Tutorial on Venn-ABERS prediction

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

Computer Learning Research Centre
Royal Holloway, University of London

6th Symposium on Conformal and Probabilistic Prediction with Applications
Stockholm, Sweden, June 13, 2017

Content

1. Introduction
2. Venn Predictors
3. Venn-ABERS predictors

Introduction: Why Probabilistic Prediction

- Machine Learning is primarily concerned with producing **Prediction rules**, i.e. functions mapping "objects" onto predicted "labels", given a training set of ("object","label") examples
- Classical statistical techniques catered for small scale, low-dimensional data.
- The advent of high-dimensional data that does not necessarily follow well-known distributions required new approaches (e.g. Support Vector Machines, Vapnik, 1995, 1998)
- Practitioners often focus on the value of the prediction, but overlook its **uncertainty**.

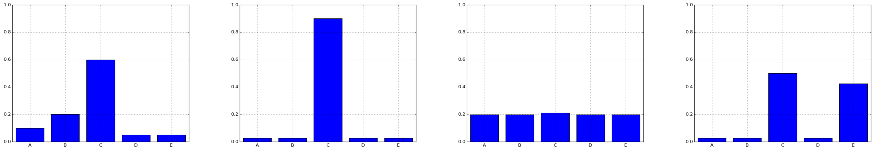
Venn Prediction offers a principled way of assigning a **probability** to predictions.

Motivation example

Suppose that, out of five possible labels A, B, C, D, and E, a classifier outputs C as prediction for an object *x*

Generally, this means that C is the most probable label for *x*, given the training data.

But this in itself does not tell us much!



In the 4 distributions above, C is the most probable label.

But the way we would act based on them might be very different!

Confidence vs. Probability

- Confidence prediction

  > "What is the probability of picking an example that is as or more contrary than the test example to the hypothesis of randomness, given the training set?"

- People generally think of the more direct question:

  > "what is the probability of the label of the test object being L given the training set?"

- **Probabilistic Predictors** answer this question

Probabilistic Prediction

- We want to predict the probability distribution of the label, given the training set and the test object
- We want the predictor to be **valid**
  - Validity is the property of a predictor that outputs probability distributions that perform well against statistical tests based on subsequent observation of the labels.
  - In particular, we are interested in calibration:

$$\mathbb{P}(Y|P) = P$$

- It can be proved that validity cannot be achieved for probabilistic prediction in a general sense.

Venn Predictors

- Venn Predictors are a form of multi-probabilistic predictors for which we can prove **validity** properties
- The impossibility result mentioned earlier is circumvented in two ways:
    - We output multiple probabilities for each label, one of which is the valid one.
    - We restrict the statistical test for validity to calibration, i.e. that probabilities are matched by observed frequencies
      (for example, a particular label should occur in about 25% of the instances in which we give it a probability of 0.25)

Venn Predictors: some detail

- Usual setting:
    - training examples $z_i = (x_i, y_i)$ forming a bag $\Lbag z_1, \dots, z_\ell \Rbag$
    - test object $x_{\ell+1}$
- Venn Predictor output:
    - For each possible label value $y \in \mathbf{Y}$:
        - a probability distribution on $|\mathbf{Y}|$
- NOTE: not a single probability for a label, but a probability distribution on the set of the labels. One of these probabilities is the valid one

Venn Predictors: some detail

- To create a Venn Predictor, one starts by defining a **Venn Taxonomy.**
  - Intuitively, the Venn Taxonomy is used to group examples that we consider sufficiently similar for the purposes of estimating label probabilities.

