

CONFORMAL PREDICTION

A Tiny Tutorial on Predicting with Confidence

Henrik Linusson, Ulf Johansson

December 3, 2014

University of Borås, Borås, Sweden

henrik.linusson@hb.se, ulf.johansson@hb.se

How good is your prediction?

You want to predict the toxicity of a new drug candidate: compound X.

How good is your prediction?

You want to predict the toxicity of a new drug candidate: compound X.

- Your classifier says the compound is non-toxic.

Is it really?

How good is your prediction?

You want to predict the toxicity of a new drug candidate: compound X.

- Your classifier says the compound is non-toxic.
Is it really?
- Your probability estimator says it's 80% likely that the compound is non-toxic.

How good is the estimate?

How good is your prediction?

You want to predict the toxicity of a new drug candidate: compound X.

- Your classifier says the compound is non-toxic.
Is it really?
- Your probability estimator says it's 80% likely that the compound is non-toxic.
How good is the estimate?
- Your regression model says the compound's toxicity level is 0.2.
How close is that to the true value?

The simple answer:

We expect past performance to indicate future performance.

The simple answer:

We expect past performance to indicate future performance.

- The model is 80% accurate on the test data, so we assume it's accurate for 80% of production data.
- The model has an AUC of 0.9 on the test data, so we assume it has an AUC of 0.9 on production data.
- The model has an RMSE of 2.0 on the test data, so we assume it has an RMSE of 2.0 on production data.

The simple answer:

We expect past performance to indicate future performance.

- The model is 80% accurate on the test data, so we assume it's accurate for 80% of production data.
- The model has an AUC of 0.9 on the test data, so we assume it has an AUC of 0.9 on production data.
- The model has an RMSE of 2.0 on the test data, so we assume it has an RMSE of 2.0 on production data.

But...

What about compound X?

What performance should we expect from the model for this particular instance?

We can use PAC (probably approximately correct) theory.

Gives us valid error bounds for the model.

But...

- Bounds are on model-level — don't consider whether instance is “easy” or “hard”.
- Bounds tend to be large.

We can use PAC (probably approximately correct) theory.

Gives us valid error bounds for the model.

But...

- Bounds are on model-level — don't consider whether instance is “easy” or “hard”.
- Bounds tend to be large.

We can use Bayesian learning.

Gives us calibrated error bounds on a per-instance basis.

But...

- Only if we know the prior probabilities.

We can use Conformal Prediction.

- Individual probabilities/error bounds per instance.
- Probabilities are well-calibrated: 80% means 80%.
- We don't need to know the priors.
- We make a single assumption — exchangeability (\sim i.i.d.)
- We can apply it to any machine learning algorithm.
- It's rigorously proven and simple to implement!

Conformal Prediction¹

Transforms classifiers and regressors into **confidence predictors**.

¹Vovk, Gammerman & Shafer (2005) Algorithmic Learning in a Random World

Conformal Prediction¹

Transforms classifiers and regressors into confidence predictors.

Predictions are multivalued **prediction regions**.

- Classification — label sets, e.g. {red, blue, green}
- Regression — intervals, e.g. [0.12, 0.19]

¹Vovk, Gammerman & Shafer (2005) Algorithmic Learning in a Random World

Conformal Prediction¹

Transforms classifiers and regressors into confidence predictors.

Predictions are multivalued prediction regions.

- Classification — label sets, e.g. {red, blue, green}
- Regression — intervals, e.g. [0.12, 0.19]

Predictions are associated with a measure of confidence.

A γ -confidence prediction region contains the true output with probability γ .

¹Vovk, Gammerman & Shafer (2005) Algorithmic Learning in a Random World

CONFORMAL REGRESSION

Assume we have:

- A regression model h , trained on Z^t .
- A **calibration set** of (labeled) examples, Z^c so that $Z^t \cap Z^c = \emptyset$.

Let's do the following:

- Use h to make predictions for Z^c and measure the absolute error for each prediction.
- Sort these error scores in descending order, and call them $\alpha_1, \dots, \alpha_q$.

