Prediction in bioinformatics applications by conformal predictors

Alex Gammerman
(joint work with Ilia Nuoretdinov and Paolo Toccaceli)

Computer Learning Research Centre
Royal Holloway, University of London

International Conference on Pharmaceutical Bioinformatics (ICPB)
Pattaya, Thailand, January 24-26, 2016

Content

Introduction

Problem of prediction: classification and regression - standard subjects in statistics and machine learning.

Classical techniques: small scale, low-dimensional data.

But conceptual and computational difficulties for high-dimensional, high-throughput data.

Modern techniques: Support Vector Machine (Vapnik, 1995, 1998; Vapnik and Chervonenkis, 1974); Kernel Methods, and so on.

Problems

Drawback: lack of useful measures of confidence in their predictions.

Solution: conformal predictors

- To “hedge” predictions: to complement with measures of their accuracy and reliability (“confidence machines” or “conformal predictors”).
- These measures are valid, informative, tailored to the individual object to be predicted.
- Automatic validity under the randomness assumption (the data are i.i.d.): they never overrate the accuracy and reliability of their predictions.
- Any classification or regression algorithm can be transformed into a conformal predictor.

Problem

Given a *training set* of *examples*

$$(x_1, y_1), \ldots, (x_l, y_l),$$

each example (x_i, y_i), $i = 1, \ldots, l$, consisting of an *object* x_i and its *label* y_i; the problem is to predict the label y_{l+1} of a new object x_{l+1}.

Two important special cases:

- *classification* with finite set of labels
  Goal: to produce a prediction $\hat{y}_{l+1}$ that is likely to coincide with the true label y_{l+1}.
- *regression* where the labels are allowed to be any real numbers.
  Goal: to produce a prediction $\hat{y}_{l+1}$ that is likely to be close to the true label y_{l+1}.

General Idea

Classification. Try every possible label Y as a candidate for x_{l+1}'s label and see how well the resulting sequence

$$(x_1, y_1), \ldots, (x_l, y_l), (x_{l+1}, Y)$$

conforms to the randomness assumption (if it does conform to this assumption, we will say that it is "random").

The ideal case is where all Ys but one lead to sequences that are not random.

We can then use the remaining Y as a *confident* (or hedged) **prediction** for y_{l+1}.

Prediction with confidence and credibility

- consider all possible values $Y \in \mathbf{Y}$ for the label y_{l+1};
- find the randomness level detected by t for every possible completion $(x_1, y_1), \ldots, (x_l, y_l), (x_{l+1}, Y)$;
- predict the label Y corresponding to a completion with the largest randomness level detected by t;
- output as the **confidence** in this prediction one minus the second largest randomness level detected by t;
- output as the **credibility** of this prediction the randomness level detected by t of the output prediction Y (i.e., the largest randomness level detected by t over all possible labels).

Intuition:

choose a conventional “significance level”, such as 1%. If the confidence in our prediction is 99% or more and the prediction is wrong, then we assume that a very rare event happened with the probability at most 1%.

Intuitively, *low credibility* means that either the training set is non-random or the test object is not representative of the training set.

Typically credibility will not be low provided the data set was generated independently from the same distribution: the probability that credibility will be less than some threshold ϵ (such as 1%) is less than ϵ.

In summary: we can trust a prediction if:

1. the confidence is close to 100% and
2. the credibility is not low (say, is not less than 5%).

Confidence Predictors

In regression problems choose a range of "confidence levels" $1 - \epsilon$, and for each of them specify a prediction set $\Gamma^\epsilon \subseteq \mathbf{Y}$

We will always consider nested prediction sets: $\Gamma^{\epsilon_1} \subseteq \Gamma^{\epsilon_2}$ when $\epsilon_1 \geq \epsilon_2$.

A confidence predictor is a function that maps each training set, each new object, and each confidence level $1 - \epsilon$ (formally, we allow ϵ to take any value in $(0, 1)$) to the corresponding prediction set Γ^ϵ.

For confidence predictor to be valid probability that the true label will fall outside the prediction set Γ^ϵ should not exceed ϵ, for each ϵ.

Conformal prediction from support vector machines

With each data set

$$(x_1, y_1), \ldots, (x_n, y_n)$$

one associates an optimization problem whose solution produces nonnegative numbers $\alpha_1, \ldots, \alpha_n$ ("Lagrange multipliers").

Each α_i, $i = 1, \ldots, n$, tells us how "strange" an element of the data set the corresponding example (x_i, y_i) is.

The elements with $\alpha_i > 0$ are called support vectors, and the large value of α_i indicates that the corresponding (x_i, y_i) is an outlier.

