Chapter 11

Training Gradient Boosting Machines using Curve-fitting and Information-theoretic Features for Causal Direction Detection

Spyridon Samothrakis, Diego Perez, Simon Lucas

Abstract D

etecting causal relationships between random variables using only matched pairs of noisy observations is a crucial problem in many scientific fields. In this paper the problem is addressed by extracting a number of features for each matched pair using a selection of curve-fitting and information theoretic features. Using these features, we train a pair of Gradient Boosting Machines whose hyperparameters we optimise using stochastic simultaneous optimistic optimisation. The results show that our method is relatively successful, gaining a 3rd place in the 2013 Kaggle's Causality Challenge. Our method is sound enough to be used in causality detection (or as part of a more comprehensive toolkit), although we believe it might be possible to considerably improve the quality of results by adding more features in the same vein.

Key words: Causality Detection, Gradient Boosting Machine, StoSOO.

Spyridon Samothrakis e-mail: ssamot@essex.ac.uk

Diego Perez e-mail: dperez@essex.ac.uk

Simon Lucas e-mail: sml@essex.ac.uk
Wivenhoe Park
Colchester
Essex CO4 3SQ, United Kingdom.

11.1 Introduction

Humanity's effort to understand causality and its relationship to knowledge can be observed in almost every academic field, including philosophy (e.g. Falcon [2012]:"we think we have knowledge of a thing only when we have grasped its cause" quoting Aristotle) or Anthropology (e.g. see the ability for associative thinking in Frazer [1936]). One can formulate the problem of attributing causality between events in the spirit of Pearl [2000] as a Markov Decision Process (MDP). A (finite) Markov Decision Process is a tuple $< S, C, T, R >$, where $c \in C$ is the set of actions an agent can perform, $s \in S$ a set of states and $R(s'|s)$ is the reward at each state/action pair. $T(s'|s,c)$ is a transition function that denotes the probability of an agent moving from state s to another state s' given an action c. To apply MDPs to the problem of causality we make the following instantiation: All actions come from two sets $C_1 = A, C_2 = B$ and states $S_1 = A, S_2 = B$. Thus, there is a transition function that has the form $T(b|s,a)$ and $T(a|s,b)$. The MDP runs for one step, with both agents being at dummy state s initially. The agent takes an action that either leads it to one group of states A or B, followed by a second action that takes it to either B or A, respectively. Let's assume we are trying to learn a generative model of the transition function, in order to use it later in some policy scheme. If $T(A|s,B) = T(A)$, we claim A does **not** cause B. Otherwise, if the actions taken from set B impact T, i.e. the previous equation does not hold, we claim that A causes B. In other words if an agent is able to effectively control a process given the possibility of doing so, we can claim that the agent's actions are causal to the states of the process.

It is not always easy to perform the controlled process mentioned above, but it might be the case that we have a number of observations of actions each agent took, following uniform random policy (i.e. nature is the agent). This in effect turns our problem into one of prediction. Does the knowledge of action A help me predict action B and/or the reverse? Obviously, assuming the transition function is stochastic, there can be no proof of this. We could possibly try to infer the causal direction if we assume some sensible set of priors over the transition function, mostly taking a view reminiscent of the work of Kolmogorov, i.e. assuming nature prefers simple mechanisms. In this paper we try to infer causal structure using a machine learning approach on features extracted from the random variables provided.

The rest of the paper is organised as follows: In Section 11.2 we present the method we used for inferring causal direction. In Section 11.3 we present some experimental results and analyse the resulting classifiers. We conclude with a short discussion in Section 11.4.

11.2 Methodology

There are some core concepts behind the methodology followed. Firstly, we are trying to find whether the mapping $F : A \rightarrow B$ is more probable than $F : B \rightarrow A$. This can be captured by trying to fit different classifiers at each direction of the data. This implicitly assumes that machine learning classifiers tend to prefer simpler models. The second concept is that information theoretic features about the data should be able to capture some of the characteristics of the underlying distributions, thus helping our overall classification task.

