JMLR: Workshop and Conference Proceedings 3: 65-76 WCCI2008 workshop on causality

Random Sets Approach and its Applications

Vladimir Nikulin VLADIMIR.NIKULIN@SUNCORP.COM.AU
Suncorp, Actuary Department
Brisbane, QLD, Australia

Editors: I. Guyon, C. Aliferis, G. Cooper, A. Elisseeff, J.-P. Pellet, P. Spirtes, and A. Statnikov

Abstract

The random sets approach is heuristic in nature and has been inspired by the growing speed of computations. For example, we can consider a large number of classifiers where any single classifier is based on a relatively small subset of randomly selected features or random sets of features. Using cross-validation we can rank all random sets according to the selected criterion, and use this ranking for further feature selection. Another application of random sets was motivated by the huge imbalanced data, which represent significant problem because the corresponding classifier has a tendency to ignore patterns with smaller representation in the training set. Again, we propose to consider a large number of balanced training subsets where representatives from both patterns are selected randomly. The above models demonstrated competitive results in two data mining competitions.

Keywords: causal relations, random forest, boosting, SVM, CLOP, cross validation

1. Introduction

It is a well known fact that for various reasons it may not be possible to theoretically analyze a particular algorithm or to compute its performance in contrast to another. The results of the proper experimental evaluation are very important as these may provide the evidence that a method outperforms alternative approaches.

Feature selection (FS) represents a very essential component of data mining, as it will help to reduce overfitting and make prediction more accurate (see, for example, Nikulin (2006)). According to (Guyon et al., 2007) causal discovery may be regarded as a next step with the aim of uncovering causal relations between features and target variable. In many cases it is theoretically impossible to solve full graphical structure of all relations between features and target variable but it may be possible to uncover and approximate some essential relations. This knowledge will help to understand data better and will give some hints which methods will be more efficient.

A graphical model is a family of probability distributions defined in terms of a directed or undirected graph (Jordan, 2004). The nodes in the graph are identified with random variables, and joint probability distributions are defined by taking products over functions defined on connected subsets of nodes. By exploiting the graph-theoretic representation, the formalism provides general algorithms for computing conditional probabilities of interest.

In line with Bayesian Networks graphical semantics (Tsamardinos et al., 2004) every edge from a feature x_1 to a feature $x_2, x_1 \rightarrow x_2$, means that x_1 probabilistically and directly causes x_2, see Figure 1. Bayesian Networks represent the joint probability distribution. For

the given target variable y, the set of parents, children and spouses (i.e. parents of common children) is called the Markov Blanket (MB) of y. Markov Blanket as a set of features is sufficient in relation to y. Any other features becomes superfluous. Ideally, we would be interested to find MB and investigate its structure (see Section 2.1 for more details).

Usually, any dataset may be viewed as a matrix with two dimensions: 1) data entries and 2) features. In the Section 3.3 we consider application of the random sets (RS) approach to data-entries.

2. Methods

Let $\mathbf{X} = (\mathbf{x}_t, y_t), t = 1..n$, be a training sample of observations where $\mathbf{x}_t \in \mathbb{R}^\ell$ is ℓ-dimensional vector of features, and y_t is binary label: $y_t \in \{-1, 1\}$. Boldface letters denote vector, whose components are labeled using a normal typeface.

In practical situation the label y_t may be hidden, and the task is to estimate it using vector of features. Area under receiver operating curve (AUC) will be used as an evaluation and optimisation criterion.

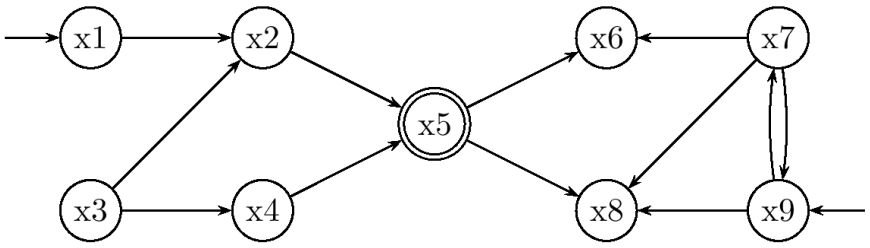

Figure 1: Illustrative Example.

