Hedging Predictions in Machine Learning

Alex Gammerman and Zhiyuan Luo

zhiyuan@cs.rhul.ac.uk

Computer Learning Research Centre
Dept of Computer Science
Royal Holloway, University of London
Egham, Surrey TW20 0EX, UK

Networks and Data Mining Session II
June 27 – July 11, 2015
Luchon, France

Last edited: July 9, 2015

Royal Holloway, University of London

Opened by Queen Victoria in 1886, it's one of the larger colleges of the University of London ...

Computer Learning Research Centre

Established in January 1998 by a decision of the College's Academic Board.

Goal: to provide a focus for fundamental research, academic leadership, and the development of commercial-industrial applications in the field of machine learning.

http://clrc.rhul.ac.uk

People

- Local members: Kalnishkan, Luo, Vovk (co-director), Watkins, Gammerman (co-director).
- Outside fellows, including several prominent ones, such as: Vapnik and Chervonenkis (the two founders of statistical learning theory), Shafer (co-founder of the Dempster–Shafer theory), Rissanen (inventor of the Minimum Description Length principle), Levin (one of the 3 founders of the theory of NP-completeness, made fundamental contributions to Kolmogorov complexity)
- RAs and PhD students

Directions of research

- Statistical learning theory (Vapnik, Chervonenkis, founders of the field)
- Conformal prediction (Gammerman, Luo, Shafer, Vovk)
- Competitive prediction (Kalnishkan, Shafer, Vovk)
- Computational and mathematical finance (Shafer, Vovk)
- Information-theoretic analysis of evolution (Watkins)
- Reinforcement learning (Watkins, one of the founders of the field)

Hedging predictions in machine learning

Hedge: protect oneself against loss on (a bet or investment) by making balancing or compensating transactions.

- The hedged predictions for the labels of new objects include quantitative measures of their own accuracy and reliability.
- These measures are provably valid under the assumption that the objects and their labels are generated independently from the same probability distribution.
- It becomes possible to control (up to statistical fluctuations) the number of erroneous predictions by selecting a suitable confidence level.
- Conformal predictors developed by Gammerman and Vovk at Royal Holloway, University of London.

Outlines

We will discuss the following topics:

- Introduction to Prediction with confidence
- Conformal Prediction
  - Transductive Conformal Prediction (TCP)
  - On-line TCP
  - Inductive Conformal Predictor (ICP)
  - Mondrian Conformal Predictor (MCP)
- Applications and conclusions

Resources

- V.Vovk, A.Gammerman and G.Shafer. “Algorithmic Learning in a Random World”, Springer, 2005.
- A. Gammerman and V. Vovk. “Hedging Predictions in Machine Learning”. The Computer Journal 2007 50(2):151-163.
- G. Shafer and V. Vovk. “A Tutorial on Conformal Prediction”. Journal of Machine Learning Research 9 (2008) 371-421.
- Tom M. Mitchell, “Machine Learning”, McGraw-Hill, 1997.
- V. Balasubramanian, S.-S. Ho and V. Vovk (eds), “Conformal Prediction for Reliable Machine Learning: Theory, Adaptations and Applications”, Morgan Kaufmann, 2014.

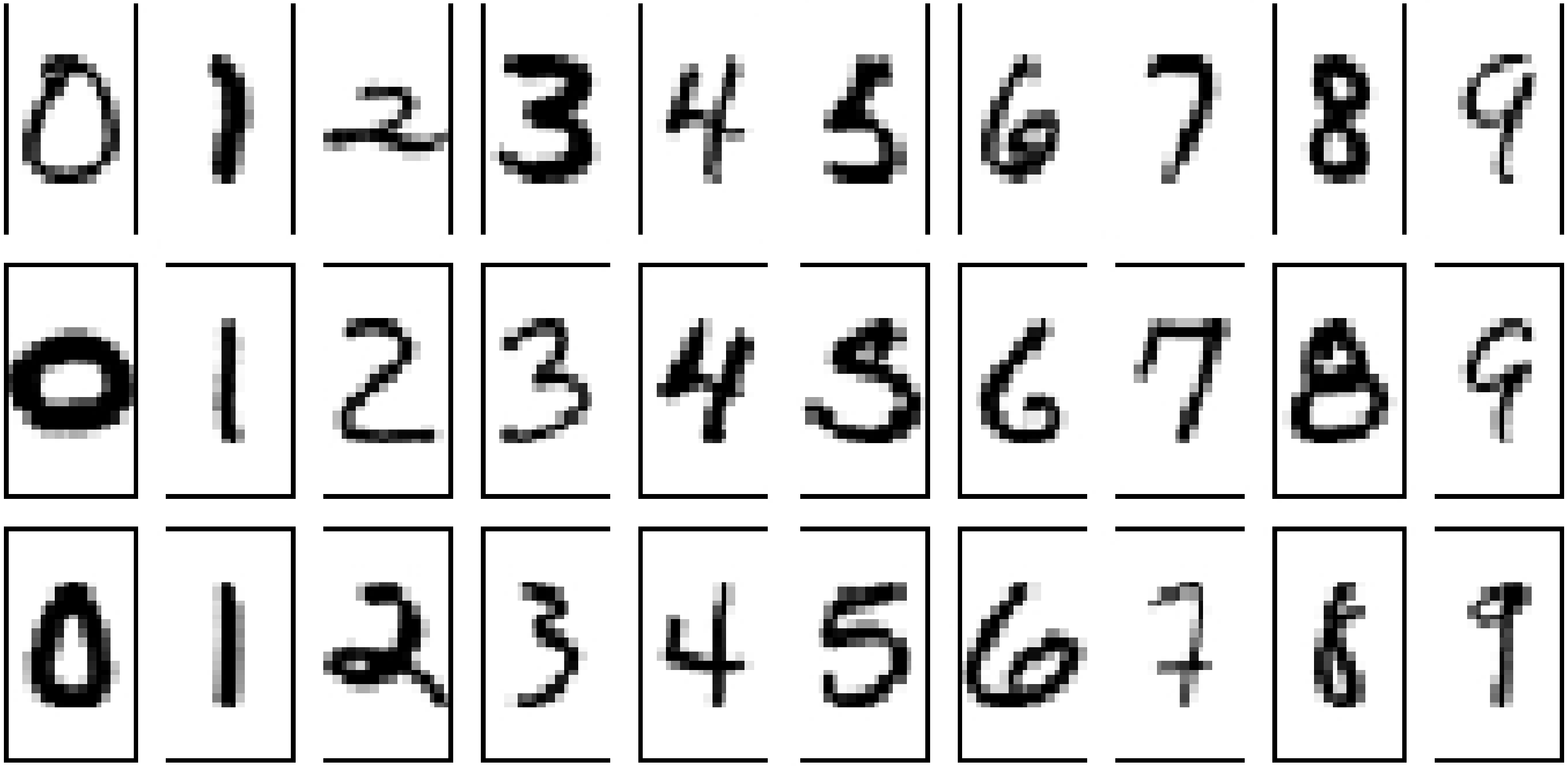
Section: Introduction to Prediction with Confidence

- Machine Learning
- Supervised learning vs unsupervised learning
- Batch vs on-line learning

Why machine learning?

- Data is cheap and abundant but knowledge is expensive and scarce
- Learning is used when:
  - Human expertise does not exist (e.g. navigating on Mars)
  - Humans are unable to explain their expertise (e.g. speech recognition, face recognition)
  - Solution changes in time (e.g. routing on a computer network)
- Build a model that is a good and useful approximation to the data.

Example: hand-written digits



USPS dataset - hand-written digits

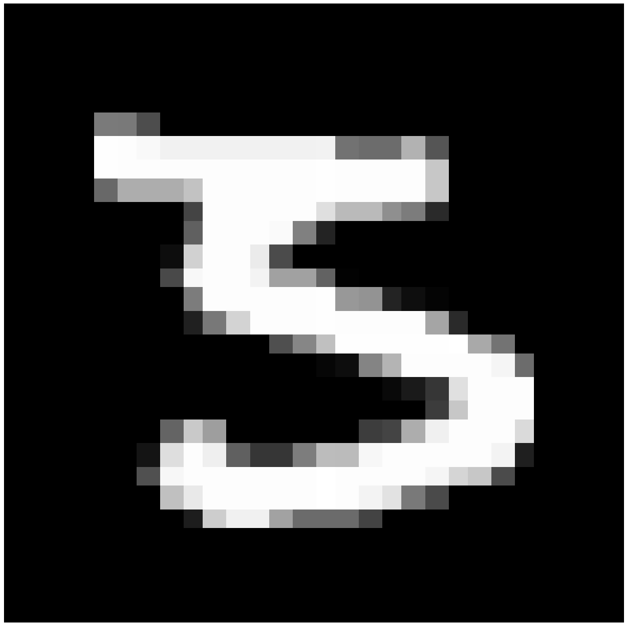
US Postal Service data set: 9298 hand-written digits (7291 training examples and 2007 test examples).

Each example consists of an image (16×16) matrix with entries in the interval (-1,1) that describe the brightness of individual pixels and its label.

For every new hand-written digit we predict a possible label (0 to 9).

Which digit?

3 or 5



Learning methodology

How can a computer perform an "intelligent" task (e.g., recognise hand-written digits)?

1. we can give the computer explicit rules and instructions
   - we may not know the rules ourselves; how would you describe a digit "2"?
   - or the explicit rules may be computationally expensive
2. we can give the computer examples (of handwritten digits) and let it learn the difference
   - this is really a universal method!
   - we just need enough examples and a method of learning

A popular definition

Machine Learning is giving computers the ability to learn without being explicitly programmed.

(Samuel, 1959)

Machine learning in the CS curriculum

The four levels of the Computer Science curriculum:

Level 1: Hardware. Performs simple operations.

Level 2: Software (programs). Makes hardware do what we want.

Level 3: Algorithms: complicated tasks expressed in high-level languages, possibly even in English.

- the author of a program or an algorithm must still foresee and analyse every eventuality

Level 4: Machine learning: the algorithms that can learn and improve themselves.

What is learning?

a working definition: A computer program is said to

- learn from experience E
- with respect to some class of tasks T and performance measure P,
- if its performance at tasks in T, as measured by P, improves with experience E.

