Experimental design of the WCCI 2006 performance prediction challenge

Isabelle Guyon – April 2005
isabelle@clopinet.com

This documents provides details about the five datasets used in the WCCI 2006 predictive modeling challenge. We used publicly available data to carve out five two-class classification problems suitable to benchmark performance prediction capabilities. By "performance prediction" we mean predicting how well a given classifier will perform on new unseen data.

How good are you at predicting how good you are?

The challenge is to obtain the **best test performance** on the five proposed datasets AND **to accurately predict ahead of time such performance**. In most real world situations, it is important both to produce a good predictor AND to assess accurately how well this predictor will perform on new unseen data: Before deploying a model in the field, one must know whether it will meet desired specifications or whether one should invest more time and resources to collect more data or develop more sophisticated models. The competitors have several months to build classifiers with provided (labeled) training data. A web server is provided to submit prediction results on additional unlabeled data. Two unlabeled datasets are used for evaluation: a small validation set used during the development period and a very large test set to do the final evaluation and the ranking of the participants. During a development period, the validation set performance is published immediately upon submission of prediction results. The test set performance remains undisclosed until the end of the competition. The labels of the validation set are published shortly before the end of the competition.

The performance prediction challenge is connected to model selection because accurate performance predictions are good model ranking criteria. The problem is particularly difficult in applications for which the amount training data is small. Training data serve both to train the models and assess their performance. A typical method used is cross-validation. Other methods consist in training with all the available training data and using theoretical performance bounds to correct the performance prediction.

We formatted a number of datasets for the purpose of benchmarking performance prediction algorithms in a controlled manner. The data sets were chosen to span a variety of domains (drug discovery, ecology, handwritten digit recognition, text classification, and marketing.) We chose data sets that had sufficiently many examples to create a large enough test set to obtain statistically significant results. The input variables are continuous or binary, sparse or dense. All problems are two-class classification problems. The similarity of the tasks allows participants to enter results on all data sets. The data characteristics are summarized in Table 1.

Table 1: Datasets of the performance prediction challenge.

Dataset	Sparsity (%)	SaveAs	Type	FracPos (%)	Tr/FN	FeatNum	Train	Valid	Test
ADA	79.4	dense	mixed	24.8	86.4	48	4147	415	41471
GINA	69.2	dense	continuous	49.2	3.25	970	3153	315	31532
HIVA	90.9	dense	binary	3.5	2.38	1617	3845	384	38449
NOVA	99.7	sparse	binary	28.5	0.1	16969	1754	175	17537
SYLVA	77.9	dense	mixed	6.2	60.58	216	13086	1308	130858

Method:

Preparing the data included the following steps:

- Preprocessing data to obtain features in the same numerical range (0 to 999 for continuous data and 0/1 for binary data).
- Randomizing the order of the patterns and the features to homogenize the data.
- Splitting the data into training, validation and test set. **The validation set is 100 times smaller than the test set** to make it 10 times less accurate to compute the performances on the basis of the validation set only. The training set is ten times larger than the validation set.

The classification performance is evaluated by the Balanced Error Rate (BER), that is the average error rate of the two classes. Both validation and test set truth-values (labels) are withheld during the benchmark. The validation set serves as development test set to give on-line performance feed-back to the participants. One month before the end of the challenge, the validation set labels are made available. At the end of the benchmark, the participants send their test set results. The scores on the test set results are disclosed simultaneously to all participants after the benchmark is over.

Data formats:

All the data sets are in the same format and include 7 files in text format:

dataname.param: Parameters and statistics about the data

dataname_train.data: Training set (a sparse or a regular matrix, patterns in lines, features in columns).

dataname_valid.data: Development test set or “validation” set.

dataname_test.data: Test set.

dataname_train.labels: Labels (truth values of the classes) for training examples.

dataname_valid.labels: Validation set labels (withheld during the benchmark).

dataname_test.labels: Test set labels (withheld during the benchmark).

The matrix data formats used are:

- For regular matrices: a space delimited file with a new-line character at the end of each line.
- For sparse matrices with binary values: for each line of the matrix, a space delimited list of indices of the non-zero values. A new-line character at the end of each line. In this challenge there are no sparse matrices with non-binary values.

The results on each dataset should be formatted in 5 ASCII files:

dataname.guess: your prediction of the BER (Balanced Error Rate) for your classifier on test data (mandatory for all submission.).

dataname_valid.resu: +-1 classifier outputs for validation set examples (mandatory for all submission.).

dataname_test.resu: +-1 classifier outputs for final test set examples (mandatory for final submissions.)
dataname_valid.conf: confidence values for validation examples (optional.)
dataname_test.conf: confidence values for test examples (optional.)

