Chapter 13

Feature importance in causal inference for numerical and categorical variables

Bram Minnaert

Abstract Predicting whether A causes B (write A $\rightarrow$ B) or B causes A from samples (X, Y) is a challenging task. Several methods have already been proposed when both A and B are numerical. However, when A and/or B are categorical, few studies have already been performed.
This paper aims to learn the causal direction between two variables by fitting the regressions of X on Y and Y on X with machine learning algorithm and giving preference to the direction that yields a better fit.
This paper will investigate which features are the most important when A/B is numerical/categorical. Via an ensemble method, it finds that the features that are important heavily depend on the different combination of numerical/categorical.

Key words: Causal inference, Deterministic causal relations, Random forest regression, Graphical models, Feature selection

13.1 Introduction

Consider the following problem: we have a set of observations of (A, B) pairs. Without any context, can we give an estimate of the causal relationship between A and B? It is possible that A causes B (A $\rightarrow$ B), that B causes A (B $\rightarrow$ A), that they are independent (A $\mid$ B) or that they have a common cause (C $\rightarrow$ A and C $\rightarrow$ B).

Recent years a number of very promising methods have been proposed to predict causal relationships [Janzing et al., 2012, Hoyer et al., 2009, Shimizu et al., 2006,

Bram Minnaert
Ghent, 9000, Belgium e-mail: bram.minnaert@gmail.com

Zhang and Hyvärinen, 2010, Daniušis et al., 2010, Mooij et al., 2010]. Most studies in the field of causal discovery require A and B to be numerical. This paper will concentrate on the differences between numerical and categorical data.

This paper will define some features and via machine learning techniques it will investigate the importances of these features and estimate the probability that A causes B:

$$P(A \to B) \in [0,1]. \tag{13.1}$$

Section 13.2 will describe the model on a high level. Section 13.3 will zoom in on the different submodels. In section 13.4 we look at the results of the model and we will focus on the importances of the features. We will focus on the differences between numerical and categorical data since only few studies have already been performed on categorical data [Sun et al., 2006]. At last, we will draw conclusions in section 13.5.

13.2 Model description

Figure 13.1 shows the architecture of the model.

After some preprocessing steps, such as data normalisation, a list of features is extracted for every $A-B$ pair, for example the correlation coefficient. The preprocessing step and feature extraction is performed on a large number of $A-B$ pairs, which were provided in a training set. Because we know the correct solution for these $A-B$ pairs in the training set (the solution in the causal relationship, for example: A $\to$ B), we can apply supervised learning methods. Via the ensemble method *Random Forest* a large collection of classification trees is generated for the training set (features $\to$ causal relationship). After training we can use this trained model to predict new $A-B$ pairs with unknown causal relationship.

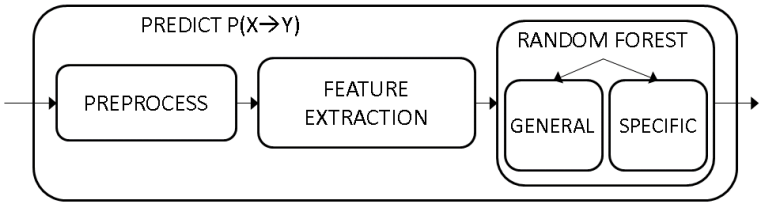

Fig. 13.1: Overview of the causal inference model

The model has been developed in Python and makes uses of the libraries Numpy, Pandas and SciPy [Jones et al., 2001–]. The figures in this paper have been made using matplotlib [Hunter, 2007].

In order to make this study replicable, the code has been made available on the following url: `https://github.com/braincomic/CauseEffectChallenge`

13.3 Model steps

13.3.1 Preprocessing

The following preprocessing steps are executed.

- Data normalisation if numerical (feature scaling). This is common in data processing. The range of A/B values can vary widely. Suppose that we would like to calculate the distances between two points. If we don't perform normalisation, this distance will be large if ranges of values are large and small if ranges of values are small.
- Reordering the categories from 0 to n if A/B is categorical in such a way that $E(B|A)$ is increasing in A. Figure 13.2 show examples. This will enable the numerical features to perform better on categorical data.