Examples (1)

Chess playing problem:

- task T: playing chess (choosing a move in a given position)
- performance measure P: percent of games won against opponents
- training experience E: playing practice games (against opponents or itself)

Examples (2)

The handwritten digits learning problem:

- task T: classifying handwritten digits from 0 to 9
- performance measure P: percentage of digits correctly classified
- training experience E: a database of handwritten digits with given classifications

Example (3)

Medical diagnosis problem:

- task T: making diagnoses among a class of possible diseases
- performance measure P: percentage of correct diagnoses
- training experience E: a set (database) of past patients records with their diagnoses

Supervised learning (1)

The handwritten digits and the medical diagnosis problems have a similar structure

- they deal with objects (or cases or instances or unlabelled examples or input variables) $\mathbf{x}$
- the task is to provide a label (or outcome or response or output variables) y for an object $\mathbf{x}$
- we learn from a set of observations (or labelled examples), which are pairs $(\mathbf{x}, y)$ consisting of an object $\mathbf{x}$ and its label y

Supervised learning (2)

The handwritten digits recognition problem:

- an object is a scanned image of a symbol
- a label belongs to the set $\{0, 1, 2, ..., 9\}$

The problem of supervised learning consists of providing labels for new (test) objects.

Supervised learning (3)

- if the set of possible labels in supervised learning is finite, the problem is called classification (and then labels are sometimes referred to as classes)
  - binary classification: two possible labels; for example: differentiating 0s from the other digits
  - multi-class classification: more than two (but finitely many) possible labels; for example: recognising digits from the set $\{0, 1, 2, ..., 9\}$
- if the set of possible labels in supervised learning is infinite (usually the set $\mathbb{R}$ of real numbers), the problem is called regression
  - example: determining the price of a house from its description

Exploration and exploitation

- supervised learning can proceed according to two protocols: batch or on-line
- in batch learning we are given a training set of observations $(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), ..., (\mathbf{x}_n, y_n)$ and we need to work out labels for the objects from a test set $\mathbf{x}_{n+1}, \mathbf{x}_{n+2}, ..., \mathbf{x}_m$.
- there are two stages:

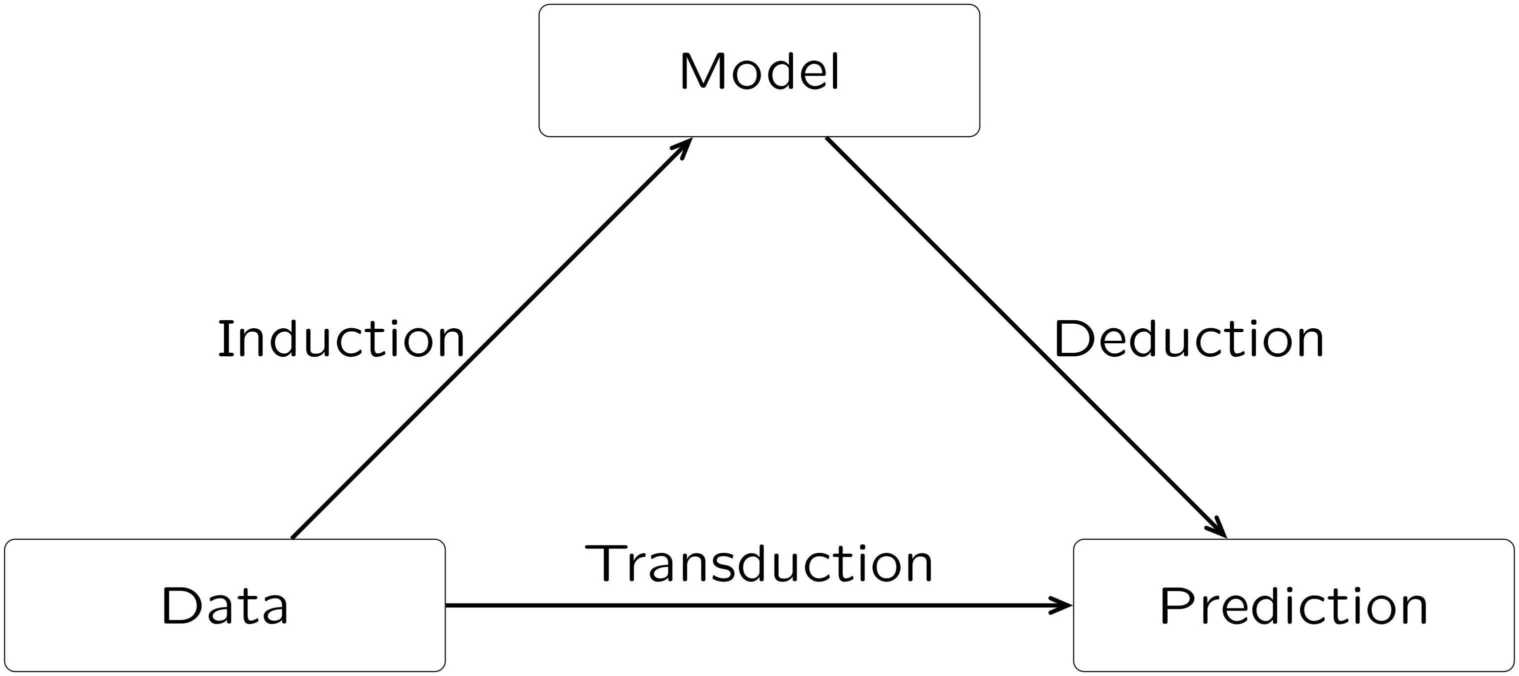
  1. the training (or exploration) stage, when we analyse the training set (and possibly find a hypothesis describing it)
  2. the exploitation stage, when we apply the hypothesis to the test data

Induction vs transduction (1)

- sometimes we do not create a hypothesis
- induction: based on our experience (data set), we arrive at a general hypothesis which tells us something about the unseen data
- transduction: we avoid a general hypothesis and deal with each instance of new data individually
- the difference can be subtle (e.g., computational)

Induction vs transduction (2)

Vapnik, The Nature of Statistical Learning Theory, 1995



On-line learning

- in on-line (supervised) learning we are given observations as follows:
  - we see $\mathbf{x}_1$
  - we work out the predicted label for $\mathbf{x}_1$
  - we see the true label y_1 for $\mathbf{x}_1$
  - we see $\mathbf{x}_2$
  - we work out the predicted label for $\mathbf{x}_2$
  - we see the true label y_2 for $\mathbf{x}_2$
  - etc.
- examples: predicting the weather or stock prices

Unsupervised Learning

Unsupervised learning is concerned with analysing data without labels, e.g., finding out the structure of the data

- for example: clustering, i.e., finding clusters (groups of similar examples) in data

Nearest Neighbours algorithms

- Nearest Neighbour (NN) is a simple algorithm for classification or regression
- suppose we are given a training set $(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), ..., (\mathbf{x}_n, y_n)$
- we need to predict the label for a test object $\mathbf{x}$
- the algorithm:
  - search for the training object that is nearest the test object $\mathbf{x}$
  - predict that the label of the new object is the same as of this nearest training object

Example (1)

- training set:
  - positive objects: (0, 3), (2, 2), (3, 3)
  - negative objects: (-1, 1), (-1,-1), (0, 1)
- test object: (1, 2)
- let us calculate the distance from the new object to each training object

Example (2)

Training object	Label	Euclidean distance
(0, 3)	+1	1.414
(2, 2)	+1	1
(3, 3)	+1	2.236
(-1, 1)	-1	2.236
(-1;-1)	-1	3.506
(0, 1)	-1	1.414

(2, 2) is the nearest object and it is positive

- we predict that our new object is positive too

Transduction

this is our first example of transduction

- we do not formulate any hypothesis; we simply output a prediction on the test object

K-Nearest Neighbours

- K-Nearest Neighbours (KNN) is an enhancement of simple Nearest Neighbours
- the algorithm for classification:
  - find the K nearest neighbours to the new object
  - take a vote between them to decide on the best label for the new object
- the algorithm for regression:
  - find the K nearest neighbours to the new object
  - predict with the average of their labels

Discussion

+ No assumptions and simple methodology

+ Very flexible method

– Potential computational problems

– Problems in high dimensions

Bare prediction algorithms

The learning machines such as KNN and decision trees are "universal": they can be used for solving a wide range of problems. They can be used for:

- hand-written digit recognition
- face recognition
- predicting house prices
- medical diagnosis

The main differences are not in the problems they can be applied to but in their efficiency in coping with those problems.

Motivation

- How good is your prediction $\widehat{y}$?
- How confident are you that the prediction $\widehat{y}$ for a new object is the correct label?
- If the label y is a number, how close do you think the prediction $\widehat{y}$ is to y?

The usual prediction goal: we want new predictions to perform as well as past predictions

Can we ...

1. Allow a user to specify a confidence level or error rate so that a method cannot perform worse than the predefined level or rate before prediction or

2. provide confidence/uncertainty level for all possible outcomes?

Why prediction with confidence

Algorithms predict labels for new examples without saying how reliable these predictions are.

Reliability of method is often given by measuring general accuracy across an independent test set.

- Accuracy is a measurement made following the learning experiment and is not subject to experimental control.
- There is no formal connection between accuracy on the test set and the confidence in a prediction on any particular new and unknown example.
- For prediction, knowing the general rate of error may not be useful, as we are interested primarily in the probability of prediction for each particular case.

Confidence intervals for Gaussian distribution

Given a sample mean μ and variance σ^2, how good an estimate is the sample mean of the true mean?

The computation of a confidence interval (CI) allows us to answer this question quantitively.

Let μ and σ be the sample mean and sample standard deviation computed from the results of a random sample from a normal population with mean μ, then a $100(1-\alpha)\%$ confidence interval for μ is $(\mu - t_{\alpha/2,n-1}\frac{\sigma}{\sqrt{n}}, \mu + t_{\alpha/2,n-1}\frac{\sigma}{\sqrt{n}})$

The t-distribution is used with $n-1$ degrees of freedom for samples of size n, to derive a t-statistic $t_{\alpha/2,n-1}$ for the significance level α.