Format for classifier outputs:
- The .guess files should have a single decimal numeric value between 0 and 1, indicating the predicted BER.
- Both .resu files should have one +-1 integer value per line indicating the prediction for the various patterns.
- Both .conf files should have one decimal positive numeric value per line indicating classification confidence. The confidence values can be the absolute discriminant values. They do not need to be normalized to look like probabilities. They will be used to compute ROC curves and Area Under such Curve (AUC).

Model formats:
There is also the possibility of submitting information about the models used. This is described in a separate document.

Result rating:
The scoring methods that have been retained are:
- Test set balanced error rate (test_ber): the average of the class error rates (the class error rates are the error rates obtained with test examples of individual classes, using the predictions provided by the participants.)
- Error of performance prediction: the difference between the test_ber and guessed_ber (the value of the BER that the participants provide at their guess of how well they will perform on test data): E_pred = abs(guessed_ber - test_ber).

Of the 10% top ranking challengers according to test_ber, the winner will be the one providing the best E_pred.
In addition to test_ber and E_pred, other statistics will be computed, but not used for scoring, including:
- AUC: Area under the ROC curve.
- Negative cross-entropy: The average negative log estimated predictive probability of the true labels (see http://predict.kyb.tuebingen.mpg.de/).

Dataset A: SYLVA

1) Topic

The task of SYLVA is to classify forest cover types. This is a two-class classification problem with 216 input variables. Each pattern is composed of 4 records: 2 true records matching the target and 2 records picked at random. Thus ½ of the features are distracters.

The forest cover type for 30 x 30 meter cells is obtained from US Forest Service (USFS) Region 2 Resource Information System (RIS) data.

2) Sources

a. Original owners

Remote Sensing and GIS Program
Department of Forest Sciences
College of Natural Resources
Colorado State University
Fort Collins, CO 80523

(contact Jock A. Blackard, jblackard/wo_ftcol@fs.fed.us
or Dr. Denis J. Dean, denis@cnr.colostate.edu)
Jock A. Blackard
USDA Forest Service
3825 E. Mulberry
Fort Collins, CO 80524 USA
jblackard/wo_ftcol@fs.fed.us

Dr. Denis J. Dean
Associate Professor
Department of Forest Sciences
Colorado State University
Fort Collins, CO 80523 USA
denis@cnr.colostate.edu

Dr. Charles W. Anderson
Associate Professor
Department of Computer Science
Colorado State University
Fort Collins, CO 80523 USA
anderson@cs.colostate.edu

Acknowledgements, Copyright Information, and Availability

b. Donor of database

This version of the database was prepared for the WCCI2006 variable and feature selection benchmark by Isabelle Guyon, 955 Creston Road, Berkeley, CA 94708, USA (isabelle@clopinet.com).

c. Date received: August 28, 1998, UCI Machine Learning Repository, under the name Forest Cover Type.

d. Date prepared for the challenge: June 2005.

3) Past usage

Blackard, Jock A. 1998. "Comparison of Neural Networks and Discriminant Analysis in Predicting Forest Cover Types." Ph.D. dissertation. Department of Forest Sciences. Colorado State University. Fort Collins, Colorado.

Classification performance with first 11,340 records used for training data, next 3,780 records used for validation data, and last 565,892 records used for testing data subset: -- 70% backpropagation -- 58% Linear Discriminant Analysis

4) Experimental design

The original data comprises a total of 581012 instances (observations) grouped in 7 classes (forest cover types) and having 54 attributes corresponding to 12 measures (10 quantitative variables, 4 binary wilderness areas and 40 binary soil type variables). The actual forest cover type for a given observation (30 x 30 meter cell) was determined from US Forest Service (USFS) Region 2 Resource Information System (RIS) data. Independent variables were derived from data originally obtained from US Geological Survey (USGS) and USFS data. Data is in raw form (not scaled) and contains binary (0 or 1) columns of data for qualitative independent variables (wilderness areas and soil types).

Variable Information

Given is the variable name, variable type, the measurement unit and a brief description. The forest cover type is the classification problem. The order of this listing corresponds to the order of numerals along the rows of the database.

