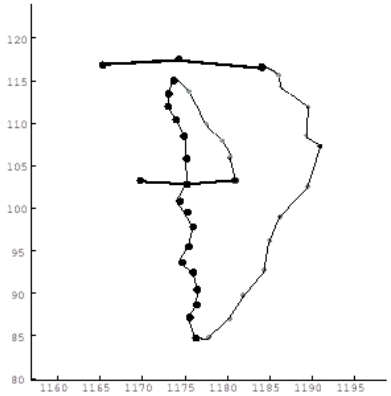
Preprocessing and Learning Systems



Lektor Dr.techn. Alexander K. Seewald
Österreichisches Forschungsinstitut
für Artificial Intelligence

alex@seewald.at / alex.seewald.at

Types of Attributes

Categorical Scale (qualitative, discrete, nominal)

- e.g. {Success, Failure}, {ORF1, SAT1, RTL, ...}
- Finite, small number of values; arbitrary ordering of values

Ordinal scale (ordered categorical ~ qualitative)

- e.g. {low, medium, high}, {slow, moderate, fast}
- Ordering is apparent (low<medium<high), but distances are meaningless.

  E.g. the distance between low and medium and between medium and high is not necessarily the same.

Interval/Ratio scale (quantitative, numeric, continuous)

- e.g. temperature in ℃/℉/K, wind speed in km/h
- Interval: Distances between values are meaningful.

  Today it is 2 ℃ colder than yesterday.
- Ratio: Ratios are also meaningful. Zero point has to be known!

  In Florida, the wind blows twice as fast as here.

WEKA only distinguishes between qualitative (categorical & ordinal scale) and quantitative (interval & ratio scale) attributes/features. Convert Qualitative ⇔ Quantitative via NominalToBinary(⇒) and Discretize(⇐)

alex@seewald.at / alex.seewald.at

Types of Datasets

Attributes can be related temporally and/or spatially:

- **Unstructured Data:** no explicit temporal/spatial structure (= no temporal/spatial relationship over attributes)
- **Time Series Data:** temporal structure
- **Text Data:** special kind of time series data
- **Spatial Data:** spatial structure
- **Spatiotemporal Data:** both spatial and temporal structure

For all learning systems, the order of attributes is **irrelevant**. Therefore, **spatial and temporal structure** between attributes **is not adressed**. If one wants or needs to take account of spatial/temporal structure in the data, this is done by appropriate **preprocessing**.

alex@seewald.at / alex.seewald.at

Unstructured Data

Each attribute measures a property. Each example represents one object with all measured properties.

- Postal zip code of home residence, age, turnaround & revenue (over a fixed period), profession, household income etc... of a given customer.

  1010 23 4500 250 Sales Representative 2100 ...
  1040 37 5700 530 Manager 3500 ...
  ...

- Sepal length and width, Petal length and width of a flower

  5.0 3.3 1.4 0.2 Iris-setosa
  7.0 3.2 4.7 1.4 Iris-versicolor
  6.3 3.3 6.0 2.5 Iris-virginica
  ...

alex@seewald.at / alex.seewald.at

Unstructured Data

No specific preprocessing is needed. Learner-specific preprocessing for attributes X is transparently handled by WEKA; filters to transform Y may need to be applied manually. Remember: a learner finds f: X → Y

Some learning algorithms only accept quantitative/numerical X

- 2 values ⇒ map to one quantitative variable as 0/1 or -1/1
- *n* values ⇒ map to *n* quantitative binary variables, only one of which is on (=1). Also called *1-of-n coding*, *dummy variables*.

Some learning algorithms only accept quantitative/numerical Y

- 2 values ⇒ map as above, and map the prediction Y back via simple treshold (0.5 for 0/1 and 0 for -1/1)
- *n* values ⇒ multiple models have to be learned; e.g. map as above and use one binary variable for each of *n* models. More complex mappings are possible.