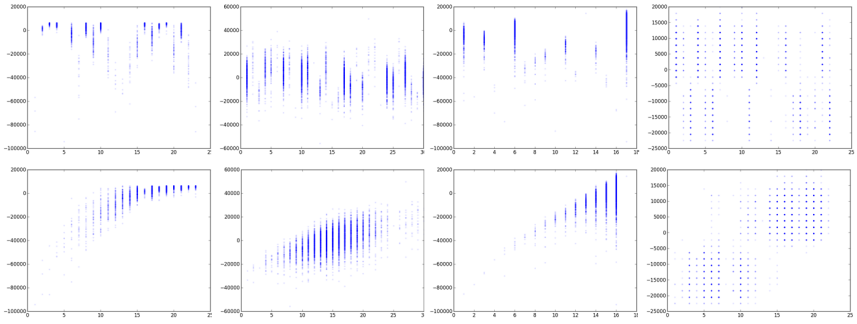

Fig. 13.2: Reordering of categorical values. The figures in the second row are the reordered versions of the corresponding figures of the first row.

We have not performed outlier removal as preprocessing step because outliers can give an indication of a causal relationship.

13.3.2 Features

In this step we extract 211 features. We will not describe the full list. Instead, we will describe groupings.

1. Number of samples: number of samples in the data set, number of unique samples, difference of unique samples A versus B, fraction of unique samples. These features serve mainly as control, we don't expect these features to matter.
2. Basic statistics: median, minimum, maximum, range, percentiles, skewness, curtosis and minimal precision.
3. Correlation: Pearson product-moment correlation coefficient, Spearman's rank correlation coefficient.
4. Polynomial regressions, as described in Hoyer et al. [2009], ranging from degree 1 (linear regression) to degree 4. One feature will determine the best degree itself by splitting the sample into a training and test set, up to degree 9. Examples are shown in figure 13.3.
5. Logistic regression.
6. Moving average: quality of the moving average function
7. Uniformity and Normality.
8. Remainder test: first a regression is made (for example, regression with degree 4) and then the difference is made of the noisy data with the regression. The resulting distribution is tested for uniformity and normality.
9. Inversibility test: specific test to check if some polynomial regression is inversible or not.
10. Outlier detection.
11. Information theory features: Shannon entropy, conditional entropy, mutual information, homogeneity, completeness, v-measure and information gain. This is all calculated after binning. For each feature we used either fixed width or fixed frequency discretisation. We choose the method that obtained the best results on the training set. This also determined the number of bins.
12. IGCI: Information Geometric Causal Inference, both the entropy based and the integral-approximation based estimator, both for uniform and gaussian noice [Janzing et al., 2012].
13. Clustering: quality of vector quantisation using k-means clustering.

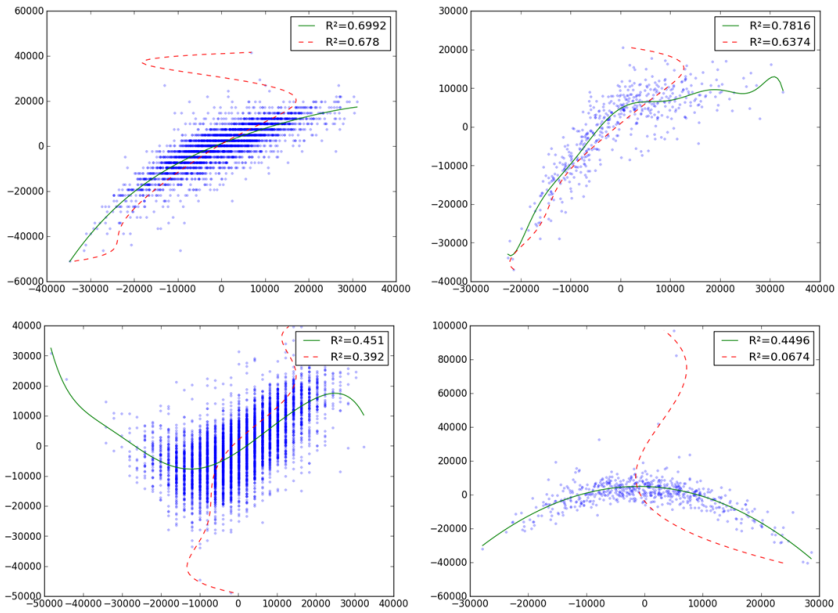

Fig. 13.3: In these figures the variable is the cause of the Y variable. We expect better regressions Y=f(X), the green line, compared to regressions X=f(Y), the red line. These figures are cherry picked to make the idea more clear, it is not always this obvious, unfortunately.