According to Holland (1986) we shall assume that there are two causes of treatment, denoted by e (the experiment) and c (the control, which may be automatic or natural).

Fundamental problem of causal inference: it is impossible to observe the values of $y_t(e)$ and $y_t(c)$ on the same unit (that meant at the same time and for the same client) and, therefore, it is impossible to observe the effect of e on y.

For example, price of premium in insurance industry represents one of the most important features. Based on this price and available alternatives the customer will make a decision whether or not to renew an insurance contract. Suppose that the customer decided to accept renewal. In this case the Company would be interested to know decision if the price will be slightly higher. In an alternative case, if the customer decided to decline the proposed contract, the Company would be interested to know decision if the price will be slightly lower.

Classical randomized designs (Rubin, 1978) stand out as especially appealing assignment mechanisms designed to make inference for causal effects.

Let us consider another example where e represents a novel year-long study of arithmetics, c represents a standard arithmetic program, and target variable y is a score on a test at the end of the year. Obviously, for any particular student we can observe $y_t(e)$ or $y_t(c)$ but not both. Respectively, it appears to be natural to split randomly available field

of students into several groups where we can apply either e or c. Absolutely similarly we can formulate example with medical applications.

During the Causal Discovery competition participants were able to take into account some specific information and assumptions. By the given definitions[1], there are two types of data: purely artificial and semi-artificial. In the latter case, there are two types of features: 1) real features and 2) probes where the last ones were artificially created variables as a functions of real features and other probes. Probes may be manipulated in order to highlight the importance of the proper feature selection. The real features and the target variable are never manipulated. As a result, in semi-artificial systems, only non-causes of the target may be manipulated.

Definition 1 *Manipulations are actions or experiments performed by an external agent on a system, whose effect disrupts the natural functioning of the system.*

Consider the example of Figure 1 where x_5 is a target variable, which graphically represents a presumed semi-artificial system. Features $x_1 - x_4$ (to the left side from x_5) cannot be manipulated as they cause (directly or indirectly) x_5. On the other hand, all features $x_6 - x_9$ (to the right side from x_5) may be manipulated because they may be viewed as consequences of the target variable. Let us consider two particular examples. Firstly, suppose that x_7 is a probe. Then, x_6, x_8 and x_9 must have the same status of probes. Secondly, suppose that x_6 is a probe. In this case, $x_7 - x_9$ may be probes or real features.

In the example of the Figure 1 MB consists of 6 members: 1) parents (x_2 and x_4); 2) children (x_6 and x_8); 3) spouses (x_7 and x_9). We know that direct causal features (parents) cannot be manipulated. Respectively, it will be the most disappointing to loose these features as a result of the filtering process.

The main assumption: we assume that direct causal features (parents) have stronger influence on the target variable and, therefore, are more likely to be selected by the Algorithms 1 and 2.

Algorithm 1 Basic Iterative Feature Selection (BIFS)

1: Input: training sample $\mathbf{X}$ including set of all features S.
2: Select loss function (or evaluation criterion) D, algorithm g for the prediction and forward threshold parameter Δ_F.
3: Set Z $= \emptyset$.
4: Select feature $f \in S$, which optimizes criterion D applied to the prediction $g(f \cup Z)$.
5: Transfer feature f from S to Z in the case if improvement is sufficient (not smaller than Δ_F).
6: Stop the algorithm if there are no sufficient improvement, or no features left in S. Alternatively, goto step 4.