Bayesian learning

Data is modelled as probability distribution

Probability as confidence

Bayes rule:

$$P(y|x) = \frac{P(x|y)P(y)}{P(x)}$$

Assumptions: The data-generating distribution belongs to a certain parametric family of distributions and the prior distribution for the parameter is known

When prior distributions are not correct, there is no theoretical base for validity of these methods

Statistical learning theory

Statistical learning theory (Vapnik, 1998) including the PAC theory (Valiant, 1984) allows us to estimate with respect to some confidence level the upper bound on the probability of error.

Three main issues:

- Bounds produced may depend on the VC-dimension of a family of algorithms or other numbers that are difficult to attain for methods used in practice.
- The bounds usually become informative when the size of the training set is large.
- The same confidence values ara attached to all examples independent of their individual properties.

Prediction with confidence

- Traditional classification methods give bare predictions. Not knowing the confidence of predictions makes it difficult to measure and control risk of error using a decision rule
- Some measure of confidence for learning algorithm can be derived using the theory of PAC (Probably Approximately Correct)
  - These bounds are often too broad to be useful
- Traditional statistical methods can be used to compute confidence intervals
  - Small sample size means the confidence intervals are often too broad to be useful
- Bayesian methods need strong underlying assumptions

Prediction with confidence goals

- A predictor is valid (or well-calibrated) if its frequency of prediction error does not exceed ε at a chosen confidence level $1 - \varepsilon$ in the long run.
- A predictor is efficient (or perform well) if the prediction set (or region) is as small as possible (tight)

Assumptions

i.i.d. = “independent and identically distributed”: there is a stochastic mechanism which generates the digits (digit=image+classification) independently of each other.

Traditional statistics: parametric families of distributions.

Bags

A bag (also called a multiset) of size $n \in \mathbb{N}$ is a collection of n elements some of which may be identical.

A bag resembles a set in that the order of its elements is not relevant, but it differs from a set in that repetition is allowed.

We write $\lbag z_1, ..., z_n \rbag$ for the bag consisting of elements $z_1, ..., z_n$, some of which may be identical with each other.

Prediction with confidence - our approach

For concreteness: the problem of digit recognition.

The problem is to classify an image which is a 16×16 matrix of pixels; it is known *a priori* that the image represents a hand-written digit, from 0 to 9. We are given a training set containing a large number of classified images. We can confidently classify the new image as, say, 7 if and only if all other classifications are excluded (and 7 is not excluded).

What does it mean that an alternative classification, such as 3, is "excluded"? We regard classification 3 excluded if the training set complemented with the new image classified as 3 contains some feature that makes it highly unlikely under the iid assumption.

Prediction with confidence

We will study the standard machine-learning problem:

- We are given a *training set* of *examples* $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_{n-1}, y_{n-1})$, every example $z_i = (\mathbf{x}_i, y_i)$ consisting of its *object* $\mathbf{x}_i$ and its *label* y_i.
- We are also given a test object $\mathbf{x}_n$; the actual label y_n is withheld from us.
- Our goal is to say something about the actual label y_n assuming that the examples $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)$ were generated from the same distribution independently.

Section: Conformal Prediction

Suppose we want to classify an image; it is known that the image represents either a male or a female face. We are given a training set containing a large number of classified (M/F, or 1/0) images.

We try all possible classifications $k = 0, 1$ of the new image; therefore, we have 2 possible *completions*: both contain the $n - 1$ training examples and the new object (classified as 0 in one completion and as 1 in the other). For every completion we solve the SVM classification problem separating 1s from 0s (male from female faces) obtaining the n Lagrange multipliers α_i for all examples in the completion.

At this point you are only required to know that Lagrange multipliers reflect the strangeness of the examples.

Nonconformity and Conformity (1)

A nonconformity (or strangeness) measure is a way of scoring how different a new example is from a bag of old examples.

Formally, a nonconformity measure is a measurable mapping

$$A : Z^{(*)} \times Z \to \mathbb{R}$$

to each possible bag of old examples and each possible new example, A assigns a numerical score indicating how different the new example is from the old ones.

Given a nonconformity measure A, a sequence $z_1, ..., z_l$ of examples and an example z, we can score how different z is from the bag $\lbag z_1, ..., z_l \rbag$: $A(\lbag z_1, ..., z_l \rbag, z)$.

Nonconformity and Conformity (2)

A conformity measure $B(\Lbag z_1, ..., z_l \Rbag, z)$ measures conformity.

Given a conformity measure B we can define a nonconformity measure A using any strictly decreasing transformation, e.g. $A := -B$ or $A := 1/B$.

When we compare a new example with an average of old examples, we usually first define a distance between the two rather than devise a way to measure their closeness.

For this reason, we emphasize nonconformity rather than conformity.

Nonconformity measure example - 1NN (1)

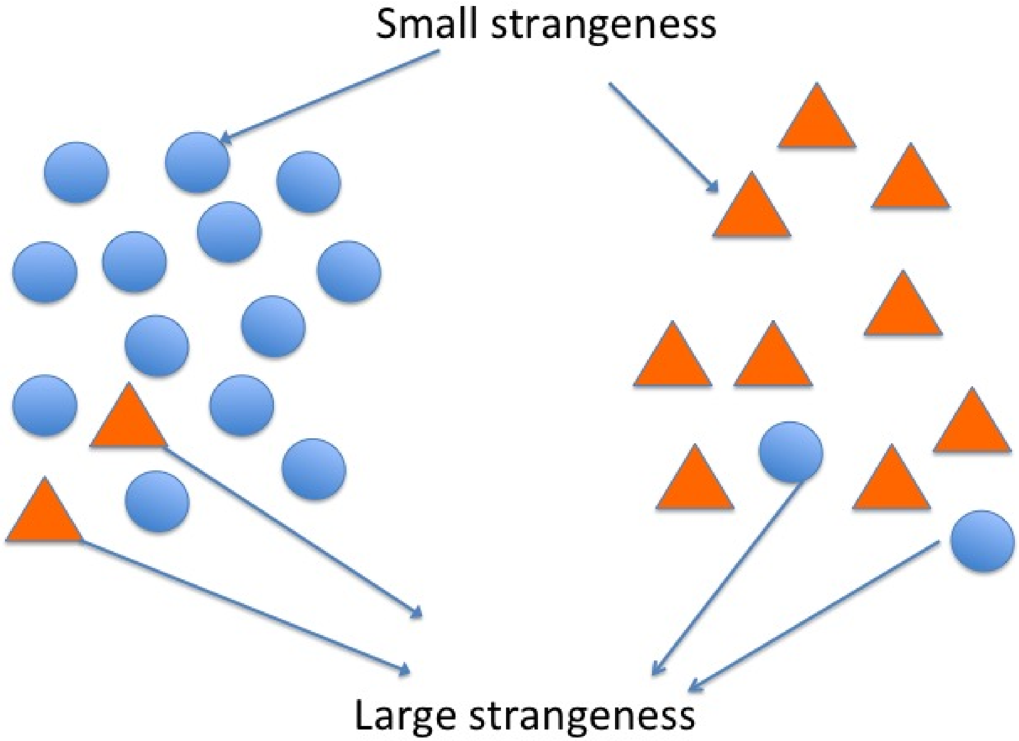
Natural individual conformity measure: αs are defined, in the spirit of the Nearest Neighbour Algorithm, as

$$\alpha_i := \frac{\min_{j \neq i : y_j = y_i} d(\mathbf{x}_i, \mathbf{x}_j)}{\min_{j \neq i : y_j \neq y_i} d(\mathbf{x}_i, \mathbf{x}_j)}$$

where d is the Euclidean distance.

An object is considered strange if it is in the middle of objects labelled in a different way and is far from the objects labelled in the same way.

Nonconformity measure example - 1NN (2)



Nonconformity measure examples for classification (1)

Support vector machine (SVM)

$$\arg\min_{\mathbf{w},b}\max_{\alpha\geq 0}\{\frac{1}{2}||\mathbf{w}||^2-\sum_{i=1}^{n}\alpha_i[y_i(\mathbf{w}\cdot\mathbf{x}_i-b)-1]\}$$

- Lagrange multipliers α

Decision tree

- After a decision tree is constructed, a conformity score $B(\mathbf{x},y)$ of the new example $(\mathbf{x},y)$ as the percentage of examples labeled as y among the training examples whose objects are classified in the same way as $\mathbf{x}$ by the decision tree

Nonconformity measure examples for classification (2)

Neural network

- When fed with an object $\mathbf{x} \in \mathbb{X}$, a neural network outputs a set of numbers $o_y, y \in \mathbb{Y}$, such that o_y reflects the likelihood that y is $\mathbf{x}$'s label.

  $$A(\mathbf{x}, y) = \frac{\sum_{y' \in \mathbb{Y} : y' \neq y} o_{y'}}{o_y + \gamma}$$

  where $\gamma \geq 0$ is a suitably chosen parameter.

Logistic regression

$$A(\mathbf{x}, y) := \begin{cases} 1 + e^{-\hat{\mathbf{w}}\mathbf{x}} & \text{if } y = 1 \\ 1 + e^{\hat{\mathbf{w}}\mathbf{x}} & \text{if } y = 0 \end{cases}$$

Hypothesis testing

A hypothesis is a conjecture about the distribution of some random variables.

- For example, a claim about the value of a parameter of the statistical model.

There are two types of hypotheses:

- The null hypothesis, H_0, is the current belief.
- The alternative hypothesis, H_a, is your belief, it is what you want to show.

Guidelines for hypothesis testing

Hypothesis testing is a proof by contradiction.

1. Assume H_0 is true
2. Use statistical theory to make a statistic (function of the data) that includes H_0. This statistic is called the test statistic.
3. Find the probability that the test statistic would take a value as extreme or more extreme than that actually observed. Think of this as: probability of getting our sample assuming is true.
4. If the probability we calculated in step 3 is high it means that the sample is likely under H_0 and so we have no evidence against . If the probability is low, there are two possibilities:
   - we observed a very unusual event, or
   - our assumption is wrong

p-value

The p-value is the probability, calculated assuming that the null hypothesis is true, of obtaining a value of the test statistic at least as contradictory to H_0 as the value calculated from the available sample.