13.3.3 Random forest

We make use of a random forest regression [Breiman, 2001]. This is an ensemble learning method that creates many classification trees when training the model.

Instead of using a random forest, we also tried gradient boosting [Friedman, 2000, 2002].

Even though gradient boosting scores slightly better than random forest in a comparison of 11 binary classification methods [Caruana and Niculescu-Mizil, 2006], random forest obtained better scores in the training set predictions, so we used random forest.

The training also showed that a random forest regression performed better than a random forest classification.

13.3.3.1 Training

We train the features using a random forest regression. In a random forest a number of features is randomly selected to train a classification tree. The training of this tree is not done on the entire training set, but only a random sample. This is done many time, so a lot of classification trees are generated, which we call a forest.

We do not only perform this training on the entire training set, but we train it on several subsets:

- Full training set.
- Numerical $\rightarrow$ Numerical.
- Categorical $\rightarrow$ Categorical.
- Categorical $\rightarrow$ Numerical.
- Numerical $\rightarrow$ Categorical.

In these subsets, binary variables are treated as Categorical.

13.3.3.2 Predicting

In the previous step we have trained our model via the ensemble method *random forest*, resulting into many classification trees. Now we can use this trained model to make predictions on new $A-B$ pairs. In the previous step we have not trained one model, we have trained five models. In order to predict a new $A-B$ pair, we need to take the weighted average of different predictions of different trained models.

1. Predict with the model trained with the full training set. We call this function

$$Pred_{\text{full}}(A,B) \tag{13.2}$$

2. Predict with the model trained with the specific training set for these types. We use T_A as the type of A (Numerical or Categorical) and T_B the type of B. For example, $Pred_{Categorical,Numerical}$.

$$Pred_{T_A,T_B}(A,B) \tag{13.3}$$

$$\text{with } T_A, T_B \in \{\text{"Numerical", "Categorical"}\} \tag{13.4}$$

3. Take the weighted average of these 2 predictions. As we will see in section 13.4, some models based on the specific training sets (types) achieve much better scores than others. Therefore the better models are given an extra benefit via a

weight factor W that depend on the type of A and B. Normalisation is added: worst possible score is 0, best is 1.

$$P(A \to B) = \frac{W_{\text{full},T_A,T_B} Pred_{\text{full}}(A,B) + W_{\text{spec},T_A,T_B} Pred_{T_A,T_B}(A,B)}{\text{MAX}(W_{\text{full},T_A,T_B} + W_{\text{spec},T_A,T_B})} \tag{13.5}$$

If we would like to predict a ternary truth value $T(A,B)$ indicating whether A is a cause of B (+1), B is a cause of A (-1), or neither (0), we simply take the following difference.

$$T(A,B) = P(A \to B) - P(B \to A) \tag{13.6}$$

13.4 Results

13.4.1 AUC score

We use the AUC (Area Under the ROC curve) as evaluation metric. The predictions of the full model are evaluated against a test set of 4050 A-B pairs. When we evaluate the ternary truth value T we use the average of the two AUC score related to $P(A \to B)$ and $P(B \to A)$

The following table summarizes the results. In the ChaLearn cause-effect pair challenge, hosted by Kaggle[et al, 2014], this AUC score resulted in the 5th place from 267 competitors.

subset	AUC score
Num $\to$ Num	0.818*
Cat $\to$ Cat	0.571*
Cat $\to$ Num	0.690*
Num $\to$ Cat	0.608*
TOTAL	0.788

*As the detailed final results of the ChaLearn cause-effect pair challenge, hosted by Kaggle[et al, 2014], has not yet been published, these lines contains the results on the cross validation set instead of the final test set. The total score on the other hand is based on the test set.