Remark 2 *Essentially, BIFS-process is not uniform: initially, overfitting is limited because size of the set Z is small, and we can apply very simple algorithm (like linear regression). Then, the size of Z will grow and we will be able to use more advanced technique (for example, SVM). But, overfitting will grow at the same time and application of the cross validation (CV) may become unavoidable.*

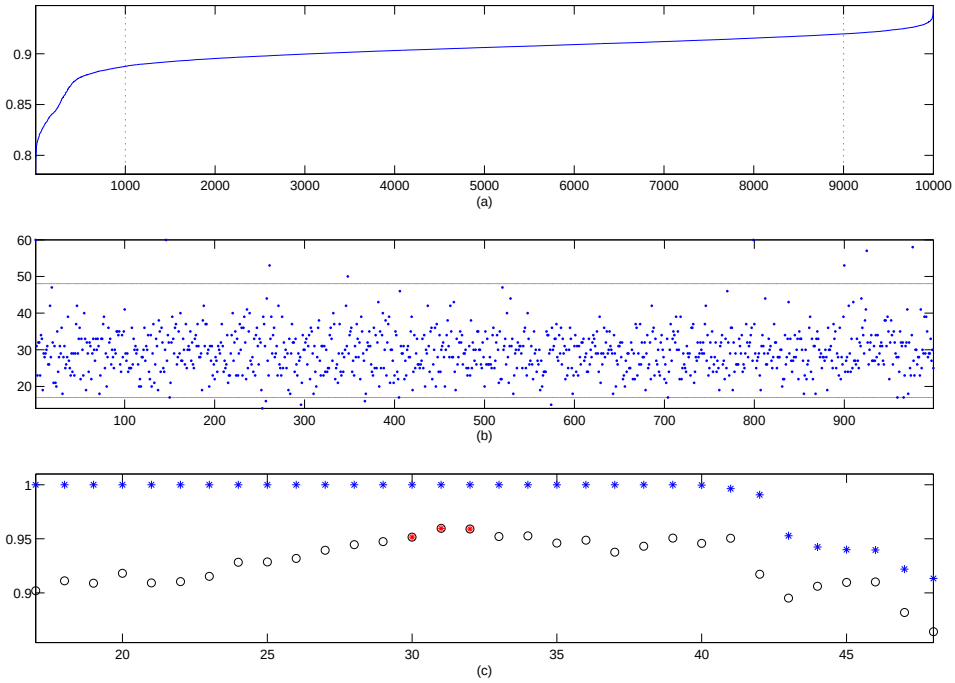

Figure 2: Illustration for RS-Algorithm 2 in the case of MARTI-set (see for more details Section 3). First, we evaluated 10000 random sets using AUC, (a) illustrates results sorted in an increasing order; (b) illustrates number of the occurrences for 999 features in the block B of 10% top performing random sets; (c) illustrates results of the secondary CV where features were selected according to the numbers of occurrences in the block B, stars correspond training results, circles correspond test results.

Algorithm 2 Random Sets (RS)

1: Evaluate long sequence of random subsets of features using CV.
2: Sort results in an increasing order (see Figure 2(a)).
3: Select block B of the best (or worst in the case of deductive strategy, see Remark 5) performing sets of features.
4: Compute for any feature number of occurrences in the block B (see Figure 2(b)).
5: Select range of occurrences for detailed investigation, which may be conducted using secondary CV (see Figure 2(c)).

Remark 3 *Note that we can add additional step (after step 5, Algorithm 1) with trimming (see Algorithm 3) or with test for independence within subset Z, see definition of HITON (Aliferis et al., 2003). On the one hand, this test will require additional computational time,*

1. http://www.causality.inf.ethz.ch/challenge.php

but, on the other hand, it may be viewed as barrier to prevent growth of Z. As a consequence, the whole procedure may quickly reach state of equilibrium (subject to the proper selection of forward and backward threshold parameters). Respectively, it will be stopped.

Remark 4 *Note that we can facilitate Algorithm 1 using projections method as it is described in (Stoppiglia et al., 2003).*

Remark 5 *By definition, random set τ represents a relatively small subset of features. In the case if classifier requires bigger than 50% of all features it will be better to apply deductive strategy. That means, we will form a new subset of features $\gamma = S \setminus \tau$, which will be used in the Step 1 of the algorithm 2. Accordingly, we can classify subset τ as the worst if subset γ was classified as the best.*

The functioning of the proposed Algorithm 2 is uniform, in difference to the Algorithm 1. As a consequence, we can apply any base algorithm, which appears to be appropriate for the given data (see for more details Figure 2 and Section 3).