Important points:

- This probability is calculated assuming that the null hypothesis is true
- The p-value is NOT the probability that H_0 is true, nor is it an error probability

Decision rule based on p-value

Clearly, if the significance level chosen is ε, then

1. Reject H_0 if p-value $\leq \varepsilon$
2. Do not reject H_0 if p-value $> \varepsilon$

Randomness – an example

According to classical probability theory, if we toss a fair coin n times, all sequence $\{0, 1\}^n$ will have the same probability $\frac{1}{2^n}$ of occurring.

We would be much more surprised to see a sequence like 11111111...1 than a sequence like 011010100...1.

The classical approach to probability theory can only give probabilities of different outcomes, but cannot say anything about the randomness of sequence.

Randomness

Assumption: examples are generated independently from the same distribution.

A data sequence is said to be random with respect to a statistical model if a test does not detect any lack of conformity between the two.

Kolmogorov's algorithmic approach to complexity: formalising the notion of a random sequence.

Complexity of a finite string z can be measured by the length of the shortest program for a universal Turing machine that outputs the string z.

Martin-Löf test for randomness

Let P_n be a set of computable probability distributions in a sample space X^n containing elements made up of n data points. A function t: $X^n \to N$, the set of natural numbers N including ∞, is a Martin-Löf test for randomness if

- t is lower semi-computable; and
- for all $n \in N$ and $m \in N$ and $P \in P_n$,

$$P[x \in X^n : t(x) \geq m] \leq 2^{-m}.$$

The connection

Using the Martin-Löf randomness test definition, one can reconstruct the critical regions in the theory of hypothesis. By transform the test t using $f(a) = 2^{-a}$, one gets

Definition: Let P_n be a set of computable probability distributions in a sample space Z^n containing elements made up of n data points. A function $t : Z^n \to (0, 1]$ is a p-value function if for all $n \in N, P \in P_n$ and $r \in (0, 1]$,

$$P[z \in Z^n : t(z) \leq r] \leq r$$

Equivalent to the statistical notion of p-value, a measure on how well the data support or discredit a null hypothesis.

Prediction via hypothesis testing

- A new example $\mathbf{x}$ is assigned a possible label y: $(\mathbf{x}, y)$.
- Hypothesis Test:
  - H_o: The data sequence $S \cup \{(\mathbf{x}, y)\}$ is random in the sense that they are generated independently from the same distribution.
  - H_a: The data sequence $S \cup \{(\mathbf{x}, y)\}$ is not random.

Transductive Conformal Prediction

TCP: a way to define a region predictor from a "bare predictions" algorithm.

Formally: "individual nonconformity measure" $\mapsto$ region predictor.

A family of measurable

$$A_n : \ (z_1, ..., z_n) \mapsto (\alpha_1, ..., \alpha_n)$$

$(n = 1, 2, ...)$ is an individual nonconformity measure if every α_i is determined by the bag $\Lbag z_1, ..., z_n \Rbag$ and z_i.

Conformal prediction (1)

We define the *p-value* associated with a completion to be

$$p_y = \frac{\#\{i : \alpha_i \geq \alpha_n\}}{n}.$$

In words: the p-value is the proportion of αs which are at least as large as the last α and has the value between $1/n$ and 1.

Example: the last α, α_n, is the largest.

- It is small (close to its lower bound $1/n$ for a large n), then the example is very nonconforming (an outlier).

If p-value is large (close to its upper bound 1), then the example is very conforming.

Conformal prediction (2)

Theorem. Every function $t(z_1, ..., z_n) = \frac{\#\{i : \alpha_i \geq \alpha_n\}}{n}$ obtained by a computable individual nonconformity measure α will satisfy equation

$$P[(z_1, ..., z_n : t(z_1, ..., z_n) \leq r] \leq r$$

Proof (Vovk and Gammerman, 1999)

Two ways to make prediction

The property means that p-values can be used as a principled approach to obtain calibrated predictions.

There are different ways to package p-values into predictions.

Two forms have been devised for TCP

- predictions with confidence and credibility
- the region predictor

Predicting with confidence and credibility

- compute the p-values p_0 and p_1 for both completions (with the tentative labels 0 and 1 for the new image, respectively);
- if p_0 is smaller [intuitively, 0 is a stranger label than 1], predict 1 with *confidence* $1 - p_0$ and *credibility* p_1;
- if p_1 is smaller [intuitively, 1 is a stranger label than 0], predict 0 with *confidence* $1 - p_1$ and *credibility* p_0.

In general, we output $\arg\max_y p(y)$ as the prediction and say that $1 - p_2$ (where p_2 is the 2nd largest p-value) is the confidence and that the largest p-value p_1 is the credibility.

Confidence and credibility

The ideal situation (“clean and easy” data set): max(p_0, p_1) close to 1; min(p_0, p_1) close to 0. In this case: both confidence and credibility close to 1.

Intuitive meaning of confidence & credibility. Noisy/small (confidence informative) and clean/large (credibility informative) data sets.

Low credibility implies either the training set is non-random (biased) or the test object is not representative of the training set.

USPS Dataset - Example

Results (in %) obtained using Support Vector Machine (SVM)

0	1	2	3	4	5	6	7	8	9	L	P	Conf	Cred
0.01	0.11	0.01	0.01	0.07	0.01	100	0.01	0.01	0.01	6	6	99.89	100
0.32	0.38	1.07	0.67	1.43	0.67	0.38	0.33	0.73	0.78	6	4	98.93	1.43
0.01	0.27	0.03	0.04	0.18	0.01	0.04	0.01	0.12	100	9	9	99.73	100

If, say, the 1st example were predicted wrongly, this would mean that a rare event (of probability less than 1%) had occurred; therefore, we expect the prediction to be correct.

The credibility of the 2nd example is low (less than 5%). From the confidence we can conclude that the labels other than 4 are excluded at level of 5%, but the label 4 itself is also excluded at the level 5%. This shows that the prediction algorithm was unable to extract from the training set enough information to allow us to confidently classify the example. Unsurprisingly, the prediction for the 2nd example is wrong.

Exercise

The training set is

X: at $(1,0)$ and $(0,1)$

O: at $(-1,0)$, $(0,0)$ and $(1,-1)$

Find the prediction, confidence and credibility using the Nearest Neighbour algorithm with Euclidean distance measure if the new example is:

- $(0.5,-2)$

Region prediction

Given a nonconformity measure, the conformal algorithm produces a prediction region Γ^{ε} for every probability of error ε (significance level).

$$R = \Gamma^{\varepsilon} = \{y \in \mathbb{Y} : p(y) > \varepsilon\}$$

The regions for different ε are nested: when $\varepsilon_1 > \varepsilon_2$, so that $(1 - \varepsilon_1)$ is a lower level of confidence than $1 - \varepsilon_2$, we have $\Gamma^{\varepsilon_1} \subseteq \Gamma^{\varepsilon_2}$.

If Γ^{ε} contains only a single label (the ideal outcome in the case of classification), we may ask how small ε can be made before we must enlarge Γ^{ε} by adding a second label; the corresponding value of $(1 - \varepsilon)$ is the confidence level we assert in the predicted label.

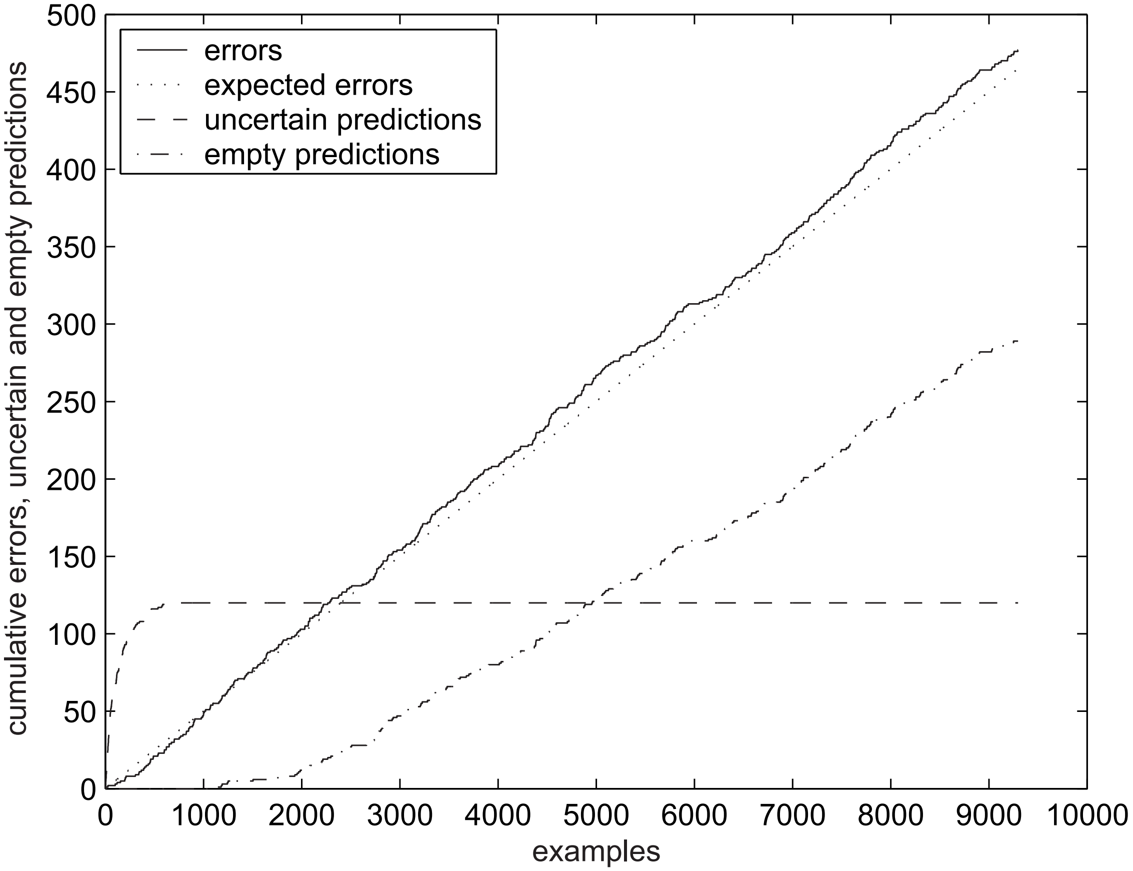
Region prediction

- Empty prediction: $|R|$=0.
- Certain prediction: $|R|$=1.
- Uncertain prediction: $|R| > 1$.