Some learning algorithms do not accept quantitative/numerical X or Y

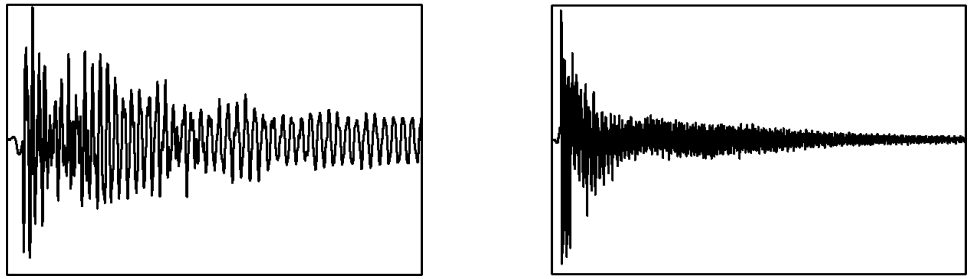
- Discretize values to a qualitative variable with ordinal scale
- Some information is inevitably lost - performance does not necessarily suffer.

alex@seewald.at / alex.seewald.at

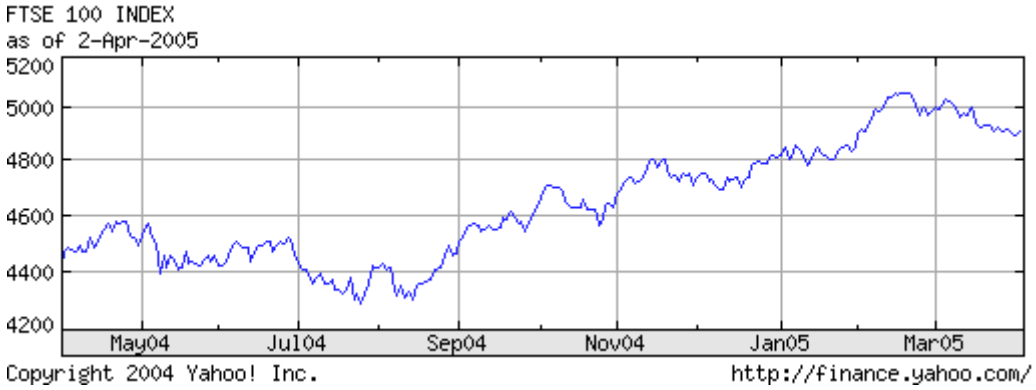
Time Series Data

Each value corresponds to a measurement at a specific time point. Multiple values of each measurement exist.

- Audio Data: music, speech, equiment noise...



- Company share / exchange index: price over time



alex@seewald.at / alex.seewald.at

Time Series Data

A large set of methods can characterize time series data:

- Store all training examples. Use instance-based learning with dynamic time warping (DTW) as distance measure.
- FFT, Wavelets, ..: transform time series into time-independent representation by describing it as a sum of distinct base signals. Specific features should be extracted.
- Windowing: split time series into smaller parts and treat each as independent example.
- Downsampling: stretch/contract time series to constant length and sample the value at a small number of points. Features can be computed on these points (e.g. increasing/decreasing series, constant series..)

...

alex@seewald.at / alex.seewald.at

Text Data

Text is represented as a sequence of characters. Transforming this into useful features via:

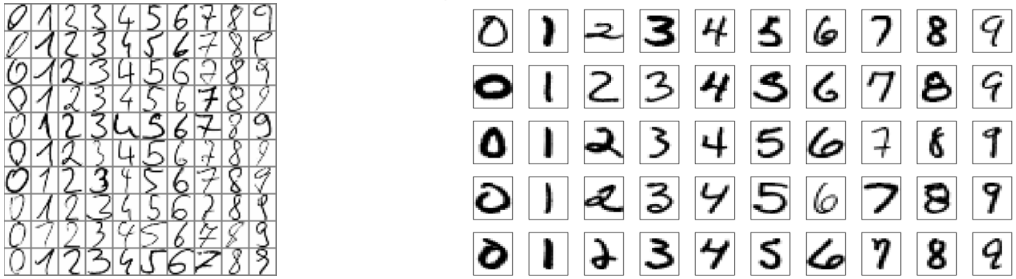
- **Tokenization:** Find the boundaries between words/sentences. Simple for German and English (i.e. whitespace), almost impossible for Chinese, Japanese and some other languages. Extract words from each sentence.
- **Bag-of-words:** one attribute per word which encodes (binary) word occurrence or (numeric) word count. Discards word order, but works surprisingly well in many text classification tasks, and does not need costly corpus.
- **Chunking and Parsing:** Create tree-like structure which describes syntactics and semantics of sentence. Instance-based learning and Hidden Markov Models are mainly used for creating and processing these representations. **Needs a training set corpus which is costly to create.**

