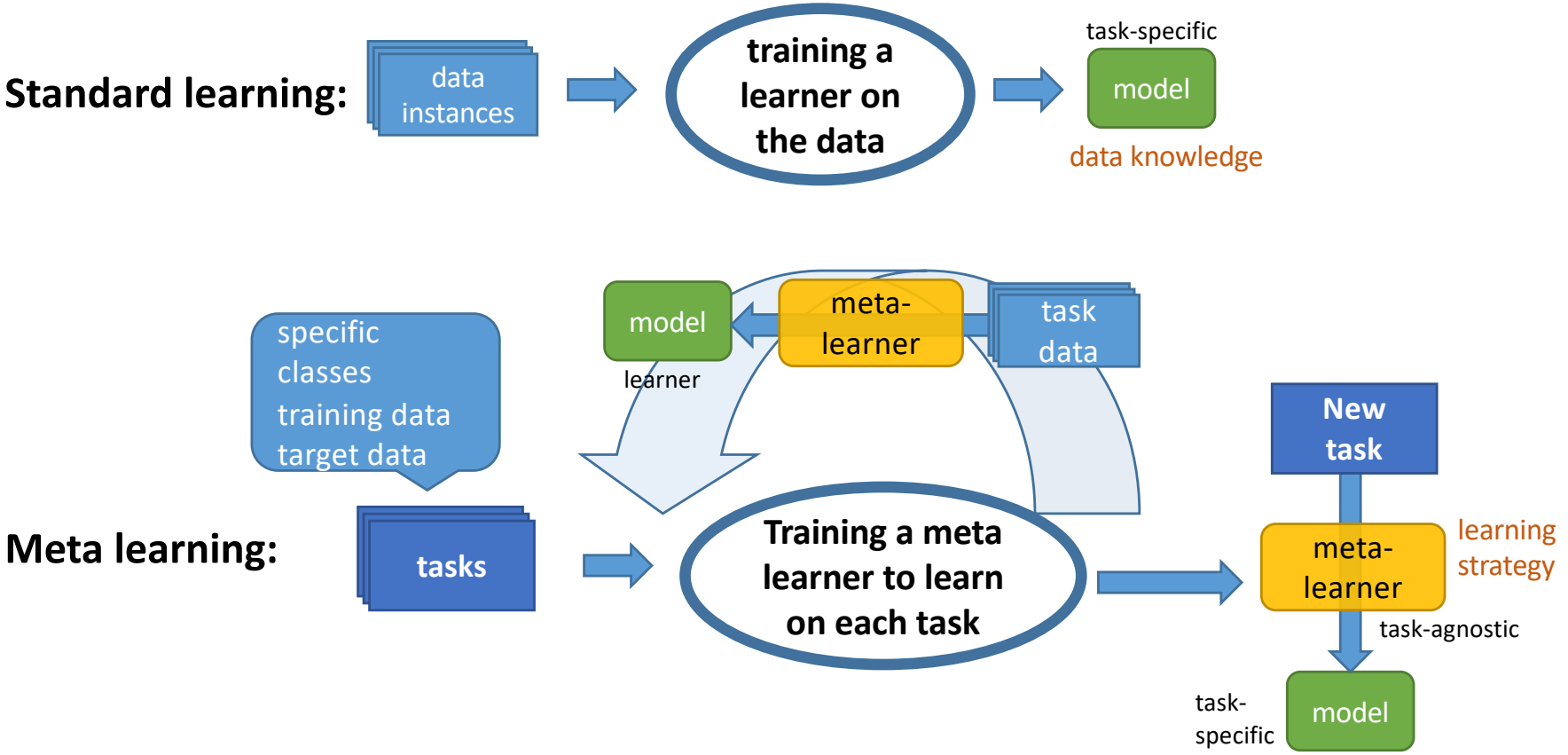
Learn to perform classification, detection, regression

Each category is represented by just a few examples

Data augmentation
Synthesize more data from the novel classes to facilitate the regular learning