Performance:

- Validity the number of errors made by the system should be $1 - \delta$, if the confidence value is given as δ
- Accuracy the quantity of predictions made correctly.
- Efficiency the size of the region prediction. We want to have small region size, with certain predictions being the most efficient predictions.

Example: region predictions at 95% confidence level for hand-written digits



Lemma

- Lemma 1: The sequences of non-conformal scores for data generated from a source satisfying the exchangeability assumption is exchangeable.
- Lemma 2: p-values from the conformal predictor on data generated from a source satisfying the exchangeability assumption are independent and uniformly distributed on [0, 1].

TCP Calibration theorem

Theorem (Vovk 2002). A transductive conformal predictor is valid in the sense that the probability of error that a correct label

$$y \notin \Gamma^{\varepsilon}(S, \mathbf{x})$$

at confidence level $1 - \varepsilon$ never exceeds ε, with the error at successive prediction trials not independent (conservative), and the error frequency is close to ε in the long run.

Comparison

Key differences between TCP and traditional learning algorithms

Performance measure	Traditional learning algorithm	Conformal predictor (region prediction)
Accuracy	Maximised	Strictly controlled by confidence level
Efficiency	Fixed	Maximized

Example: region prediction

Given: $p_{y=1} = 0.3$, $p_{y=2} = 0.2$, $p_{y=3} = 0.7$, $p_{y=4} = 0.9$, $p_{y=5} = 0.4$, $p_{y=6} = 0.6$, $p_{y=7} = 0.7$, $p_{y=8} = 0.8$, $p_{y=9} = 0.5$, $p_{y=0} = 0.8$.

$\Gamma^{0.85} = \{4\}$ (confidence level 15%)

$\Gamma^{0.75} = \{4, 8, 0\}$ (confidence level 25%)

$\Gamma^{0.65} = \{4, 8, 0, 3, 7\}$ (confidence level 35%)

$\Gamma^{0.05} = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$ (confidence level 95%)

Exercise 1 – region predictions

Given the following p-values (in %)

0	1	2	3	4	5	6	7	8	9	Label
0.01	0.11	0.01	0.01	0.07	0.01	100	0.01	0.01	0.01	6
0.32	0.38	1.07	0.67	1.43	0.67	0.38	0.33	0.73	0.78	6
0.01	0.27	0.03	0.04	0.18	0.01	0.04	0.01	0.12	100	9
0.11	0.23	5.03	0.04	0.18	0.01	0.04	0.01	23.12	0.01	8

What are region predictions at the following confidence level

- 99%
- 95%
- 80%

Exercise 2 – region prediction

The training set is

X: at $(1, 0)$ and $(0, 1)$

O: at $(-1, 0)$, $(0, 0)$ and $(1, -1)$

Find the region prediction at confidence level 95% and 80% respectively, using the Nearest Neighbour algorithm with Euclidean distance measure if the new example is:

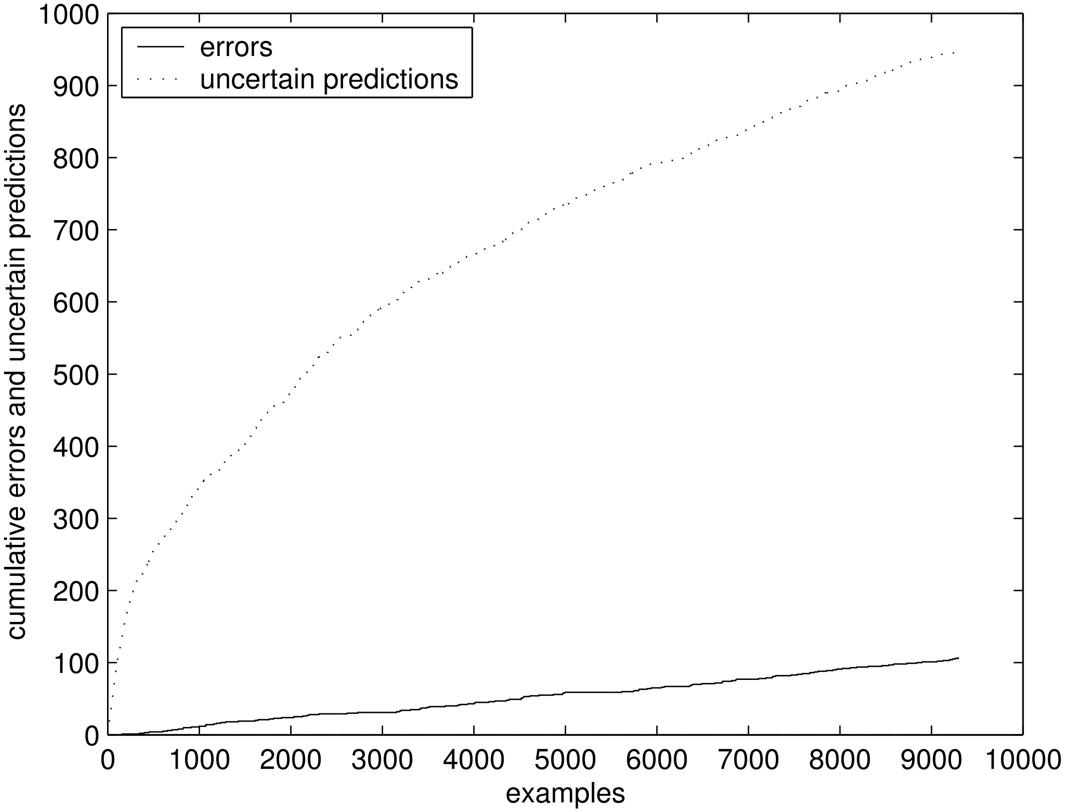
- $(0.5, -2)$

Section: On-line TCP

On-line learning protocol

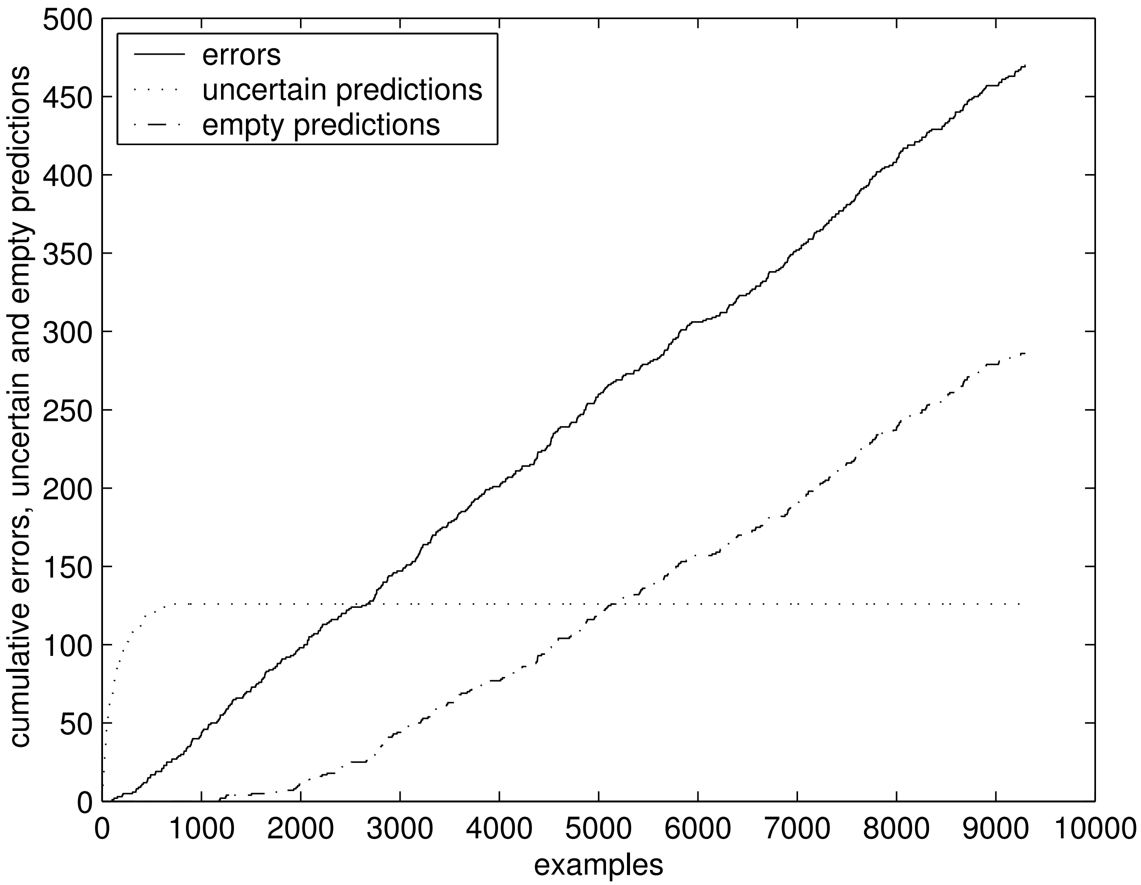
```
$\mathrm{Err}_0 := 0$
$\mathrm{Unc}_0 := 0$
FOR $n = 1, 2, \dots$:
    Nature outputs $\mathbf{x}_n \in \mathbb{X}$
    Learner outputs $\Gamma_n \subseteq \mathbb{Y}$
    Nature outputs $y_n \in \mathbb{Y}$
    $\mathrm{err}_n := \begin{cases} 1 & \text{if } y_n \notin \Gamma_n \\ 0 & \text{otherwise} \end{cases}$
    $\mathrm{Err}_n := \mathrm{Err}_{n-1} + \mathrm{err}_n$
    $\mathrm{unc}_n := \begin{cases} 1 & \text{if } |\Gamma_n| > 1 \\ 0 & \text{otherwise} \end{cases}$
    $\mathrm{Unc}_n := \mathrm{Unc}_{n-1} + \mathrm{unc}_n$
END FOR
```

On-line TCP at confidence level 99%



The solid line shows the cumulative number of errors, dotted the cumulative number of uncertain predictions.

On-line TCP at confidence level is 95%



Evaluation

Since all on-line conformal predictors are valid, the main criterion for comparing different predictors is their efficiency, i.e., the size of output prediction region.

Clearly a smaller prediction region is more informative.