We seem to have have achieved the best job in predicting Numerical → Numerical. Categorical → Categorical on the other hand didn't work out well.

13.4.2 Feature importances

Random forest regressions come with a very handy scoring of the features: the variable importance. For every feature we have calculated the sum of the variable importances and we have listed them in figure 13.4.

Some features aways perform badly:

- Logistic regression, even on the categorical data.
- Moving average
- Outlier detection. It seems we should have performed outlier removal in the preprocessing steps, or at least treated them separately.
- IGCI
- Clustering

	SPECIFIC				FULL
Feature category	Num->Num	Cat-->Cat	Cat->Num	Num->Cat	
Number samples	0.2	0.5	0.4	0.7	0.3
Basic stats	1.5	11.7	4.4	16.2	1.8
Correlation	0.4	1.7	4.9	6.8	0.8
Polynomial regression	90.6	47.0	7.7	52.1	92.6
Logistic regression	0.0	2.0	0.2	0.7	0.0
Moving average	0.1	1.0	0.5	0.6	0.1
Uniformity and Normality test	4.7	5.7	0.9	3.1	1.9
Remainder test	0.5	1.2	0.5	1.0	0.3
Inversibility	0.1	0.4	0.1	0.2	0.0
Outlier detection	0.2	0.7	0.7	0.8	0.2
Information Theory	1.4	26.7	78.9	14.3	1.6
IGCI	0.1	0.3	0.4	0.5	0.1
Clustering	0.1	0.9	0.4	3.1	0.1
Total	100.0	100.0	100.0	100.0	100.0

Fig. 13.4: Importances of feature categories in the different submodels in sum of the percentages of the features in this category.

13.4.2.1 Numerical → Numerical feature importance

Numerical to numerical is totally ruled by the polynomial regression features. The most important features are listed below.

- The most important feature is the R^2 value of a polynomial regression with variable degree. The degree is determined by splitting the data set into a training set to find the best polynomial regression and a cross-validation set to measure the quality of the regression. It's feature importance is 75.1%.
- The degree of this polynomial has feature importance 9.9%.

Interpretation: if A causes B, then the regression $B = f_1(A) + \varepsilon_1$ will have a higher quality (better R^2) and will be simpler (lower degree) than the regression $A = f_2(B) + \varepsilon_2$.

13.4.2.2 Categorical → Categorical feature importance

- Surprisingly the polynomial regressions do very well on categorical data. When we introduced reordering categories as described in section 13.3.1, the variable importance of these features raised dramatically. All polynomial features sum up to 47.1%.
- Several features of information theory score pretty well. The best are the mutual information and correlated entropy, defined as the mutual information divised by the Shannon entropy. The corresponding feature importances are 12.6% and 7.5%.

13.4.2.3 Categorical → Numerical feature importance

- The features of information theory score very high. The best are the mutual information and the v-measure, having importances of 49.3% and 23.8%

13.4.2.4 Numerical → Categorical feature importance

- The polynomial regressions again have the highest importance, summing to 52.1%.
- From information theory, the mutual information has 9.1% importance.

13.5 Conclusion

When predicting causal relationships, the category of features that gives the most information heavily depends on the type of the variables: numerical or categorical.

Thanks to category reordering the numerical features are still important to categorical data. The following table summarizes the most important categories of features.

subset	Most important feature categories
Num → Num	Polynomial regression
Cat → Cat	Polynomial regression, information theory
Cat → Num	Information theory
Num → Cat	Polynomial regression, information theory
TOTAL	Polynomial regression

Acknowledgments

I would like to thank Kaggle and Chalearn to stir my interest into this topic Guyon [2013] and I thank Isabelle Guyon and Mehreen Saeed for their assistance to make my source code portable.

References

Leo Breiman. Random forests. *Mach. Learn.*, 45(1):5–32, October 2001. ISSN 0885-6125. doi: 10.1023/A:1010933404324. URL http://dx.doi.org/10.1023/A:1010933404324.

