Appendix I:

Causation and Prediction Challenge Fact Sheets

Causation and Prediction Challenge: FACT SHEET

Title: Feature selection, redundancy elimination, and gradient boosted trees.
Author: Alexander Borisov
Address: INTEL Corporation, Advanced Analytics team
Email: alexander.borisov@intel.com
Acronym of your best entry: ACE+GBT
Performance graphs generated by the organizers:
http://clopinet.com/isabelle/Projects/WCCI2008/Reports/Alexander_Borisov.html

References:

[1] Borisov A., K.Torkkola, Tuv E. (2006) "Best Subset Feature Selection for Massive Mixed – Type Problems". 7th International Conference on Intelligent Data Engineering and Automated Learning, IDEAL-2006, Lecture Notes in Computer Science Series, Vol. 4224, 1048-1056, Springer 2006.
[2] Tuv E., Borisov A., Runger G., Torkkola K. "Best Subset Feature Selection with Ensembles, Artificial Variables, and Redundancy Elimination". Submitted to Journal of Machine learning Research, 2008

Method:
No preprocessing was done.
Feature selection method contains 2 steps. For unbalanced datasets, all classifiers (RF, GBT) use stratified sampling to compensate, i.e. for each tree in ensemble 60% samples of rare class and same quantity of frequent class are selected as input.
1. Feature selection using ensemble classifiers (ACE FS). Contrast variables that are permutation of original features are added. Importance of each variable in RF ensemble is compared versus importance of probes using t-test over several ensembles. Variables that are more important in statistical sense then most of probes are selected as important. Variables are ordered according to sum of Gini index reduction in tree splits.
2. Variable masking is estimated on important variables with GBT ensemble using surrogate splits (if more important variable has surrogate on less important one, the second variable is masked by the first). Again, statistically significant masking pairs are selected, then subset of mutually non-masked variables with high importance is selected
3. Effect of found variables is removed using RF ensemble.
Steps 1-3 are repeated until no more important variables remain.
Variables are sorted by cumulative variable importance (computed as usual for ensemble of trees, i. e importance of feature is sum of split weights on this feature) in ensembles constructed on step 3. Then top 1, 2, 4,… and so on variables are used to build GBT model. For more than 100 features we used embedded feature selection in GBT that reduces the running time. The idea is that with redundant feature elimination probes will be recognized as redundant, and will have zero or very small importance.

The following parameters of GBT were selected empirically for all datasets:
800 iterations, tree depth = 8, shrinkage = 0.01
For FS, #of trees in series = 50, #series = 20, importance and masking quantile = 0.75, tree depth = 6.

Results:

Table 1: Result table. The two stars next to the feature number indicate that the submission included a sorted list of features and multiple results for nested subsets of features. Top Ts refers to the best score among all valid last entries made by participants. Max Ts refers to the best score reachable, as estimated by reference entries using the knowledge of true causal relationships not available to participants.

Dataset	Entry	Method	Fnum	Fscore	Tscore (Ts)	Top Ts	Max Ts	<Tscore>	Rank
REGED0	171	ACE+GBT	512/999 **	0.6969	0.9996±0.0009	0.9998	1		
REGED1	171	ACE+GBT	512/999 **	0.6838	0.9095±0.0038	0.9888	0.998	0.8331	8
REGED2	171	ACE+GBT	8/999 **	0.9107	0.5902±0.0059	0.86	0.9534		
SIDO0	171	ACE+GBT	256/4932 **	0.4653	0.9337±0.0076	0.9443	0.9467		
SIDO1	171	ACE+GBT	2048/4932 **	0.468	0.6908±0.0136	0.7532	0.7893	0.7333	8
SIDO2	171	ACE+GBT	4932/4932 **	0.468	0.5756±0.0130	0.6684	0.7674		
CINA0	171	ACE+GBT	64/132 **	0.6312	0.9755±0.0029	0.9765	0.9788		
CINA1	171	ACE+GBT	64/132 **	0.6085	0.8236±0.0048	0.8691	0.8977	0.8328	5
CINA2	171	ACE+GBT	64/132 **	0.6085	0.6993±0.0043	0.8157	0.891		
MARTI0	171	ACE+GBT	512/1024 **	0.4841	0.8872±0.0050	0.9996	0.9996		
MARTI1	171	ACE+GBT	32/1024 **	0.5188	0.7005±0.0061	0.947	0.9542	0.7638	7
MARTI2	171	ACE+GBT	128/1024 **	0.5998	0.7036±0.0063	0.7975	0.8273		

Quantitative advantages

Method is fast (~a minute for one FS iteration on largest dataset)

Time complexity is proportional to (Fsel+Fimpvar)*N*logN*Ntrees*Nensembles*Niter + Niter*Fimpvar^2,

Niter - #of iteration of ACE FS algorithm always < 10, usually 3-4

Nensembles = 20 (number of ensembles for t-test)

Ntrees = 50 (number of trees in RF or ensemble)

N - number of samples,

Fsel = number of selected important variabless per tree split (sqrt(total number features) or less)

Fimvar – total number of selected important variable.

Works with any variable types, mixed values, requires no preprocessing.

Qualitative advantages .

Requires no investigation of causal structure.

It is not a push-button application. ACE is a part of internally developed at Intel machine learning toolset called IDEAL not available for external usage.

Keywords:

- Preprocessing or feature construction: no.
- Causal discovery: indirect trough redundant feature elimination strategy in ACE method. Probes should be more likely to be redundant and go at the end of the feature sorted list..
- Feature selection: embedded feature selection using tree ensembles.
- Classifier: RF, GBT (tree ensembles)
- Hyper-parameter selection: used defaults that work well on most data sets.
- Other: ensemble method.

Causation and Prediction Challenge: FACT SHEET

Title: Regularized and Averaged Selective Naïve Bayes Classifier
Author: Marc Boullé,
Address: France Telecom R&D, 2, avenue Pierre Marzin, 22307 Lannion cedex – France
Email: marc.boulle@francetelecom.com
Acronym of best entry: SNB(CMA), IID assumption
Performance graphs generated by the organizers:
http://clopinet.com/isabelle/Projects/WCCI2008/Reports/MB.html

References

[1] M. Boullé. Compression-Based Averaging of Selective Naive Bayes Classifiers. Journal of Machine Learning Research, 8:1659-1685, 2007
[2] M. Boullé. MODL: a Bayes optimal discretization method for continuous attributes. Machine Learning, 65(1):131-165, 2006

Method

IID assumption

The method is based on the IID assumption and ignores causal discovery.
Although its results make sense only on the initial datasets, it was also applied on the manipulated datasets to challenge the causal methods.

Noise filtering for MARTI

The data samples of MARTI were preprocessed to remove the correlated noise as follows:

- The 2-dimensional nature of the patterns was reconstructed using the variable indices
- The low frequency noise was removed by convolving the image thus obtained with a 2-d Gaussian filter to obtain the "background", then subtracting this background from the image. Specifically, we used the kernel ker=[1 4 6 4 1]'*[1 4 6 4 1]; ker=ker./sum(sum(ker)); without tuning its width. Better results might be obtained with other kernels or bay adjusting the width.
- To alleviate border effects, the image was first extrapolated by tiling the borders with average values of near border variables.
- To alleviate the problem of high intensity outliers, we detected points whose value was more that one standard deviation away from the mean of their neighbors and replaced them by that mean before computing the background.
- Finally, a bias value was added to the resulting filtered image such that the average of the calibrants is the same as that is test data (namely 1).

Compression-based averaging of selective naïve Bayes classifiers

Our method is based on the naïve Bayes assumption, and incorporates optimal preprocessing, feature selection and model averaging as follows:

- All the input features are preprocessed using the Bayes optimal MODL discretization method, which results in a reliable and accurate estimation of the univariate class conditional probabilities.
- Feature selection is performed using a Bayesian approach to find a trade-off between the number of selected features and the performance of the selective naïve Bayes classifier: this provides a regularized feature selection criterion. The feature selection search is performed using alternate forward selection and backward elimination searches on randomly ordered feature sets: this provides a fast search heuristic, with super-linear time complexity with respect to the number of instances and features.
- The method exploits a variant of feature selection: feature "soft" selection. Whereas feature "hard" selection gives a "Boolean" weight to the features according to whether they selected or not, the method gives a continuous weight between 0 and 1 to each feature. This weighing schema of the features comes from a new classifier averaging method, derived from Bayesian Model Averaging, with a logarithmic smoothing of the posterior distribution of the models.

Advantages

- Bayesian regularization technique (for preprocessing and feature selection): all the available data is used for training, with no need for validation or cross-validation
- fully automatic
- highly scalable (train and deploy)
- accurate and reliable
- easy interpretation
- compute the posterior probabilities

Limitations

- the naïve Bayes assumption might be harmful is no subset of variables in the initial representation is compliant with the conditional independence assumption: this can be leveraged by feature construction to extend the representation space
- no causal discovery

Results

For each of the four datasets, one single model was trained and applied on the initial test set (0) and the two manipulated test sets (1 and 2).

The results are very good on the initial test sets, which conform to the IID assumption: our method gets the best Tscore on REGED0 and CINA0, and is within 1% of the best performance for the two other datasets.

Surprisingly, the results are good on some tests sets 1, with the best Tscore on REGED1 and CINA1.

This might be explained by two features of our method:

- the optimal preprocessing is highly reliable: any input noise variable is almost surely detected as irrelevant and discarded
- the model averaging accounts for the uncertainty on model selection: whereas one single maximum a posteriori (MAP) model might select a wrong subset of variables with respect to causation, averaging a large number of models leverages the effect of irrelevant features

Not surprisingly, the results are very poor on the test sets 2, which are heavily manipulated. Our method based on the IID assumption is clearly outperformed by the causal methods.

Table 1: Result table. The star following the feature number indicates that the feature set was sorted. Top Ts refers to the best score among all valid last entries made by participants. Max Ts refers to the best score reachable, as estimated by reference entries using the knowledge of true causal relationships not available to participants.

Dataset	Entry	Method	Fnum	Fscore	Tscore (Ts)	Top Ts	Max Ts	<Tscore>
REGED0	321	SNB(CMA), IID assumption	122/999 *	0.8352	1.0000±0.0002	1	1	
REGED1	321	SNB(CMA), IID assumption	122/999 *	0.7946	0.9980±0.0015	0.998	0.998	0.8462
REGED2	321	SNB(CMA), IID assumption	122/999 *	0.991	0.5407±0.0061	0.86	0.9534	
SIDO0	321	SNB(CMA), IID assumption	1592/4932 *	0.6831	0.9297±0.0070	0.9443	0.9467	
SIDO1	321	SNB(CMA), IID assumption	1592/4932 *	0.3922	0.6337±0.0132	0.7532	0.7893	0.7104
SIDO2	321	SNB(CMA), IID assumption	1592/4932 *	0.3922	0.5678±0.0129	0.6684	0.7674	
CINA0	321	SNB(CMA), IID assumption	90/132 *	0.8913	0.9788±0.0029	0.9788	0.9788	
CINA1	321	SNB(CMA), IID assumption	90/132 *	0.4542	0.8977±0.0043	0.8977	0.8977	0.8694
CINA2	321	SNB(CMA), IID assumption	90/132 *	0.4542	0.7318±0.0043	0.8157	0.891	
MARTI0	386	SNB(CMA), IID assumption (F)	22/1024 *	0.7097	0.9848±0.0031	0.9996	0.9996	
MARTI1	386	SNB(CMA), IID assumption (F)	22/1024 *	0.6716	0.8891±0.0043	0.947	0.9542	0.8869
MARTI2	386	SNB(CMA), IID assumption (F)	22/1024 *	0.9936	0.7868±0.0058	0.7975	0.8273	

Code

Our implementation was done in C++.
The software is available as a shareware on http://perso.rd.francetelecom.fr/boulle/.

Keywords

- Preprocessing or feature construction: Bayes optimal discretization
- Causal discovery: none
- Feature selection: Bayesian regularization, fast forward backward feature selection
- Classifier: naive Bayes, compression-based model averaging
- Hyper-parameter selection: none, automatic

Causation and Prediction Challenge: FACT SHEET

Title: A Strategy for Making Predictions Under Manipulation

Author, address, email:

Laura Brown, Eskind Biomedical Library 4th floor, 2209 Garland Ave.
Nashville, TN 37232 USA laura.e.brown@vanderbilt.edu
Ioannis Tsamardinos, FORTH-ICS, N. Plastira 100, Vassilika Vouton GR-700 13
Heraklion Crete, GREECE tsamard@ics.forth.gr

Acronym of your best entry: final test
Performance graphs generated by the organizers:
http://clopinet.com/isabelle/Projects/WCCI2008/Reports/LEBYT.html
Complete paper:
A Strategy for Making Predictions Under Manipulation
Laura E. Brown and Ioannis Tsamardinos; JMLR W&CP 3:35-52, 2008.