11.2.1 Data and Data Pre-processing

Our data source was the union of all samples provided by the "Kaggle Causality Challenge" and can be found here: `http://www.kaggle.com/c/cause-effect-pairs/data`. The amount of data provided is doubled by reversing all the examples provided. The total number of labelled data is 32399 samples. Each labelled sample belongs to either class 1 (A causes B), class -1 (B causes A) or class 0 (where respectively the events are independent; are influenced by a third cause or we cannot tell). The type of variable in each data sample is also known (i.e. categorical, binary or continuous).

11.2.2 Feature Extraction

What follows is a brief exposition of the features used:

1. *Spearman* ρ: The correlation coefficient ρ.
2. *Number of Unique Samples A*: Number of unique samples of variable A.
3. *Number of Unique Samples B*: Number of unique samples of variable B.
4. *Noise Independence* $A \rightarrow B$ *(trees)*: The mutual information of an additive noise model[Hoyer et al., 2009]. Uses k-means++(Arthur and Vassilvitskii [2007]) to discretise noise. Modelling is performed using Regression or Decision Trees.
5. *Noise Independence* $B \rightarrow A$ *(trees)*: As in feature 4, but trying to predict A using B.
6. *Noise Independence* $A \rightarrow B$ *(SVM)*: As in feature 4, with a support vector classifier or regressor as the modelling function.
7. *Noise Independence* $B \rightarrow A$ *(SVM)*: As in feature 5, but trying to predict A using B.

8. *Noise Independence A → B (trees) - spearman*: As in feature 4 but, instead of mutual information, using spearman ρ as an independence test.
9. *Noise Independence B → A (trees) - spearman*: As in feature 5, but trying to predict A using B.
10. *Entropy A*: Entropy of Variable A. If the variable is continuous, k-means is performed and distance is measured from closest centre as a method for descretisation.
11. *Entropy B*: Entropy of Variable B. Same discretisation method as with feature 10.
12. *Uncertainty Coefficient A* Uncertainty Coefficient of Variable A. In case of continuous variable, the k-means trick from feature 10 is used.
13. *Uncertainty Coefficient B* Uncertainty Coefficient of Variable B. In case of continuous variable, the k-means trick from feature 10 is used.
14. *Predicts A → B (trees)*: Fraction of correctly classified examples or R^2, depending on whether B is categorical or continuous. In all tree examples a decision tree regressor or a decision tree classifier is used.
15. *Predicts B → A (trees)*: As in feature 14, but trying to predict A using B.
16. *Predicts U → B (trees)*: Predict B using just random variables that come from a distribution as close to A as possible.
17. *Predicts U → A (trees)*: As above but with reversed direction.
18. *Predicts A → B (SVM)*: Exactly as in the case of labelit:pred, but this time with support vector machines.
19. *Predicts B → A (SVM)*: See above.
20. *Predicts U → B (SVM)*: See above.
21. *Predicts U → A (SVM)*: See above.
22. *Uniform Symmetrised Divergence A*: Symmetrised KL Divergence between A and the Uniform distribution. As usual, discretisation is performed using k-means.
23. *Uniform Symmetrised Divergence B*: Symmetrised KL Divergence between B and the Uniform distribution.
24. *KL Divergence from Normal A*: KL Divergence of A from the normal distribution.
25. *KL Divergence from Normal B*: KL Divergence of B from the normal distribution.
26. *KL Divergence from Uniform A*: KL Divergence of A from the uniform distribution.
27. *KL Divergence from Uniform B*: KL Divergence of A from the uniform distribution.
28. *LiNGAM*: The LiNGHAM causality coefficient [Shimizu et al., 2006], implemented by the original author of this method.
29. *ICGI - Normal Integration*: IGCI Gaussian-Integration coefficient [Janzing et al., 2012], implemented by the original authors of this method.
30. *ICGI - Uniform Integration*: IGCI Uniform-Integration coefficient, as above.

11.2.3 Classifier

Two Gradient Boosting Machines (GBM see Friedman [2001]) have been used, with 3000 trees at each one. Each tree in each GBM has a maximum depth of 12 and a learning rate of approximately 0.0063640. The minimum samples required for each tree split is 5. The first GBM_1 is trained using only samples from the class 1 vs everything else, where everything else forms class 0). The other GBM_{-1} is trained using samples of class -1 vs everything else. To denote the probability of a sample belonging to a specific class P_{GBM} is used. the score of each sample is set to $S = P_{GBM_1}(1) - P_{GBM_{-1}}(-1)$. In other words the score attributed to each sample is the probability of having causal direction from A to B minus the probability of having causal direction B to A.