Algorithm 3 Trimming

1: Input: training sample $\mathbf{X}$ including set of all features S.
2: Select loss function (or evaluation criterion) D, algorithm g for the prediction, block of features $Z \subset S$ and backward threshold parameter Δ_B.
3: Compute $\alpha = D(g(Z))$ - initial optimal value of the target function.
4: Select feature $f \in Z$, which optimises criterion D applied to the prediction $g(Z \setminus f)$.
5: Compute $\beta = D(g(Z \setminus f))$ - new optimal value of the target function;
6: $Z := Z \setminus f$ if $|\alpha - \beta| < \Delta_B$, $\alpha = \beta$, and goto Step 4 if $Z \neq \emptyset$;
7: stop the algorithm if $Z = \emptyset$ or $|\alpha - \beta| \geq \Delta_B$.

Algorithm 3 may be used independently or in conjunction with Algorithms 1 or 2. The role of threshold parameters Δ_F and Δ_B is important and similar to the role of regularisation. Essentially, the online combination of the Algorithms 1 and 3 represents a modification of the Iterative Associative Markov Blanket (IAMB) algorithm (Aliferis et al., 2002).

2.1 Bayesian framework for the Markov blanket construction

Binary data-sets represent an ideal case for the illustration of the concepts of the Bayesian approach. Suppose that events $x_6 = 1$ and $y = 1$ represent coughing and lung cancer (see Figure 1). Using available data we can calculate two empirical probabilities:

$$1)\mathbb{P}(y = 1, x_6 = 1|y = 1); \quad 2)\mathbb{P}(y = 1, x_6 = 1|x_6 = 1).$$

We can expect that the first probability will be significant in difference to the second probability. As a next step, we can check stability of the values using standard bootstrapping technique. Based on the results of our analysis, we can make conclusion that there is a relation between y and x_6 where y is a parent (lung cancer) and x_6 is a child (coughing). However, there may be some complications. For example, x_6 may be a child of a child. The target of the Algorithm 3 is to detect and to resolve such problems. Similarly, we can

investigate relations of the target variable with all other features. As an outcome we will obtain subset of features which have direct relations with target variable either as children or as parents where the last ones are the most important. Finally, we can detect field of spouses considering any particular child as a target variable.

Similarly, we can consider any discreet features. Note that consideration of continuous (numerical) features may be much more difficult. In this case we can apply transformation with several splitters for any particular feature. It works similarly to the method of classification trees.

3. Experiments

The list of 6 datasets which were used during WCCI-2008 Causal Discovery competition is given in the Table 1.

The case of *MARTI*-set appears to be the most complicated because of the 25 given calibrants: the training set was perturbed by a zero-mean correlated noise model. As far as the test sets have no added noise, we used linear regression model in order to filter noise from the training set. Then, we considered sequence of 10000 sets with 40 randomly

Data	# Train (positive)	# Test	ℓ	Method	Software
LUCAS	2000 (1443)	10000	11	neural+gentleboost	MATLAB-CLOP
LUCAP	2000 (1443)	10000	143	neural+gentleboost	MATLAB-CLOP
REGED	500 (59)	20000	999	SVM-RBF	C
SIDO	12678 (452)	10000	4932	binaryRF	C
CINA	16033 (3939)	10000	132	adaBoost	R
MARTI	500 (59)	20000	1024	svc+standardize	MATLAB-CLOP

Table 1: *List of datasets including sizes and main methods plus software which were used during the competition.*

selected features (without repeats). Based on some preliminary experiments, we applied *svc* function from MATLAB-CLOP (deductive strategy: means, we used all features without features from random set, see Remark 5) for the evaluation. We sorted all sets in an increasing order (see Figure 2(a)) according to the meanTestAUC (used CV with 20 folds), and computed number of occurrences for any particular feature according to the block B of the worst 10% sets (see Figure 2(b)). Based on the visual consideration, we conducted detailed examination of the subinterval [17..48]. In this experiment features were selected according to the condition: $n_j \geq a, a \in [17..48]$ where n_j is number of repeats in the block B for the feature j.