    More precisely, the taxonomy is a partition of the space $Z^{(\ell)} \times Z$
  - Let denote with $A(\wr z_1, \ldots, z_\ell \int, z)$ the element of the partition that contains $(\wr z_1, \ldots, z_\ell \int, z)$
  - Two examples z_i and z_j belong to the same category iff:

$$A(\wr z_1, \ldots, z_{\ell+1} \int / z_i, z_i) = A(\wr z_1, \ldots, z_{\ell+1} \int / z_j, z_j)$$

  - Example: Taxonomy based on Nearest Neighbour

    two examples are assigned to the same category if their nearest neighbours have the same label

Venn Predictors: some detail

- Given the test object $x_{\ell+1}$
- For every possible value y of the label:
  - We form the bag $\wr z_1, \dots, z_{\ell+1} \int$, with the hypothetical example $z_{\ell+1} = (x_{\ell+1}, y)$
  - Identify the category T to which the example $(x_{\ell+1}, y)$ belongs.
  - The empirical probability distribution p_y of the labels in category T is obtained as:

$$p_y(y') := \frac{|\{(x^*, y^*) \in T : y^* = y'\}|}{|T|}$$

  - In words: for every possible value y' of the label, we calculate the fraction of examples in category T that have label y'

One comment on calibration

- Calibration is desirable, but it is not the only property we seek in predictions
- Example: weather prediction
  - If we asked to predict with what probability it will rain tomorrow, we can simply always respond with the long-term average probability of rain.
  - It would be a calibrated prediction, but it is hardly useful.
  - What we want to have a more **specific** prediction.
- Venn Predictors offer a theoretically-backed framework in which we no longer have to worry about calibration; we can focus only on making predictions more specific.

Venn Predictors: Nearest Neighbour example

- Let’s create a taxonomy with this rule:
  "two examples are assigned to the same category if their nearest neighbours have the same label"

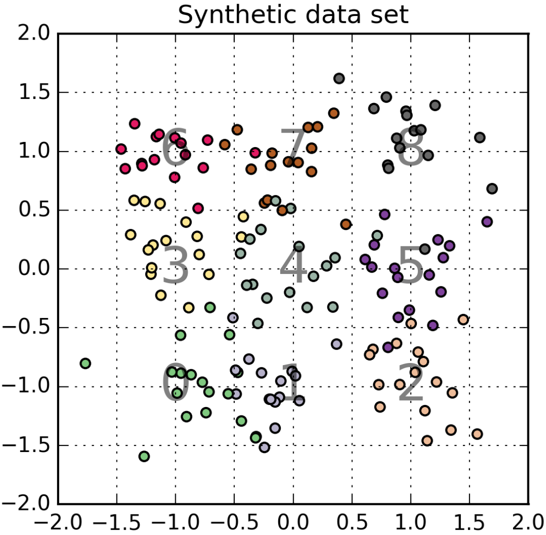
- Given a test object x_i
  - For each possible label value y
    - we create an example (x_i, y)
    - Identify the category T to which the hypothetical example (x_i, y) belongs.

      i.e. in this example find the label y_{NB} of the nearest neighbour of (x_i, y)
    - We compute the empirical probability distribution p_y of the labels in category T

      i.e. find all examples that have nearest neighbour with label y_{NB}; then for each possible label, count how many of those examples have that label; then normalize the counts to obtain relative frequencies.

- The result can be represented with a matrix, in which each row contains the probability distribution over the labels associated with the hypothetical label assignment.

Venn Predictors: Nearest Neighbour example



Venn Predictors: Nearest Neighbour example

Example: matrix for point (-1,-1)

	0	1	2	3	4	5	6	7	8
0	0.647059	0.294118	0.0000	0.058824	0.0000	0.0000	0.0000	0.0000	0.0000
1	0.562500	0.375000	0.0000	0.062500	0.0000	0.0000	0.0000	0.0000	0.0000
2	0.562500	0.312500	0.0625	0.062500	0.0000	0.0000	0.0000	0.0000	0.0000
3	0.562500	0.312500	0.0000	0.125000	0.0000	0.0000	0.0000	0.0000	0.0000
4	0.562500	0.312500	0.0000	0.062500	0.0625	0.0000	0.0000	0.0000	0.0000
5	0.562500	0.312500	0.0000	0.062500	0.0000	0.0625	0.0000	0.0000	0.0000
6	0.562500	0.312500	0.0000	0.062500	0.0000	0.0000	0.0625	0.0000	0.0000
7	0.562500	0.312500	0.0000	0.062500	0.0000	0.0000	0.0000	0.0625	0.0000
8	0.562500	0.312500	0.0000	0.062500	0.0000	0.0000	0.0000	0.0000	0.0625