alex@seewald.at / alex.seewald.at

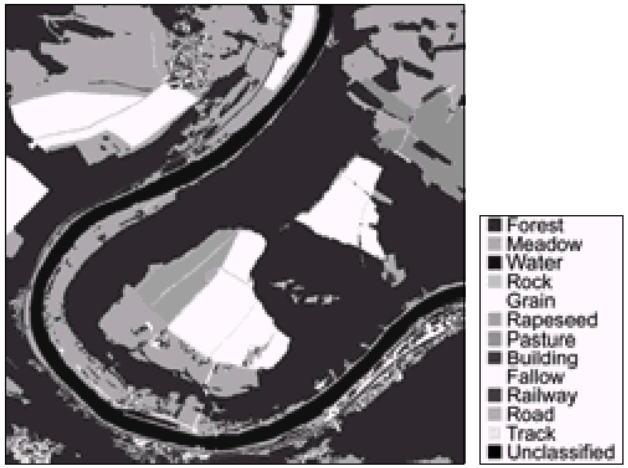
Spatial Data

Each value corresponds to a measurement at a specific spatial location (2D or 3D).

- Handwritten digit recognition based on scanned 2D input



- Satellite images to predict area utilization (segment)



alex@seewald.at / alex.seewald.at

Spatial Data

A large set of methods can characterize spatial data:

- Downsampling: each example is sampled to constant small size (e.g. 8x8 pixels). Each pixel value gives an attribute.
- Histograms: sum of pixel values for each row / column. Features such as position of maximum / minimum values, energy of Wigner-Ville distribution, entropy etc.. are used.
- Pixel-based (e.g. for area utilization): each pixel is treated as an example. Colorspace coordinates or derived attributes are used as features.
- Image Segmentation: algorithms extract segments (i.e. set of connected pixels) with constant color / brightness. Area, point-of-gravity, length of boundary etc.. can be used as features.

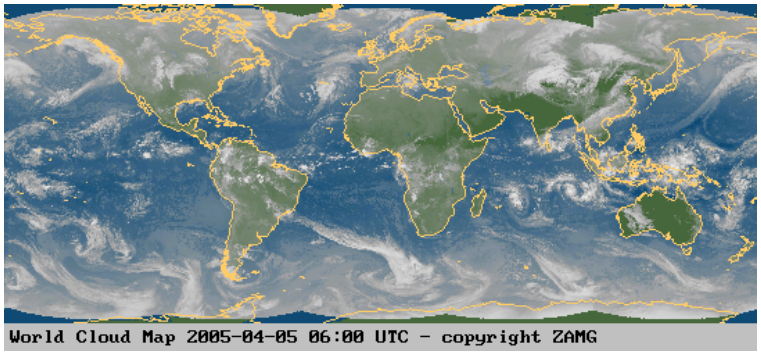
...

alex@seewald.at / alex.seewald.at

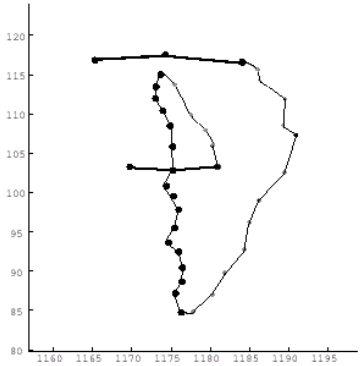
Spatiotemporal Data

Each value corresponds to a measurement in space and time. Often spatial and temporal locations cannot be uniformly sampled. This is called sparse sampling.