Figure 2(c) illustrates the final CV experiment where blue-stars correspond to the training and black-circle to the test results. Some marginal numerical values: 17) 994, 0.9019; 31) 410, 0.9597; 48) 8, 0.8641 where first and second numbers indicate number of the selected features and meanTestAUC. We can see some decline after point $a = 31$ (as a consequence of overfitting). Accordingly, the cases of $a \in [30..32]$ may be suitable for the submission in the normal situation when training and test samples have the same probability distribution.

Data	Submission	CASE0	CASE1	CASE2	Mean	Rank
REGED	vn14	0.9989	0.9522	0.7772	0.9094	4
SIDO	vn14	0.9429	0.7192	0.6143	0.7588	6
CINA	vn14a	0.9764	0.8617	0.7132	0.8504	2
MARTI	vn14	0.9889	0.8953	0.7364	0.8736	4
LUCAS	vn1	0.9209	0.9097	0.7958	0.8755	validation
LUCAP	vn10b+vn1	0.9755	0.9167	0.9212	0.9378	validation
CINA	vn1	0.9765	0.8564	0.7253	0.8528	all features
CINA	vn11	0.9778	0.8637	0.718	0.8532	CE

Table 2: *Results of the final submissions in terms of AUC (first 4 lines). LUCAS and LUCAP were used for validation and learning only.*

It is interesting to note that in the initial submission *"vn1"* for CINA-set we used all 132 features. The best CINA-result was obtained using committee of experts (CE) method (*"vn11"*) applied to the following 7 submissions: *"vn1"* and *"vn10-vn10e"*.

Random Forest (Breiman, 2001) model proved to be the most suitable in the case of SIDO-set. We used RF model with 1000 trees where 70 randomly selected features were used for any splitter. Then, we computed number of occurrences in the RF-object for any particular feature. These occurrences were used for further feature selection. For example, we used in the final submission 1030 features for SIDO0, 517 features for SIDO1 and only 203 features for SIDO2.

Data	Submission	# features	Fscore	TrainAUC	TestAUC
REGED1	vn14	400	0.7316	1	0.9522
REGED1	vn11d	150	0.8223	1	0.9487
REGED1	vn1	999	0.5	1	0.9445
REGED1	vn8	899	0.5145	1	0.9436
MARTI1	vn12c	500	0.5784	1	0.8977
MARTI1	vn14	400	0.5554	1	0.8953
MARTI1	vn3	999	0.5124	1	0.8872
MARTI1	vn7	899	0.4895	1	0.8722
SIDO0	vn9	203	0.5218	0.9684	0.946
SIDO0	vn9a	326	0.536	0.9727	0.9459
SIDO0	vn1	1030	0.5785	0.9811	0.943
SIDO0	vn14	527	0.5502	0.9779	0.9429

Table 3: *Some additional results.*

Lists of 100 manipulated features were given in the cases of REGED1 and MARTI1, but, according to our experience, this information was not really helpful, see Table 3 where submissions "vn1" (REGED) and "vn3" (MARTI) represent cases with all features. After removal of the manipulated features, test-results were slightly worse: see submissions "vn8" (REGED) and "vn7" (MARTI). Also, we have noticed surprising fact that value of Fscore for "vn7" is smaller comparing with Fscore for "vn3" (MARTI). Respectively, FS was