The prediction is obtained as follows:

- Define as *quality* of a column its minimum entry
- Choose as *best* column the column with the highest quality
- Output the label of the column as label prediction and $\left[\min_i p_{i,j_{best}}, \max_i p_{i,j_{best}}\right]$

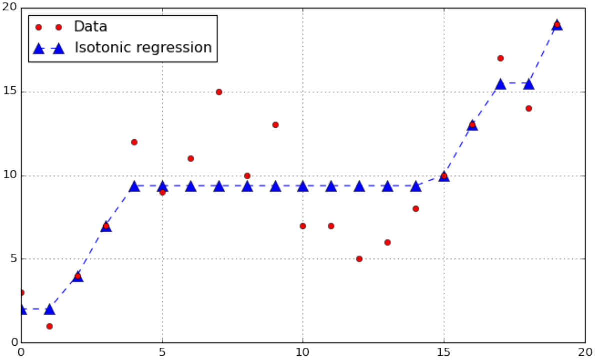
In this example, the prediction is 0 and the interval for the probability is [0.5625; 0.647]

Venn-ABERS[1] predictors

- Let's restrict our attention to binary classification
- Many machine learning algorithms for classification are in fact *scoring classifiers*: they output a prediction score $s(x)$ and the prediction is obtained by comparing the score to a threshold.
- One could apply a function g to $s(x)$ to calibrate the scores so that $g(s(x))$ can be used as predicted probability.
- Let's make only one assumption:
  that $g()$ be an non-decreasing function.

[1] ABERS: acronym of Ayer, Brunk, Ewing, Reid, Silverman. Authors of 1954 paper that first suggested the underlying technique

Isotonic Regression example



Non-decreasing function that minimizes sum of square residues

Venn-ABERS predictors

- The *isotonic calibrator* g for $((s(x_1), y_1), (s(x_2), y_2), \ldots, (s(x_\ell), y_\ell))$ is the non-decreasing function on $s(x_1), s(x_2), \ldots, s(x_\ell)$ that maximizes the likelihood

  $$\prod_{i=1,2,\ldots,\ell} p_i$$

  where:

  $$p_i = \begin{cases} g(s(x_i)) & \text{if } y_i = 1 \\ 1 - g(s(x_i)) & \text{if } y_i = 0 \end{cases}$$

- The isotonic calibrator can be found as isotonic regression on $(s(x_1), y_1), (s(x_2), y_2), \ldots, (s(x_\ell), y_\ell))$.

Venn-ABERS predictors

- Let $s_0(x)$ be the scoring function for $(z_1, z_2, \dots, z_\ell, (x, 0))$, $s_1(x)$ be the scoring function for $(z_1, z_2, \dots, z_\ell, (x, 1))$, $g_0(x)$ be the isotonic calibrator for

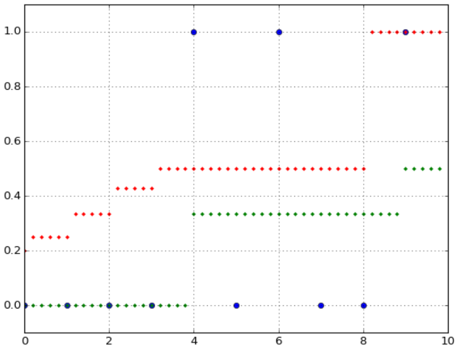
  $$((s_0(x_1), y_1), (s_0(x_2), y_2), \dots, (s_0(x_\ell), y_\ell), (s_0(x), 0))$$

  and $g_1(x)$ be the isotonic calibrator for

  $$((s_1(x_1), y_1), (s_1(x_2), y_2), \dots, (s_1(x_\ell), y_\ell), (s_1(x), 1))$$