References:

Bach's – Bach, F.R. and Jordan, M.I. NIPS, 2002
FCI – Spirtes, P. et al. 1993
HITON – Aliferis, C.F. et al. AMIA, 2003
MMHC – Tsamardinos, I. et al. Machine Learning, 2006
MMPC, MMMB – Tsamardinos, I. et al. SIGKDD, 2003
RFE – Guyon et al. Machine Learning, 2002
Regions of Interest – Tsamardinos et al., Tech Report DSL-03-02, DBMI, Vanderbilt University

Method:

Preprocessing: The preprocessing was tailored to each data set. For the REGED data set each variable was normalized so its mean was zero and standard deviation was one. For the SIDO data set, the variables were binary and no preprocessing was performed. For the CINA data set, variables that were not binary were treated as continuous and normalized; binary variables were all set to values of zero and one. For the MARTI data set, the calibrant variables were used to fit a spline across the training array estimating the correlated noise model. The estimated noise was then subtracted from the training samples.

Causal discovery: We addressed the following problems in turn (a) finding the Markov Blanket of the target even under some non-faithfulness conditions (e.g., parity functions) (b) reducing the problems to a size manageable by subsequent algorithms (c) identifying and orienting the network edges (d) identifying causal edges (i.e. not confounded) and (e) selecting the causal Markov Blanket of the target in the manipulated distribution.

Once the initial data sets have been pre-processed, the next step of our procedure was to identify the Markov Blanket (MB) from the non-manipulated data sets, i.e., the parents, children, and spouses of the target. Several variable selection techniques, mostly causally-based, were applied to this problem in order to both identify the MB and also attempt to gain insight into the predictive variables in each domain. The published methods used included MMPC (for identifying the parents and children of a target variable, PC(T)), MMMB (for identifying the Markov Blanket of a target variable), HITON-MB (for identifying the Markov Blanket of a target variable), and RFE (variable selection method to identify predictive variables). All of the above causally-based methods assume that if a variable belongs in the neighbor's set of the target, it will have a detectable pairwise association with the target. RFE is able to additionally identify variables that participate in strong multivariate associations, even if they have no detectable pairwise association (e.g., parity functions). A new technique under development (recently submitted for publication), called Feature Space Markov Blanket (FSMB) combines kernel-based methods with causally-based methods to identify the neighbor's set in feature space, where multivariate associations may become pairwise associations. Any additional multivariate associations identified by FSMB were added to the Markov Blanket and participated in subsequent analysis. At this point, we know that our Markov Blanket set contains all variables need for calculation of the Causal Markov Blanket in any *manipulated* distribution (plus false positives depending on the type of manipulations).

In the second step, starting from the above Markov Blanket we identified the skeleton structure of the Bayesian Network around the target variable recursively using the MMPC algorithm, up to three edges away from the target. This region of interest makes it practical to apply causal algorithms that cannot scale up to the sizes of all the networks in the challenge. There are theoretical reasons why a network region of depth 3 allows most inferences about the orientation of the edges to be made. The idea of region learning was first described in Tsamardinos et al. 2003 (DSL-03-02). Further theoretical and experimental results are about to be submitted.

In the third step of our analysis we tried to orient the edges and discover whether an edge appears in the network due to a hidden confounder. The orientations of the edges and the confounded edges are necessary to identify the Causal Markov Blanket of the target, i.e., the Markov Blanket in the manipulated distribution. For the case of continuous or mixed data, an adaptation of Bach's algorithm was used. For the case of binary data, MMHC was used to find the top scoring network. The final network was converted into a PDAG to find the orientation of the compelled edges. To obtain suspected hidden confounders we used the FCI algorithm and developed our own extension of the Y-structures' identification algorithm (see Mani, et. al. UAI 2006) for the purposes of the challenge. Our extension is based on tests of independence rather than scoring and is able to handle confounders of the top variables in the Y-structure. We suspect this way we can identify Y-structures in more general conditions that those described in Mani.

For the non-manipulated data set, the Markov Blanket was selected as the variables to include in the variable list. The members were sorted first by parents, children, and spouses. For the manipulated data set where manipulations were known, the variable list consisted of the Causal Markov Blanket. This we defined to be the *effective* parents, the non-manipulated children, and the *effective* spouses of the target. The effective parents are the parent variables of the target that are still predictive of the target in the manipulated distribution. They are the direct *causes* of the target (i.e., parents found not to be confounded) plus the parents of the target that are not manipulated. The effective spouses are the effective parents of the non-manipulated children. For the manipulated data set where the manipulations were unknown, the variable list consisted of the causes of the target node, i.e., parents found not to be confounded. Weighting the evidence of the orientation of an edge and whether it is due to a hidden confounder or not by the above methods was done based on methods under development and submission for publication. For some edges the above methods failed to provide evidence whether they are causal or not (i.e., confounded) or about their direction. Thus, some guesswork was necessary that gave rise to different variable subsets that we have tried.

Classification and Model Selection: Once the variable list was determined for each problem and data set, a final classification model was trained using only the variables of the feature list. The models trained for this task were SVMs. An n-fold cross validation design was used to select the optimal parameters (type of kernel, kernel parameters, and C value). The value of n ranged from 5 to 10 based on the sample size available in the training sample. Once the best parameters were selected, a final SVM model was trained and used to predict the values for the test data sets.

Results:

Table 1: Result table. The star following the feature number indicates that the feature set was sorted. Top Ts refers to the best score among all valid last entries made by participants. Max Ts refers to the best score reachable, as estimated by reference entries using the knowledge of true causal relationships not available to participants. This entry obtained best average score for REGED among all valid last entries.

Dataset	Entry	Method	Fnum	Fscore	Tscore (Ts)	Top Ts	Max Ts	<Tscore>
REGED0	1491	final test	15/999 *	0.8571	0.9998±0.0010	0.9998	1	
REGED1	1491	final test	9/999 *	0.7851	0.9673±0.0036	0.9888	0.998	0.9423
REGED2	1491	final test	3/999 *	1	0.8600±0.0053	0.86	0.9534	
SIDO0	1491	final test	13/4932 *	0.5015	0.9230±0.0069	0.9443	0.9467	
SIDO1	1491	final test	4/4932 *	0.5003	0.6073±0.0027	0.7532	0.7893	0.6909
SIDO2	1491	final test	4/4932 *	0.5003	0.5426±0.0027	0.6684	0.7674	
CINA0	1491	final test	101/132 *	0.8496	0.9721±0.0031	0.9765	0.9788	
CINA1	1491	final test	5/132 *	0.4716	0.5113±0.0053	0.8691	0.8977	0.6015
CINA2	1491	final test	5/132 *	0.4716	0.3210±0.0025	0.8157	0.891	
MARTI0	1491	final test	24/1024 *	0.5869	0.9681±0.0037	0.9996	0.9996	
MARTI1	1491	final test	17/1024 *	0.5643	0.7837±0.0056	0.947	0.9542	0.8083
MARTI2	1491	final test	3/1024 *	0.4985	0.6730±0.0060	0.7975	0.8273	

The methods described above generally resulted in a compact variable list representing either the Markov Blanket or Causal Markov Blanket. The results on CINA were very low and are indicative to the inappropriateness of the statistical tests used in MMPC and MMMB when mixed data was used. The MMPC and MMMB algorithms have statistical tests provided for when the data is entirely binary or continuous (with a binary target); the mixed data set did not therefore match well to these methods.

The methods described above were implemented in Matlab. The MMPC, MMMB, and MMHC methods are available from the Causal Explorer library, www.dsl-lab.org (please note, we were in part the developers of these methods and may have slightly extended or modified the code from the precise implementation available in Causal Explorer). The

SVMs were created using the LibSVM software (http://www.csie.ntu.edu.tw/~cjlin/libsvm/). Our method combined many different approaches and is not currently available as a push-button application although we are working on automating this process.

Keywords:

- Preprocessing or feature construction: normalization
- Causal discovery: Bayesian Network,
- Feature selection: filter
- Classifier: SVM
- Hyper-parameter selection: K-fold cross-validation

Causation and Prediction Challenge: FACT SHEET

Title: Causation, Prediction, Feature Selection and Regularization
Author: Gavin Cawley
Address: School of Computing Sciences, UEA, Norwich, U.K.,
Email: gcc@cmp.uea.ac.uk
Acronym of your best entry: final models
Performance graphs generated by the organizers:
http://clopinet.com/isabelle/Projects/WCCI2008/Reports/Gavin_Cawley.html

Method:

Preprocessing: All continuous features are standardized to have zero mean and unit variance. For MARTI, the correlated noise was reduced by fitting a multi-output kernel ridge regression model, with a Gaussian RBF kernel, to the training that predicts the data as a function of the x and y co-ordinates. No special use was made of the calibration points, so the method was probably sub-optimal.

Causal discovery: The CausalExplorer package was used to detect feature sets representing the Markov blanket, direct causes + effects and direct causes only. This made use of the HITON_MB, PC and MMHC algorithms.

Feature selection: Sparse logistic regression with Bayesian regularization using a Laplace prior (BLogReg) was used for non-causal feature selection for comparison purposes. Models using the full feature set were also used to determine if regularization alone were sufficient. Also for the final submission using the multi-column format, some predictions are made by models using the feature weightings from other ridge regression models, so there is also a crude form of RFE used in some cases.

Classification: Ridge regression was used for MARTI, REGED and SIDO, in cases where there were more features than patterns, kernel ridge regression with a linear kernel was used for computational efficiency. For CINA, the BLogReg algorithm was used as this seemed to produce better results under cross-validation.

Model selection/hyper-parameter tuning: Virtual leave-one-out cross-validation using Allen’s PRESS statistic was used for hyper-parameter selection throughout.

Performance evaluation: The model skill for the un-manipulated datasets was performed via 100-fold repeated hold out experiments. The predictions submitted for the challenge represent the mean of the resulting ensemble of 100 models. The number features used by individual models is generally much smaller than that reported on the challenge website as not all features are used by all 100 models.

Fuller description of the methods used in "finalsubmission".

CINA:

Sparse logistic regression with Bayesian regularisation using a Laplace prior was used as the base classifier for all CINA datasets. An ensemble of models from 100-fold repeated hold-out was used to make predictions. All features were standardised to have zero mean and unit variance. HITON_MB (k=2, 'z' statistic, threshold=0.05) was used to pre-select the Markov blanket independently in each trial of the hold-out procedure for CINA0 and CINA1. HITON_MB (k=5, 'z' statistic, threshold=0.05) was used for CINA2. Not all of the features are used by all of the members of the ensemble, so the feature set contains some features that are barely used (if at all).

SIDO:

An optimally regularised ridge regression model with no feature selection was used for all datasets. An ensemble of models from 100-fold repeated hold-out was used to make predictions.

REGED:

Optimally regularised kernel ridge regression used as the base classifier for all datasets, all features standardised, again an ensemble of 100 models is used to make predictions.

REGED0: HITON_MB (k=2, 'z' statistic, threshold=0.05) used to identify the Markov blanket in each fold.

REGED1: Features known to be manipulated are discarded, HITON_MB (k=4, 'z' statistic, threshold=0.05) used to identify the Markov blanket in each fold. The PC algorithm in CausalExplorer (k=16, 'z' statistic, threshold=0.05) was then used to find the direct causes and direct effects from the features comprising the Markov blanket.

REGED2: This is a hybrid model created to interpolate between more formal
models:

PART #022 - All features used by all 100 models for REGED0 were identified. The PC (k=16, 'z' statistic, threshold=0.05) algorithm was then used to identify the direct causes using the entire training set. This identified two features (the reason for training additional models without ensembling was to populate the first few columns of the multi-column prediction matrix).

PART #021 - All features used by all 100 models for REGED0 were identified. The PC (k=16, 'z' statistic, threshold=0.05) algorithm was then used to identify the direct causes and direct effects using the entire training set rather than using an ensemble approach. This identified twelve features.