- Meterological data for weather prediction



- Handwritten character recognition based on pen traces



Spatiotemporal Data

Uses methods from temporal and spatial preprocessing

(e.g. Downsampling, Windowing, Histograms, FFTs...)

Additional issue: Sparse Sampling

It is not always possible to sample all data points at uniform temporal and spatial resolution.

(e.g. satellite with high-resolution camera moving over earth: high spatial, low temporal resolution; airplane with NO_x sensor: low spatial, high temporal resolution etc..)

So there is a clear trade-off, which means that only a small part of the possible temporal/spatial positions can be measured. The non-measured points usually have to be interpolated from known points.

alex@seewald.at / alex.seewald.at

Preprocessing - Summary

For each type of dataset, specific preprocessing is needed:

- **Unstructured Data:** no specific preprocessing is needed (learner-specific preprocessing is mostly done by WEKA)
- **Temporal Data:** Use instanced-based learning with DTW, and/or characterize each example via FFT/Wavelet, and/or downsample / window...
- **Text Data:** bag-of-words or corpus-based approaches.
- **Spatial Data:** Downsample to fixed size, horizontal/vertical histograms, treat each pixel as separate example, image segmentation...
- **Spatiotemporal Data:** mix temporal and spatial preprocessing. Suffers from sparse sampling (very few of all possible spatial and temporal points can be measured), so missing data needs to be interpolated from given data.

⇒ **Replace spatial/temporal structure with less structure.**

alex@seewald.at / alex.seewald.at

State-of-the-Art Learning Systems

Linear Methods

- Linear Regression
- Logistic Regression
- Support Vector Machines with linear & nonlinear kernels

Non-Linear Methods from Statistics

- NaïveBayes
- Instanced-Based learning

Non-Linear Methods from Machine Learning

- Decision Trees
- Rules Learning (RIPPER)

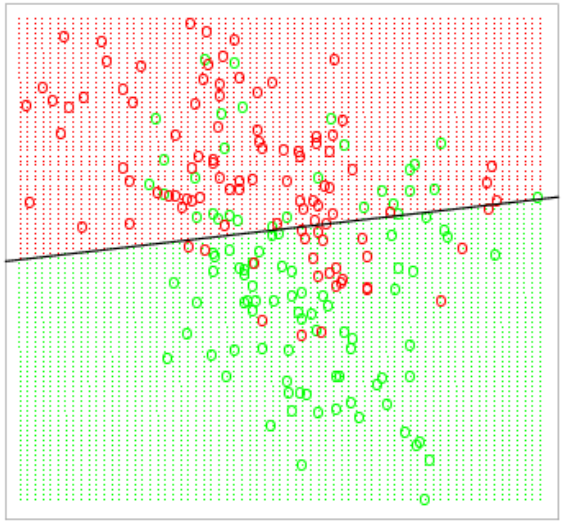
Simple & Fast Baseline Methods

- OneR
- ZeroR

alex@seewald.at / alex.seewald.at

Linear Methods

All linear methods have high bias and low variance. The decision surface which splits the data into positive and negative example is always a hyperplane (in 2D: a line)

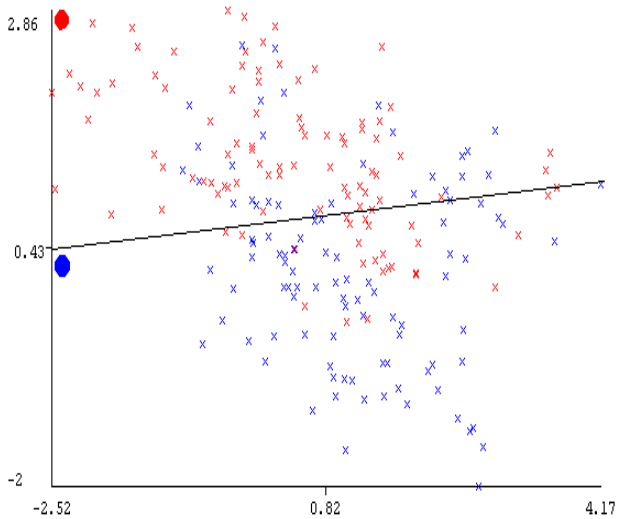


Linear Regression

- Minimizes mean squared error
- Very fast, but susceptible to outliers

Logistic Regression

- Regularization: Estimate class probabilities via logistic function, and ensure they sum to 1. Maximizes log-likelihood of model given training data, i.e. *P(f|TD)*
- Quite fast; less susceptible to outliers