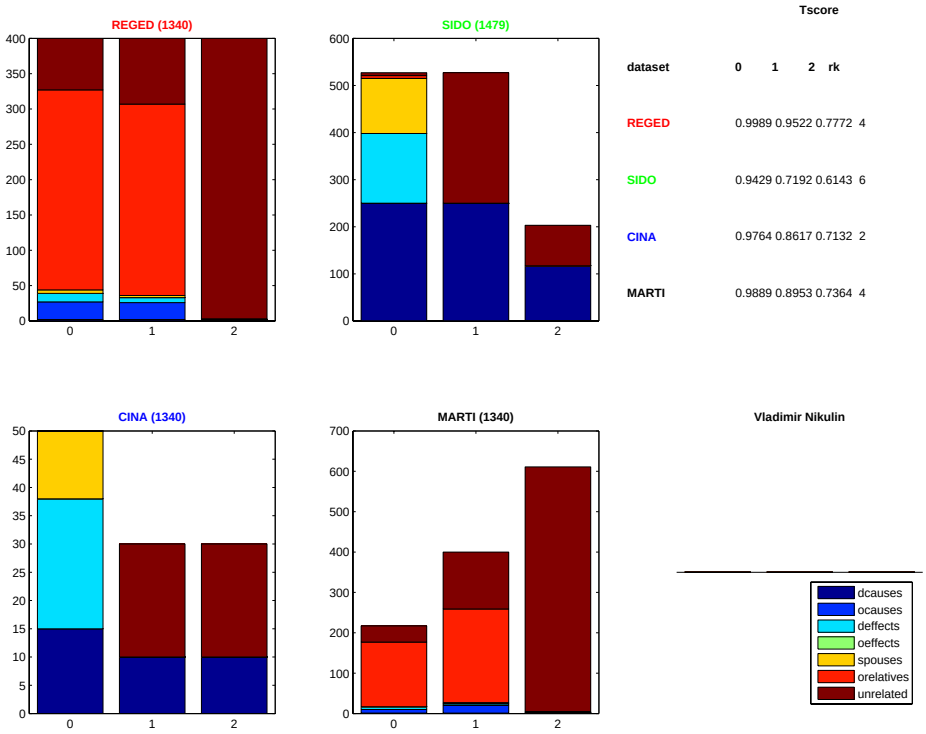

Figure 3: Histograms of selected features (evaluated by the competition organizers) where dcause: direct cause, deffect: direct effects, ocauses: other causes, oeffects: other effects, spouses: parent of a direct effect, orelatives: other relatives, unrelated: completely irrelevant.

conducted in the space of all features for the final submission "vn14". In both cases of REGED and MARTI we used 400 features including 33 manipulated features for REGED and 42 manipulated features for MARTI. It appears that in case of SIDO0 Fscore reflects rather relations with TrainAUC but not with TestAUC, see submissions "vn9" and "vn1".

Remark 6 *As a feedback the participants were able to view colour of their submission. For example, all TestAUC for SIDO0 in the Table 3 were green (means top 25% of all current results). Generally, this feedback appears to be too rough, and, definitely, cannot be accepted as a sufficient in the case when distributions of the training and test datasets are different.*

During competition we made the following number of full submissions (given in brackets): CINA(8), REGED(12), MARTI(32) and SIDO(6) plus some partial submissions. We did not use an opportunity of nested submissions (that means picking up the best out of the table of results) during the competition, and have found afterwards that this option may give significant advantage (see Table 4). Note that the methods which we used did not

orient edges and cannot discover Market blanket as an expected outcome. Also, we did not use HITON-algorithm or similar as a component of our methods.

Figure 3 was downloaded from the web-site of the competition. Similar histograms of Jianxin Yin and Prof. Zhi Gengs Group (who won best overall contribution award) demonstrate much larger proportions of *dcauses* and *ocauses*. However, Jianxin Yin and his team did not produce significant improvement in terms of average AUC.

We have found that our results against unmanipulated datasets are quite competitive (see column "CASE0" in the Table 2). In particular, CINA0-result is the best.