The features found by PART #022 were sorted in decreasing order of the magnitude of the weights of the model found in PART #022. Then the additional features found in PART #021 were added, sorted in order of the magnitude of the weights of the PART #021 model. The first eight features on this list were used to train a single model using the full training set. Roughly, this model contains the direct causes and a selection of the better correlated direct causes.

MARTI:

The pre-processing step described in the paper (based on iteratively re-weighted kernel ridge regression) was used to remove the noise. All features were then standardised. Optimally regularised kernel ridge regression used as the base classifier. Again, some models were constructed to interpolate between more formal feature selection methods. A feature list was constructed as before from the following parts:

PART #009 HITON_MB (k=2,'z' statistic,threshold=2) used to find the Markov blanket, PC (k=16, 'z' statistic, threshold=0.05) used to determine the direct causes (3 features)

PART #009 HITON_MB (k=2,'z' statistic,threshold=2) used to find the Markov blanket, PC (k=16, 'z' statistic, threshold=0.05) used to determine the direct causes and effects (15 features)

PART #011 BlogReg for non-causal feature selection (44 features).

PART #007 HITON_MB (k=5,'z' statistic,threshold=2) used to find the Markov blanket (131 features).

PART #003 Full model trained on all features.

The features were ranked by PART and then by the magnitude of the corresponding weight of the model in the PART where first encountered. All of these PARTS used an ensemble of 100 models, and so the feature sets are relatively large.

MARTI0: Model trained on the first 128 elements of the feature list. This is likely to contain all direct causes and effects, some highly correlated features and most (if not all) of the true Markov blanket.

MARTI1: Model trained on the first 32 elements of the feature list. This will be all direct causes and all direct effects, plus some highly correlated features.
MARTI2: The BLogReg algorithm was used for non-causal feature selection. An ensemble of 100 models was used for predictions.

All submissions use a single classifier with nested subsets, but not all subsets correspond to simple atomic feature selection policies. I added the interpolating models to fill in the gaps as this could not decrease my chances of winning, even if they didn't help. However, I also performed more formal experiments so that some more solid conclusions could be drawn about the value of causal and non-causal feature selection methods (I am convinced I should learn more about them!).

Results:

Table 1: Result table. The two stars next to the feature number indicate that the submission included a sorted list of features and multiple results for nested subsets of features. Top Ts refers to the best score among all valid last entries made by participants. Max Ts refers to the best score reachable, as estimated by reference entries using the knowledge of true causal relationships not available to participants.

Dataset	Entry	Method	Fnum	Fscore	Tscore (Ts)	Top Ts	Max Ts	<Tscore>	Rank
REGED0	1373	final models	128/999 **	0.941	0.9997±0.0012	0.9998	1		
REGED1	1373	final models	32/999 **	0.8393	0.9787±0.0036	0.9888	0.998	0.9276	2
REGED2	1373	final models	8/999 **	0.9985	0.8045±0.0056	0.86	0.9534		
SIDO0	1373	final models	4928/4932 **	0.589	0.9427±0.0070	0.9443	0.9467		
SIDO1	1373	final models	4928/4932 **	0.5314	0.7532±0.0137	0.7532	0.7893	0.7881	1
SIDO2	1373	final models	4928/4932 **	0.5314	0.6684±0.0130	0.6684	0.7674		
CINA0	1373	final models	128/132 **	0.5166	0.9743±0.0031	0.9765	0.9788		
CINA1	1373	final models	128/132 **	0.586	0.8691±0.0046	0.8691	0.8977	0.8488	3
CINA2	1373	final models	64/132 **	0.586	0.7031±0.0047	0.8157	0.891		
MARTI0	1373	final models	128/1024 **	0.8697	0.9996±0.0012	0.9996	0.9996		
MARTI1	1373	final models	32/1024 **	0.8064	0.9470±0.0039	0.947	0.9542	0.9147	1
MARTI2	1373	final models	64/1024 **	0.9956	0.7975±0.0059	0.7975	0.8273		

Much of the MATLAB code used is available from my website, BLogReg is available from http://theoval.cmp.uea.ac.uk/cbl/blogreg/ and the KRR model is implemented in the GKM toolbox, http://theoval.cmp.uea.ac.uk/~gcc/projects/gkm/. Scripts were written to perform the repeated hold-out validation etc and to distribute the work across the parallel HPC facility.

Keywords: Put at *least one keyword in each category*. Try some of the following keywords and add your own:

- Preprocessing or feature construction: standardization, regression.
- Causal discovery: Bayesian Network, Information Theoretic Method.
- Feature selection: Embedded feature selection, feature ranking,RFE.
- Classifier: kernel-method, least-square, ridge regression, L1 norm regularization, L2 norm regularization, logistic regression, ensemble method.
- Hyper-parameter selection: cross-validation.
- Other: ensemble method.

Answers to questions asked by the reviewers:

How was the information about manipulations used in REGED1?

The final submission for REGED1 discarded all features known to be manipulated.

Some algorithms, e.g. MMHC, require discrete data. How was the discretization performed?

I didn't do any discretization of continuous variables, but used the PC algorithm instead for problems with continuous features.

Algorithms like PC do not typically scale to datasets with more than 100 variables.

Yes, the simulations using the PC algorithm did take quite a long time! However, I used HITON_MB to find an estimate of the Markov blanket and then used the PC algorithm to direct the edges.

Similarly, it may be extremely computationally expensive to apply MMHC to some datasets with >1000-5000 variables. Did the author perform any pre-filtering to apply these algorithms?

My plan for SIDO was to find the Markov Blanket first (using HITON_MB) and then use MMHC to direct the edges in the Markov blanket, but I didn't finish the experiments in time for the close of the challenge. I hope to have completed them for Table 2 of the compete paper to give a more complete comparison of feature selection methods.

Causation and Prediction Challenge: FACT SHEET

Title: SVM-Based Feature Selection for Causation and Prediction Challenge
Author: Yin-Wen Chang
Address: Department of Computer Science, No. 1, Sec. 4, Roosevelt Road, Taipei, 106, Taiwan
E-mail: b92059@csie.ntu.edu.tw

Acronym of your best entry: final submission
Performance graphs generated by the organizers:
http://clopinet.com/isabelle/Projects/WCCI2008/Reports/Yin-Wen_Chang.html
Complete paper:
Feature Ranking Using Linear SVM
Yin-Wen Chang and Chih-Jen Lin; JMLR W&CP 3:53-64, 2008.

Method:

- Preprocessing
  - We scale the numerical data sets and conduct instance-wise normalization on the binary data sets as preprocessing. Gaussian filter is used to eliminate the low frequency noise in the MARTI data sets.
- Feature selection
  - We experiment with various SVM-based feature selection methods and have several interesting findings. Feature ranking via linear SVM models seems to be useful for these data sets. Checking AUC with/without removing each feature gives similar rankings.
  - During the development period, we experiment with various methods including feature ranking based on F-score, linear SVM weights, AUC/ACC change of removing a feature. After comparing the cross-validation AUC on training sets and the performance on toy examples, we use the feature ranking based on linear SVM weight in our final submission. We rank features according to the absolute value of weight corresponding to each feature.
  - The models for all versions of each task are the same since we tried to obtain a general and simple model for the problem.
  - In addition to cross-validation on training sets and the performance of toy examples, the quartile information is used since only one method is in the first quartile for all datasets.
  - Discovering that nested subsets would results in better performance when the selected features are the same, we used nested subsets of features from the slist we submitted. The reason might be that the feature rank we give in slist is good enough.
  - We did not use any knowledge derived from the test set to make the submissions.
- Classification
  - We use L2-loss linear SVM to train the classifier.
- Model selection/hyperparameter selection
  - Grid search is used to select the parameter of the SVM classifier.

Results:

Table 1: Result table. The two stars next to the feature number indicate that the submission included a sorted list of features and multiple results for nested subsets of features. Top Ts refers to the best score among all valid last entries made by participants. Max Ts refers to the best score reachable, as estimated by reference entries using the knowledge of true causal relationships not available to participants.

Dataset	Entry	Method	Fnum	Fscore	Tscore (Ts)	Top Ts	Max Ts	<Tscore>	Rank
REGED0	1452	final submission	16/999 **	0.8526	0.9998±0.0009	0.9998	1		
REGED1	1452	final submission	16/999 **	0.8566	0.9556±0.0040	0.9888	0.998	0.9316	1
REGED2	1452	final submission	8/999 **	0.997	0.8392±0.0052	0.86	0.9534		
SIDO0	1452	final submission	1024/4932 **	0.6516	0.9432±0.0074	0.9443	0.9467		
SIDO1	1452	final submission	4096/4932 **	0.5685	0.7523±0.0137	0.7532	0.7893	0.773	2
SIDO2	1452	final submission	2048/4932 **	0.5685	0.6235±0.0129	0.6684	0.7674		
CINA0	1452	final submission	64/132 **	0.6	0.9715±0.0032	0.9765	0.9788		
CINA1	1452	final submission	64/132 **	0.7053	0.8446±0.0047	0.8691	0.8977	0.8773	1
CINA2	1452	final submission	4/132 **	0.7053	0.8157±0.0052	0.8157	0.891		
MARTI0	1452	final submission	256/1024 **	0.8073	0.9914±0.0025	0.9996	0.9996		
MARTI1	1452	final submission	256/1024 **	0.7279	0.9209±0.0045	0.947	0.9542	0.891	3
MARTI2	1452	final submission	2/1024 **	0.9897	0.7606±0.0062	0.7975	0.8273		

Comment about the following:

- quantitative advantages: simplicity
- qualitative advantages: SVM feature selection method comparison.

Our implementation consists of python and matlab codes, and the LIBLINEAR software is used to train and predict.

Keywords:

- Preprocessing or feature construction: scaling.
- Feature selection: filter, feature ranking.
- Classifier: SVM.
- Hyper-parameter selection: grid-search.

Answers to the organizer’s questions:

What else did you try besides the method you submitted last? What do you think was a critical element of success compared to other things you tried?

We have tried several approaches. We used a kernel function to measure association between variables, but it takes too long time for a large data set. We also used several methods for predictions, such as Naïve Bayes, Boosting, SVM, Lasso et al. But they may be good for some of data sets but bad for others. Finally we select the L1 penalized logistic regression approach which performed averagely well for all of these data sets.

We focused on causal discovery and prediction models, especially we tried to minimize the number of features (ulist) selected for prediction. We should take advantage of a slist of features to improve TScore by chance.

In what do the models for the versions 0, 1, and 2 of the various tasks differ?

The structure learning is all the same to version 0, 1 and 2. But the selection of variables from the learned graph is different in the cases 0, 1 and 2 since they were differently manipulated. Nothing more is different among these three tasks.

Did you rely on the quartile information available on the web site for model selection or did you use another scheme?

The quartile is useful auxiliary information for us to consider whether it is necessary to improve our prediction. But they are not determinate. We check whether our model behaves better or worse by observing the main output indicators. We didn't use any other scheme.

In the result table you submitted, did you use nested subsets of features from the slist you submitted?

We did not use any nested subsets of features from slist, and we used the ulist only.

Did you use any knowledge derived from the test set to make your submissions, including simple statistics and visual examination of the data?

We did not use any knowledge from the test data.

Causation and Prediction Challenge: FACT SHEET

Title: Boosting Probabilistic Network for causality prediction
Author: Louis Duclos-Gosselin
Address: 205 Gosselin street, St-Agapit, Québec, g0s 1z0, Canada
Email: louis.gosselin@hotmail.com
Acronym of your best entry: Bayes Method
Performance graphs generated by the organizers:
http://clopinet.com/isabelle/Projects/WCCI2008/Reports/Louis_Duclos-Gosselin.html

Reference:
These recent years Bayesian Analysis became a subject of great interest for many practitioners and researchers. The 2008 causality challenge permits me to test one of my best Bayes methods. In fact, I use a special case of boosting probabilistic networks (Bayes Network) controlled with genetic algorithm and simulated annealing. More precisely, the construction of the network works this way. First, I use conventional probabilistic network architecture in conjunction with genetic algorithm and simulated annealing for controlling those elements : learning algorithms (M.A.P. and L.M.), number of neurons, number of links in the network, number of layer, type of kernel, transfer and activation function and the predictors to be in the models. Second, I use the idea of boosting (weighted re sampling) to construct an ensemble of probabilistic network. Third, during the process, Bayes Analysis (prior) helped to produce posterior probability. Finally, the joint distribution between the predictors and the joint distribution between the predictors and the target variable was used.