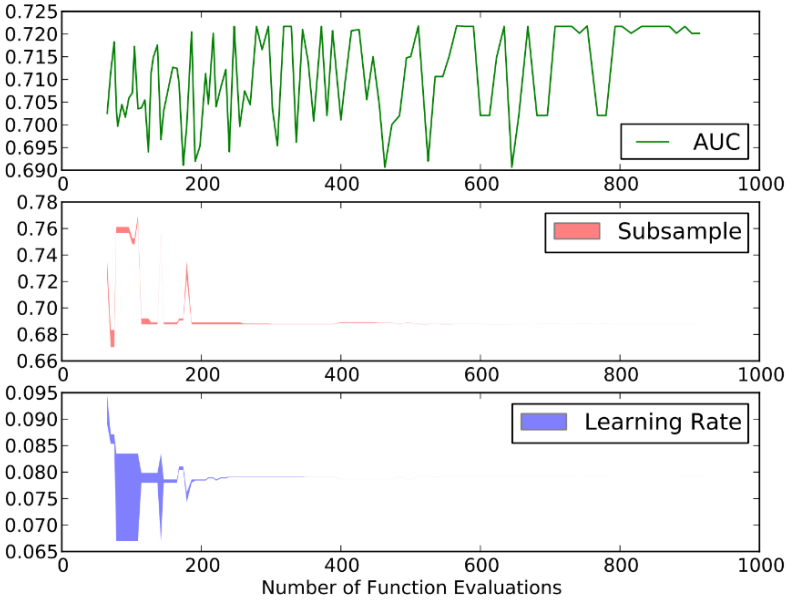


Fig. 11.1: Hyper-parameter Optimization Progress. Notice that both hyper-parameters affect regularization. Learning rate affects speed of convergence and sub-sample affects the portion of samples used at each iteration of each GBP learning cycle

11.2.4 Hyperparameter Optimisation

A modified version of Stochastic Simultaneous Optimistic Optimisation [Valko et al., 2013] (StoSOO) was used to optimise the learning rate and the subsampling percentage (i.e. the samples to be used in a bagging-like procedure) for the two GBMs. StoSOO is a tree-like algorithm that samples the hyperparameter space by iteratively splitting it into smaller segments, which it then samples, until some cut-off point.

11.3 Experiments & Analysis

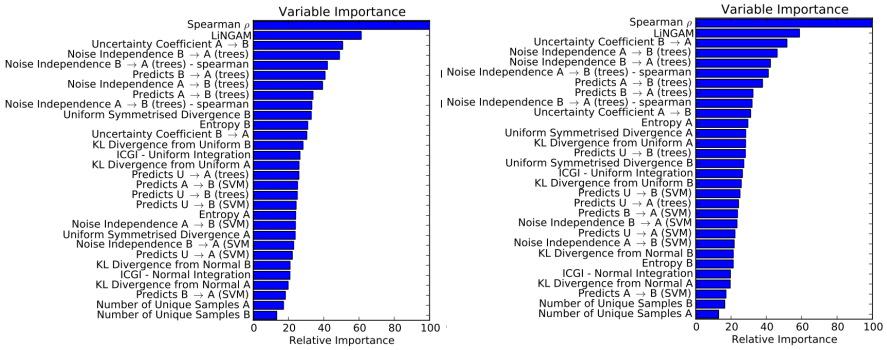
The resulting classifier and meta-optimisation technique is analysed in this section. Note that the AUC score of our classifier in the Kaggle's causality challenge test set is 0.79957. This gave us the third place in the competition out of 69 participants.

11.3.1 Hyperparameter Optimisation

A number of hyperparameters were optimized by a hyperparameters by a combination of hand-tuning and small runs of StoSOO. A sample run can be seen in Figure 11.1, on a subset (10%) of the experimental data, 100 trees in our GBM and a maximum tree depth of 10. Notice StoSOO improving AUC using just a subset of the data. The AUC is obtained by doing 3-fold cross validation over a random selected subset of that data (i.e. dataset splits are NOT fixed in every iteration). Notice the randomness of AUC score (within certain bounds), but the convergence of subsample and learning rate GBM attributes. Also notice the uncertainty concerning hyperparameter values early in the run.