Taking the completion

$$(x_1, y_1), \ldots, (x_l, y_l), (x_{l+1}, Y)$$

as our data set, we can find the corresponding $\alpha_1, \ldots, \alpha_{l+1}$.
If Y is different from the actual label y_{l+1}, we expect (x_{l+1}, Y) to be an outlier in the completion and so α_{l+1} be large as compared with $\alpha_1, \ldots, \alpha_l$.
A natural way to compare α_{l+1} to the other αs is to look at the ratio

$$p_Y := \frac{|\{i = 1, \ldots, l+1 : \alpha_i \geq \alpha_{l+1}\}|}{l+1},$$

which we call the p-value associated with the possible label Y for x_{l+1}.
In words, the p-value is the proportion of the αs which are at least as large as the last α.

Conformal Prediction from Nearset Neighbours

Many other prediction algorithms can be used as underlying algorithms for hedged prediction. For example, we can use the nearest neighbours technique to associate

$$\alpha_i := \frac{\sum_{j=1}^{k} d_{ij}^{+}}{\sum_{j=1}^{k} d_{ij}^{-}}, \quad i = 1, \ldots, n,$$

with the elements (x_i, y_i) of our data set, where d_{ij}^{+} is the jth shortest distance from x_i to other objects labelled in the same way as x_i, and d_{ij}^{-} is the jth shortest distance from x_i to the objects labelled differently from x_i.

The **intuition** behind this definition: a typical object x_i labelled by, say, y will tend to be surrounded by other objects labelled by y; and if this is the case, the corresponding α_i will be small.

In the untypical case that there are objects whose labels are different from y nearer than objects labelled y, α_i will become larger. Therefore, calculated α reflect the strangeness of examples.

General definition

A nonconformity measure is a function that assigns to every data sequence a sequence of numbers $\alpha_1, \ldots, \alpha_n$, called nonconformity scores, such that: interchange of any two examples (x_i, y_i) and (x_j, y_j)

leads to the interchange of the corresponding nonconformity scores (with all the other nonconformity scores unchanged).

The corresponding conformal predictor maps each data set

$$(x_1, y_1), \ldots, (x_l, y_l),$$

$l = 0, 1, \ldots$, each new example x_{l+1}, and each confidence level $1 - \epsilon \in (0, 1)$, into the prediction set

$$\Gamma^{\epsilon}(x_1, y_1, \ldots, x_l, y_l, x_{l+1}) := \{Y \in \mathbf{Y} : p_Y > \epsilon\},$$

where p_Y are defined by

$$p_Y := \frac{|\{i = 1, \ldots, l+1 : \alpha_i \geq \alpha_{l+1}\}|}{l+1}$$

with $\alpha_1, \ldots, \alpha_{l+1}$ being the nonconformity scores corresponding to the completion

$$(x_1, y_1), \ldots, (x_l, y_l), (x_{l+1}, Y).$$

Associating with each completion its p-value gives a randomness test. Therefore: for each l the probability of the event

$$y_{l+1} \in \Gamma^{\epsilon}\left(x_1, y_1, \ldots, x_l, y_l, x_{l+1}\right)$$

is at least $1 - \epsilon$.
In the case of classification we can summarize the prediction sets Γ^{ϵ} by two numbers: the confidence

$$\sup\left\{1 - \epsilon : \left|\Gamma^{\epsilon}\right| \leq 1\right\}$$

and the credibility

$$\inf\left\{\epsilon : \left|\Gamma^{\epsilon}\right| = 0\right\}.$$

Theoretical Analysis

From the experimental results it is tempting to guess that the probability of error at each step of the on-line protocol is ϵ and that errors are made independently at different steps. This is not literally true.
If the p-values are redefined as

$$p_Y := \frac{|\{i\,\alpha_i > \alpha_{l+1}\}| + \eta\,|\{i\,\alpha_i = \alpha_{l+1}\}|}{l+1},$$

where $i \in \{1, \ldots, l+1\}$ (i ranges over $\{1, \ldots, l+1\}$ and $\eta \in [0,1]$ is generated randomly from the uniform distribution on $[0,1]$, we obtain *smoothed conformal predictors.*

Theorem

For any smoothed conformal predictor working in the on-line prediction protocol and any confidence level $1-\epsilon$*, the random variables* $\mathrm{err}_1^\epsilon, \mathrm{err}_2^\epsilon, \ldots$ *are independent and take value 1 with probability* ϵ*.*

Combining with the strong law of large numbers:

$$\lim_{n\to\infty} \frac{\mathrm{Err}_n^\epsilon}{n} = \epsilon$$

holds with probability one for smoothed conformal predictors. (They are "well calibrated".)