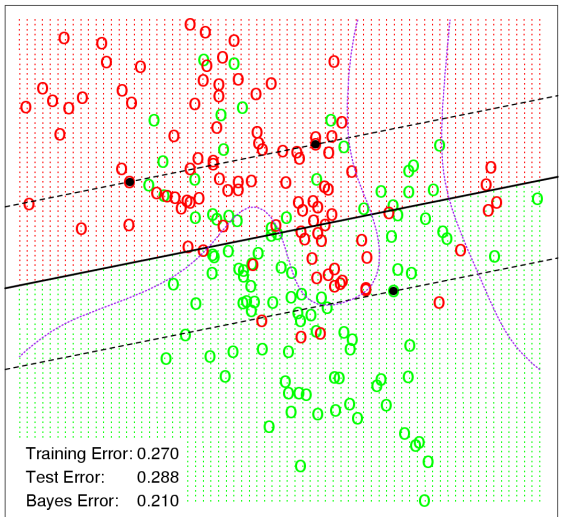


alex@seewald.at / alex.seewald.at

Linear Methods (2)

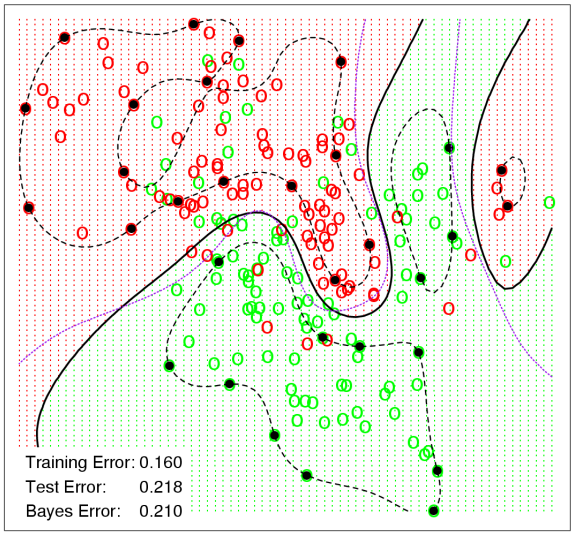
Support Vector Machine with linear kernel

- Regularization by defining a well-posed optimization problem: finding the *maximum margin hyperplane* (i.e. maximizing the margin under constraints on misclassifications)
- Fast; least susceptible to outliers
- Similar to logistic regression for two-class tasks.



Support Vector Machine with nonlinear kernel

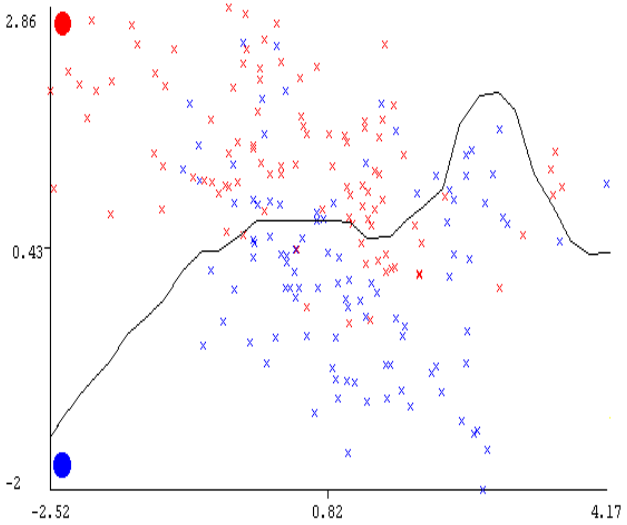
- A linear model in high-dimensional (feature) space defined by a nonlinear kernel. Decision boundary will usually be non-linear in the original feature space.
- Quite fast for polynomial and RBF kernel; slow for complex kernels (e.g. String/Graph kernel)



alex@seewald.at / alex.seewald.at

Non-linear Methods from Statistics

The following non-linear models are low bias and high variance. The decision boundary looks differently in each case and is usually non-linear.