11.3.2 Training and Classifier Analysis

In Figure 11.2 one can see the relevant score of each variable plotted, with 100 being the most important variable. Feature importance signifies the average importance of each variable, as measured by how high in the tree the variable is (being higher in the tree means affecting more samples). In an ensemble of features produced by GPM the normalised average of these variables is what is plotted. From Figure 11.2 one



(1) Feature Importances of GBM_1 (2) Feature Importances of GBM_{-1}

Fig. 11.2: Relative feature importance for each classifier

can see that the most important feature is Spear's correlation, presumably GBM is first throwing away cases that are uncorrelated. The least important feature involved (predictably) if the number of unique variables for variable B.

11.4 Conclusion

A method for detecting causality has been presented. Obvious improvements to the method include creating more curve fitting features and introducing more information theoretic features. One could, for example, add trees of different sizes, plus a number of SVMs with different kernels/kernel parameters. Fitting linear classifiers/regressors or higher level polynomials would be another option. Finally, at the beginning of this paper we emphasised the decision theoretic aspects of causality detection. It might be possible to directly tackle the problem using decision theoretic methods (e.g. standalone StoSOO or Monte Carlo Tree Search).

Acknowledgment

This work was supported by EPSRC grant EP/H048588/1 entitled:"UCT for Games and Beyond".

References

David Arthur and Sergei Vassilvitskii. k-means++: The advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 1027–1035. Society for Industrial and Applied Mathematics, 2007.

Andrea Falcon. Aristotle on causality. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Winter 2012 edition, 2012.

James George Frazer. *The Golden Bough: A Study in Magic and Religion. Vol. 13, Aftermath: a Supplement to the Golden Bough*. Macmillan, 1936.

Jerome H Friedman. Greedy function approximation: a gradient boosting machine. *Annals of Statistics*, pages 1189–1232, 2001.

Patrik O Hoyer, Dominik Janzing, Joris M Mooij, Jonas Peters, and Bernhard Schölkopf. Nonlinear causal discovery with additive noise models. 2009.

Dominik Janzing, Joris Mooij, Kun Zhang, Jan Lemeire, Jakob Zscheischler, Povilas Daniušis, Bastian Steudel, and Bernhard Schölkopf. Information-geometric approach to inferring causal directions. *Artificial Intelligence*, 182:1–31, 2012.

Judea Pearl. *Causality: models, reasoning and inference*, volume 29. Cambridge Univ Press, 2000.

Shohei Shimizu, Patrik O Hoyer, Aapo Hyvärinen, and Antti Kerminen. A linear non-gaussian acyclic model for causal discovery. *The Journal of Machine Learning Research*, 7:2003–2030, 2006.

Michal Valko, Alexandra Carpentier, and Rémi Munos. Stochastic Simultaneous Optimistic Optimization. In *30th International Conference on Machine Learning*, Atlanta, États-Unis, February 2013. URL http://hal.inria.fr/hal-00789606.

Appendix A. Causality challenge

Title: Training Gradient Boosting Machines using Curve-fitting and Information theoretic features for Causal Direction Detection.
Participant name, address, email and website: Spyridon Samothrakis, Diego Perez, https://github.com/ssamot/causality.
Task(s) solved: Kaggle Competition.
Reference: This paper.
Method: A combination of feature extraction from the sample data, Gradient boosting machines and StoSOO meta-optimisation.

- Preprocessing: Exploit Symmetries.
- Causal discovery: Gradient Boosting Machine, Curve fitting/Information theoretic features.
- Feature selection: Feature Ranking.
- Classification: Gradient Boosting Machine
- Model selection/hyperparameter selection: Cross-validation, Stochastic Simultaneous Optimistic Optimisation.

Results:

Dataset/Task	Score
Test Set	0.79957

Table 11.1: Result table.

- quantitative advantages: The method and ideas behind our method are relatively simple. We advocate a feature extraction strategy based on curve fitting + information theoretic features.
- qualitative advantages: There are some elements of novelty, mostly in the ideas behind extracting features and doing hyper-parameter optimisation.

Code and installation instructions can be found here: https://github.com/ssamot/causality