Name	Data Type	Measurement	Description
Elevation	quantitative	meters	Elevation in meters
Aspect	quantitative	azimuth	Aspect in degrees azimuth
Slope	quantitative	degrees	Slope in degrees
Horizontal_Distance_To_Hydrology	quantitative	meters	Horz Dist to nearest surface water features
Vertical_Distance_To_Hydrology	quantitative	meters	Vert Dist to nearest surface water features
Horizontal_Distance_To_Roadways	quantitative	meters	Horz Dist to nearest roadway
Hillshade_9am	quantitative	0 to 255 index	Hillshade index at 9am, summer solstice
Hillshade_Noon	quantitative	0 to 255 index	Hillshade index at noon, summer soltice
Hillshade_3pm	quantitative	0 to 255 index	Hillshade index at 3pm, summer solstice
Horizontal_Distance_To_Fire_Points points	quantitative	meters	Horz Dist to nearest wildfire ignition
Wilderness_Area (4 binary columns)	qualitative	0 (absence) or 1 (presence)	Wilderness area designation
Soil_Type (40 binary columns)	qualitative	0 (absence) or 1 (presence)	Soil Type designation
Cover_Type (7 types)	integer	1 to 7	Forest Cover Type designation

Code Designations

Wilderness Areas:

1 -- Rawah Wilderness Area
2 -- Neota Wilderness Area
3 -- Comanche Peak Wilderness Area
4 -- Cache la Poudre Wilderness Area

Soil Types:

1 to 40 : based on the USFS Ecological Landtype Units for this study area.

Forest Cover Types:

1 -- Spruce/Fir
2 -- Lodgepole Pine
3 -- Ponderosa Pine

4 -- Cottonwood/Willow
5 -- Aspen
6 -- Douglas-fir
7 – Krummholz

Class Distribution

Number of records of Spruce-Fir:	211840
Number of records of Lodgepole Pine:	283301
Number of records of Ponderosa Pine:	35754
Number of records of Cottonwood/Willow:	2747
Number of records of Aspen:	9493
Number of records of Douglas-fir:	17367
Number of records of Krummholz:	20510
Total records:	581012

Data preprocessing and data split

We carved a binary classification task out these data. We decided to separate Ponderosa pine from all others. To disguise the data and render the task more challenging, we created patterns containing the concatenation of 4 patterns: two of the target class and two randomly chosen from either class. In this way there are pairs of redundant features and ½ of the features are non-informative.

5) Number of examples and class distribution

	Positive ex.	Negative ex.	Total	Check sum
Training set	805	12281	13086	238428868
Validation set	80	1228	1308	23852055
Test set	8053	122805	130858	2382587223
All	8938	136314	145252	2644868146

6) Type of input variables and variable statistics

Real variables	Random probes	Total
108	108	216

All variables are **integer** quantized on 1000 levels. There are **no missing values**. The data is not very sparse, but for data compression reasons, we thresholded the values. Approximately 78% of the variable values are zero. The data was saved as a **dense** matrix.

7) Results of the run of the zarbi method

train, BER= 4.11 +- 0.49%, guess_error= 0.00%
valid, BER= 2.73 +- 1.29%, guess_error= 1.39%
test, BER= 4.47 +- 0.16%, guess_error= 0.36%
A linear SVM (C=0.1) obtains a test error of 3.98%.

Dataset B: GINA

1) Topic

The task of GINA is handwritten digit recognition. We chose the problem of separating the odd numbers from even numbers. We use 2-digit numbers. Only the unit digit is informative for that task, therefore at least ½ of the features are distracters. This is a two-class classification problem with sparse continuous input variables, in which each class is composed of several clusters. It is a problems with heterogeneous classes.

2) Sources

a. Original owners

The data set was constructed from the MNIST data that is made available by Yann LeCun of the NEC Research Institute at http://yann.lecun.com/exdb/mnist/.
The digits have been size-normalized and centered in a fixed-size image of dimension 28x28. We show examples of digits in Figure B1.