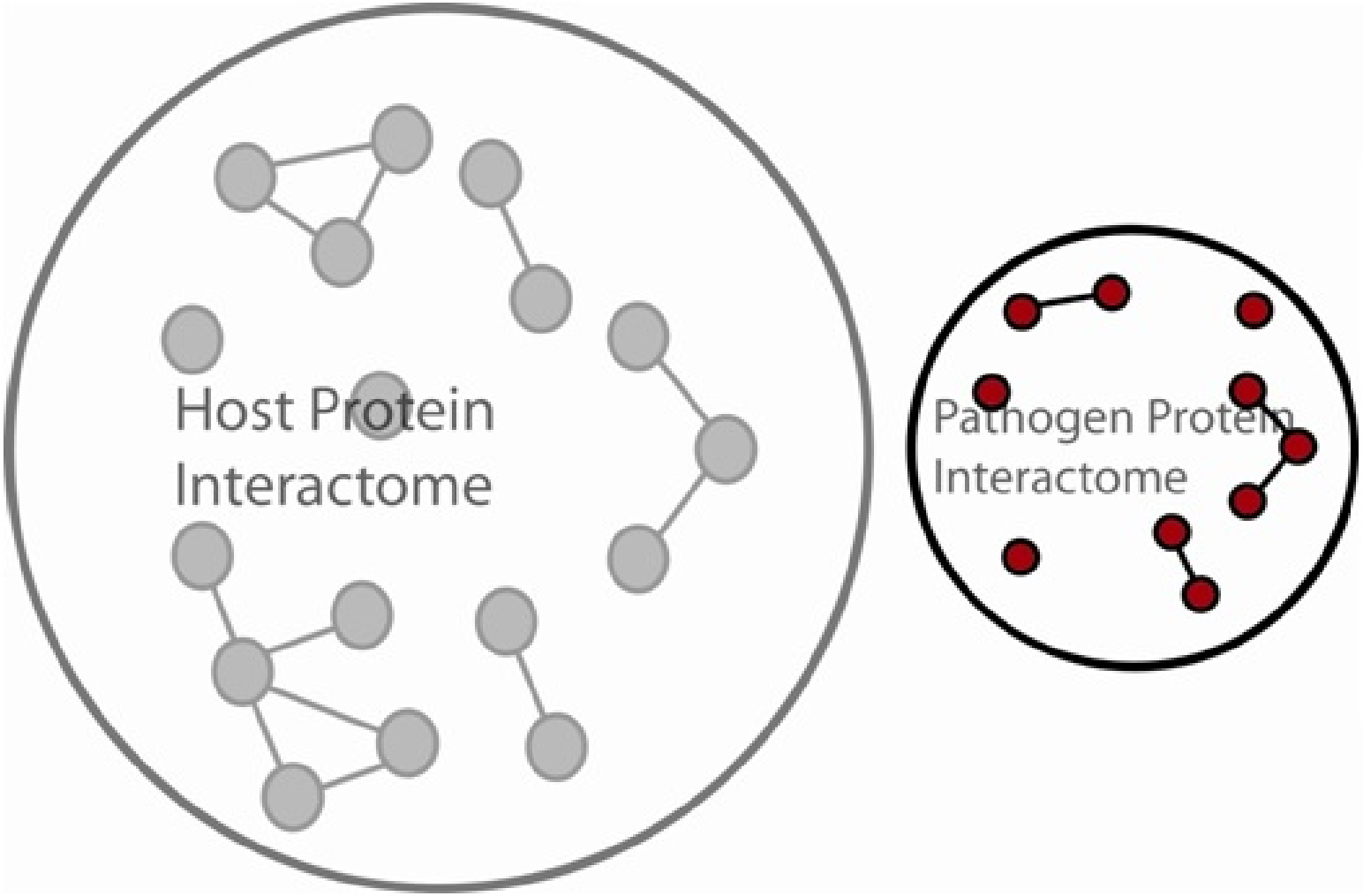
Protein-Protein Interactions

I.Nouretdinov, A.Gammerman, Y.Qi and J.Klein-Seetharaman. Determining confidence of predicted interactions between HIV-1 and human proteins using conformal method.

Pacific Symp Biocomput. 2012:311-22.

Protein-protein interactions

Fundamental unit of communication



Goal: Identify interactome

Supervised learning

Protein pair is described with a feature vector and a class label:

$$(\vec{x}_i, y) \quad y \in \{\text{'Interact','Not Interact'}\}$$

Each feature summarizes a biological information Given data learn a function that would map feature space into one of the two classes:

$$f : X \rightarrow Y$$

NIAID database:
http://www.ncbi.nlm.nih.gov/RefSeq/HIVInteractions

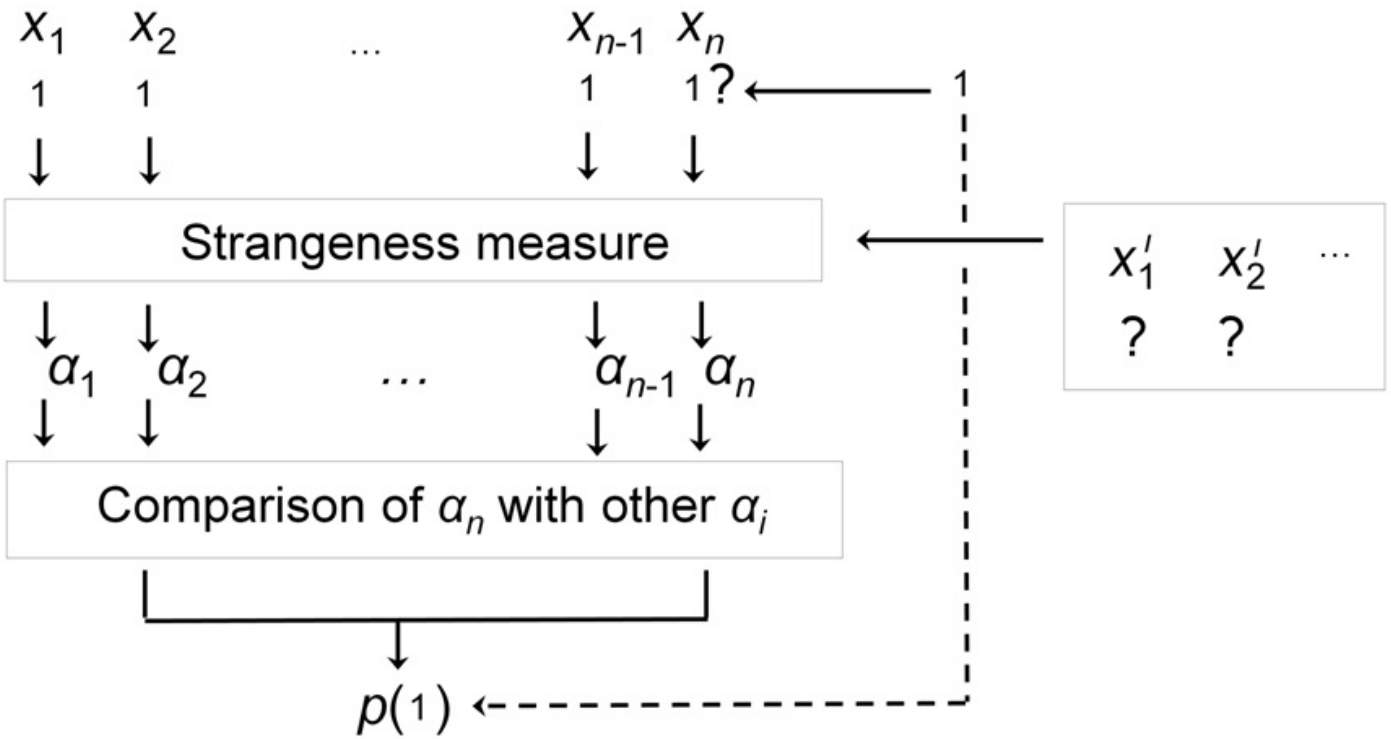
Application of conformal predictor to protein-protein interaction data

Data description:
Protein-protein interactions (17x20873 pairs with 35 attributes).
1063 of them are labelled as already known interactions.
Others are unlabelled: may be interactions or not.

Details

- A large pool of objects (protein-protein interactions);
- A group of them are known to be interacting (training set);
- Aim: is to find new interacting proteins.
- Problem: non-conformity measure is applied to objects instead of (object,label) pairs.

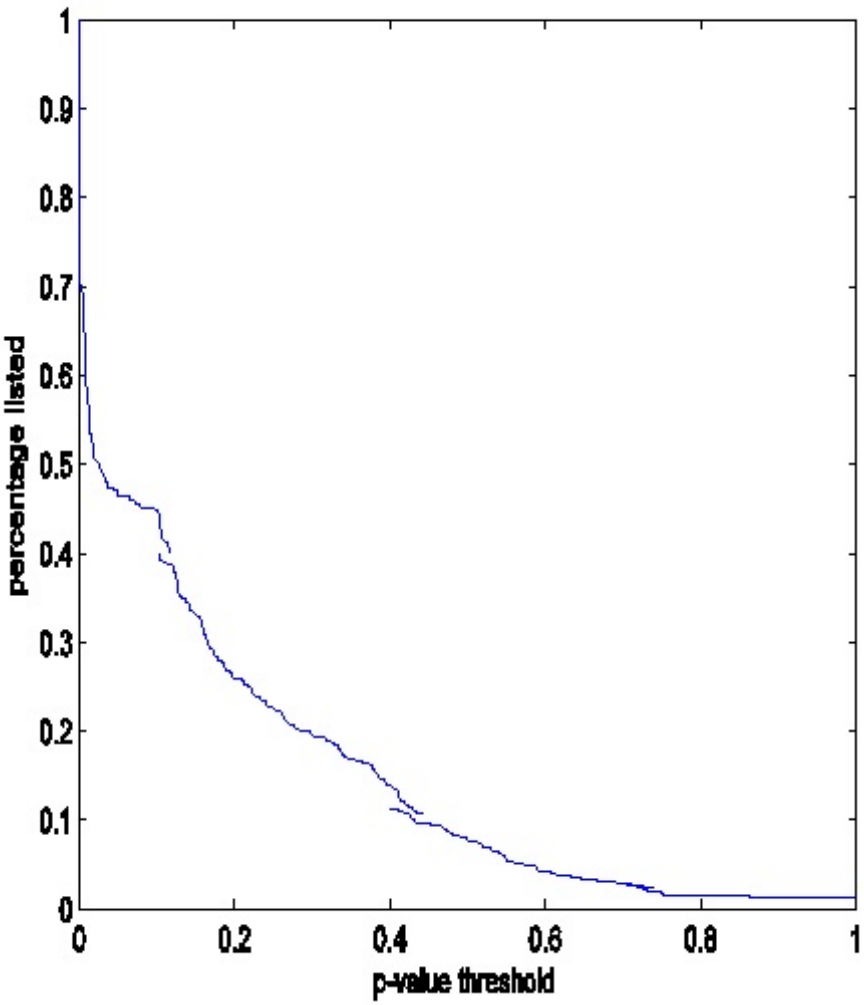
Conformal predictor for protein interaction search