Given that we have

q calibration set scores $\alpha_1, \dots, \alpha_q$ and

prediction $\hat{y}_k = h(\mathbf{x}_k)$ for a new test example,

and that

the examples in $Z^c \cup Z^t$ are exchangeable,

we can easily calculate error bounds for \hat{y}_k .

Given that we have

q calibration set scores $\alpha_1, \dots, \alpha_q$ and

prediction $\hat{y}_k = h(\mathbf{x}_k)$ for a new test example,

and that

the examples in $Z^c \cup Z^t$ are exchangeable,

we can easily calculate error bounds for \hat{y}_k .

20%	20 %	20%	20%	20%
α_1	α_2	α_3	α_4	

$$P(y_k \in \hat{y}_k \pm \alpha_1) = 0.8.$$

$$P(y_k \in \hat{y}_k \pm \alpha_2) = 0.6.$$

Given that we have

q calibration set scores $\alpha_1, \dots, \alpha_q$ and

prediction $\hat{y}_k = h(\mathbf{x}_k)$ for a new test example,

and that

the examples in $Z^c \cup Z^t$ are exchangeable,

we can easily calculate error bounds for \hat{y}_k .

20%	20 %	20%	20%	20%
0.78	0.65	0.33	0.24	

$$P(y_k \in 0.29 \pm 0.78) = 0.8.$$

$$P(y_k \in 0.29 \pm 0.65) = 0.6.$$

The values $\alpha_1, \dots, \alpha_q$ are called **nonconformity scores**.

They measure the “strangeness” of an example, using a **nonconformity function**.

We can use **any** function as a nonconformity function.

(For regression problems, the absolute error function is simply convenient.)

CONFORMAL CLASSIFICATION

Predicting the toxicity of compound X

1. Train an RF classifier on part (e.g. 80%) of the training data.
2. Make predictions for the remainder of the training data.
 - This is the held-out calibration set.
 - Calculate their nonconformity as, e.g., $\alpha_i = 1 - \hat{P}_h(y_i | \mathbf{x}_i)$.

Calibration nonconformity scores $A = \alpha_1, \dots, \alpha_q$.

$A = [0.28, -0.12, \dots, 0.99, -0.3]$

Predicting the toxicity of compound X

For every possible class, {no-toxicity, low-toxicity, high-toxicity},
make a tentative classification:

- $(X, \text{no-toxicity})$
- $(X, \text{low-toxicity})$
- $(X, \text{high-toxicity})$

Measure the nonconformity of the three tentative classifications using $\alpha_i = 1 - \hat{P}_h(y_i | \mathbf{x}_i)$.

$$\alpha^{\text{no}} = 0.02$$

$$\alpha^{\text{low}} = 0.28$$

$$\alpha^{\text{high}} = 0.92$$

So we measured nonconformity – now what?

Remember – is (x_k, \tilde{y}) particularly nonconforming compared to the calibration examples? Then \tilde{y} is probably an incorrect classification. Otherwise, include it in the prediction region.

To get statistically valid prediction regions, we simply use hypothesis testing to compare nonconformity scores!

Comparing nonconformity scores.

We have the nonconformity scores of the calibration examples, $A = \alpha_1, \dots, \alpha_q$, and the nonconformity of our tentatively classified example, $\alpha_k^{\tilde{y}}$. So, we calculate a p-value

$$p(x_k, \tilde{y}) = \frac{\#\{\alpha_i \in A \mid \alpha_i \geq \alpha_k\}}{\#A},$$

and compare $p(x_k, \tilde{y})$ to a predefined significance level ϵ .

- If $p < \epsilon$ then \tilde{y} is probably incorrect, and we reject it.
- Otherwise, we include it in the prediction region Γ_k^ϵ .

With probability $1 - \epsilon$, Γ_k^ϵ contains y_k .

Predicting the toxicity of compound X

Choose a significance level. We'll choose $\epsilon = 0.05$.

Calculate the p-values:

$$p(X, \text{no-toxicity}) = 0.35$$

$$p(X, \text{low-toxicity}) = 0.07$$

$$p(X, \text{high-toxicity}) = 0.02$$

At $\epsilon = 0.05$, we can reject high-toxicity, i.e.:

With 95% probability, compound X has either none or low toxicity!