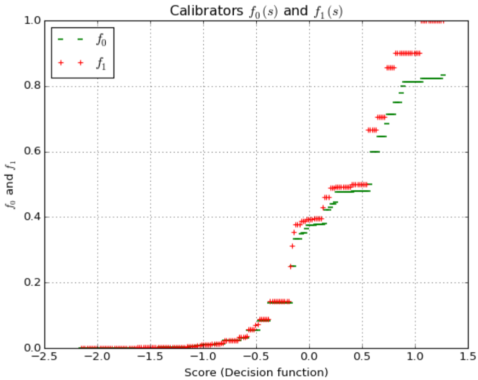
- The multiprobabilistic prediction output by the Venn-Abers predictor is:
  (p_0, p_1), where $p_0 := g_0(s_0(x))$ and $p_1 := g_1(s_1(x))$

Toy example



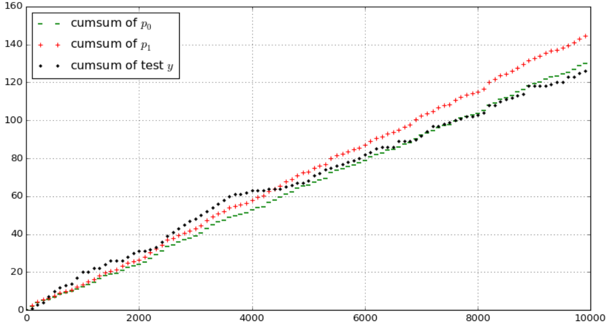
- Example of the two Isotonic Regressions in Venn Prediction
  - Blue dots: data (+1 or 0)
  - green dots: $g_0(s)$, red dots: $g_1(s)$

Real-life example



- Venn-ABERS Calibrators for Compound Activity Prediction
    - Applied to SVM decision function
    - green dots: $g_0(s)$, red dots: $g_1(s)$

Real-life example



- Venn-ABERS Multi-probabilistic Prediction
  - The red and green traces are random walks around the black trace

Venn-ABERS predictors

- Calibration via Isotonic regression is known to "overfit"
- Venn-ABERS predictors lessen this tendency to overfit and inherit the validity guarantee of Venn predictors.
- Compared with bare Isotonic Regression, the multi-probabilistic output also provides an indication of the reliability of the probability estimates.
  - If the probabilities differ, this can be taken as an indication of the uncertainty on the probability estimate itself.
- Compared with Platt's Scaling (fitting a sigmoid as calibrator), Venn-ABERS predictors do not make any assumption on the shape (functional form) of the calibrator.

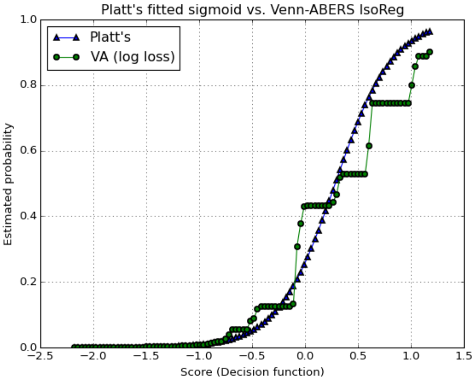
Single value prediction

- Can we deal with a single-valued probability instead?
- One possibility is the following:
  - In some types of problems, we are given the test object, we make a probability prediction and take a course of action based on it, then the actual label of the test object is revealed.
  - Regret: the loss incurred by using the prediction
  - Given the Venn-ABERS predictor, what probability value should we use so that *regret* is minimized?