Efficiency is typically measured as the average number of labels in the prediction sets.

Section: Inductive Conformal Prediction (ICP)

Large data set: TCPs can be computationally inefficient.

ICP: sacrifices (in typical cases) predictive accuracy for computational efficiency and provide a decision rule.

The idea of the Inductive Conformal Prediction (ICP):

- Divide the training set into the proper training set and the calibration set.
- Construct a decision rule from the proper training set.

Inductive Conformal Prediction (ICP)

“individual nonconformity measure” $\mapsto$ (“inductive algorithm” , “discrepancy measure”)

$\hat{\mathbb{Y}}$: prediction space (often $\hat{\mathbb{Y}} = \mathbb{Y}$)

Inductive algorithm:

$$D : \Lbag z_1, \dots, z_n \Rbag \mapsto (D_{\Lbag z_1, \dots, z_n \Rbag} : \mathbb{X} \to \hat{\mathbb{Y}})$$

($D_{\Lbag z_1, \dots, z_n \Rbag}$: decision rule).

Discrepancy measure $\Delta : \mathbb{Y} \times \hat{\mathbb{Y}} \to \mathbb{R}$

Inductive conformal prediction

- For every tentative label of the test example do the following:
  - For every example i in the calibration set and for the test example with its tentative label compute α_i, the distance from the decision rule to example i ($i = 1, 2, \ldots, m$; $m - 1$ is the size of the calibration set; the test example has number m).
  - Compute the p-value $\frac{\#\{i=1,2,\ldots,m:\alpha_i \geq \alpha_m\}}{m}$, where, again, $m - 1$ is the size of the calibration set and α_m is the test example's α.
- Compute the predicted label, confidence and credibility or region prediction as before.

An Example

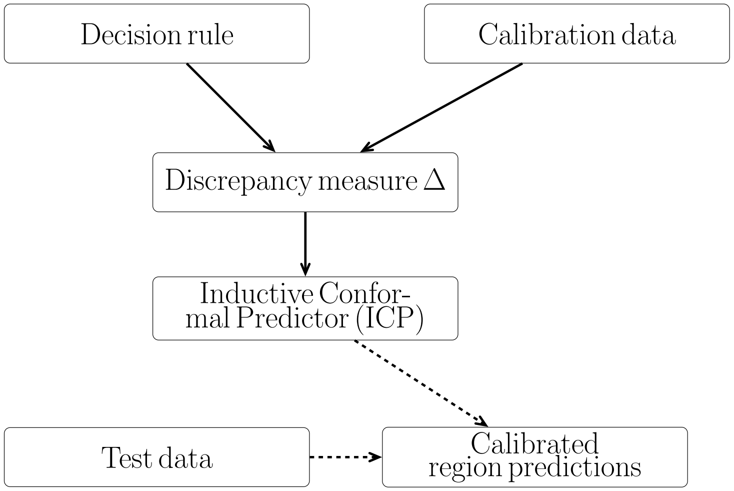
Inductive algorithm: SVM ($D(\mathbf{x}) : \widehat{y} = \mathbf{w} \cdot \mathbf{x} + b$)

Discrepancy measure $\Delta = -y(\mathbf{w} \cdot \mathbf{x} + b)$

- This value is higher for labels which deviate greatly from the decision made by SVM

We define $\alpha_i = \Delta(y_i, D(\mathbf{x}_i))$.

ICP: Flow chart



ICP: Nonconformity measure

D and Δ define an individual nonconformity measure:

$$\alpha_i = \Delta(y_i, D_{\lbag(\mathbf{x}_1,y_1),\ldots,(\mathbf{x}_n,y_n)\rbag}(\mathbf{x}_i))$$

Alternatively

$$\alpha_i = \Delta(y_i, D_{\lbag(\mathbf{x}_1,y_1),\ldots,(\mathbf{x}_{i-1},y_{i-1}),(\mathbf{x}_{i+1},y_{i+1}),\ldots,(\mathbf{x}_n,y_n)\rbag}(\mathbf{x}_i))$$

Inductive algorithms: "proper inductive algorithms" vs "transductive algorithms" (Vapnik, 1995).

- Proper inductive algorithms: $D_{\lbag z_1,\ldots,z_n\rbag}$ can be "computed"; after that, computing $D_{\lbag z_1,\ldots,z_n\rbag}(\mathbf{x})$ for a new $\mathbf{x}$ is fast.
- Transductive algorithms: little can be done before seeing $\mathbf{x}$

ICP algorithm

Fix a finite or infinite sequence $m_1 < m_2 < \ldots$ (called update trials); if finite, set $m_i := \infty$ for $i >$ length. ICP based on D, Δ and $m_1, m_2, \ldots$:

- if $n \le m_1$, $\Gamma(\mathbf{x}_1, y_1, \ldots, \mathbf{x}_{n-1}, y_{n-1}, \mathbf{x}_n, 1-\varepsilon)$ is found using TCP;
- otherwise, find the k such that $m_k < n \le m_{k+1}$ and set

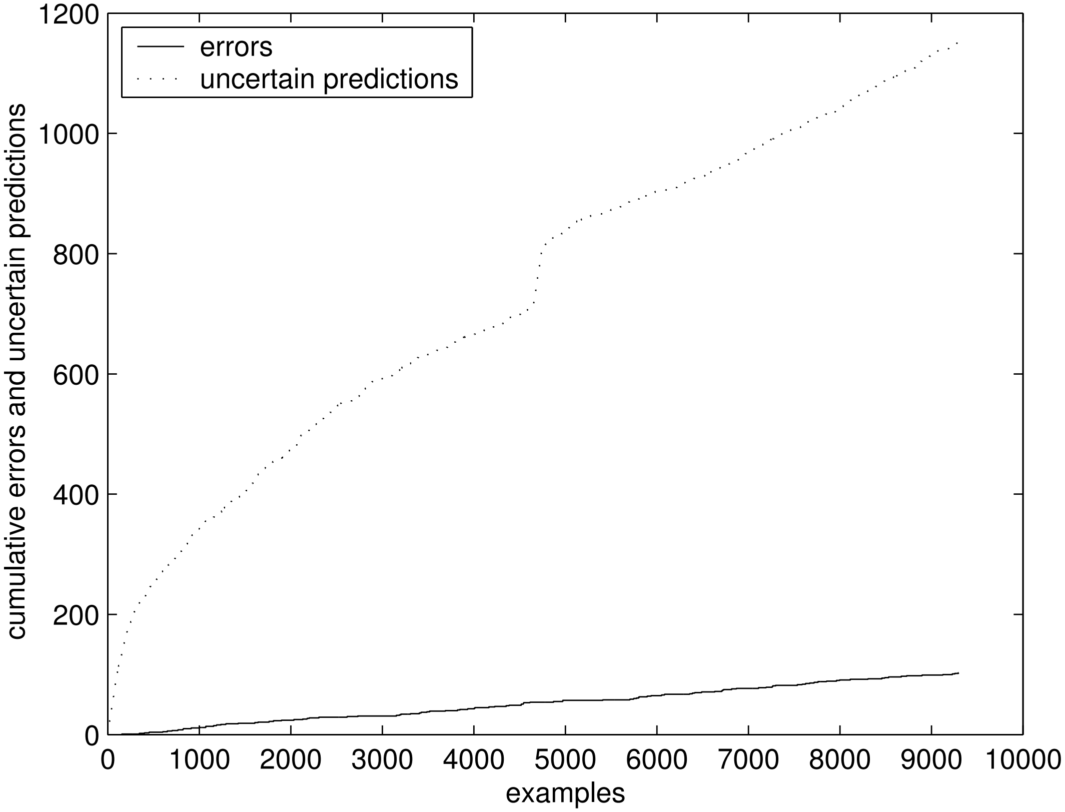
  $$\Gamma(\mathbf{x}_1, y_1, \ldots, \mathbf{x}_{n-1}, y_{n-1}, \mathbf{x}_n, 1-\varepsilon) := \left\{y : \frac{\#\{j = m_k + 1, \ldots, n : \alpha_j \ge \alpha_n\}}{n - m_k} > \varepsilon\right\}$$

  where the αs are defined by

  $$\alpha_j := \Delta(y_j, D_{\Lbag(\mathbf{x}_1,y_1),\ldots,(\mathbf{x}_{m_k},y_{m_k})\Rbag}(\mathbf{x}_j)), \quad j = m_k + 1, \ldots, n-1$$

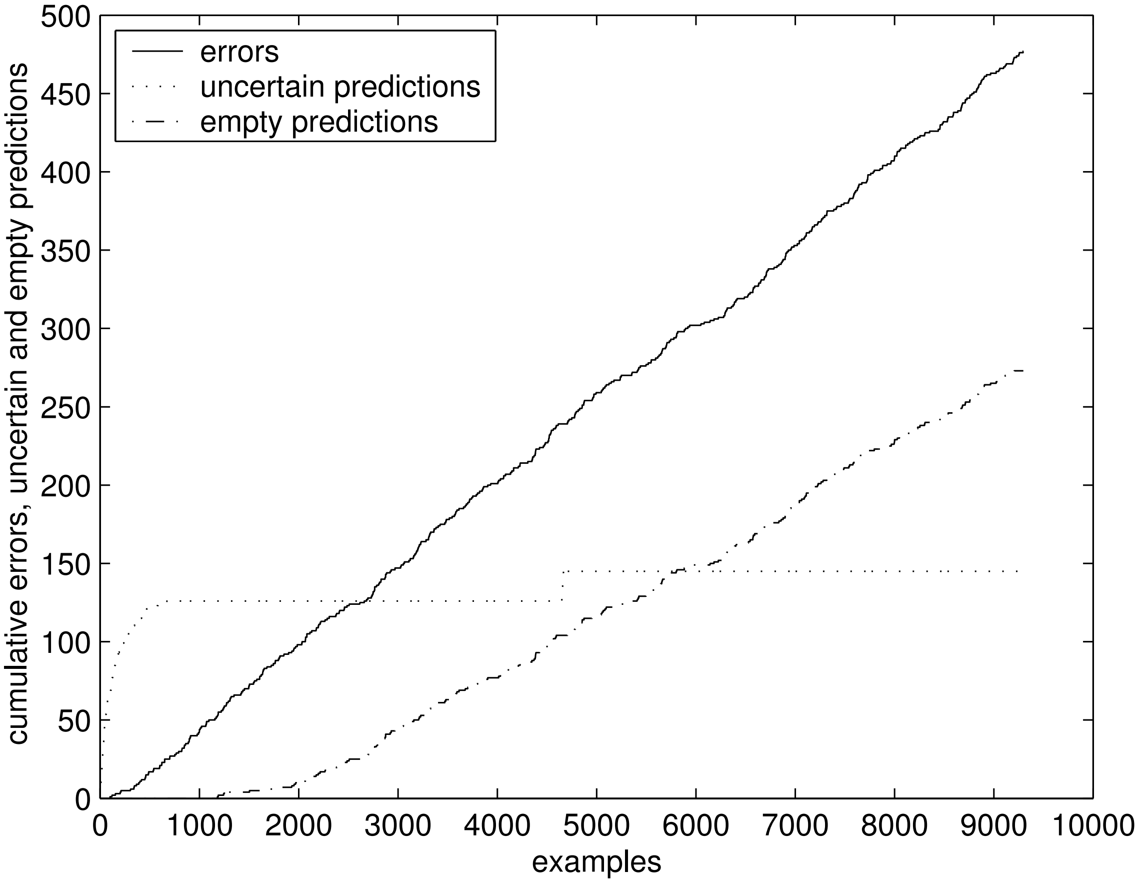
  $$\alpha_n := \Delta(y, D_{\Lbag(\mathbf{x}_1,y_1),\ldots,(\mathbf{x}_{m_k},y_{m_k})\Rbag}(\mathbf{x}_n))$$