SOME IMPORTANT MODIFICATIONS

Evaluation

Since all conformal predictors are valid, the main criterion when comparing different conformal predictors is their **efficiency**, i.e., the sizes of output prediction regions.

- Clearly a smaller prediction region is more informative, so efficiency is typically measured as:
 - Classification: The (average) number of labels present in the prediction sets
 - Regression: The (average) size of the prediction intervals

With standard nonconformity functions, the conformal regressor will produce prediction intervals of the same size for every x_j ; i.e., it does not consider the difficulty of x_j

With **normalized** nonconformity functions, the absolute error is scaled using the expected accuracy of the underlying model;

The motivation for this, from a conformal prediction standpoint, is that if two instances have identical absolute errors, an “easier” instance is actually stranger than a “harder”.

Using a normalized nonconformity function, the resulting prediction intervals will be smaller for instances that are deemed “easy” and larger for “harder” instances.

There are many possible ways to estimate the difficulty of a specific instance:

- A model (e.g., an ANN) could be trained to predict the (log) error of a test instance.
- We could use the average training error of the k nearest neighbors.
- For specific models, internal measures could be used. As an example - an instance is deemed easier for a kNN model if its k neighbors are close and if they agree in their predictions.

Naturally, when using normalized nonconformity measures, we expect the average prediction interval to be smaller, i.e., the efficiency is increased.

Purpose

Mondrian conformal predictors divide the problem space into several disjoint subcategories $\kappa_1, \dots, \kappa_m$ and provide a confidence $1 - \epsilon$ for each κ_j separately.

Purpose

Mondrian conformal predictors divide the problem space into several disjoint subcategories $\kappa_1, \dots, \kappa_m$ and provide a confidence $1 - \epsilon$ for each κ_j separately.

Examples:

- For a classification problem, each κ_j can be mapped to a possible class label, thus providing guarantees for each class, i.e., the errors will be evenly distributed over the classes.

Purpose

Mondrian conformal predictors divide the problem space into several disjoint subcategories $\kappa_1, \dots, \kappa_m$ and provide a confidence $1 - \epsilon$ for each κ_j separately.

Examples:

- For a classification problem, each κ_j can be mapped to a possible class label, thus providing guarantees for each class, i.e., the errors will be evenly distributed over the classes.
- The problem space can be divided w.r.t. to the feature space:

Purpose

Mondrian conformal predictors divide the problem space into several disjoint subcategories $\kappa_1, \dots, \kappa_m$ and provide a confidence $1 - \epsilon$ for each κ_j separately.

Examples:

- For a classification problem, each κ_j can be mapped to a possible class label, thus providing guarantees for each class, i.e., the errors will be evenly distributed over the classes.
- The problem space can be divided w.r.t. to the feature space:
 - In a decision list (rule set) - each rule will be independently valid
 - In a tree - each leaf (path) will be independently valid

Purpose

Mondrian conformal predictors divide the problem space into several disjoint subcategories $\kappa_1, \dots, \kappa_m$ and provide a confidence $1 - \epsilon$ for each κ_j separately.

Examples:

- For a classification problem, each κ_j can be mapped to a possible class label, thus providing guarantees for each class, i.e., the errors will be evenly distributed over the classes.
- The problem space can be divided w.r.t. to the feature space:
 - In a decision list (rule set) - each rule will be independently valid
 - In a tree - each leaf (path) will be independently valid

To construct a Mondrian conformal predictor, α_p is selected not from the full list of calibration scores $\alpha_1, \dots, \alpha_q$, but from a subset $\alpha_i \in \alpha_1, \dots, \alpha_q : \kappa(\mathbf{x}_i) = \kappa(\mathbf{x}_{k+1})$. The rest of the procedure remains unchanged.