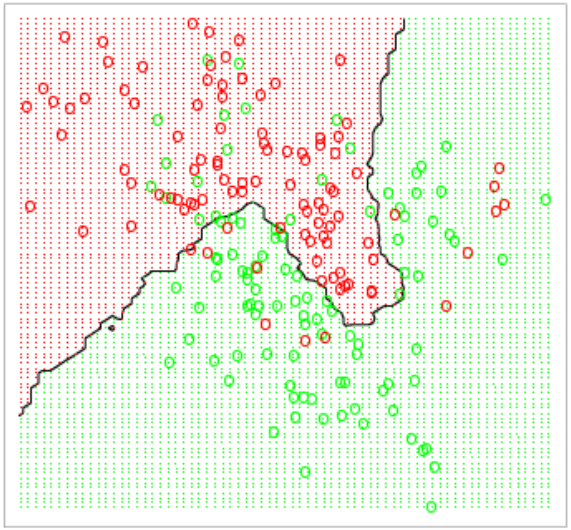


NaïveBayes

- Estimates class probabilities directly from TD. Assumes each attribute contributes independently to the final class probablities.
- Very fast. Less suitable for quantitative attributes

Instance-based learning (nearest neighbor)

- Classify by similarity with training examples.
- *Universal approximator*: Can learn any concept to abitrary precision given sufficient data
- **But** sufficient data size grows exponentially with the number of input dimensions (*curse-of-dim.*)
- Fast training, slow testing (~$O(N^2)$)

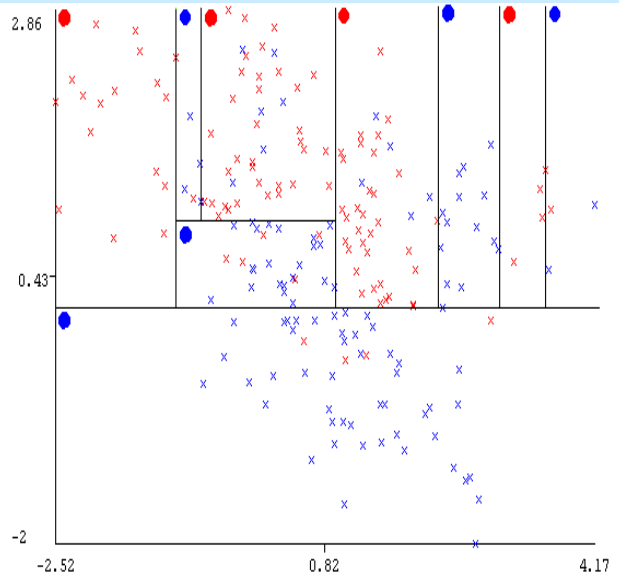


alex@seewald.at / alex.seewald.at

Non-linear Methods from ML

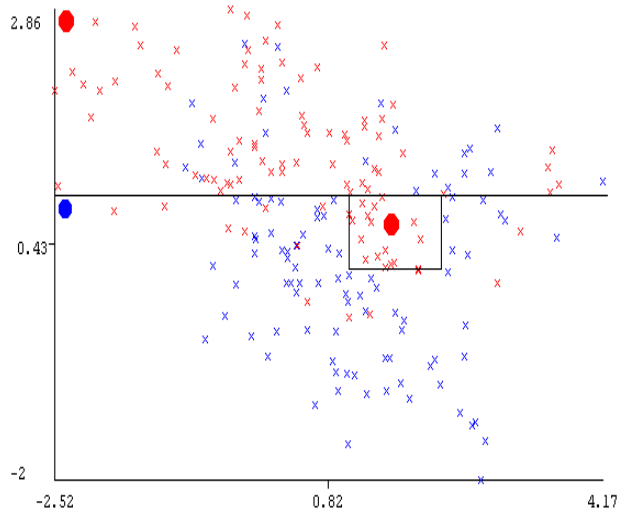
Decision Trees (C4.5)