An object x_n is listed as a suspected interaction if p is higher than the significance level ϵ.
Validity guarantees that a real interaction will be suspected with the probability $1 - \epsilon$.
Prediction is more efficient if size of suspected interaction list is smaller.

Results

At a given threshold for p-value, how many objects are listed as suspected interactions?



Results

Example: if confidence level is 90%,
interactions with p-value at least 10% are listed;
a true interaction is listed with 90% chance,
34.2% of all interactions are listed.

confidence level	pred.interactions (amongst known interactions)	pred.interactions (amongst other pairs)
0%	1/1063(0.1%)	4/353778(0%)
1%	11/1063(1%)	37/353778(0%)
5%	54/1063(5.1%)	241/353778(0.1%)
10%	107/1063(10.1%)	604/353778(0.2%)
20%	213/1063(20%)	2185/353778(0.6%)
30%	320/1063(30.1%)	5446/353778(1.5%)
40%	426/1063(40.1%)	10380/353778(2.9%)
50%	533/1063(50.1%)	18988/353778(5.4%)
60%	639/1063(60.1%)	31412/353778(8.9%)
70%	745/1063(70.1%)	49019/353778(13.9%)
80%	852/1063(80.2%)	72743/353778(20.6%)
90%	957/1063(90%)	121091/353778(34.2%)
95%	1010/1063(95%)	150711/353778(42.6%)
99%	1053/1063(99.1%)	189432/353778(53.5%)
100%	1063/1063(100%)	353778/353778(100%)

Conformal Predictors for Drugs Design

Paolo Toccaceli, Ilia Nouretdinov, Zhiyuan Luo
Vladimir Vovk, Lars Carlsson and Alex Gammerman
Excape WP1 internal report. Conformal Predictors

Aims

- to apply TCP and ICP to strongly imbalanced datasets of compounds in order to select active and non-active compounds;
- to experiment with very large sets of high-dimensional data to find the most accurate and computationally efficient algorithms.

The questions we want to consider are:

- What is the most effective learning strategy when training classifiers on highly imbalanced datasets in a supervised learning?
- What would be the most useful underlying algorithms?
- What kernels could be used?
- What non-conformity measures can we use for efficient and accurate predictions?
- How does the performance of the algorithms vary with the class distribution ratio?

Mondrian Conformal Predictors

There are two main problems The problem of applying CP to strongly imbalanced data set is that the error rate for the class of a smaller size could be disproportionately higher. To overcome this we have applied *Mondrian Conformal Predictors*.

Mondrian Conformal Predictors split all examples (x_n, y_n) into categories $k(n, x_n, y_n)$ and set a separate significance level ϵ_k for each category. Here we shall consider so-called *label-conditional Conformal Predictors* where $k(n, x_n, y_n) = y_n$.

The fundamental advantage of Mondrian Conformal Predictors is that it holds *validity* property: it guarantees that in the long run the p-values assigned to objects of each type k are smaller than ϵ_k with probability at most ϵ_k.

p-values for Mondrian CP

For the label-conditional CP the *p*-values are:

$$p(y) = \frac{|\{i = 1, \ldots, (\ell + 1) : y_i = y, \alpha_i \geq \alpha_{\ell+1}\}|}{|\{i = 1, \ldots, (\ell + 1) : y_i = y\}|}$$

The difference with respect to the earlier definition of *p*-value is that the comparisons are restricted to the α_i associated with training examples with the **same** label as the hypothetical completion.

Mondrian Inductive Conformal Prediction

The Mondrian CP is our method to deal with **imbalanced** set of data. In order to deal with the problem of **large** data we applied Inductive CP.

In the Inductive Conformal Prediction, the training set is divided into a *proper training set* and a *calibration set*. The proper training set is used to train the underlying Machine Learning algorithm and the calibration set is used to compute the α_i on the basis of the trained underlying machine learning algorithm.