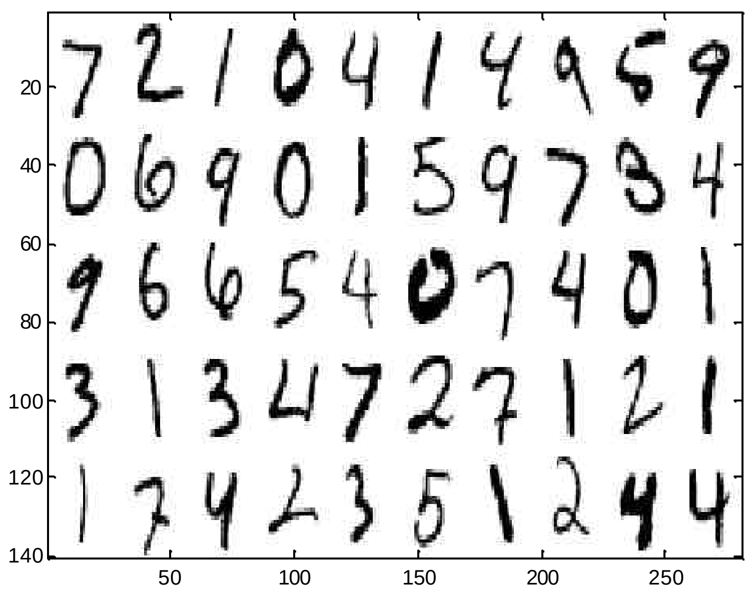

Figure B1: Examples of digits from the MNIST database.

Table 1: Number of examples in the original data

Digit	0	1	2	3	4	5	6	7	8	9	Total
Training	5923	6742	5958	6131	5842	5421	5918	6265	5851	5949	60000
Test	980	1135	1032	1010	982	892	958	1028	974	1009	10000
Total	6903	7877	6990	7141	6824	6313	6876	7293	6825	6958	70000

b. Donor of database

This version of the database was prepared for the WCCI performance prediction challenge by Isabelle Guyon, 955 Creston Road, Berkeley, CA 94708, USA (isabelle@clopinet.com).

c. Date prepared for the challenge: June 2005.

3) Past usage

Many methods have been tried on the MNIST database, in its original data split (60,000 training examples, 10,000 test examples, 10 classes.) Here is an abbreviated list from http://yann.lecun.com/exdb/mnist/:

METHOD	TEST ERROR RATE (%)
linear classifier (1-layer NN)	12.0
linear classifier (1-layer NN) [deskewing]	8.4
pairwise linear classifier	7.6
K-nearest-neighbors, Euclidean	5.0
K-nearest-neighbors, Euclidean, deskewed	2.4
40 PCA + quadratic classifier	3.3
1000 RBF + linear classifier	3.6
K-NN, Tangent Distance, 16x16	1.1
SVM deg 4 polynomial	1.1
Reduced Set SVM deg 5 polynomial	1.0
Virtual SVM deg 9 poly [distortions]	0.8
2-layer NN, 300 hidden units	4.7
2-layer NN, 300 HU, [distortions]	3.6
2-layer NN, 300 HU, [deskewing]	1.6
2-layer NN, 1000 hidden units	4.5
2-layer NN, 1000 HU, [distortions]	3.8
3-layer NN, 300+100 hidden units	3.05
3-layer NN, 300+100 HU [distortions]	2.5
3-layer NN, 500+150 hidden units	2.95
3-layer NN, 500+150 HU [distortions]	2.45
LeNet-1 [with 16x16 input]	1.7
LeNet-4	1.1
LeNet-4 with K-NN instead of last layer	1.1
LeNet-4 with local learning instead of ll	1.1
LeNet-5, [no distortions]	0.95
LeNet-5, [huge distortions]	0.85
LeNet-5, [distortions]	0.8
Boosted LeNet-4, [distortions]	0.7
K-NN, shape context matching	0.67

Reference:
Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. "Gradient-based learning applied to document recognition." Proceedings of the IEEE, 86(11):2278-2324, November 1998. http://yann.lecun.com/exdb/publis/index.html#lecun-98

The dataset restricted to a selection of digits “4” and “9” was used in the NIPS 2003 feature selection challenge http://clopinet.com/isabelle/Projects/NIPS2003/ and http://www.nipsfsc.ecs.soton.ac.uk/, under the name GISETTE.

4) Experimental design

To construct the dataset, we performed the following steps:
- We removed the pixels that were 99% of the time white. This reduced the original feature set of 784 pixels to 485.
- The original resolution (256 gray levels) was kept.
- In spite of the fact that the data are rather sparse (about 30% of the values are non-zero), we saved the data as a dense matrix because we found that it can be compressed better in this way (to 19 MB.)
- The feature names are the (i,j) matrix coordinates of the pixels (in a 28x28 matrix.)
- We created 2 digit numbers by dividing the datasets into to parts and pairing the digits at random.
- The task is to separate odd from even numbers. The digit of the tens being not informative, the features of that digit act as distracters.