Method:

- Preprocessing : Informational theory, entropy
- Causal discovery : Probabilistic networks
- Feature selection : Genetic algorithm and simulated annealing
- Classification : Boosting Probabilistic networks (learned with M.A.P. and L.M.)
- Model selection/hyperparameter selection : Genetic algorithm, simulated annealing, bayes analysis

Results: The strength of this method is the use of boosting probabilistic networks controlled with simulated annealing and genetic algorithm. In addition, the uses of bayes analysis add something interesting.

Table 1: Result table. The star following the feature number indicates that the feature set was sorted. Top Ts refers to the best score among all valid last entries made by participants. Max Ts refers to the best score reachable, as estimated by reference entries using the knowledge of true causal relationships not available to participants.

Dataset	Entry	Method	Fnum	Fscore	Tscore (Ts)	Top Ts	Max Ts	<Tscore>	Rank
REGED0	361	Bayes Method	66/999 *	0.6973	0.9311±0.0040	1	1		
REGED1	361	Bayes Method	10/999 *	0.6761	0.8406±0.0054	0.998	0.998	0.7582	14
REGED2	361	Bayes Method	66/999 *	0.7317	0.5030±0.0016	0.86	0.9534		
SIDO0	361	Bayes Method	6/4932 *	0.5007	0.8956±0.0082	0.9443	0.9467		
SIDO1	361	Bayes Method	25/4932 *	0.4994	0.5244±0.0081	0.7532	0.7893	0.6254	10
SIDO2	361	Bayes Method	6/4932 *	0.5005	0.4562±0.0059	0.6684	0.7674		
CINA0	361	Bayes Method	106/132 *	0.644	0.9337±0.0030	0.9788	0.9788		
CINA1	361	Bayes Method	109/132 *	0.6689	0.7419±0.0052	0.8977	0.8977	0.7453	11
CINA2	361	Bayes Method	109/132 *	0.6689	0.5602±0.0052	0.8157	0.891		
MARTI0	361	Bayes Method	10/1024 *	0.495	0.9196±0.0041	0.9996	0.9996		
MARTI1	361	Bayes Method	2/1024 *	0.499	0.6658±0.0060	0.947	0.9542	0.7539	9
MARTI2	361	Bayes Method	2/1024 *	0.499	0.6764±0.0062	0.7975	0.8273		

Quantitative advantages : This method is really long to compute, but it has the advantage to explore all the possibility and it uses the full power of bayes analysis.

Qualitative advantages This method provide a lot of new elements. In fact, the idea of using boosting with probabilistic network is pretty new. In addition, the use of bayes analysis, simulated annealing and genetic algorithm to control all the processus is really special. All this make that method really unique. In brief, this method should be explore by researcher.

This method can be easily implanted in many system with a SAS code, C code or C++ code.

Keywords: Bayesian Analysis, Bayes Network, Boosting, Causality Prediction, Genetic Programming, Probabilistic Network, Simulated Annealing

- Preprocessing or feature construction: Entropy, information theory
- Causal discovery: Bayesian Network, Probabilistic network, boosting
- Feature selection: Genetic algorithm, simulated annealing
- Classifier: boosting probabilistic network
- Hyper-parameter selection: Genetic algorithm, simulated annealing, bayes analysis
- Other: ensemble method

Causation and Prediction Challenge: FACT SHEET

Title: Dimensionality reduction through unsupervised learning
Author: Nistor Grozavu (Nist in challenge)
Address: LIPN, Institut Galilée, 99 Av. J.B. Clément, F-93430 Villetaneuse
Email: nistor_grozavu@yahoo.com
Acronym of your best entry: Som (by Nist)
Performance graphs generated by the organizers:
http://clopinet.com/isabelle/Projects/WCCI2008/Reports/Nistor_Grozavu.html

Method:

Profile of my methods:

- Preprocessing : Data normalization;
- Causal discovery : Self Organizing Maps (SOM – Kohonen Map) adapted for supervised learning; Statistical Test (Cattell Scree Test) using acceleration stop criteria.
- Feature selection : Statistical Test (Cattell Scree Test) using acceleration stop criteria for each cluster.
- Classification : SOM + CAH (or K-means)

Results:

Table 1: Result table. The two stars next to the feature number indicate that the submission included a sorted list of features and multiple results for nested subsets of features. Top Ts refers to the best score among all valid last entries made by participants. Max Ts refers to the best score reachable, as estimated by reference entries using the knowledge of true causal relationships not available to participants.

Dataset	Entry	Method	Fnum	Fscore	Tscore (Ts)	Top Ts	Max Ts	<Tscore>	Rank
REGED0	1447	Som	4/999	0.498	0.5000±0.0000	1	1		
REGED1	1447	Som	4/999	0.498	0.5000±0.0000	0.998	0.998	0.5	15
REGED2	1447	Som	4/999	0.498	0.5000±0.0000	0.86	0.9534		
SIDO0	137	fd	16/4932 **	0.5068	0.5057±0.0056	0.9443	0.9467		
SIDO1	137	fd	128/4932 **	0.493	0.5161±0.0068	0.7532	0.7893	0.5098	16
SIDO2	137	fd	4/4932 **	0.499	0.5076±0.0057	0.6684	0.7674		
MARTI0	1447	Som	71/1024	0.4889	0.5000±0.0000	0.9996	0.9996		
MARTI1	1447	Som	71/1024	0.5011	0.5000±0.0000	0.947	0.9542	0.5	12
MARTI2	1447	Som	71/1024	0.7158	0.5000±0.0000	0.7975	0.8273		

Quantitative advantages (e.g. compact feature subset, simplicity, computational advantages): Compact feature subset (71 for MARTI), rapid in cost time.
Qualitative advantages (e.g. compute posterior probabilities, theoretically motivated, has some elements of novelty): elements of novelty : SOM provide a nice visualization; Using Cattell Statistical Test for each cluster we can give a good cluster characterization.

Implementation:
I implemented the model in Matlab and I used Statistical Toolbox and SOM Toolbox to facilitate the implementation.

Keywords:

- Preprocessing or feature construction: normalization.
- Causal discovery: Supervised SOM, Scoring.
- Feature selection: statistical test, weighting.
- Classifier: neural networks, CAH.

Causation and Prediction Challenge: FACT SHEET

Title: Markov blanket of the target and Norm1 linear SVM
Author: Cristian Grozea
Address: Fraunhofer Institute FIRST, Kekulestrasse 7, 12489 Berlin, Germany.
Email: cristian.grozea@first.fraunhofer.de
Acronym of your best entry: darum
Performance graphs generated by the organizers:
http://clopinet.com/isabelle/Projects/WCCI2008/Reports/Cristian_Grozea.html

Method:

- Cina and the toy problems: norm1 linear svm
- Other pbs: norm1 linear svm on the features in the Markov blanket of the target

The features have been ranked by the strength of the corresponding weight in the final classifier. The same model has been applied to all three subproblems. On the problems where the Markov blanket has been used as a feature selection method, the results without these selection were initially bad on the colored quartiles.Hold-out test set has also been used to measure the performance of the training. Nested subsets of features have not been used.

For MARTI (where I have also used at training spatial filters) I have looked at the first few entries in the test set in order to understand the phrases :
The test sets have no added noise. This situation simulates a case where we would be using different instruments at "training time" and "test time", e.g. we would use DNA microarrays to collect training data and PCR for testing.

Erratum: The graphs from http://clopinet.com/isabelle/Projects/WCCI2008/Reports/Cristian_Grozea.html confirm what I suspected, that is that I sent the wrong features indexes for Marti.
What I did was keeping only the estimated Markov blanket features, then running my existing code that reported the index of the features ordered by their absolute weight in the final classifier, but not taking into account the original index of the features.
I have to admit that I didn't pay much attention to this as it was seemingly not important for the ranking. For the next problems I preferred to "kill" the unwanted features by zeroing them, such that I wouldn't have to change the code and still get the right indexes.
The reason for writing this is to avoid the impression that you could do well with the wrong features.

Implementation:
Matlab, CVX and Causal explorer have been used

Keywords:

- Preprocessing or feature construction: causal
- Causal discovery: Markov blanket
- Classifier: SVM, L1 norm regularization
- Hyper-parameter selection: sweep, hold-out test

Causation and Prediction Challenge: FACT SHEET

Title: An Energy-based Model for Feature Selection
Authors: H. Jair Escalante, Luis Enrique
Address: Erro #1, Tonantzintla, Puebla, 72840, México
Email: hugo.jair@gmail.com, hugojair@ccc.inaoep.mx
Acronym of your best entry: DRF-LM-PSMS Final Run 2
Performance graphs generated by the organizers:
http://clopinet.com/isabelle/Projects/WCCI2008/Reports/H_Jair_Escalante.html

Method:
We propose an energy-based (random-field like) model for the selection of predictive features. The user specifies *k*, the size of the subset of features they want to obtain. Then a random-field with *k* nodes is defined; each node representing a feature. Each of the *k* features depends on the other *k-1* features and on the target variable Y. In Figure 1 the graphical model of the proposed approach is shown for a value of *k*=6.

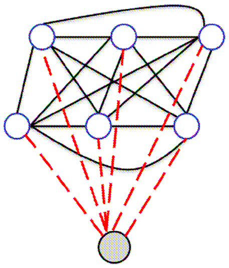

Fig. 1. Graphical model of the proposed EBM for a value of *k* = 6.

An energy value is assigned to each combination of *k*-features, according an energy function. This function assigns low values to *good* configurations of features taking into account the following information:

1. The rank position of each individual feature though the ranking lists returned by eight ranking-based feature selection methods (those in the CLOP package [2]). Different ranking lists are merged considering the position of features along the lists.
2. The predictive power of each individual feature, measured by the CV- balanced error rate obtained by an arbitrary classifier using a single feature for predicting Y.
3. The combined predictive power of the *k*-features, measured as above using the *k*-features for predicting Y.
4. Global Markov blanket (MB) information: those features appearing in the MB are weighted higher. The global MB is calculated using all the features in the training data with the Causal Explorer [1].
5. Local MB information: those features appearing in the local MB receive an extra weight. The local MB is calculated using only the *k*-features in the training data with the Causal Explorer [1].

The feature selection problem reduces to find the configuration of *k*-features that minimizes the energy function. This configuration will be that offering the best tradeoff among the considered information (1-5) . A simple iterative procedure called iterative conditioned modes (ICM) is used for minimization of the energy function. For those entries containing PSMS in their name we applied particle swarm model selection at the end of the feature selection process. This method is used for searching for the best classifier and hyperparameters for each subset of features *k*. Therefore, different classifiers were considered for different subset sizes.

I tried several combinations of the sources of information we considered (1-5). The key elements of the proposed approach were the rank of individual features according several feature selection methods (1) and the predictive power of individual features (2). There is not a significant difference (neither positive nor negative) of using only (1-2) or including causal information (1-5). This result is interesting because by simply combining the ranked lists of features from different methods and taking into account the individual predictive power of features we can obtain competitive results. For SIDO the *PC_HITON* algorithm could not be applied because it was running too slow. For CINA this algorithm was not able to infer the MB.

- Preprocessing
  - ▪ *No preprocessing was applied to data.*
- Causal discovery
  - ▪ *For some experiments I used the PC_HITON implementation ([1]) from the Causal Explorer for obtaining the Markov Blanket of the target variable.*
- Feature selection
  - ▪ *I used the following feature selection methods from the Challenge Learning Object Package (CLOP) (See [2] for a description of these methods):*
    - *s2n,gs,relief, svcrfe, aucfs, f-test, t-test, Pearson*
- Classification
  - ▪ *Kernel ridge regression and Naïve Bayes (CLOP implementations) were considered for classification. The latter method was used (extensively) during the optimization process and the former for computing initial and final predictions.*
- Model selection/hyperparameter selection
  - ▪ *For most of the entries, default parameters were considered for the methods above described.*
  - ▪ *For a few runs it was used PSMS (a population-based search strategy for model selection) for the selection of a classifier at the end of the feature selection process.*

Results:

Table 1: Result table. The two stars next to the feature number indicate that the submission included a sorted list of features and multiple results for nested subsets of features. Top Ts refers to the best score among all valid last entries made by participants. Max Ts refers to the best score reachable, as estimated by reference entries using the knowledge of true causal relationships not available to participants.