A FINAL EXAMPLE

Models

- We start with a (tiny) regression tree producing point predictions
- A standard ICP - all intervals have the same size
- Normalized ICP - intervals have different sizes.
 - Here, the normalization is per leaf (not per instance) since we want to keep the interpretability.
 - The difficulty of a leaf is estimated as the standard deviation of the training predictions
- A Mondrian ICP - intervals have different sizes and all leaves are independently valid.

```

x6<10.58
| x6<7.25
| | x5<4.54
| | | y=0.078
| | x5>=4.54
| | | y=0.185
| x6>=7.25
| | x5<7.195
| | | y=0.293
| | x5>=7.195
| | | y=0.394
x6>=10.58
| y=0.689
    
```

(a) Regression tree for mortgage data set

```

x6<10.58
| x6<7.25
| | x5<4.54
| | | y=[0, 0.227]
| | x5>=4.54
| | | y=[0.035, 0.335]
| x6>=7.25
| | x5<7.195
| | | y=[0.143, 0.443]
| | x5>=7.195
| | | y=[0.244, 0.544]
x6>=10.58
| y=[0.539, 0.839]
    
```

(b) ICP for mortgage. $\epsilon = 0.1$

```

x6<10.58
| x6<7.25
| | x5<4.54
| | | y=[0, 0.227]
| | x5>=4.54
| | | y=[0.035, 0.335]
| x6>=7.25
| | x5<7.195
| | | y=[0.143, 0.443]
| | x5>=7.195
| | | y=[0.244, 0.544]
x6>=10.58
| y=[0.539, 0.839]
    
```

(a) ICP for mortgage. $\epsilon = 0.1$

```

x6<10.58
| x6<7.25
| | x5<4.54
| | | y=[0, 0.216]
| | x5>=4.54
| | | y=[0.050, 0.320]
| x6>=7.25
| | x5<7.195
| | | y=[0.157, 0.429]
| | x5>=7.195
| | | y=[0.257, 0.531]
x6>=10.58
| y=[0.539, 0.839]
    
```

(b) NICPs for mortgage. $\epsilon = 0.1$

```

x6<10.58
| x6<7.25
| | x5<4.54
| | | y=[0, 0.216]
| | x5>=4.54
| | | y=[0.050, 0.320]
| x6>=7.25
| | x5<7.195
| | | y=[0.157, 0.429]
| | x5>=7.195
| | | y=[0.257, 0.531]
x6>=10.58
| y=[0.539, 0.839]
    
```

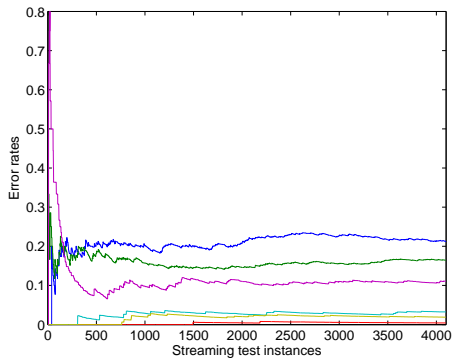
(a) NICPs for mortgage. $\epsilon = 0.1$

```

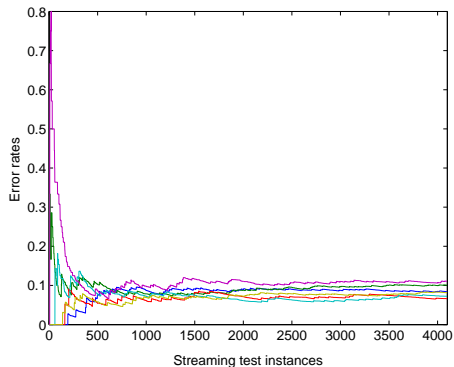
x6<10.58
| x6<7.25
| | x5<4.54
| | | y=[0.011, 0.144]
| | x5>=4.54
| | | y=[0.144, 0.226]
| x6>=7.25
| | x5<7.195
| | | y=[0.231, 0.355]
| | x5>=7.195
| | | y=[0.332, 0.456]
x6>=10.58
| y=[0.468, 0.910]
    
```

(b) MICP for mortgage. $\epsilon = 0.1$

CONFORMAL REGRESSION TREES



(a) ICP leaf errors, $\epsilon = 0.1$.



(b) MICP leaf errors, $\epsilon = 0.1$.

CONCLUDING REMARKS

Key people

- Vladimir Vovk and Alexander Gammerman, Royal Holloway, UK
- Harris Papadopoulos, Frederick University, Cyprus

What can we use it for?