3.1 Post-challenge submissions

Using an opportunity of post-challenge submissions we were able to improve all results against manipulated sets significantly. It is interesting to note that very competitive results

Data	Method	NoF	Fscore	TrainAUC	TestAUC	NoS	BestChAUC
REGED1	LR	8	0.7133	0.9855	0.9861	9	0.9787
REGED2	LR	5	0.9985	0.9571	0.9467	8	0.8392
REGED1	Exp.	8	0.7133	0.9885	0.9867	10	0.9787
REGED2	Exp.	5	0.9985	0.9605	0.9513	9	0.8392
SIDO1	RF	128	0.5348	0.8681	0.7512	28	0.7532
SIDO2	RF	128	0.5348	0.8681	0.7359	28	0.6684
CINA1	AdaBoost	4	0.5455	0.8758	0.8694	5	0.8691
CINA2	AdaBoost	4	0.5455	0.8758	0.872	5	0.8157
MARTI1	LR	4	0.6429	0.8433	0.9407	8	0.947
MARTI2	LR	3	0.9995	0.7542	0.8049	9	0.7975
MARTI1	Exp.	4	0.6429	0.845	0.9469	9	0.947
MARTI2	Exp.	3	0.9995	0.7613	0.8296	10	0.7975

Table 4: *Results of the post-challenge submissions against manipulated sets where the following abbreviations were used: 1) NoF - number of used features; 2) NoS - number of previous submissions; 3) LR - linear regression with squared loss function; 4) RF - random forest; 5) Exp. - optimisation with exponential loss function (1); 6) BestChAUC - best challenge AUC.*

for REGED and MARTI-sets (see Table 4) were produced using the most simplest linear regression. Regularization was not necessary here because of the ultimate reduction of the number of features. The property when TrainAUC is smaller comparing with TestAUC (MARTI-set) may be viewed as a very interesting side effect of manipulation. Also, we were trying to use AdaBoost algorithm against data with the same feature selection as in the Table 4 but results were very poor.

Based on our experience, feature selection was the most important in order to achieve all results of the Table 4. Also, we have found that the Algorithm 1 is particularly efficient if we have prior information that the number of required features should be very small, and, consequently, classification algorithm may be very simple. It appears that design of the SIDO1 and SIDO2-sets was essentially different. Respectively, the number of features

in the best submission was a quite significant, and the numbers of previous submissions were three times greater comparing with other sets.

3.2 An exponential loss function

The following exponential loss function

$$\exp\{-\rho \cdot y_t \cdot u_t\},\ u_t = \sum_{j=1}^{\ell} w_j \cdot x_{tj},\ \rho > 0, \tag{1}$$

appears to be more natural comparing with squared loss function which over-punish large values of the decision function. However, application of the loss function (1) may not be simple because we cannot optimize step size in the case of the gradient-based optimization. Respectively, we will need to maintain low level of the step size in order to ensure stability of the algorithm. As a result, convergence of the algorithm may be very slow. In the cases of REGED or MARTI training sets with 3-8 features (see Table 4) we don't need to be worried about time problem: 100000 iterations until full convergence were conducted within 3min.

3.3 UCF-2008 data-mining competition

This recent competition was organized by the department of statistics and actuarial science of the University of Central Florida[2].

The available data are strongly imbalanced: 858620 (where 9737 positive and 848883 negative) units for training (labeled) and 95960 for testing (unlabeled). Any data-entry includes label, id and 61 features which are not necessarily numerical. Using special Perl software we transformed data into sparse format with 530 binary features. Then, we split the labeled data into 2 parts for training (90%) and testing (10%), and applied $k = 1000$ balanced training subsets where representatives from the larger pattern were selected randomly. As an outcome, the system produced matrix of linear regression coefficients M where rows represent random subsets and columns represent features. Based on this matrix we made an assessment of how stable is influence of the particular features. It is proposed to keep in the model only features with stable influence (the ratio of the mean to StDev must be bigger or equal comparing with selected value of threshold parameter $\Delta = 0.5$). As a consequence, number of binary features was reduced to 320.

Our entry produced $AUC = 0.6645$ - third best result.