Dataset	**Entry**	**Method**	**Fnum**	**Fscore**	**Tscore (Ts)**	**Top Ts**	**Max Ts**	**<Tscore>**	**Rank**
REGED0	1485	DRF-LM-PSMS Final Run 2	32/999 **	0.8778	0.9996±0.0010	0.9998	1		
REGED1	1485	DRF-LM-PSMS Final Run 2	128/999 **	0.7996	0.9448±0.0039	0.9888	0.998	0.8985	5
REGED2	1485	DRF-LM-PSMS Final Run 2	64/999 **	0.7638	0.7512±0.0060	0.86	0.9534		
SIDO0	1485	DRF-LM-PSMS Final Run 2	1024/4932 **	0.8442	0.9352±0.0075	0.9443	0.9467		
SIDO1	1485	DRF-LM-PSMS Final Run 2	4932/4932 **	0.4675	0.6913±0.0134	0.7532	0.7893	0.7474	7
SIDO2	1485	DRF-LM-PSMS Final Run 2	4932/4932 **	0.4675	0.6157±0.0128	0.6684	0.7674		
CINA0	1485	DRF-LM-PSMS Final Run 2	132/132 **	0.955	0.9670±0.0035	0.9765	0.9788		
CINA1	1485	DRF-LM-PSMS Final Run 2	132/132 **	0.4982	0.7873±0.0049	0.8691	0.8977	0.7675	9
CINA2	1485	DRF-LM-PSMS Final Run 2	128/132 **	0.4982	0.5481±0.0044	0.8157	0.891		
MARTI0	1485	DRF-LM-PSMS Final Run 2	1024/1024 **	0.5446	0.9673±0.0036	0.9996	0.9996		
MARTI1	1485	DRF-LM-PSMS Final Run 2	512/1024 **	0.4711	0.8636±0.0054	0.947	0.9542	0.8691	5
MARTI2	1485	DRF-LM-PSMS Final Run 2	8/1024 **	0.7055	0.7764±0.0061	0.7975	0.8273		

- Quantitative advantages (e.g. compact feature subset, simplicity, computational advantages)
    - *It is computationally efficient: by considering the two sources of information that worked well we will obtain competitive results very fast and in a simpler way, that does not requires specialized knowledge.*
    - *The method can be applied to any data set without an adhoc modification; particularly, the things that worked well (1-2) can be used directly in any binary classification data set.*
    - *The method is easy to implement: even when taking into account all of the sources of information it is not complicated to implement it. Furthermore, the energy-based modeling framework allows us introducing other sources of information, not considered here, with little effort.*

   - *The method may (or may not) take into account causal information into the feature selection process. Causal information could be very useful for improving the feature selection process.*

   - *It can return subsets of features of size k; the user is able to set this parameter (k). Furthermore, we can return a ranked list of features according their importance.*

- <u>Qualitative advantages</u>

   - *The energy-based model we propose is a new way to approach the feature selection problem. Since it is based on the energy-minimization framework it is a very general approach that can be easily modified. The proposed model can, even, be considered a template under which several sources of information and different form of potentials can be tested. This will motivate further research in several directions, particularly on the appropriate ad-hoc definition of potentials and on learning the energy function from data.*

   - *The fusion of the ranking lists of diverse feature selection methods proved to be very useful for feature selection. Information fusion has been proved to be very effective in a number of fields, most notably in machine learning (boosting, bagging) and information retrieval (multi-modal retrieval of video and images). The results obtained by merging different lists give evidence that the fusion of the outputs of diverse feature selection methods has practical advantages that motivate further research.*

   - *Domain knowledge and further information (both causal and non-causal) can be easily introduced into our model, this is also related to the generality of energy-based modeling.*

   - *For this implementation we have used the simpler potentials one can use for this problem and the simplest algorithm for energy minimization (Iterated Conditioned Modes, ICM). Therefore, better results are expected by defining more elaborate potentials and by using faster and better convergence optimization algorithms (e.g. the graph cuts algorithm). Furthermore, fixed classifiers and default hyperparameters have been used in most of the experiments.*

Implementation:

The method has been implemented in Matlab, it requires the CLOP toolbox and the causal explorer (if causal information is considered). The implementation is very simple and it can be considered a push-button application that can be applied to any domain without a significant modification.

Keywords:

- Causal discovery:
  - *Markov blanket information, HITON algorithm.*
- Feature selection:
  - *Feature ranking, combination of feature selection methods, s2n,gs,relief, svcrfe, aucfs, f-test, t-test, Pearson.*
- Classifier:
  - *Ridge regression, Naïve Bayes classifier.*
- Hyper-parameter selection:
  - *PSMS.*
- Other:
  - *Energy-based models, random field modeling, ICM.*

References:

[1] **C.F. Aliferis, I. Tsamardinos, A. Statnikov**, *HITON: A Novel Markov Blanket Algorithm for Optimal Variable Selection, In FLAIRS 2003.*

[2] **A. Safari and I. Guyon**, *Quickstart guide for CLOP. Technical report, Graz University of Technology
and Clopinet, May 2006.* http://www.ymer.org/research/files/clop/QuickStartV1.0.pdf.

Causation and Prediction Challenge: FACT SHEET

Title: Translate Binary Variable to Continuous Variable
Author: Jinzhu Jia
Address: School of Mathematical Sciences,
Peking University, Beijing, P.R.China**,** 100871
Email: jinjinjia@gmail.com
Acronym of your best entry: Final
Performance graphs generated by the organizers:
http://clopinet.com/isabelle/Projects/WCCI2008/Reports/Jinzhu_Jia.html

Method:

- EM Algorithm
- Causal discovery
- Elastic net
- SVM

quantitative advantages :Simple and fast. After translating the binary variable to continuous variable, we can use modern model selection methods to deal with this problem, such as Lasso, Elastic, etc. All of these method are computationally fast.

qualitative advantages : Very novel method. I construct the correlation between a binary variable and a continuous variable and then I transform a binary variable into a continuous variable without lose the information between the two variables.

Implementation:
For the ``Reged" data set, we think that the target variable Y comes from a hidden variable H with a normal distribution and ``Y=1" corresponds to ``H>a" for some fixed real number a. Based on this assumption, we can use EM algorithm to construct the correlation between Y and each of the predictors.

After obtaining the correlation matrix, we run a ridge regression and get the regression coefficients of Y on X, then we construct Y. But the solution of ridge regression is not sparse, then we run elastic net to get a sparse coefficient. Those predictors with non-zero coefficients are our approximated Blank variables and we use these variables to do predictions.

We use the approximated "blanket" variable selected from elastic net and Y to construct a causal network, by the software of TETRAD and then get the parents and children variables of Y.

For data set Reged0, since it is not manipulated, we use all the parents and children variables to do predictions.

For data set Reged1, we use all the parents and those children variables which are not manipulated to do predictions.

For the data set Reged 2, we just use the parents nodes to do predictions, for the son nodes have been changed a lot and thus there is little information to do predictions.

When do predictions, we use three methods: 1.linear regression, 2. SVM 3. SVM regression, and then the three results are used to give a final result. If more than two of the methods give Y=1, then Y=1, or else, Y=-1.

This is not a push-button application. Since we have to use the software TETRAD to decide which variables are parents.

Table 1: Result table. Top Ts refers to the best score among all valid last entries made by participants. Max Ts refers to the best score reachable, as estimated by reference entries using the knowledge of true causal relationships not available to participants.

Dataset	Entry	Method	Fnum	Fscore	Tscore (Ts)	Top Ts	Max Ts	<Tscore>	Rank
REGED0	1487	Final	14/999	0.7847	0.9954±0.0010	1	1		
REGED1	1487	Final	11/999	0.748	0.8158±0.0056	0.998	0.998	0.8118	12
REGED2	1487	Final	10/999	0.996	0.6242±0.0053	0.86	0.9534		
SIDO0	311	test	60/4932	0.507	0.8968±0.0082	0.9443	0.9467		
SIDO1	311	test	4932/4932	0	0.5000±0.0028	0.7532	0.7893	0.6123	11
SIDO2	311	test	4932/4932	0	0.4401±0.0027	0.6684	0.7674		

Keywords:
Preprocessing or feature construction: centering, standardization
Causal discovery: Bayesian Network, Probabilistic Graphical Models
Feature selection: Penalized regression, Lasso, Elastic net.
Classifier: SVM, least-square, ridge regression, L1 norm regularization, L2 norm regularization, ensemble method.
Hyper-parameter selection: cross-validation, K-fold.

Causation and Prediction Challenge: FACT SHEET

Title: Univariate feature ranking and SVM classifier
Author: Jianming Jin
Address: HP Labs, China, 112 JianGuo Road, ChaoYang District, HP Building, Beijing, China, 100022
Email: jian-ming.jin@hp.com
Acronym of your best entry: HPLC
Performance graphs generated by the organizers:
http://clopinet.com/isabelle/Projects/WCCI2008/Reports/Jianming_Jin.html

Method:

- Preprocessing: We map the causation and prediction challenge to a document classification task. Here, a dataset is a group of documents, the feature dimension is the size of lexicon, each feature is corresponding to a word in the lexicon, and the value of a feature is the word appearance number in a document (TF).
- Feature selection: Weighted feature vector is calculated by multiply the original feature vector with a weighting vector. The weighting value of each dimension is mainly determined by the TF distribution variance in positive training data and negative training data.
- Classification: SVMLight is used for training and classification on the weighted feature vectors.
- Model selection/hyperparameter selection: The model is trained on the provided training set, the training parameters are optimized according to the classification result on the provided testing set. The model with the best F1 value on the provided testing set is chosen as the final model.

There is no special optimization for a peculiar dataset, and there is no need for human's participation during the whole process.

Results:

Table 1: Result table. Top Ts refers to the best score among all valid last entries made by participants. Max Ts refers to the best score reachable, as estimated by reference entries using the knowledge of true causal relationships not available to participants.

Dataset	Entry	Method	Fnum	Fscore	Tscore (Ts)	Top Ts	Max Ts	<Tscore>	Rank
REGED0	1088	exp 1	999/999	0.5	0.9932±0.0022	1	1		
REGED1	1088	exp 1	999/999	0.5	0.9340±0.0042	0.998	0.998	0.8868	8
REGED2	1088	exp 1	999/999	0.5	0.7331±0.0059	0.86	0.9534		
SIDO0	1089	exp 1	4932/4932	0.5	0.9320±0.0096	0.9443	0.9467		
SIDO1	1089	exp 1	4932/4932	0.5	0.7307±0.0136	0.7532	0.7893	0.7662	3
SIDO2	1089	exp 1	4932/4932	0.5	0.6359±0.0133	0.6684	0.7674		
CINA0	1088	exp 1	132/132	0.5	0.9566±0.0034	0.9788	0.9788		
CINA1	1088	exp 1	132/132	0.5	0.6528±0.0056	0.8977	0.8977	0.6883	13
CINA2	1088	exp 1	132/132	0.5	0.4556±0.0035	0.8157	0.891		
MARTI0	1088	exp 1	1024/1024	0.5	0.8967±0.0047	0.9996	0.9996		
MARTI1	1088	exp 1	1024/1024	0.5	0.7597±0.0060	0.947	0.9542	0.7848	7
MARTI2	1088	exp 1	1024/1024	0.5	0.6979±0.0063	0.7975	0.8273		

Implementation:

The implementation is a java package, which using SVMLight for training and classification. It’s a function module without user interface.

Keywords:

- Preprocessing or feature construction: none.
- Causal discovery: none.
- Feature selection: filter, weighting.
- Classifier: SVM.
- Hyper-parameter selection: grid-search.

Causation and Prediction Challenge: FACT SHEET

Title: Collider scores
Authors: Ernest Mwebaze and John Quinn
Address: Faculty of Computing & Information Technology, Makerere University, Kampala, Uganda.
Emails: emwebaze@cit.mak.ac.ug, jquinn@cit.mak.ac.ug
Acronym of your best entry: submission
Performance graphs generated by the organizers:
http://clopinet.com/isabelle/Projects/WCCI2008/Reports/E_MwebazeJ_Quinn.html

Method:

Preprocessing

None, raw data used directly.

Causal discovery

HITON_PC used to estimate neighbouring variables. For manipulated datasets we further narrow down the feature set by computing two scores to select strong causes:

1) To score a variable A as a cause of target variable T using supporting variable B_i , use ratio of partial correlation of $(A,B_i \mid T)$ and correlation of (A,B_i).
2) For the second score, calculate the difference of:
    1. evidence that target T is a collider for causes A and B_i, looking for high correlation between (A,target) and (B_i ,target) and low correlation between (A,B_i)
    2. evidence that variable A is a collider for causes T and B_i , using the equivalent pattern of correlation.