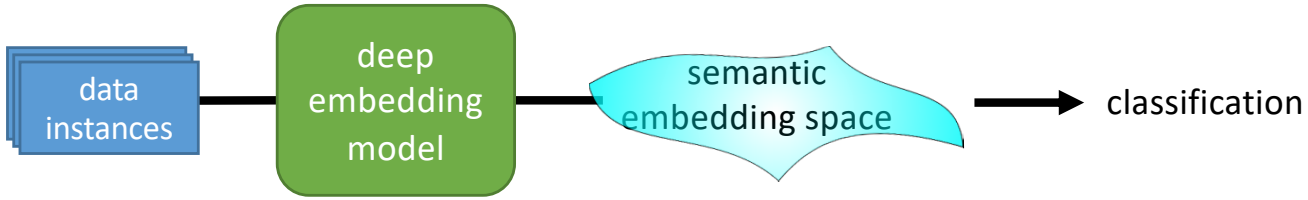
Few-Shot Learning

Meta-Learning



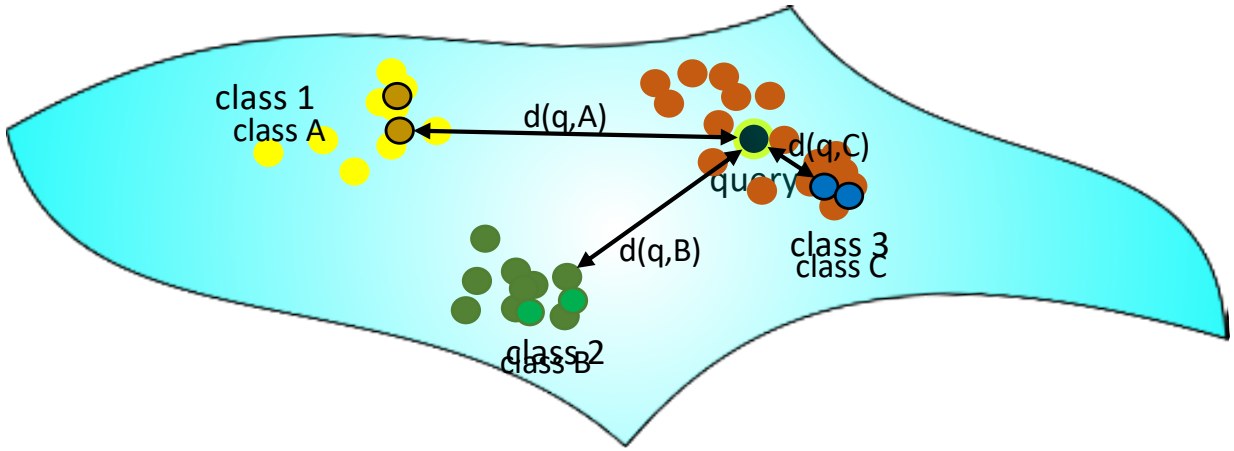
Few-Shot Learning

Metric Learning



Offline training

New task data



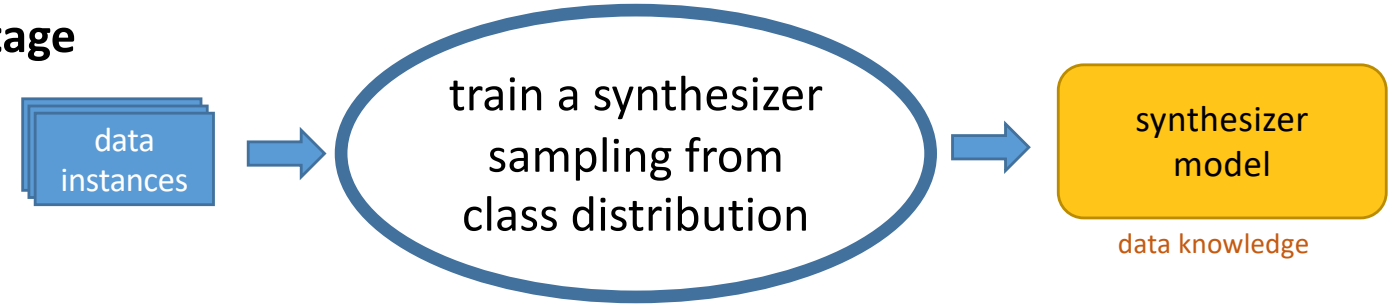
Training: achieve good distributions for offline categories