5) Number of examples and class distribution

	Positive ex.	Negative ex.	Total	Check sum
Training set	1550	1603	3153	164338762
Validation set	155	160	315	16568705
Test set	15504	16028	31532	1647222288
All	17209	17791	35000	1828129755

6) Type of input variables and variable statistics

Real variables	Random probes	Total
485	485	970

All variables are **integer** quantized on 256 levels. There are **no missing values**. The data is rather **sparse**. Approximately 69% of the entries are zero. The data was saved as a **dense** matrix, because it compresses better in that format.

7) Results of the run of the zarbi method

train, BER=18.84 +- 0.70%, guess_error= 0.00%
valid, BER=18.53 +- 2.21%, guess_error= 0.31%
test, BER=20.34 +- 0.23%, guess_error= 1.50%
Note: a linear SVM (C=0.1) obtains a test BER of 18.9%.

Dataset C: NOVA

1) Topic

The task of NOVA is text classification from the 20-Newsgroup data. We selected the separation of politics and religion topics from all the other topics. This is a two-class classification problem with sparse binary input variables using a bag-of-word representation with a vocabulary of approximately 17000 words.

2) Sources

a. Original owners

Tom Mitchell
School of Computer Science
Carnegie Mellon University
tom.mitchell@cmu.edu

Available from the UCI machine learning repository. The version we are using was preprocessed by Ron Bekkerman http://www.cs.technion.ac.il/~ronb/thesis.html into the "bag-of-words" representation.

b. Donor of database

This version of the database was prepared for the WCCI 2006 challenge on performance prediction by Isabelle Guyon, 955 Creston Road, Berkeley, CA 94708, USA (isabelle@clopinet.com).

c. Date prepared for the challenge: June 2005.

3) Past usage

T. Mitchell. Machine Learning, McGraw Hill, 1997.

T. Joachims (1996). A probabilistic analysis of the Rocchio algorithm with TFIDF for text categorization, Computer Science Technical Report CMU-CS-96-118. Carnegie Mellon University.

Ron Bekkerman, Ran El-Yaniv, Naftali Tishby, and Yoad Winter. Distributional Word Clusters vs. Words for Text Categorization. JMLR 3(Mar):1183-1208, 2003.

4) Experimental design

We selected 8 newsgroups relating to politics or religion topics as our positive class (Table C.1.) Vocabulary selection includes the following filters:
- remove words containing digits and convert to lowercase
- remove words appearing less than twice in the whole dataset.
- remove short words with less than 3 letters.
- exclude ~2000 words found frequenly in all documents.
- truncate the words at a max of 7 letters.

Table C.1: Twenty newsgroup database.

Newsgroup	Number of examples
alt.atheism	1114
comp.graphics	1002
comp.os.ms-windows.misc	1000
comp.sys.ibm.pc.hardware	1028
comp.sys.mac.hardware	1002
comp.windows.x	1000
misc.forsale	1005
rec.autos	1004
rec.motorcycles	1000
rec.sport.baseball	1000
rec.sport.hockey	1000
sci.crypt	1000
sci.electronics	1000
sci.med	1001
sci.space	1000
soc.religion.christian	997
talk.politics.guns	1008
talk.politics.mideast	1000
talk.politics.misc	1163
talk.religion.misc	1023

5) Number of examples and class distribution

	Positive ex.	Negative ex.	Total	Check sum
Training set	499	1255	1754	100410
Validation set	50	125	175	8841
Test set	4990	12547	17537	988659
All	5539	13927	19466	1097910

6) Type of input variables and variable statistics

Real variables	Random probes	Total
16969	0	16969

All variables are **binary**. There are **no missing values**. The data is very **sparse**. Over 99% of the entries are zero. The data was saved as a **sparse-binary** matrix.

7) Results of the run of the zarbi method

train, BER=16.15 +- 1.16%, guess_error= 0.00%
valid, BER=28.80 +- 4.53%, guess_error=12.65%
test, BER=20.45 +- 0.40%, guess_error= 4.31%
A linear SVM (C=0.1) obtains a test BER of 5.56%.

Dataset D: HIVA

1) Topic

The task of HIVA is to predict which compounds are active against the AIDS HIV infection. This is a two-class classification problem with about 2000 sparse binary input variables. The variables represent properties of the molecule inferred from its structure. The problem is therefore to relate structure to activity (a QSAR=quantitative structure-activity relationship problem) to screen new compounds before actually testing them (a HTS=high-throuput screening problem.)