    For log-loss:

$$p = \frac{p_1}{1 - p_0 + p_1}$$

Platt scaling vs. Venn-ABERS



- Platt scaling vs. (log-loss) Venn-ABERS
  - Imbalanced data set (class 1 was ≈ 1%)
  - Platt's scaling is possibly less accurate for high probs

Large-scale Venn-ABERS predictors

- Venn-ABERS appear much more computationally demanding than Isotonic Regression or Platt's Scaling.
  - For every evaluation, we have to retrain the underlying machine learning algorithm and recompute Isotonic Regressions
  - This would not scale to large data sets.
- Inductive Venn-ABERS Predictors
  - The training set is split into a proper training set and a calibration set.
  - The proper training set is used to train the underlying ML algorithm once.
  - The Isotonic Regression is calculated only on the calibration set.
  - This sacrifices the theoretical validity guarantee to make the computation feasible.
  - Despite the loss of the guarantee, it works well in practice

Large-scale Venn-ABERS predictors

- Inductive Venn-ABERS predictors are a step in the right direction but they still are prohibitive for large data sets.
  - For every evaluation, it seems we have to recompute 2 Isotonic Regressions on the calibration set.
- In actual fact, it can be computed very efficiently
  - It is possible to exploit the fact that only one data point is added to an otherwise fixed calibration set
  - Most computation occurs once for $g_0()$ and once for $g_1()$.
  - The evaluation requires only a binary search in a pre-computed data structure

Conclusions

- Complementing a prediction with its probability can enable better decision making
- Venn Predictors are (multi)probabilistic predictors with validity guarantee
- Venn-ABERS Predictors are Venn Predictors that can be applied on top of a Scoring Classifier
- VAP do not assume a functional form (e.g. sigmoid) for the relationship between score and probability

References

- Vladimir Vovk, Alex Gammerman, and Glenn Shafer.
  Algorithmic Learning in a Random World.
  Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2005.

- Vladimir Vovk, Ivan Petej.
  Venn-Abers Predictors
  In *Proceedings of 30th Conference on Uncertainty in Artificial Intelligence*, 2014
  `http://alrw.net/articles/07.pdf`

- Vladimir Vovk, Ivan Petej, and Valentina Fedorova.
  Large-scale probabilistic predictors with and without guarantees of validity.
  Technical Report `arXiv:1511.00213` [cs.LG], `arXiv.org` e-Print archive, November 2015.

Back-up material

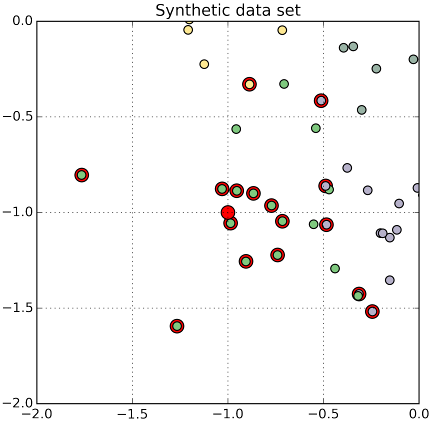
Why Venn?

- John Venn [1834-1923] was a logician and philosopher.
- In 1866 he published *The Logic of Chance*, in which he laid the foundations for the frequentist notion of probability.
- He's credited to have been the first to formulate explicitly and study the **reference class problem**.

  "It is obvious that every individual thing or event have an indefinite number of properties or attributes observable in it, and might therefore be considered as belonging to an indefinite number of different classes of things"
- Which class do we take when calculating relative frequencies?
- Venn also thought that "the more special the statistics, the better". But this leads to a dilemma: the more specific the class is, the fewer the element in the class.
- The same dilemma applies when designing a taxonomy.

Nearest Neighbour example

Zoom on the lower left quadrant



Test object at (-1,-1) with hypothetical label 0 (pale green, not drawn).
Large red dot denotes examples belonging to category 0.