The training set is split at index h so that examples with index $i \leq h$ constitute the proper training set and examples with index $i > h$ (and $i \leq \ell$) constitute the calibration set.

$$p(y) = \frac{|\{i = h+1, \ldots, \ell+1 : y_i = y, \alpha_i \geq \alpha_{\ell+1}\}|}{|\{i = h+1, \ldots, \ell+1 : y_i = y\}|}$$

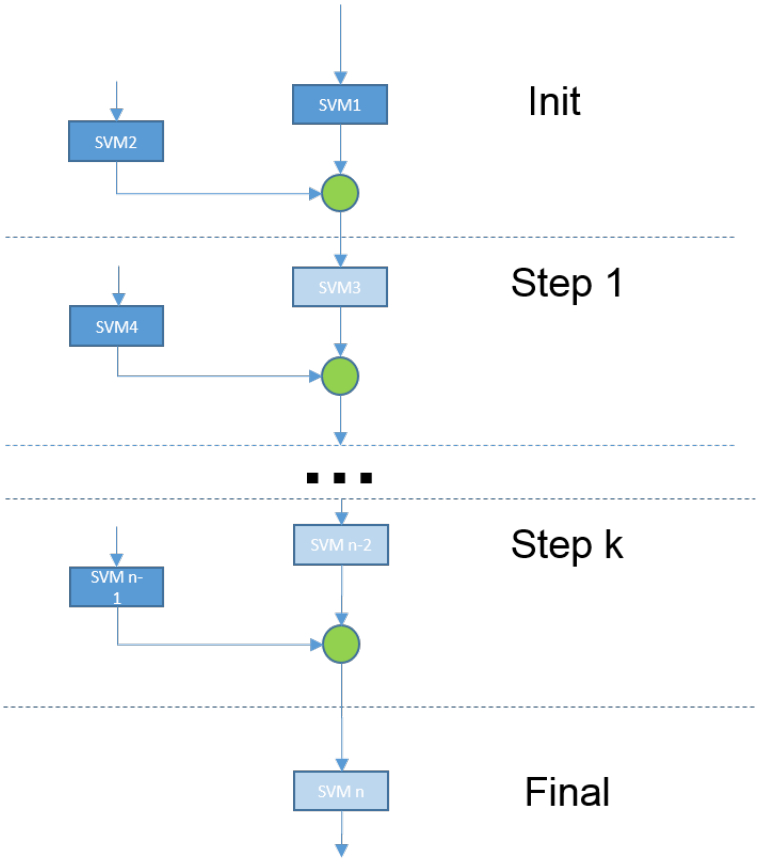
The fundamental advantage of Mondrian Inductive CP for a large and imbalanced data is that it holds **validity** property.

Underlying Algorithms: SVM and Cascade SVM

Several algorithms have been used as underlying algorithms in order to extract the non-conformity measure (NCM). Here we outline our use of SVM and Cascade SVM developed by V.Vapnik.

CascadeSVM decomposes the training over a large set in an appropriate sequence of SVM trainings over smaller sets that form a partition of the training set. The end result is, under certain conditions, the same as the one that would train over the entire set in one go. The CascadeSVM has also the desirable property that is possible to obtain a sub-optimal result by stopping early in the sequence, thereby trading off accuracy against computational effort.

Cascade SVM



Linear Cascade SVM

At each step, the set of Support Vectors from the previous stage is merged with a block of training examples from the partition of the original training set. This is used as training set for an SVM, whose SVs are then fed to the next stage.

Kernels

A key factor for the performance of the SVM classifier is the choice of the kernel.

We experimented with the Tanimoto similarity and then composed it with the Polynomial Kernel and the Gaussian RBF Kernel.

Tanimoto Coefficient	$T(A,B) = \frac{\sum \min(a_i, b_i)}{\sum a_i + \sum b_i - \sum \min(a_i, b_i)}$
Tanimoto with Polynomial	$TP(A,B) = (T(A,B) + 1)^d$
Tanimoto with Gaussian RBF	$TG(A,B) = e^{-\frac{\lvert T(A,A)+T(B,B)-2T(A,B)\rvert}{\gamma}}$