2) Sources

a. Original owners

The data is made available by the National Cancer Institute (NCI), via the DTP AIDS Antiviral Screen program at: http://dtp.nci.nih.gov/docs/aids/aids_data.html.

The DTP AIDS Antiviral Screen has checked tens of thousands of compounds for evidence of anti-HIV activity. Available are screening results and chemical structural data on compounds that are not covered by a confidentiality agreement.

b. Donor of database

This version of the database was prepared for the WCCI 2006 performance prediction challenge by Isabelle Guyon, 955 Creston Road, Berkeley, CA 94708, USA (isabelle@clopinet.com).

c. Date prepared for the challenge: June 2005.

3) Past usage

An earlier release of the database was uses in an Equbits case study: http://www.limsfinder.com/community/articles_comments.php?id=1553_0_2_0_C75. The feature set was obtained by a different method.

4) Experimental design

The screening results of the May 2004 release containing the screening results for 43,850 compounds were used. The results of the screening tests are evaluated and placed in one of three categories:

- **CA** - Confirmed active
- **CM** - Confirmed moderately active
- **CI** - Confirmed inactive

We converted this into a two-class classification problem: Inactive (CI) vs. Active (CA or CM.)

Chemical structural data for 42,390 compounds was obtained from the web page. It was converted to structural features by the program ChemTK version 4.1.1, Sage Informatics LLC. Four compounds failed parsing.

The 1617 features selected include:

- unbranched_fragments: 750 features
- pharmacophores: 495 features
- branched_fragments: 219 features
- internal_fingerprints: 132 features

- ring_systems: 21 features

Only binary features having a total number of ones larger than 100 (>400 for unbranched fragments) and at least 2% of ones in the positive class were retained. In all cases, the default program settings were used to generate keys (except for the pharmacophores for which "max number of pharmacophore points" was set to 4 instead of 3; the pharmacophore keys for Hacc, Hdon, ExtRing, ExtArom, ExtAliph were generated, as well as those for Hacc, Hdon, Neg, Pos.) The keys were then converted to attributes.

We briefly describe the attributes/features:

Branched fragments: each fragment is constructed through an "assembly" of shortest-path unbranched fragments, where each of the latter is required to be bounded by two atoms belonging to one or more pre-defined "terminal-atom".

Unbranched fragments: unique non-branching fragments contained in the set of input molecules.

Ring systems: A ring system is defined as any number of single or fused rings connected by an unbroken chain of atoms. The simplest example would be either a single ring (e.g., benzene) or a single fused system (e.g., naphthalene).

Pharmacophores: ChemTK uses a type of pharmacophore that measures distance via bond connectivity rather than a typical three-dimensional distance. For instance, to describe a hydrogen-bond acceptor and hydrogen-bond donor separated by five connecting bonds, the corresponding key string would be "HAcc.HDon.5". The pharmacophores were generated from the following features:

Neg. Explicit negative charge.

Pos. Explicit positive charge.

HAcc. Hydrogen-bond acceptor.

HDon. Hydrogen-bond donor.

ExtRing. Ring atom having a neighbor atom external to the ring.

ExtArom. Aromatic ring atom having a neighbor atom external to the ring.

ExtAliph. Aliphatic ring atom having a neighbor atom external to the ring.

Internal fingerprints: small, fixed catalog of pre-defined queries roughly similar to the MACCS key set developed by MDL.

We matched the compounds in the structural description files and those in the compound activity file, using the NSC id number. We ended up with 42678 examples.

5) Number of examples and class distribution

	Positive ex.	**Negative ex.**	**Total**	**Check sum**
Training set	135	3710	3845	569450
Validation set	14	370	384	56326
Test set	1354	37095	38449	5669451
All	1503	41175	42678	6295227

6) Type of input variables and variable statistics

Real variables	**Random probes**	**Total**
1617	0	1617

All variables are binary. The data was saved as a **non-spase** matrix, even though it is 91% sparse because dense matrices load faster in Matlab and the ASCII format compresses well.

7) Results of the run of the zarbi method

train, BER=27.81 +- 2.73%, guess_error= 0.00%
valid, BER=40.71 +- 9.28%, guess_error=12.91%
test, BER=31.47 +- 0.89%, guess_error= 3.66%

A linear SVM (C=0.1) obtains a test BER of 27.54%.

Dataset E: ADA

1) Topic

The task of ADA is to discover high revenue people from census data. This is a two-class classification problem with sparse binary input variables. The task was cut out of the Adult database available from the UCI machine learning repository.