- Classification and regression with guaranteed maximum error rates [Vovk et al., 2005, Papadopoulos, 2008, Johansson et al., 2013a].
- Anomaly detection with guaranteed maximum false positive rates [Laxhammar and Falkman, 2010].
- Concept drift detection / i.i.d. checking with maximum false positive rates [Fedorova et al., 2012].
- Semi-supervised learning [Zhu et al., 2013]

Research opportunities

Although the foundation is very solid there are a number of interesting topics.

Specifically, most of the presentations are very mathematical, and often not adapted for specific machine learning algorithms.

Sample tasks

- Novel nonconformity functions, typically adapted for the underlying algorithm
- Some fundamental questions regarding parameter choices (e.g., calibration set size) are not solved
- Novel uses of the frame work - we are using it for rule extraction with guaranteed fidelity
- Applications

Conformal prediction using decision trees

ICDM [Johansson et al., 2013a]

Effective utilization of data in inductive conformal prediction using ensembles of neural networks

IJCNN [Löfström et al., 2013]

Evolved decision trees as conformal predictors

CEC [Johansson et al., 2013b]

Regression conformal prediction with random forests

Machine Learning [Johansson et al., 2014c]

Regression trees for streaming data with local performance guarantees

IEEE BigData [Johansson et al., 2014a]

Rule extraction with guaranteed fidelity

CoPA [Johansson et al., 2014b]






Efficiency comparison of unstable transductive and inductive conformal classifiers






CoPA [Linusson et al., 2014b]

Signed-error conformal regression


PAKDD [Linusson et al., 2014a]


THANK YOU!


-  Fedorova, V., Gammerman, A., Nouretdinov, I., and Vovk, V. (2012).
Plug-in martingales for testing exchangeability on-line.
arXiv preprint arXiv:1204.3251.
-  Johansson, U., Boström, H., and Löfström, T. (2013a).
Conformal prediction using decision trees.
In Data Mining (ICDM), 2013 IEEE 13th International Conference on, pages 330–339.
IEEE.
-  Johansson, U., Boström, H., Löfström, T., and Linusson, H. (2014a).
Regression conformal prediction with random forests.
Machine Learning, 97(1-2):155–176.
-  Johansson, U., König, R., Linusson, H., Löfström, T., and Boström, H. (2014b).
Rule extraction with guaranteed fidelity.
In Artificial Intelligence Applications and Innovations, pages 281–290. Springer Berlin
Heidelberg.
-  Johansson, U., König, R., Löfström, T., and Boström, H. (2013b).
Evolved decision trees as conformal predictors.
In Evolutionary Computation (CEC), 2013 IEEE Congress on, pages 1794–1801. IEEE.

-  Johansson, U., Sönströd, C., Boström, H., and Linusson, H. (2014c).
Regression trees for streaming data with local performance guarantees.
In IEEE International Conference on Big Data. IEEE, In press.
-  Laxhammar, R. and Falkman, G. (2010).
Conformal prediction for distribution-independent anomaly detection in streaming vessel data.
In Proceedings of the First International Workshop on Novel Data Stream Pattern Mining Techniques, pages 47–55. ACM.
-  Linusson, H., Johansson, U., Boström, H., and Löfström, T. (2014a).
Efficiency comparison of unstable transductive and inductive conformal classifiers.
In Artificial Intelligence Applications and Innovations, pages 261–270. Springer.
-  Linusson, H., Johansson, U., and Löfström, T. (2014b).
Signed-error conformal regression.
In Advances in Knowledge Discovery and Data Mining, pages 224–236. Springer.
-  Löfström, T., Johansson, U., and Boström, H. (2013).
Effective utilization of data in inductive conformal prediction using ensembles of neural networks.

In Neural Networks (IJCNN), The 2013 International Joint Conference on, pages 1–8. IEEE.

 Papadopoulos, H. (2008).
Inductive conformal prediction: Theory and application to neural networks.
Tools in artificial intelligence, 18(315-330):2.

 Vovk, V., Shafer, G., and Gammerman, A. (2005).
Algorithmic learning in a random world.
Springer, New York.

 Zhu, X., Schleif, F.-M., and Hammer, B. (2013).
Secure semi-supervised vector quantization for dissimilarity data.
In Advances in Computational Intelligence, pages 347–356. Springer.