3.3.1 Uncertainty estimation

Using above matrix of regression coefficients M we can estimate uncertainty associated with any particular data entry. First, we compute k predictions where k is number of random sets. Then, we can measure the corresponding standard deviation or empirical probabilities of deviation from the sample mean for any given margin.

2. http://dms.stat.ucf.edu/competition08/home.htm

3.4 Computation Time and Used Hardware

A Dell desktop with 3GB RAM, 2.4GHZ INTEL CORE 2 DUO, was used for the most of computations. For example, experiment with 10000 random sets as it is described in the Section 3 took about 17 hours according to the special program written in C. We spent about 2 hours in order to generate random forest for SIDO-set with 1000 trees where each tree had up to 8 levels of depth.

4. Concluding Remarks

Computational statistics is a relatively new scientific area, which may be viewed as one of the most promising areas of contemporary science. High technologies are generating large data sets and new problems, which must be addressed. Data mining competitions represent a rapidly growing and very important part of computational statistics. Practically any large commercial company in the world has data mining department, which is responsible for data analysis and modeling. Additionally, companies are hiring consultants in order to produce an alternative solutions and check effectiveness of their own results. These activities may be quite expensive, but unavoidable.

Generally, practical experience is the best way to learn, and participation in data mining competitions may be useful for wide range of researchers including academics, consultants and students in particular.

We understand that Causal Discovery competition was motivated by some interesting theoretical papers. However, in practical applications we are dealing not with pure probability distributions, but with mixtures of distributions, which reflect changing in time trends and patterns. Accordingly, it appears to be more natural to form training set as an unlabeled mixture of subsets derived from different (manipulated) distributions, for example, REGED1, REGED2,..,REGED9. As a distribution for the test set we can select any "pure" distribution.

Another point, "blind learning" (case when training and test data-sets have different distributions) appears to be interesting as a form of gambling. But in most practical applications proper organized validation is the most important. Respectively, it will be good to apply traditional strategy: split randomly available test-set into 2 parts 50/50 where one part will be used for validation, second part for the testing.

We considered in this paper several methods which may be used independently or in conjunction. We cannot expect that any of the methods may demonstrate an absolute superiority against the others. Therefore, performance of the particular method depends on the dataset, and the main strength of our approach rests on flexibility.

Acknowledgments

I would like to thank actuarial department of Suncorp for the valuable support. The paper was improved thanks to the comments and suggestions of the reviewers and organizers of the WCCI-2008 Causal Discovery competition.

References

C. Aliferis, I. Tsamardinos, and A. Statnikov. Large-scale feature selection using markov blanket induction for the prediction of protein-drug binding. In *Technical Report DSL 02-06*, 2002.

C. Aliferis, I. Tsamardinos, and A. Statnikov. HITON: a novel Markov blanket algorithm for optimal feature selection. pages 21–25. AMIA 2003, 2003.

L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

I. Guyon, C. Aliferis, and A. Elisseeff. Causal feature selection. In H. Liu and H. Motoda, editors, *Computational Methods of Feature Selection*. Chapman and Hall, 2007.

P. Holland. Statistics and causal inference. *Journal of the American Statistical Association*, 81(396): 945–960, 1986.

M. Jordan. Graphical model. *Statistical Science*, 19(1):140–155, 2004.

V. Nikulin. Learning with mean-variance filtering, SVM and gradient-based optimization. In *International Joint Conference on Neural Networks, Vancouver, BC, Canada, July 16-21*, pages 4195–4202. IEEE, 2006.

D. Rubin. Bayesian inference for causal effects: the role of randomisation. *The Annals of Statistics*, 6(1):34–58, 1978.

H. Stoppiglia, G. Dreyfus, R. Dubois, and Y. Oussar. Ranking a random feature for variable and feature selection. *Journal of Machine Learning Research*, 3:1399–1414, 2003.

I. Tsamardinos, C. Aliferis, and A. Statnikov. Time and sample efficient discovery of Markov blankets and direct causal relations. In *Technical Report DSL 03-06*, 2004.