2) Sources

a. Original owners

This data was extracted from the census bureau database found at
http://www.census.gov/ftp/pub/DES/www/welcome.html
Donor: Ronny Kohavi and Barry Becker,
Data Mining and Visualization
Silicon Graphics.
e-mail: ronnyk@sgi.com for questions.

The information below is exerpted from the UCI machine learning repository:

Extraction was done by Barry Becker from the 1994 Census database. The prediction task is to determine whether a person makes over 50K a year. The attributes are:
age: continuous.
workclass: Private, Self-emp-not-inc, Self-emp-inc, Federal-gov, Local-gov, State-gov, Without-pay, Never-worked.
fnlwgt: continuous.
education: Bachelors, Some-college, 11th, HS-grad, Prof-school, Assoc-acdm, Assoc-voc, 9th, 7th-8th, 12th, Masters, 1st-4th, 10th, Doctorate, 5th-6th, Preschool.
education-num: continuous.
marital-status: Married-civ-spouse, Divorced, Never-married, Separated, Widowed, Married-spouse-absent, Married-AF-spouse.
occupation: Tech-support, Craft-repair, Other-service, Sales, Exec-managerial, Prof-specialty, Handlers-cleaners, Machine-op-inspct, Adm-clerical, Farming-fishing, Transport-moving, Priv-house-serv, Protective-serv, Armed-Forces.
relationship: Wife, Own-child, Husband, Not-in-family, Other-relative, Unmarried.
race: White, Asian-Pac-Islander, Amer-Indian-Eskimo, Other, Black.
sex: Female, Male.
capital-gain: continuous.
capital-loss: continuous.
hours-per-week: continuous.
native-country: United-States, Cambodia, England, Puerto-Rico, Canada, Germany, Outlying-US(Guam-USVI-etc), India, Japan, Greece, South, China, Cuba, Iran, Honduras, Philippines, Italy, Poland, Jamaica, Vietnam, Mexico, Portugal, Ireland, France, Dominican-Republic, Laos, Ecuador, Taiwan, Haiti, Columbia, Hungary, Guatemala, Nicaragua, Scotland, Thailand, Yugoslavia, El-Salvador, Trinadad&Tobago, Peru, Hong, Holand-Netherlands.
income: >50K, <=50K.

```
Split into train-test using MLC++ GenCVFiles (2/3, 1/3 random).
 48842 instances, mix of continuous and discrete    (train=32561, test=16281)
 45222 if instances with unknown values are removed (train=30162, test=15060)
 Duplicate or conflicting instances : 6
 Class probabilities for adult.all file
 Probability for the label '>50K'  : 23.93% / 24.78% (without unknowns)
 Probability for the label '<=50K' : 76.07% / 75.22% (without unknowns)

 Description of fnlwgt (final weight)
 The weights on the CPS files are controlled to independent estimates of the
 civilian noninstitutional population of the US.  These are prepared monthly
 for us by Population Division here at the Census Bureau.  We use 3 sets of
 controls. People with similar demographic characteristics should have
 similar weights.
```

b. Donor of database

This version of the database was prepared for the WCCI 2006 performance prediction challenge by Isabelle Guyon, 955 Creston Road, Berkeley, CA 94708, USA (isabelle@clopinet.com).

c. Date prepared for the challenge: June 2005.

3) Past usage

```
 First cited in:
 @inproceedings{kohavi-nbtree,
    author={Ron Kohavi},
    title={Scaling Up the Accuracy of Naive-Bayes Classifiers: a
           Decision-Tree Hybrid},
    booktitle={Proceedings of the Second International Conference on
               Knowledge Discovery and Data Mining},
    year = 1996}
 Error Accuracy reported as follows, after removal of unknowns from
    train/test sets):
    C4.5       : 84.46+-0.30
    Naive-Bayes: 83.88+-0.30
    NBTree     : 85.90+-0.28
 The following algorithms were later run with the following error rates,
    all after removal of unknowns and using the original train/test split.
    All these numbers are straight runs using MLC++ with default values.

    Algorithm               Error
 -- ----------------        -----
 1  C4.5                    15.54
 2  C4.5-auto               14.46
 3  C4.5 rules              14.94
 4  Voted ID3 (0.6)         15.64
 5  Voted ID3 (0.8)         16.47
 6  T2                      16.84
 7  1R                      19.54
 8  NBTree                  14.10
 9  CN2                     16.00
 10 HOODG                   14.82
 11 FSS Naive Bayes         14.05
 12 IDTM (Decision table)   14.46
 13 Naive-Bayes             16.12
 14 Nearest-neighbor (1)    21.42
 15 Nearest-neighbor (3)    20.35
 16 OC1                     15.04
 17 Pebls                   Crashed.  Unknown why (bounds WERE increased)
```

Note: The performances reported are error rates, not BER. We tried to reproduce these performances and obtained 15.62% error with a linear ridge regression classifier. The performances slightly degraded when we tried to group features (15.67% when we replace the country code by a binary US/nonUS value and 16.40% with further reduction to 33 features.)