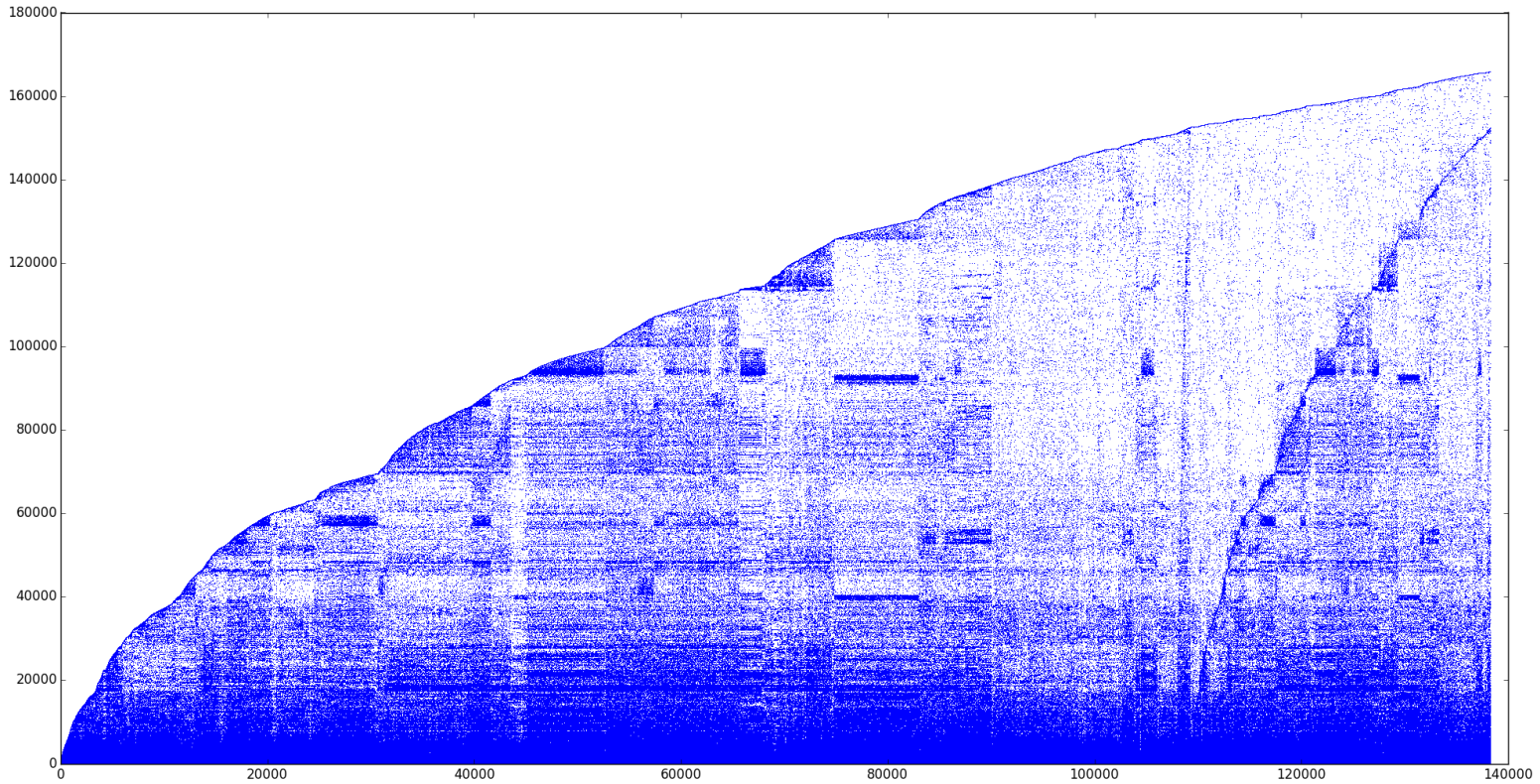
where $A = (a_1, a_2, \ldots, a_p)$, $B = (b_1, b_2, \ldots, b_p)$, $a_i, b_i \in \mathbb{N}^0$.

The data

The data set originates from BioAssay AID 827 in the PubChem public repository. For this experiment, each tested compound is described by a variable number of *signature descriptors*. Each signature corresponds to the number of occurrences of a given labelled subgraph in the molecule graph.

Total number of examples = 138,287
Number of features = 165,786
Number of non-zero entries = 7,711,571
Density of the data set = 0.00034
Active compounds = 1,658
Inactive compounds = 136,629
Unique set of signatures = 137,901

Matrix of the training data



The *x*-axis lists the compounds and the *y*-axis the attributes. A dot corresponds to a non-zero value for a feature

Application

Here are results of some experiments using the Mondrian Inductive Conformal Prediction with 3 different underlying ML algorithms for ICP:

- SVM (Cascade) with Tanimoto+Gaussian RBF kernel
- k Nearest Neighbours with Tanimoto distance
- Naive Bayes (Multinomial)

The split between training, calibration and test sets:
Total number of examples = 138,287
Number of test examples =10,000
Number of training examples = 90,000
Calibration set size = 10,000 (23,520 for SVM)

Non-Conformity Measures

Underlying	**NCM α_i**	**Comment**
SVM	$-y_i d(x_i)$	distance from separating hyperplane (in the "wrong" direction); kernel=[Tani+RBF]
kNN	$\frac{agg_{j \neq i : y_j = y_i} d(x_j, x_i)}{agg_{j \neq i : y_j \neq y_i} d(x_j, x_i)}$	here *agg* is either mean of the k smallest values, or max of the k smallest values
Naïve Bayes	$-\log p(y_i = c \mid x_i)$	p is the posterior probability estimated by NB

Performance of CP region predictor on 10,000 test examples for $\epsilon = 0.01$

Underlying	A as A	NA-A	A-NA	NA-NA	Empty	Unc.
Casc SVM	46	103	0	208	0	9643
3NN mean	36	83	4	785	0	9090
Naïve Bayes	38	85	2	245	0	9630