Both scores are aggregated over the B_i 's.

Feature selection

For unmanipulated datasets, use the features estimated to be neighbouring the targets. For manipulated datasets, choose the subset of features with highest mean scores above.

Classification

For REGED, k-nn classification. For SIDO and CINA, shallow decision trees with naive Bayes classifiers in the leaves (single trees only – no ensemble methods).

Results:

Estimation of neighbouring variables uses the HITON_PC implementation in the Matlab 'Causal Explorer' library. All other code (for calculating scores, learning and classification etc) written in Python using the Numpy libraries.

The scores are simple to implement and quick to calculate (on the order of seconds for all datasets).

The utility of the scores is dependent on the success of estimating variables which are neighbours to the target. The inclusion of other variables, particularly outside the Markov blanket, can confound the result.

Table 1: Result table. The two stars next to the feature number indicate that the submission included a sorted list of features and multiple results for nested subsets of features. Top Ts refers to the best score among all valid last entries made by participants. Max Ts refers to the best score reachable, as estimated by reference entries using the knowledge of true causal relationships not available to participants.

Dataset	Entry	Method	Fnum	Fscore	Tscore (Ts)	Top Ts	Max Ts	<Tscore>	Rank
REGED0	1444	submission	14/999 **	0.8088	0.9933±0.0014	0.9998	1		
REGED1	1444	submission	1/999 **	0.7122	0.9528±0.0028	0.9888	0.998	0.8559	7
REGED2	1444	submission	14/999 **	0.9935	0.6216±0.0019	0.86	0.9534		
SIDO0	1444	submission	10/4932 **	0.5003	0.9325±0.0074	0.9443	0.9467		
SIDO1	1444	submission	6/4932 **	0.5009	0.6660±0.0133	0.7532	0.7893	0.7509	6
SIDO2	1444	submission	6/4932 **	0.5009	0.6541±0.0131	0.6684	0.7674		
CINA0	1444	submission	8/132 **	0.7575	0.9430±0.0033	0.9765	0.9788		
CINA1	1444	submission	46/132 **	0.5885	0.7381±0.0047	0.8691	0.8977	0.7832	8
CINA2	1444	submission	8/132 **	0.5235	0.6685±0.0042	0.8157	0.891		

Keywords:

- Preprocessing or feature construction: none.
- Causal discovery: Structural Equation Models, heuristic.
- Feature selection: feature ranking.
- Classifier: nearest neighbors, tree classifier, naive Bayes.
- Hyper-parameter selection: cross-validation.

Causation and Prediction Challenge: FACT SHEET

Title: Random Sets Approach and its Applications
Author: Vladimir Nikulin
Address: Suncorp, Brisbane, Australia
Email: vladimir.nikulin@suncorp.com.au
Acronyms of your final entries: *"vn14"* and *"vn14a"* (for SIDO)
Performance graphs generated by the organizers:
http://clopinet.com/isabelle/Projects/WCCI2008/Reports/Vladimir_Nikulin.html
Complete paper:
Random Sets Approach and its Applications
Vladimir Nikulin; JMLR W&CP 3:65-76, 2008.

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 represents a very essential component of data mining, as it will help reduce overfitting and make prediction more accurate. Causal discovery may be regarded as a next step with aim to uncover 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.

Table 1: Result table. Top Ts refers to the best score among all valid last entries made by participants. Max Ts refers to the best score reachable, as estimated by reference entries using the knowledge of true causal relationships not available to participants. These entries used unsorted lists of features.

Dataset	Entry	Method	Fnum	Fscore	Tscore (Ts)	Top Ts	Max Ts	<Tscore>	Rank
REGED0	1340	vn14	400/999	0.7576	0.9989±0.0016	0.9998	1		
REGED1	1340	vn14	400/999	0.7316	0.9522±0.0038	0.9888	0.998	0.9094	4
REGED2	1340	vn14	400/999	0.8004	0.7772±0.0059	0.86	0.9534		
SIDO0	1479	vn14a	527/4932	0.5502	0.9429±0.0075	0.9443	0.9467		
SIDO1	1479	vn14a	527/4932	0.5339	0.7192±0.0138	0.7532	0.7893	0.7588	5
SIDO2	1479	vn14a	203/4932	0.5225	0.6143±0.0132	0.6684	0.7674		
CINA0	1340	vn14	50/132	0.7174	0.9764±0.0031	0.9765	0.9788		
CINA1	1340	vn14	30/132	0.5	0.8617±0.0047	0.8691	0.8977	0.8504	2
CINA2	1340	vn14	30/132	0.5	0.7132±0.0043	0.8157	0.891		
MARTI0	1340	vn14	217/1024	0.5863	0.9889±0.0025	0.9996	0.9996		
MARTI1	1340	vn14	400/1024	0.5554	0.8953±0.0048	0.947	0.9542	0.8736	4
MARTI2	1340	vn14	611/1024	0.7021	0.7364±0.0062	0.7975	0.8273		

Method:
Random sets approach has heuristic nature and has been inspired by the growing speed of computations. For example, we can consider large number of classifiers where any single classifier (base classifier or model) is based on the subset of relatively small number 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.

Table 2: List of the base models, which were used during WCCI-2008 data-mining competition.

Data	Model	Software
LUCAS	neural+doubleboost	MATLAB-CLOP
LUCAP	neural+doubleboost	MATLAB-CLOP
REGED	SVM-RBF (special software)	C
SIDO	binaryRF (special software)	C
CINA	adaBoost	R
MARTI	svc+standardize	MATLAB-CLOP

In the case of SIDO-set, Random Forest model proved to be the most suitable. Note that RF model appears to be very relevant to the subject of this paper. However, approach of RF is far from the same comparing with RS approach. We used RF model with 1000 trees where 70 randomly selected features were used for any splitter. The splitting process was stopped if size of the current node was smaller than 10 (anyway, no more than 8 splitting levels were used). The SIDO-set is binary, and this property simplified implementation of the RF-algorithm essentially. Next, we computed for any particular feature number of repeats in the above RF-object. These repeats 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.

Preprocessing: 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. The test sets have no added noise. We used linear regression model in order to filter noise from the training set. As a target variables we used remaining 999 features.

Another application of random sets was motivated by the huge imbalanced data, which represent significant problem because the corresponding classifier has tendency to ignore patterns with smaller representation in the training set. We propose to consider large number of balanced training subsets where representatives from both patterns are selected randomly.

Keywords: Causal relations, graphical models, random forest, boosting, svm, CLOP, cross validation.

Causation and Prediction Challenge: FACT SHEET

Title: Optimally Compressive Regression
Author: Florin Popescu
Address: Fraunhofer-Institut FIRST, Kekulestrasse 7, 12489 Berlin Germany
Email: florin.popescu@first.fraunhofer.de
Acronym of your best entry: optimally_compressive_regression
Performance graphs generated by the organizers:
http://clopinet.com/isabelle/Projects/WCCI2008/Reports/Florin_Popescu.html
Reference:
Florin Popescu. Identification of Sparse Multivariate Autoregressive Models. EUSIPCO 2008 (in press).

Method:
Models (Gaussian probability distributions) are derived each class using auto-regression and minimum description length (MDL) regularization, where details of MDL have been derived by the author to allow mixtures of binary and non-binary valued data. Initially the method was meant for classification of time series data, but is applicable also to stationary data sets by essentially building large, sparse covariance matrices. MDL sparsifies automatically and in itself is a conduit for causality inference: the best explanation (e.g. causal chain) is the one that compresses the data the most.

Preprocessing: The features were scaled such that their quantization level is 1.
Causal discovery: Linear regressions were done to make a list of predictability of each feature (how compressible it is given knowledge of all other features: a *full* regression). MDL gives sparse results so each feature thereby has a set of predictor features.
Feature selection: The union of the best *X*% predicted variables and all necessary predictors ordered by predictability.
Model selection/hyperparameter selection: The model (for each class) was a strictly upper triangular auto-regression (AR) matrix between selected features (with bias and scaling). This is called *causal* regression because it enforces a causal chain. The feature set was *sorted* by the resulting predictability and the causal regression re-computed until the predictability is strictly increasing. The MDL regression sparsifies the AR matrix and therefore may further reduce the feature set – it also produces a directed *acyclic* graph of causal factors (the causal chain). The method 'works' with the parameter *X* set to 0 (naïve method: meaning only predictors are used) but it was set at a higher level. By the MDL principle it is the final MDL score that counts: hyperparameters such as *X* only serve to make the necessary (non-convex) MDL optimization faster or more likely to reach the global minimum, they do not embody a statistical principle *per se*. There is no cross-validation.
Classification: Once the class models are obtained, the classification is trivial: each new example corresponds to a likelihood (or probability) determined by the model, and the label is the class of the highest likelihood model.

Results:

Table 1: Result table. Top Ts refers to the best score among all valid last entries made by participants. Max Ts refers to the best score reachable, as estimated by reference entries using the knowledge of true causal relationships not available to participants. These entries used unsorted lists of features.

Dataset	Entry	Method	Fnum	Fscore	Tscore (Ts)	Top Ts	Max Ts	<Tscore>	Rank
CINA0	1495	optimally compressive regression	25/132 *	0.6087	0.7769±0.0051	0.9788	0.9788		
CINA1	1495	optimally compressive regression	25/132 *	0.7639	0.7322±0.0052	0.8977	0.8977	0.7488	10
CINA2	1495	optimally compressive regression	25/132 *	0.7639	0.7372±0.0052	0.8157	0.891		

- quantitative advantages The feature set is automatically generated and for the naïve method was very compact (11 features selected for CINA0)
- qualitative advantages The method, as explained, computes posterior probabilities *as well as* a directed causal chain, is theoretically motivated, can be fully automatic and is (to the author's knowledge) fully novel.

The method was implemented in Matlab and C. The linear regressions were performed using standard methods (albeit organized such that large regressions can be done within memory limitations). The MDL optimization is computationally expensive but is *not* an exhaustive feature subset modeling technique, rather it is programmed using iterative heuristics for sparsification of features and structural models.

Keywords:

- Preprocessing or feature construction: scaling.
- Causal discovery: Structural Equation Models, Probabilistic Graphical Models, Information Theoretic Method.
- Feature selection: filter, feature ranking.
- Classifier: likelihood-based
- Hyper-parameter selection: none
- Other: minimum description length.

Causation and Prediction Challenge: FACT SHEET

Title: Markov blanket and kernel ridge regression
Author: Marius Popescu
Address: University of Bucharest
Department of Computer Science
Academiei 14, 70109 Bucharest, Romania
Email: popescunmarius@gmail.com
Acronym of your best entry: MB_Kcomb1
Performance graphs generated by the organizers:
http://clopinet.com/isabelle/Projects/WCCI2008/Reports/Marius_Popescu.html

Method:
I approached the challenge from the position of someone with experience in machine learning, but a completely newcomer in causality. As learning method I used Kernel Ridge Regression. For prediction (and training) I used only features from the Markov Blanket (MB) of the target variable, but I also tried to exploit the structure of the MB. The structure of MB was exploited by defining two separate kernels: one over the parents (direct causes) and another one over the children (direct effects) and spouses. The kernel used in Ridge regression was a linear combination of these two kernels.