4) Experimental design

We performed the following steps:

- Convert the features to 14 numeric values $a \in 1 \ldots n$.
- Convert the numeric values to binary codes (a vector of n zeros with value one at the a^{th} position. This results in 88 features. The missing values get an all zero vector.
- Downsize the number of features to 48 by replacing the country code by a binary US/nonUS feature.
- Randomize the feature and pattern order.
- Remove the entries with missing values for workclass.

Table E.1. Features of the ADA datasets.

Feature name	**min**	**max**	**numval**	**comments**
age	0.19	1	continuous	No missing value.
workclass_Private	0	1	2	2799 missing values (corresponding entries removed.)
workclass_Self_emp_not_inc	0	1	2	
workclass_Self_emp_inc	0	1	2	
workclass_Federal_gov	0	1	2	
workclass_Local_gov	0	1	2	
workclass_State_gov	0	1	2	
workclass_Without_pay	0	1	2	
workclass_Never_worked	0	1	2	
fnlwgt	0.008	1	continuous	No missing value.
educationNum	0.06	1	16	Number corresponding to 16 discrete levels of education
maritalStatus_Married_civ_spouse	0	1	2	No missing value.
maritalStatus_Divorced	0	1	2	
maritalStatus_Never_married	0	1	2	
maritalStatus_Separated	0	1	2	
maritalStatus_Widowed	0	1	2	
maritalStatus_Married_spouse_absent	0	1	2	
maritalStatus_Married_AF_spouse	0	1	2	
occupation_Tech_support	0	1	2	2809 missing values (corresponding entries removed.)
occupation_Craft_repair	0	1	2	
occupation_Other_service	0	1	2	
occupation_Sales	0	1	2	
occupation_Exec_managerial	0	1	2	
occupation_Prof_specialty	0	1	2	
occupation_Handlers_cleaners	0	1	2	
occupation_Machine_op_inspct	0	1	2	
occupation_Adm_clerical	0	1	2	
occupation_Farming_fishing	0	1	2	
occupation_Transport_moving	0	1	2	

occupation_Priv_house_serv	0	1	2	
occupation_Protective_serv	0	1	2	
occupation_Armed_Forces	0	1	2	
relationship_Wife	0	1	2	No missing value.
relationship_Own_child	0	1	2	
relationship_Husband	0	1	2	
relationship_Not_in_family	0	1	2	
relationship_Other_relative	0	1	2	
relationship_Unmarried	0	1	2	
race_White	0	1	2	No missing value.
race_Asian_Pac_Islander	0	1	2	
race_Amer_Indian_Eskimo	0	1	2	
race_Other	0	1	2	
race_Black	0	1	2	
sex	0	1	2	0=female, 1=male. No missing value.
capitalGain	0	1	continuous	No missing value.
capitalLoss	0	1	continuous	No missing value.
hoursPerWeek	0.01	1	continuous	No missing value.
nativeCountry	0	1	2	0=US, 1=non-US. 857 missing values replaced by 1.

5) Number of examples and class distribution

	Positive ex.	Negative ex.	Total	Check sum
Training set	1029	3118	4147	6785414
Validation set	103	312	415	676623
Test set	10290	31181	41471	67954509
All	11422	34611	46033	75416546

6) Type of input variables and variable statistics

Real variables	Random probes	Total
48	0	48

Six variables are continuous, the others are binary. There are **no missing values**. The data is 80% **sparse**. The data was saved as a **dense** matrix because once compressed it makes almost no difference and it loads much faster.

7) Results of the run of the zarbi method

train, BER=26.31 +- 0.97%, guess_error= 0.00%
valid, BER=29.00 +- 3.16%, guess_error= 2.70%
test, BER=27.37 +- 0.31%, guess_error= 1.07%
With a linear SVM (C=0.1), we obtained: a test BER = 19.75% and a test error rate of 15.26%.