ICP at confidence level 99%

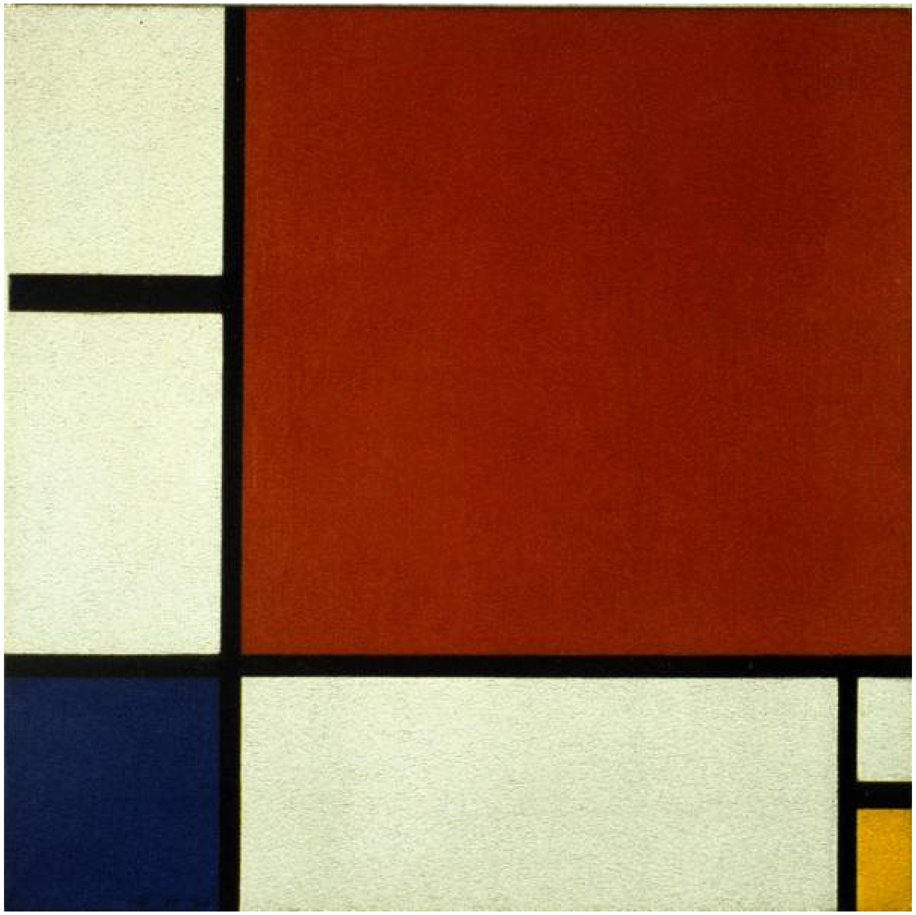


Before (and including) example 4649: TCP; after that the calibration set consists of examples $4649,\ldots,n-1$.

ICP at confidence level 95%



Piet Mondrian



Section: Mondrian conformal prediction

Our starting point is a natural devision of examples into several categories: different categories can correspond to different labels.

Conformal predictors do not guarantee validity within categories (classes).

Mondrian conformal predictors (MCPs) represent a wide class of conformal predictors which is the generalization of TCP and ICP with a new property - validity within categories.

Mondrian conformal predictor

Validity within categories (or conditional validity) is especially relevant in the situation of asymmetric classification, where errors for different categories of examples have different consequences.

In this case, we cannot allow low error rates for some categories to compensate excessive error rates for other categories.

Mondrian conformal predictor

We are given a division of the Cartesian product $\mathbb{N} \times Z$ into categories: a measurable function

$$\kappa : \mathbb{N} \times Z \to K$$

maps each pair (n, z) to its category, where z is an example and n will be the ordinal number of this example in the data sequence $z_1, z_2, \ldots$.

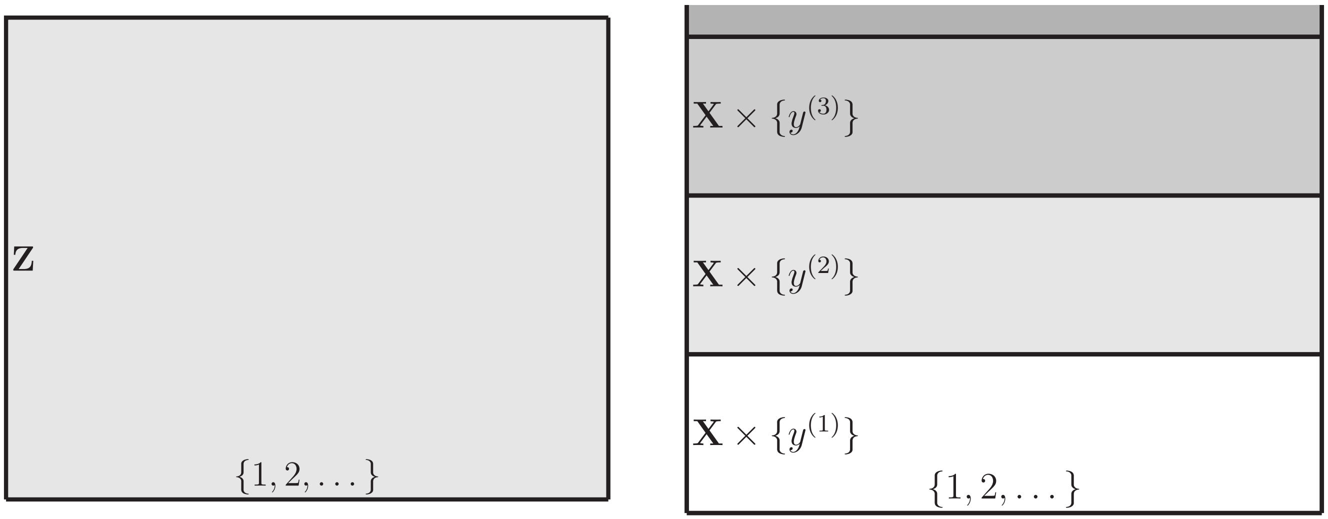
Given a Mondrian taxonomy κ, we can define Mondrian nonconformity measure

$$A_n : K^{n-1} \times (Z^{(*)})^K \times K \times Z \to \mathbb{R}$$

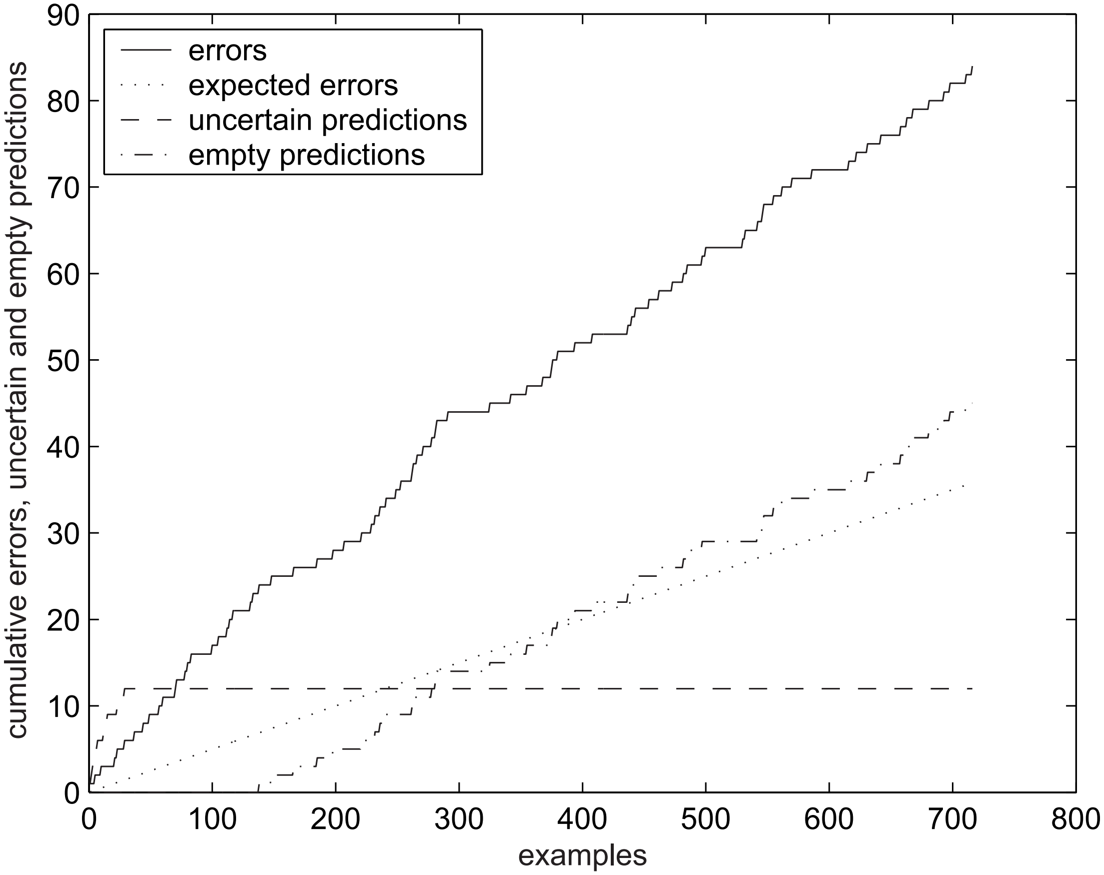
Mondrian taxonomies

left: Conformal prediction taxonomy

right: Label-conditional taxonomy


$\mathbf{Z}$
$\{1, 2, \dots\}$
$\mathbf{X} \times \{y^{(3)}\}$
$\mathbf{X} \times \{y^{(2)}\}$
$\mathbf{X} \times \{y^{(1)}\}$
$\{1, 2, \dots\}$