Inference: Nearest Neightbour in the embedding space

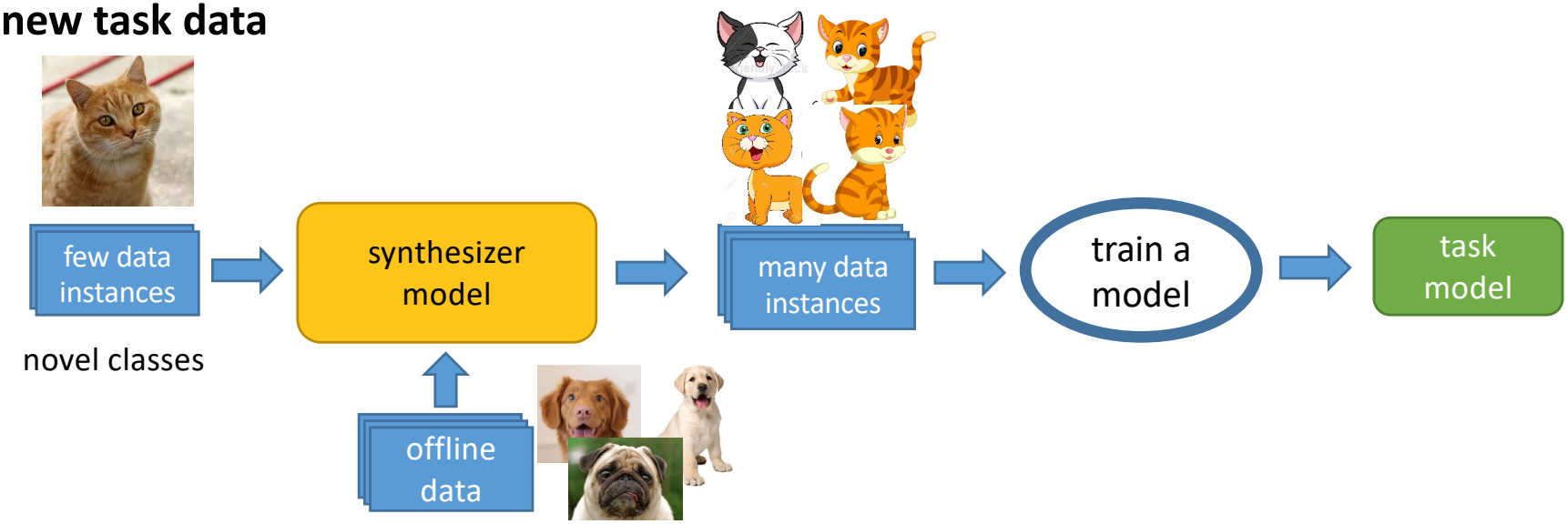
Few-Shot Learning

Data Aumentation through Sample synthesis

Offline stage



On new task data



Complementary References

- Jing Zhao, Xijiong Xie, Xin Xu, Shiliang Sun: Multi-view learning overview: Recent progress and new challenges. Inf. Fusion 38: 43-54 (2017).
- Xin Geng: Label Distribution Learning. IEEE Trans. Knowl. Data Eng. 28(7): 1734-1748 (2016).
- Pedro Antonio Gutiérrez, María Pérez-Ortiz, Javier Sánchez-Monedero, Francisco Fernández-Navarro, César Hervás-Martínez: Ordinal Regression Methods: Survey and Experimental Study. IEEE Trans. Knowl. Data Eng. 28(1): 127-146 (2016).
- José Ramón Cano, Pedro Antonio Gutiérrez, Bartosz Krawczyk, Michal Wozniak, Salvador García: Monotonic classification: An overview on algorithms, performance measures and data sets. Neurocomputing 341: 168-182 (2019).
- Juan Luis Suárez, Salvador García, Francisco Herrera. A Tutorial on Distance Metric Learning: Mathematical Foundations, Algorithms, Experimental Analysis, Prospects and Challenges. Neurocomputing (in press, 2021).
- Yaqing Wang, Quanming Yao, James Kwok, Lionel M. Ni. Generalizing from a Few Examples: A Survey on Few-Shot Learning. arXiv:1904.05046v3, 2020.

SOMACHINE

Machine Learning, Big Data, and Deep Learning in Astronomy

Singular Problems in ML

Salvador García

Andalusian Research Institute of Data Science and Computational Intelligence (DaSCI)

Dpto. Ciencias de la Computación e I.A.

Universidad de Granada

salvagl@decsai.ugr.es

http://sci2s.ugr.es