Obtaining MB and its structure
To obtain the Markov blanket and its structure I relied on "Causal Explorer". To obtain the variables of the MB I used HITON_MB method. To obtain the structure I used TPDA method over the variables in the MB and target variable. Because TPDA can leave some edges undirected and because MB is not a general Bayesian network (it has a special structure around the target node) I used the following heuristic to direct all the edges: all nodes for which TPDA find a directed link from target node to it (1 in the adjacency matrix) are considered children, all others nodes for which TPDA find connection (1 or 2 in the adjacency matrix) to target node are considered parents node, remaining nodes are considered spouses. This heuristics prefers to introduce false parents than to miss some parents (we consider direct causes very important). The details can be seen in the following MATLAB script (p is the index of parents, c the index of children and s the index of spouses):

```
load reged0_train
Y = load('reged0_train.targets');
Y = 0.5*(Y+1);
XY= [X,Y];
mb=Causal_Explorer('HITON_MB', XY, 1000, [], 'z', 0.05, 3)
XY2 = XY(:, [mb,1000]);
A=Causal_Explorer('TPDA', XY2, [], 'z', 0.05, 1, 1)

ic = find(A(end, 1:end-1) == 1);
ip = find(A(1:end-1, end) ~= 0);
is = setdiff(1:length(mb), union(ip,ic));
p = mb(ip);
c = mb(ic);
s = mb(is);

save mb mb A ic ip is p c s;
```

Training and Prediction

The kernel used in the Kernel Ridge Regression was a linear combination of two quadratic (normalized) kernels:

K = (eta * K1) + ((1 - eta) * K2)

K1 being a quadratic normalized kernel over the set of parents:

```
K1 = (X(:, p) * X(:, p)' + 0.5) .^ 2;
XN1 = sqrt(diag(K1));
K1 = K1 ./ (XN1 * XN1');
```

And K2 a quadratic normalized kernel over the set of children and spouses:

```
K2 = (X(:, [c,s]) * X(:, [c,s])' + 0.5) .^ 2;
XN2 = sqrt(diag(K2));
K2 = K2 ./ (XN2 * XN2');
```

The parameter eta of the linear combination is chosen taking into account how "good" is each kernel of the combination. The "goodness" is measured by "kernel alignment", more precisely the alignment of the kernel with the ideal kernel YY'. Again the details (including the setting of parameters) can be seen in training MATLAB script:

```
lambda = 10 ^ (-6);

load lim;
load reged0_train
X = scale(X, ll, ul);
Y = load('reged0_train.targets');

load mb;

n = size(X, 1);

K1 = (X(:, p) * X(:, p)' + 0.5) .^ 2;
XN1 = sqrt(diag(K1));
K1 = K1 ./ (XN1 * XN1');

K2 = (X(:, [c,s]) * X(:, [c,s])' + 0.5) .^ 2;
XN2 = sqrt(diag(K2));
K2 = K2 ./ (XN2 * XN2');

align1 = (Y' * K1 * Y) / (n * norm(K1, 'fro'))
align2 = (Y' * K2 * Y) / (n * norm(K2, 'fro'))
eta = align1 / (align1 + align2);

K = (eta * K1) + ((1 - eta) * K2);

w = inv(K+(n*lambda)*eye(n)) * Y;

Yh = K * w;
%Yh = sign(Yh);
%Yh(find(Yh == 0)) = 1;

%err = length(find(Yh ~= Y)) / n
```

```
fid = fopen('reged0_feat.ulist','w');
fprintf(fid,'%g ',[p,c,s]);
fprintf(fid,'\n');
fclose(fid);

fid = fopen('reged0_train.predict','w');
fprintf(fid,'%g\n',Yh);
fclose(fid);

save model w eta XN1 XN2;
```

Treating manipulation

When the list of manipulated variables is available (REGED1) from the MB are removed the children of the target that are manipulated. Also are removed all the spouses that remain without children. The script is:

```
load mb;

tbelm = [20, 27, 36, 70, 82, 83, 85, 91, 118, 125, 139, 143, 160, 169,
176, 185, 191, 204, 219, 224, 229, 239, 243, 251, 252, 269, 281, 282,
295, 297, 301, 319, 320, 321, 342, 350, 357, 359, 361, 378, 387, 407,
409, 412, 429, 430, 469, 472, 499, 501, 507, 512, 540, 545, 552, 561,
566, 572, 580, 586, 593, 618, 622, 637, 651, 663, 674, 681, 683, 686,
690, 702, 727, 754, 762, 764, 773, 786, 805, 815, 835, 861, 872, 873,
877, 880, 889, 904, 935, 936, 939, 942, 949, 962, 977, 985, 989, 991,
992, 994];

se = intersect(tbelm,s);
ce = intersect(tbelm,c);

[tmp, ise] = ismember(se, mb);
[tmp, ice] = ismember(ce, mb);

ice2 = [];

for x = ise
   ice2 = union(ice2, find(A(ic,x) ~= 0)');
end

ice2 = ic(ice2);
ice = union(ice, ice2);

ictmp = setdiff(ic, ice);

ise2 = [];

for x = is
   if isempty(find(A(ictmp,x) ~= 0))
     ise2 = [ise2, x];
   end
end

ise = union(ise, ise2);

icn = setdiff(ic, ice);
isn = setdiff(is, ise);

cn = mb(icn);
sn = mb(isn);

save newmb p cn sn;
```

In the case of REGED2 (when all variable excepting parents are manipulated) that mean that will remain only one kernel (instead of a combination of two kernels), the quadratic kernel over the set of parents.

Results:

Table 1: Result table. Top Ts refers to the best score among all valid last entries made by participants. Max Ts refers to the best score reachable, as estimated by reference entries using the knowledge of true causal relationships not available to participants.

Dataset	Entry	Method	Fnum	Fscore	Tscore (Ts)	Top Ts	Max Ts	<Tscore>	Rank
REGED0	77	MB_Kcomb1	24/999	0.9012	0.9931±0.0017	1	1		
REGED1	77	MB_Kcomb1	11/999	0.7842	0.9888±0.0026	0.998	0.998	0.9032	5
REGED2	77	MB_Kcomb1	6/999	0.7475	0.7278±0.0060	0.86	0.9534		

Keywords:

- Preprocessing or feature construction: centering, scaling, standardization, PCA.
- Causal discovery: Bayesian Network, Structural Equation Models, Probabilistic Graphical Models, Markov Decision Processes, Propensity Scoring, Information Theoretic Method.
- Feature selection: filter, wrapper, embedded feature selection, feature ranking, etc.
- Classifier: neural networks, nearest neighbors, tree classifier, RF, SVM, kernel-method, least-square, ridge regression, L1 norm regularization, L2 norm regularization, logistic regression, ensemble method, bagging, boosting, Bayesian, transduction.
- Hyper-parameter selection: grid-search, pattern search, evidence, bound optimization, cross-validation, K-fold.
- Other: ensemble method, transduction.

Causation and Prediction Challenge: FACT SHEET

Title: Markov Blanket Filtering using Mixture Models

Author: Mehreen Saeed
Address: FAST National University of Computer & Emerging Sciences, Lahore Campus, Pakistan.
Email: mehreen.saeed@nu.edu.pk

Acronym of your best entry: Final Entry 2
Performance graphs generated by the organizers:
http://clopinet.com/isabelle/Projects/WCCI2008/Reports/Mehreen_Saeed.html
Complete paper:
Bernoulli Mixture Models for Markov Blanket Filtering and Classification
Mehreen Saeed; JMLR W&CP 3:77:91, 2008.

Method:
Summarize the algorithms you used in a way that those skilled in the art should understand what to do. Profile of your methods as follows:

- Preprocessing: Normalize and Standardize
- Causal discovery: Markov blanket filtering with Bernoulli mixtures in case of SIDO. In case of CINA this method was only applied to binary features but this was not our last submitted entry.
- Feature selection: Subset feature selection using forward selection algorithm. The heuristic used to guide the search was the accuracy obtained from the Naïve Bayes' classifier.
- Classification: Naïve Bayes' classifier, Bernoulli mixtures + ensemble of neural nets in case of SIDO and ensemble of neural nets in case of CINA,
- Model selection/hyperparameter selection: Cross validation

Results:
Table 1: Result table. The two stars next to the feature number indicate that the submission included a sorted list of features and multiple results for nested subsets of features. Top Ts refers to the best score among all valid last entries made by participants. Max Ts refers to the best score reachable, as estimated by reference entries using the knowledge of true causal relationships not available to participants.

Dataset	Entry	Method	Fnum	Fscore	Tscore (Ts)	Top Ts	Max Ts	<Tscore>	Rank
SIDO0	1413	final entry 2	8/4932 **	0.4971	0.9391±0.0079	0.9443	0.9467		
SIDO1	1413	final entry 2	4096/4932 **	0.7242	0.7246±0.0137	0.7532	0.7893	0.7618	3
SIDO2	1413	final entry 2	512/4932 **	0.7242	0.6216±0.0132	0.6684	0.7674		
CINA0	1413	final entry 2	32/132 **	0.5069	0.9751±0.0031	0.9765	0.9788		
CINA1	1413	final entry 2	16/132 **	0.7858	0.8248±0.0045	0.8691	0.8977	0.8289	6
CINA2	1413	final entry 2	16/132 **	0.7858	0.6867±0.0041	0.8157	0.891		

- Quantitative advantages
    - FOR SIDO: Feature transformation to new probability space significantly reduces the dimensionality of data, use of Bernoulli mixtures for Markov blanket filtering drastically reduces the computational cost

    - FOR CINA: Subset feature selection using forward selection algorithm in CINA is extremely simple and intuitive.
- Qualitative advantages
    - Methods based upon mixture densities which model the data very effectively.

Implementation:
Code was implemented by adding modules to CLOP's library. C++ code was written for mixture models with an interface to matlab.

Keywords: Put at *least one keyword in each category*. Try some of the following keywords and add your own:

- Preprocessing or feature construction: probability space provided by mixture models, normalization, standardization
- Causal discovery: Markov blanket filtering using Bernoulli mixtures
- Feature selection: hill climbing search,
- Classifier: neural networks, ensemble method, Naïve bayes', mixture models
- Hyper-parameter selection: 2-fold cross validation.
- Other: None

Other methods Tried:
SVM, Bernoulli mixtures combined with SVM and simple neural networks.

What do you think was a critical element of success compared to other things you tried?
Quartile information gave a big clue as to which method was performing better. For some methods CV accuracy was a little misleading on versions 1 and 2 of datasets as it doesn't tell us much about a dataset where the features have been manipulated by external sources.

In what do the models for the versions 0, 1, and 2 of the various tasks differ?
SIDO: Models 1 and 2 were created with the same method and models, i.e., Markov blanket filtering using Bernoulli mixtures was used to find a feature list. Model 0 was created with subset feature selection using forward algorithm. Classification was done using Naïve Bayes'.
CINA: In case of CINA all versions were created with the same model.

Did you rely on the quartile information available on the web site for model selection or did you use another scheme?
Yes, we did use quartile information for model selection. Within a quartile we used 2 fold cross validation accuracy.

In the result table you submitted, did you use nested subsets of features from the slist you submitted? Yes

Did you use any knowledge derived from the test set to make your submissions, including simple statistics and visual examination of the data? No

Causation and Prediction Challenge: FACT SHEET

Title: Ensemble Machine Learning Method

Author: Ching-Wei Wang
Address: Department of Computing and Informatics University of Lincoln Brayford Pool Lincoln LN6 7TS United Kingdom
Email: cweiwang@lincoln.ac.uk

Acronym of your best entry: c, 10, 15
Performance graphs generated by the organizers:
http://clopinet.com/isabelle/Projects/WCCI2008/Reports/Ching-Wei_Wang.html

References:

1. Wang, C.-W. (2006) New Ensemble Machine Learning Method for Classification and Prediction on Gene Expression Data, Proceedings of the 28th international conference of the IEEE Engineering in Medicine and Biology Society. pp. 3478-3481. ISBN 1424400333.
2. Fayyad, U. M. & Irani, K. B. (1993). Multi-interval discretization of continuous-valued attributes for classification learning, 13th International Joint Conference of Artificial Intelligence, 1022-7
3. Weka, http://www.cs.waikato.ac.nz/ml/weka/

Results:

Table 1: Result table. Top Ts refers to the best score among all valid last entries made by participants. Max Ts refers to the best score reachable, as estimated by reference entries using the knowledge of true causal relationships not available to participants.

Dataset	Entry	Method	Fnum	Fscore	Tscore (Ts)	Top Ts	Max Ts	<Tscore>	Rank
REGED0	1240	c	999/999	0	0.9938±0.0013	1	1		
REGED1	1240	c	999/999	0	0.9022±0.0044	0.998	0.998	0.8512	10
REGED2	1240	c	999/999	0	0.6577±0.0057	0.86	0.9534		
SIDO0	455	10	4932/4932	0	0.6259±0.0117	0.9443	0.9467		
SIDO1	455	10	4932/4932	0	0.5023±0.0021	0.7532	0.7893	0.5402	14
SIDO2	455	10	4932/4932	0	0.4925±0.0006	0.6684	0.7674		
CINA0	599	15	132/132	0	0.9246±0.0034	0.9788	0.9788		
CINA1	599	15	132/132	0	0.6778±0.0052	0.8977	0.8977	0.7249	12
CINA2	599	15	132/132	0	0.5722±0.0050	0.8157	0.891		
MARTI0	599	15	1024/1024	0	0.6828±0.0056	0.9996	0.9996		
MARTI1	599	15	1024/1024	0	0.6294±0.0058	0.947	0.9542	0.6527	10
MARTI2	599	15	1024/1024	0	0.6459±0.0060	0.7975	0.8273		

Implementation:
The learning algorithm is previously implemented in java and can be referred to [1]. The feature selection algorithms can be found in the weka machine learning package. Portion of binary discretization for testing data is implemented in c#.