- *Divide-And-Conquer*: Recursive partitioning of training data by attribute values. Creates decision tree with class values at the leaves.
- Only allows axis-parallel splits
- Fast & easy to understand (if tree is small)



Rule Learning (RIPPER)

- *Separate-And-Conquer*: Successively partition training data by rules = sets of conditions over attribute values.
- Yields compact and modular descriptions = rule sets. Decision boundaries for each rule are still axis-parallel.
- Slower, but even easier to understand since rules can be analyzed separately.



alex@seewald.at / alex.seewald.at

Simple & Fast Methods

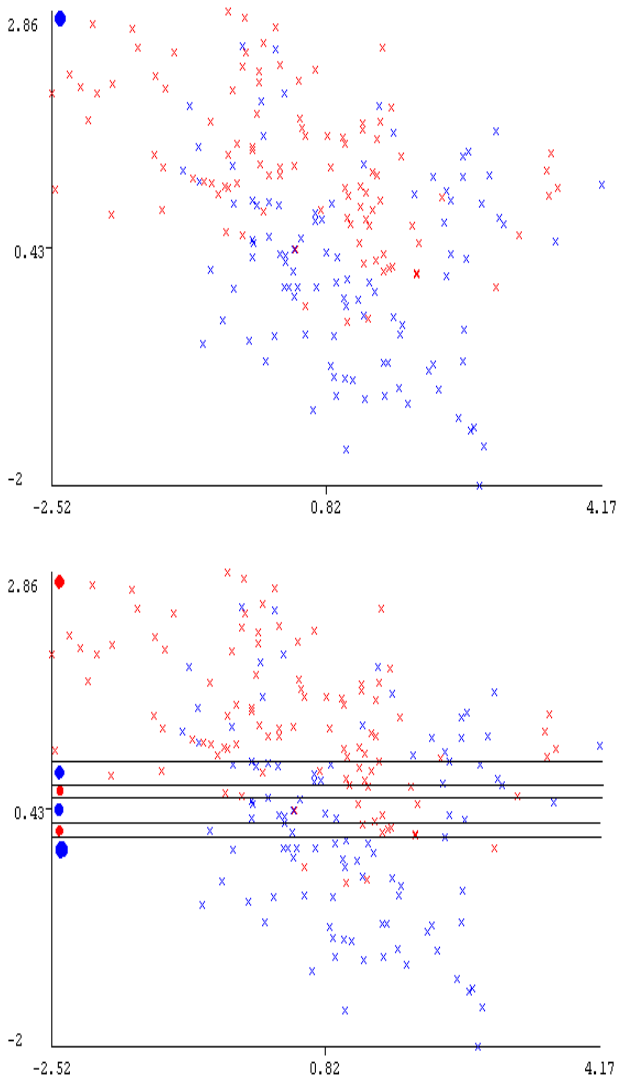
These methods are very fast even for large datasets, and have very high bias & very low variance.

ZeroR (gives Baseline Error/Accuracy)

- Predicts most common class from TD, or arithmetic mean for regression tasks
- Measures complexity of learning problem based on class distributions.

OneR

- Outputs the best rule based on values of a single attribute.
- Measures complexity of learning problem based on class distributions and the distributions of values from the most predictive attribute.
- Less useful for quantitative attributes.



alex@seewald.at / alex.seewald.at

Learning Systems - Summary

Characteristic	Lin. & Log.R	SVM	Naïve Bayes	Inst. Based	Dec. Trees	Rule Learn.
Natural handling of "mixed" data	–	–	o	–	+	+
Handling of MVs	–	–	+	+	+	+
Robustness to outliers	–/o	o	o	+	+	+
Insensitive to monotone transform.	–	–	–	–	+	+
Scalability	+	o/+	o/+	–	+	o
Robstness concerning irrelevant inputs	–	–	o	–	+	+
Interpretability	+	–	o	–	o	+
Predictive Power	o	+	–	+	o	o

alex@seewald.at / alex.seewald.at