The conducted experiments included various performance criteria:

- error rate
- number of uncertain predictions
- precision
- recall
- AUC

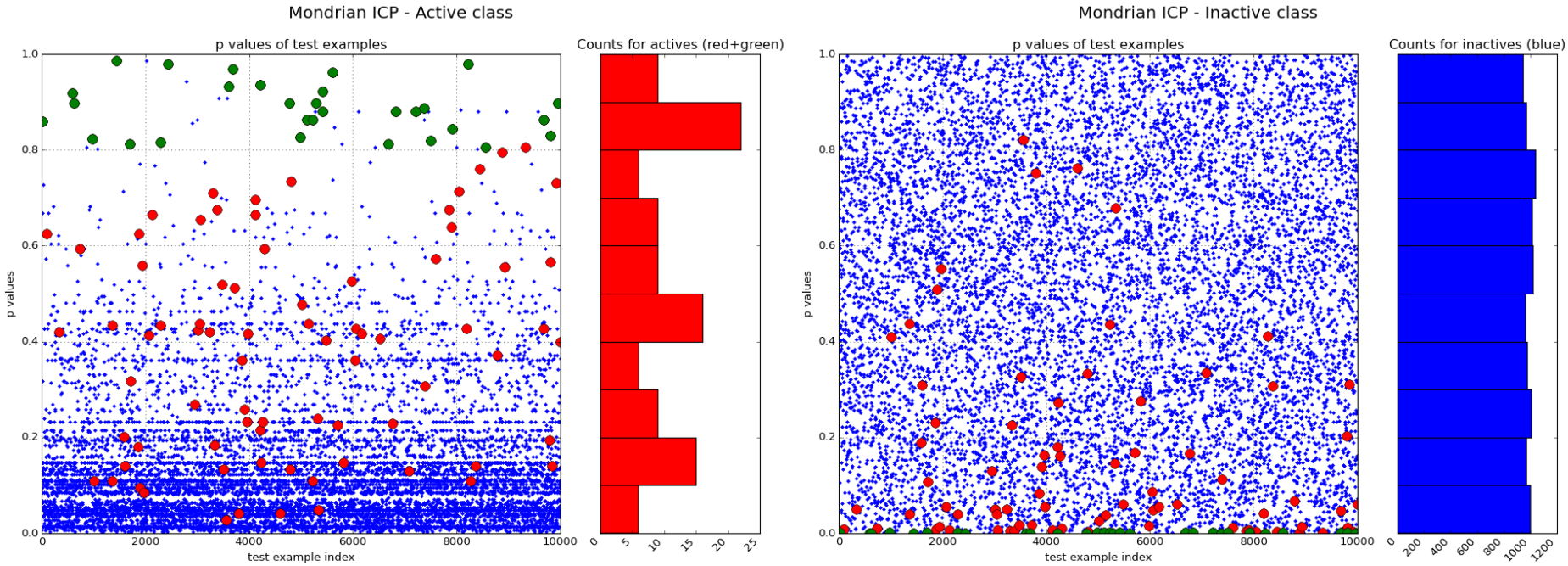
What is important here and the main difference with traditional ML algorithms) is that each of this preformance criteria depend on the **required level of confidence**: the error rate (and the rest of the criteria) will be higher if we demand 99% of confidence and lower under 95% of confidence.

Confusion Matrix given significance (confidence) level ϵ

Region prediction vs. significance level for ICP with CascadeSVM with Tanimoto+RBF kernel:

ϵ	**A as A**	**NA-A**	**A-NA**	**NA-NA**	**Empty**	**Unc.**
0.01	46	103	0	208	0	9643
0.05	64	503	4	3247	0	6182
0.10	75	1001	6	5120	0	3798
0.15	80	1451	18	7285	0	1166
0.20	86	1886	21	7923	84	0
0.25	79	1408	18	7403	1092	0

p-values (for SVM with Tanimoto+RBF)



The blue dots correspond to Inactive compounds. The larger dots are Active compounds, with the green colour denoting those that were correctly classified by the underlying SVM. Ideally, on the first plot the red and green dots would be uniformly distributed in $[0, 1]$, whereas the blue dots would be at the bottom; on the second plot the blue dots would be uniformly distributed in $[0, 1]$, whereas the red and green dots would be at the bottom.

Advantages of Conformal Predictors

- can be used in high-dimensional problems where number of features number of examples;
- does not make any additional assumption about the data beyond the iid assumption: the examples are independent, and identically distributed;
- it allows estimation of confidence of the prediction for individual examples;
- it makes valid predictions in the sense that when used in online mode the confidence measures are well-calibrated:
- we can guarantee that given certain confidence, say 99%, the number of errors will not exceed 1% ('hedged' predictions);
- can be used as a region predictor, with a number of possible classes.