Rich Caruana and Alexandru Niculescu-Mizil. An empirical comparison of supervised learning algorithms. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, pages 161–168, New York, NY, USA, 2006. ACM. ISBN 1-59593-383-2. doi: 10.1145/1143844.1143865. URL http://doi.acm.org/10.1145/1143844.1143865.

Povilas Daniušis, Dominik Janzing, Joris M. Mooij, Jakob Zscheischler, Bastian Steudel, Kun Zhang, and Bernhard Schölkopf. Inferring deterministic causal relations. In *Proceedings of the 26th Annual Conference on Uncertainty in Artificial Intelligence (UAI-10)*, 2010. URL http://event.cwi.nl/uai2010/papers/UAI2010_0121.pdf.

Isabelle Guyon et al. Results and analysis of the 2013 chalearn cause-effect pair challenge. 2014.

Jerome H. Friedman. Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, 29:1189–1232, 2000.

Jerome H. Friedman. Stochastic gradient boosting. *Comput. Stat. Data Anal.*, 38(4):367–378, February 2002. ISSN 0167-9473. doi: 10.1016/S0167-9473(01)00065-2. URL http://dx.doi.org/10.1016/S0167-9473(01)00065-2.

Isbelle Guyon. Cause-effect pairs challenge, 2013. Isabelle Guyon (ChaLearn) and Ben Hamner (Kaggle) and Alexander Statnikov (NYU) and Mikael Henaff (NYU) and Vincent Lemaire (Orange) and Bernhard Shoelkopf (MPI).

Patrik O. Hoyer, Dominik Janzing, Joris M. Mooij, Jonas Peters, and Bernhard Schölkopf. Nonlinear causal discovery with additive noise models. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Advances in Neural Information Processing Systems 21 (NIPS*2008)*, pages 689–696, 2009.

J. D. Hunter. Matplotlib: A 2d graphics environment. *Computing In Science & Engineering*, 9(3): 90–95, 2007.

Dominik Janzing, Joris Mooij, Kun Zhang, Jan Lemeire, Jakob Zscheischler, Povilas Daniušis, Bastian Steudel, and Bernhard Schölkopf. Information-geometric approach to inferring causal directions. *Artif. Intell.*, 182-183:1–31, May 2012. ISSN 0004-3702. doi: 10.1016/j.artint.2012.01.002. URL http://dx.doi.org/10.1016/j.artint.2012.01.002.

Eric Jones, Travis Oliphant, Pearu Peterson, et al. SciPy: Open source scientific tools for Python, 2001–. URL http://www.scipy.org/.

Joris M. Mooij, Oliver Stegle, Dominik Janzing, Kun Zhang, and Bernhard Schölkopf. Probabilistic latent variable models for distinguishing between cause and effect. In J. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R.S. Zemel, and A. Culotta, editors, *Advances in Neural Information Processing Systems 23 (NIPS*2010)*, pages 1687–1695, 2010. URL http://books.nips.cc/papers/files/nips23/NIPS2010_1270.pdf.

Shohei Shimizu, Patrik O. Hoyer, Aapo Hyvärinen, and Antti Kerminen. A linear non-gaussian acyclic model for causal discovery. *J. Mach. Learn. Res.*, 7:2003–2030, December 2006. ISSN 1532-4435. URL http://dl.acm.org/citation.cfm?id=1248547.1248619.

Xiaohai Sun, Dominik Janzing, and Bernhard Schölkopf. Causal inference by choosing graphs with most plausible markov kernels. In *ISAIM*, 2006. URL http://dblp.uni-trier.de/db/conf/isaim/isaim2006.html#SunJS06.

K Zhang and A Hyvärinen. Distinguishing causes from effects using nonlinear acyclic causal models. In I Guyon, D Janzing, and B Schölkopf, editors, *JMLR Workshop and Conference Proceedings, Volume 6*, pages 157–164, Cambridge, MA, USA, 2010. MIT Press. URL http://www.is.tuebingen.mpg.defileadmin/user_upload/files/publications/2012/zhang-hyv{ä}rinen_2010.pdf.