TCP on USPS data - “5” digit images at 95% confidence level



Mondrian conformal predictor

$$p_n = \frac{|\{i : \kappa_i = \kappa_n \& \alpha_i \geq \alpha_n\}|}{|\{i : \kappa_i = \kappa_n\}|}$$

The randomized MCP:

$$p_n = \frac{|\{i : \kappa_i = \kappa_n \& \alpha_i > \alpha_n\}| + \tau |\{i : \kappa_i = \kappa_n \& \alpha_i = \alpha_n\}|}{|\{i : \kappa_i = \kappa_n\}|}$$

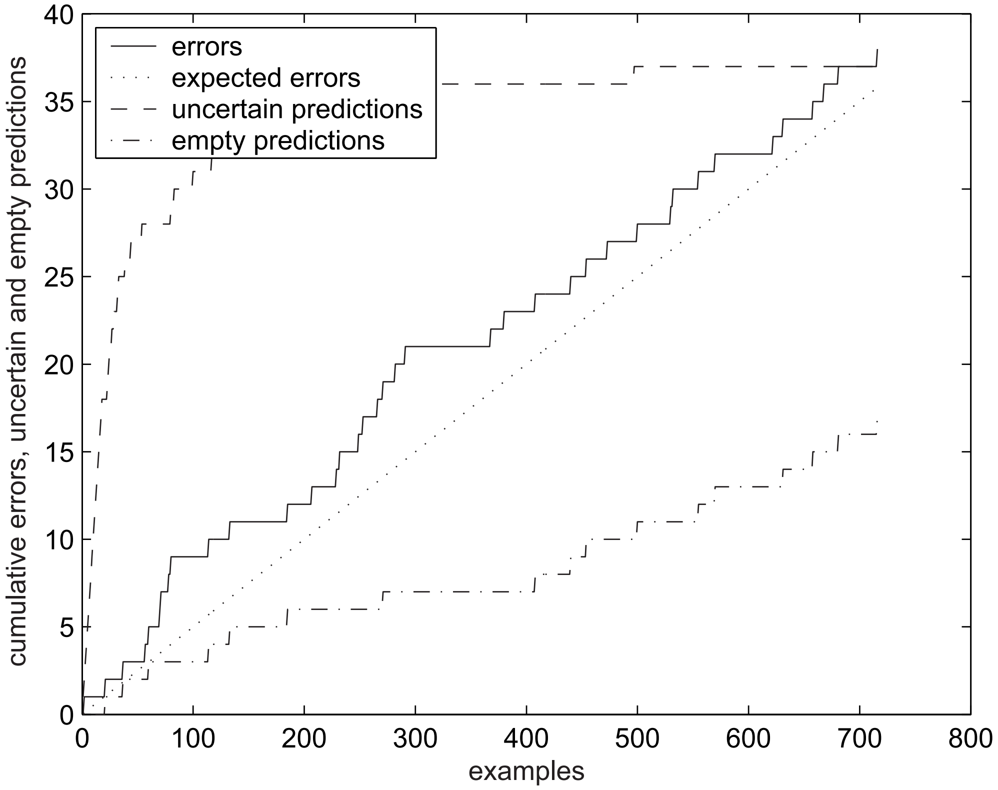
where i ranges over $\{1, ..., n\}$, $\kappa_i = \kappa(i, z_i)$ and $z_i = (\mathbf{x}_i, y_i)$.

USPS dataset

Percentage of errors at the 95% confidence level and the corresponding p-value

class	size	errors	error rate (%)	p-value
0	1553	13	0.84	3.35×10^{-20}
1	1269	12	0.95	1.02×10^{-15}
2	929	52	5.60	0.22
3	824	69	8.37	2.87×10^{-5}
4	852	90	10.56	4.29×10^{-11}
5	716	84	11.73	8.68×10^{-13}
6	834	23	2.76	9.24×10^{-4}
7	792	36	4.55	0.31
8	708	67	9.46	6.80×10^{-7}
9	821	31	3.78	0.06

MCP on USPS data - “5” digit images at 95% confidence level



MCP gives 5.31% of errors.

Section: Applications

Biological/Medical Data

- Cancer prediction (e.g. childhood acute leukaemia, ovarian cancer, breast cancer)
- Chronic gastritis diagnosis
- Abdominal pain diagnosis

  (demo http://turing.cs.rhul.ac.uk/ ~leo/)
- EEG hypoxia recognition
- Cardiac decision support
- Plant promoter prediction
- Depression MRI diagnosis

Childhood acute leukaemia (1)

Affymetrix U133A with 22,283 gene probes

- SVM is used as the linear classifier without kernels.
- The NC strangeness measure is implemented with the Euclidean distance.
- Feature selection is applied with CP using the FDR filter with number of features per class label, $t = 100$.
- The Barts 120 database (94 Acute Lymphoblastic Leukaemia and 26 Acute Myeloid Leukaemia) is used, classifying subtypes ALL or AML. This forms a binary classification problem.
- 10CV learning environment.

Childhood acute leukaemia (2)

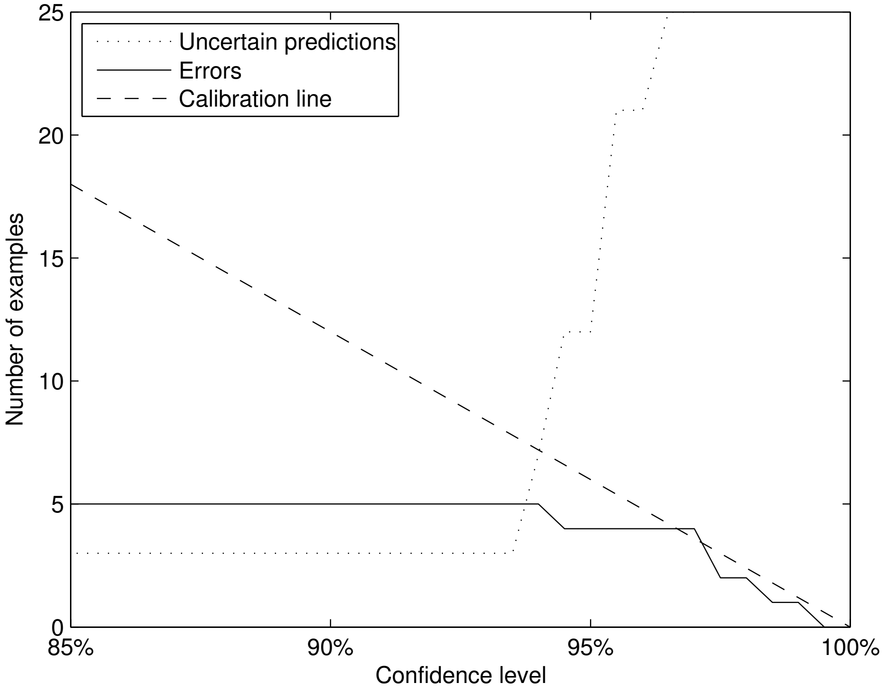
	90%		95%		97.5%	
Method	Acc.	Eff.	Acc.	Eff.	Acc.	Eff.
CP-NC	0.942	0.992	0.967	0.950	0.992	0.900
CP-SVM	0.958	0.950	0.958	0.883	0.983	0.792

Acc. is test accuracy.

Eff. is efficiency: ratio of certain predictions.

Childhood acute leukaemia (3)

Off-line CP-NC with confidence levels 85–100%



Depression MRI diagnosis

- Predicting clinical response of patient with depression who receive anti-depression medication.
- Feature selection using t-test criterion
- SVM conformal prediction

Applications – Image Data

- Head pose estimation
- Open-set face recognition
- Image Classification Problem in the TJ-II Thomson Scattering Charged Coupled Device (TS CCD) Camera

Applications – Time Series

Network Traffic Demand Prediction

- Traffic flow volume prediction for the next time period given a set of previous traffic demand observation in a network.
- Extended to time series data
- Assume no long-term dependence between observations
- Use K-NN for non-conformal scores
- Mean value of the k neighbours' label/value as the predicted label/value.

Conformal prediction framework: extensions and adaptations

- Active learning
- Model selection
- Feature selection
- Anomaly detection
- Change detection
- Quality assessment
- etc ...

Conformal prediction in a nutshell

- Given an error probability ε, together with a method that makes a prediction Y of a label y, it produces a set of labels, typically containing y with probability $1 - \varepsilon$.

- (original) CP works in an online setting in which the labels are predicted successively, each one being revealed before the next is predicted. If successive examples are sampled independently from the same distribution, then the successive predictions will be right $1 - \varepsilon$ of the time, even though they are based on an accumulating data sequence rather than on an independent data set.

Summary

Main advantages of the conformal prediction approach to prediction with confidence:

- New kind of guarantees.
- As compared to the standard theory of machine learning, TCP error bounds are practically useful.
- As compared to statistics and the theory of *Bayesian* learning, we do not assume anything beyond iid.
- There are many interesting real applications of CP.

Acknowledgment

Some of the figures and slides are taken from Prof A. Gammerman's and Prof V. Vovk's lecture notes.

We are grateful for the financial support from UK EPSRC grant EP/K033344/1 "Mining the Network Behaviour of Bots".