Method:

<u>Model selection/hyperparameter selection:</u> 10-fold cross validation

	Preprocessing	Feature select	Classification
Reged0, 1, 2	Binary discretization[2] to change feature values and filter out the features, which can not be discretized. Produce 10 features to train. Features* include {82,250,320,408,452,592,738,824,929,938}. The features are selected using reged0_train.data.		cw-Boost[1], containing one hundred C4.5 decision trees
Cina0	none	Use raw data	cw-Boost[1], containing seventy C4.5 decision trees (100 iterations may perform better)
Cina1,2	Feature selection	[3]: -E weka.attributeSelection.CfsSubsetEva –S weka.attributeSelection.BestFirst –D 1 –N 5	
Marti0	Apply Feature selection 1 times, producing 295 features to train.	[3]: -E weka.attributeSelection.CfsSubsetEva –S weka.attributeSelection.GeneticSearch –Z 20 –G 20	
Marti1,2	Apply Feature selection 3 times, producing 14 features to train.		
Sido0,1, 2	The data is too big for the equipment (CPU 2.4GHz and 1G RAM only). Systematically divide the data to 10 sub-files, and apply feature selection to the first two files. Train the two files and combine the prediction results. The performance is poor since the data preprocessing is poor here due to limitation of PC memory.	AttributeSelection [3]: -E weka.attributeSelection.CfsSubsetEva –S weka.attributeSelection.BestFirst –D 1 –N 5	

<u>*feature index: start from 0</u>

Keywords:

- <u>Preprocessing or feature construction</u>: binary discretization.
- <u>Causal discovery</u>: Information Gain, Decision Tree.
- <u>Feature selection</u>: filter.
- <u>Classifier</u>: ensemble method, boosting, cw-boost.
- <u>Hyper-parameter selection</u>: 10-fold cross-validation.
- <u>Other</u>: ensemble method.

Causation and Prediction Challenge: FACT SHEET

Title: Partial Orientation and Local Structural Learning of DAGs for Prediction

Authors:

Jianxin Yin	jianxinyin@gmail.com
Zhi Geng	zgeng@math.pku.edu.cn
You Zhou	zhouyou@pku.edu.cn
Changzhang Wang	changzhang@pku.edu.cn
Ping He	sunhp@pku.edu.cn
Cheng Zheng	zzhengccheng@pku.edu.cn
Zheyu Wang	wangzy853@sohu.com
Simeng Han	hansimeng@pku.edu.cn
Lingzhou Xue	michaelxlz@gmail.com
Shaopeng Wang	wangshop@gmail.com
Zhenguo Wu	wuzhenguo@gmail.com
Wei Yan	yanwei1982@pku.edu.cn
Manabu Kuroki	mkuroki@sigmath.es.osaka-u.ac.jp
Zhihong Cai	cai@pbh.med.kyoto-u.ac.jp

Address: School of Mathematical Sciences, Peking University, Beijing 100871, China

Acronym of our best entry: final submission

Performance graphs generated by the organizers:
http://clopinet.com/isabelle/Projects/WCCI2008/Reports/J_YinZ_Geng_Gr.html

Complete paper:
Partial orientation and local structural learning of causal networks for prediction
Jianxin Yin, You Zhou, Changzhang Wang, Ping He, Cheng Zheng, and Zhi Geng;
JMLR W&CP 3:93-105, 2008.

Method:

- **Preprocessing :**

For the case with noise (e.g., MARTI), we filter the noise using a two steps process. At the first step, we correct the global noise pattern. We find a regression model for each of 999 gene expression features, in which 25 calibrate features, are treated as explanatory variables and the gene expression as the response variable. At the second step, we locally filter the noise of each gene expression feature with neighbor features. Given a new micro-array, we first correct it with the global regression models, and then filter its noises with local models.

For a data set with very high dimensional space (e.g., SIDO), we first screen features using sure independence screening method to reduce the dimensionality to a tractable size (e.g., 1000 features for SIDO). This screen step is not necessary for other data sets and even for a higher dimensional data set if the CPU time or memory for the following computations is not a problem.

- **Causal discovery:**

We propose an approach for local structural learning and partial orientation of the edges connected to the target. In the approach, we first find the parent-children set PC(T) of the target T, and then we find the PC(X) for each feature X □ PC(T). We find local v-structures and try to orient the edges connected to the target T as much as possible. If all of the edges connected to the target T are oriented, we obtain all causal relationships to the target T that are necessary for prediction. If some edges have not oriented yet, we can repeat the process to find PC(X) for other features X □ PC(PC(T)) until all edges connected to T are oriented or we have checked all features or we have tried the maximum number of steps that we set for a very large graph. Theoretically we can show that the proposed approach is correct, that is, it can correctly find at each step local v-structures of the global DAG.

- **Feature selection:**

For the data set without manipulation (numbered 0), all the variables in the Markov blanket (MB) are used to predict the target T. For the data set with a known manipulated variable set (numbered 1), we drop the manipulated variables in the set of children and drop the spouses of T whose children common with T have been all dropped, and we use all parent variables and unmanipulated children and the parents of unmanipulated children in the MB of T. For the data set with an unknown manipulated variable set (numbered 2), only the parent variables of the target are used. When the variable sets that are used for prediction are sensitive to significance levels and other parameters, we may use a union of these sets and then predict the target with a shrinkage method to remove the redundant variables.

- **Classification:**

We use the L1 penalized logistic regression model to fit the prediction model. We use the estimated conditional probability of the target variable for each individual in the test set for its classification.

- **Model selection/hyperparameter selection:**

Given the variable set obtained at the causal discovery, we use a penalized approach which implements both estimation and selection. In the penalized approach, we use a 5-fold of cross validation (CV) method only with the training data set to select the hyper-parameter λ in the solution path. We use the CV curve to diagnose the stableness of the selected model. Three main values that are recorded every time for comparison are the minimal value of CV error, the corresponding norm fraction and the ratio of the selected variable set to the candidate variable set.

Results: The reader should also know from reading the fact sheet what the strength of the method is. To that end, we will provide a comparison table in the following format:

Table 1: Result table. Top Ts refers to the best score among all valid last entries made by participants. Max Ts refers to the best score reachable, as estimated by reference entries using the knowledge of true causal relationships not available to participants.

Dataset	Entry	Method	Fnum	Fscore	Tscore (Ts)	Top Ts	Max Ts	<Tscore>	Rank
REGED0	1475	final submission	15/999	0.8571	0.9997±0.0010	0.9998	1		
REGED1	1475	final submission	14/999	0.8189	0.9517±0.0033	0.9888	0.998	0.9133	3
REGED2	1475	final submission	11/999	0.9955	0.7885±0.0056	0.86	0.9534		
SIDO0	1475	final submission	16/4932	0.5019	0.9443±0.0075	0.9443	0.9467		
SIDO1	1475	final submission	16/4932	0.5035	0.6976±0.0137	0.7532	0.7893	0.7609	4
SIDO2	1475	final submission	16/4932	0.5035	0.6408±0.0132	0.6684	0.7674		
CINA0	1475	final submission	22/132	0.5957	0.9736±0.0032	0.9765	0.9788		
CINA1	1475	final submission	24/132	0.5852	0.8577±0.0047	0.8691	0.8977	0.833	4
CINA2	1475	final submission	18/132	0.5852	0.6676±0.0044	0.8157	0.891		
MARTI0	1475	final submission	11/1024	0.689	0.9985±0.0016	0.9996	0.9996		
MARTI1	1475	final submission	11/1024	0.6394	0.8911±0.0050	0.947	0.9542	0.8955	2
MARTI2	1475	final submission	11/1024	0.9956	0.7969±0.0060	0.7975	0.8273		

Quantitative Advantages:

For causal discovery, the approach of partial orientation and local structural learning can greatly reduce computational complexity of structural learning. On the other hand, statistical test is more powerful for the local structural learning approach than the global learning. For the prediction, we use the L1 penalized generalized logistic model to shrink the parameters at the training stage, which can reduce mean squared error (MSE) of prediction.

Qualitative Advantages:

The approach of partial orientation and local structural learning is efficient for large causal networks if we are interested only in the prediction of the target. We can theoretically show that the approach can correctly obtain the edges connected to the target and their orientations. Although the Markov blanket is useful for prediction without manipulation, it cannot be used for prediction with manipulation, and more importantly it is not sufficient to orient the edges connected to the target. The two stage filtering is efficient for the case with noise and calibrates features. The sure independence screening is a useful preprocess for ultra-high dimensional feature space.

Keywords:

-Preprocessing/feature construction: Regression model, Global and local filtering.
-Causal discovery: Causal networks, Directed acyclic graphs, Local structural learning, Partial orientation.
-Feature Selection: Parents and children, Feature ranking, Markov blanket.
-Classifier: L1 penalization, Logistic regression, Solution path.
-Hyper-parameter selection: Cross validation, K-fold.

Causation and Prediction Challenge: FACT SHEET

Title: Causative Feature Selection by PC Algorithm and SVMs
Author: Wu Zhili
Address: OEW801, Department of Computer Science, HKBU, HK
E-mail: vincent@comp.hkbu.edu.hk
Acronym of your best entry: temp 7
Performance graphs generated by the organizers:
http://clopinet.com/isabelle/Projects/WCCI2008/Reports/Wu_Zhili.html

Method:

- Preprocessing: We divide each column of feature by the maximum value
- Feature selection: We test the pcalg package in R, which as an implementation to PC algorithm can provide us a MB set. We also try the HITON_MB and HITON_PC in CausalityExplorer package. They can produce a similar causative feature set. For REGED data, the MB set we obtained leads to an improved Fscore. But for MARTI, the MB set returned doesn't help much. In early submission, feature selection based on weights calculated from linear SVMs is conducted preliminarily, like the submission for SIDO. We did not use any knowledge derived from the test set to make the submissions.
- Classification: We use standard SVM to train the final classifier. RBF and high-degree polynomial kernels are used.
- Model selection/hyperparameter selection: Cross validation with the criteria of balanced error rate is used, but not intensively conducted. Though we use separate weights for the unbalanced positive and negative classes, we haven't gain much performance improvement for these by model selection.

Results:

Table 1: Result table. The star following the feature number indicates that the feature set was sorted. Top Ts refers to the best score among all valid last entries made by participants. Max Ts refers to the best score reachable, as estimated by reference entries using the knowledge of true causal relationships not available to participants.

Dataset	Entry	Method	Fnum	Fscore	Tscore (Ts)	Top Ts	Max Ts	<Tscore>	Rank
REGED0	1051	temp 7	73/999 *	0.8758	0.9820±0.0012	1	1		
REGED1	1051	temp 7	73/999 *	0.8213	0.8454±0.0051	0.998	0.998	0.8117	13
REGED2	1051	temp 7	73/999 *	0.9564	0.6077±0.0056	0.86	0.9534		
SIDO0	529	wzl-03-27-v3	128/4932 *	0.5021	0.6446±0.0122	0.9443	0.9467		
SIDO1	529	wzl-03-27-v3	4932/4932 *	0.5088	0.4994±0.0002	0.7532	0.7893	0.5475	13
SIDO2	529	wzl-03-27-v3	4932/4932 *	0.5088	0.4985±0.0003	0.6684	0.7674		
CINA0	979	temp 2	66/132 *	0.7504	0.9210±0.0033	0.9788	0.9788		
CINA1	979	temp 2	66/132 *	0.492	0.6233±0.0048	0.8977	0.8977	0.6668	14
CINA2	979	temp 2	66/132 *	0.492	0.4562±0.0042	0.8157	0.891		
MARTI0	988	temp 3	19/1024 *	0.4905	0.6541±0.0058	0.9996	0.9996		
MARTI1	988	temp 3	19/1024 *	0.4906	0.6411±0.0060	0.947	0.9542	0.6488	11
MARTI2	988	temp 3	19/1024 *	0.4907	0.6513±0.0062	0.7975	0.8273		

Quantitative advantages: easy to try based on existing packages.
Qualitative advantages: explore the combination of MB selection with SVM.

Our implementation consists of Matlab, R, and also utilizes the LibSVM package.

Keywords:

- Preprocessing or feature construction: scaling.
- Feature selection: MB and PC algorithm, feature ranking.
- Classifier: SVM.
- Hyper-parameter selection: cross validation.