An Efficient Parameter-Free Method for Large Scale Offline Learning

Marc Boullé

ICML 2008 Large Scale Learning Workshop - Helsinki
Outline

- Data Mining in Industrial Context
- Averaging of Selective Naive Bayes Classifiers
  - Principles
  - Scalable algorithms
- Evaluation on the Large Scale Learning Challenge
- Conclusion
Data Mining methodology

- CRISP-DM: Cross-Industry Standard Process for Data Mining

- Data Mining does not reduce to Modeling

- Data Preparation is critical
  - 80% of the process
  - Requires skilled data analysts
Data Mining in France Telecom

- Many domains
  - Marketing
  - Text mining
  - Web mining
  - Traffic classification
  - Sociology
  - Ergonomics

- Many scales
  - Tens to millions of instances
  - Tens to tens of thousands of variables

- Many types of data
  - Numerical
  - Categorical
  - Text
  - Image
  - Relational databases

- Many tasks
  - Data exploration
  - Supervised
  - Unsupervised

- Data constraints
  - Heterogeneous
  - Missing values
  - Multiple classes
  - Heavily unbalanced distributions

- Training requirements
  - Fast data preparation and modeling

- Model requirements
  - Reliable
  - Accurate
  - Parsimonious (few variables)
  - Understandable

- Deployment requirement
  - Fast deployment
  - Up to real time classification in network devices

- Business requirement
  - Return of investment for the whole process
Data Mining under Limited Resources

- Data Mining in Industrial Context
  - Applicable in a large variety of contexts
  - Vast demand but slow spread

- Resource
  - Disk space: fast growth
  - RAM: medium growth
  - CPU: medium growth
  - Skilled data analysts: steady

- Bottleneck to a wide spread of data mining solutions
  lack of data analysts
Data Mining Challenges

- Primary issue: automation
  - Automatic data preparation
  - Parameter free methods
  - Generic methods

- Other issues
  - Statistical efficiency
    - Reliability
    - Accuracy
    - Understandability
  - Computational efficiency
    - Scalability: train and deployment
    - Resources: database access, network, RAM, CPU
Averaging of
Selective Naive Bayes Classifiers
Naive Bayes classifier: principles

- The Bayesian classifier is optimal
  - assigns the most probable class given the data
  - but not computable

- Naive Bayes assumption
  - the input variables are independent within each class label
  - "idiot" assumption, but easy to compute

- Reported performances
  - robust and often effective on many real data applications
  - good ranking of the class conditional probabilities
Naive Bayes classification: three major improvements

- Optimal evaluation of univariate class conditional probabilities

- Optimal evaluation of variable selection given naive Bayes assumption

- Efficient compression-based averaging method
Evaluation of conditional probabilities

- Discretization: non-parametric model of density estimation

- Main issues
  - Informational quality
    - good fit of the data
  - Statistical quality
    - good generalization
Discretization: Model Selection

Which model is the best one?
MODL Discretization Method

- Model parameters: discrete and data dependent
  - Number of intervals
  - Interval bounds
  - Multinomial distribution in each interval

- Model Selection: Bayesian approach \((\max p(M)p(D|M))\)
  - Hierarchical prior, uniform at each stage of the hierarchy
  - Multinomial conditional likelihood
  - Exact analytical criterion

- Optimization: combinatorial algorithms
  - Greedy top-down merge heuristic
  - Time complexity: \(O(N \log N)\)
Selective Naive Bayes: objectives

- Leverage the naive Bayes assumption
  - Discard redundant variables
  - Discard non informative variables

- Control overfitting caused by the variable selection

- Control the time complexity of the algorithm
Selective Naive Bayes: our approach

- Model parameters
  - Number of variables
  - Subset of variables

- Model Selection: Bayesian approach \((\max p(M)p(D|M))\)
  - Hierarchic prior, uniform at each stage of the hierarchy
  - Conditional likelihood given by the naive Bayes formula
  - Exact analytical criterion

- Optimization: combinatorial algorithms
  - Repeat log \(KN\) times \((K: \text{number of variables}, N: \text{number of instances})\)
    - Fast forward variable selection
    - Fast backward variable selection
  - Time complexity: \(O(KN \log KN)\)
Averaging of Selective Naive Bayes
Bayesian Model Averaging (BMA)

- **Objective**
  - Account for model uncertainty in variable selection

- **Model averaging**
  - Bayesian Model Averaging approach
    - models weighted by their posterior probability
  - Optimization algorithm
    - exploits the models collected during variable selection

- **Limits**
  - The posterior distribution of the models is sharply peaked
  - BMA reduces to selecting the MAP model
Averaging of Selective Naive Bayes
Our approach

Objective
- Better exploit the whole distribution of models
- Trade-off between
  - Bayesian Model Averaging approach: the winner takes (almost) all
  - Bagging approach: all models have the same weight

Compression-based Model Averaging
- Use of compression coefficient
  - ability of each model to compress the class labels given the input data
- Optimization algorithm
  - exploits the models collected during variable selection

Weighting many Selective Naive Bayes models reduces to one single Naive Bayes with weighted variables
Method overview

- Naive Bayes assumption, with three improvements
  - Bayes optimal evaluation of univariate class conditional probabilities
  - Bayes optimal evaluation of variable selection
  - Efficient compression-based averaging method

- Algorithmic complexity \((K: \text{number of variables}, N: \text{number of instances})\)
  - Train: \(O(KN \log KN)\)
    - Univariate preprocessing: \(K\) times \(O(N \log N)\)
    - Variable selection: \(O(KN \log KN)\)
    - Model averaging: no overhead
  - Deployment (per instance): \(O(K \log N)\)
    - Preprocessing (per variable): \(O(\log N)\)
    - Naive Bayes formula: \(K\) terms
Scalability
When dataset does not fit into the RAM

- Our offline method memory requirements: $O(KN)$

- Dataset organization
  - Sequential: instances*variables

- Main algorithmic alternatives
  - Process all the data simultaneously: scalability issue
  - Process instances sequentially: one-pass learning
  - Process variables sequentially: chunking issues
Key performance indicators

- **Scale: one billion operations** (one million instances, one thousand variables)
  - How long is one billion operations?

- **CPU micro-bench** (P4 3.2 Ghz)
  - With integers: 0.5 s
  - With doubles: 5 s
  - With logs: 30 s
  - With parsed doubles (atof): 1000 s

- **Hard-drive micro-bench** (transfer rate: 100 Mo/s)
  - Read a file of size 1 Go: 20 s

- **Key performance indicators**
  - RAM access: 10 nanoseconds
  - Disk access:
    - Sequential: 10 nanoseconds
    - Random: 10 milliseconds (one millions time slower!)

- **Conclusion:** chunking via sequential disk access is practicable
Scalability: our strategy

- **Preprocessing**
  - Process variables sequentially
  - Read dataset as many times as necessary
  - For each logical chunk (subset of variables)
    - Read dataset
    - Parse, load and process chunk variables only
  - **Step number: $O(KN / M)$ ($M$=RAM)**

- **Data recoding**
  - Read and parse all informative variables, instance per instance
  - Write as many physical chunks as necessary
  - For each physical chunk (subset of variables)
    - Write integer offsets in conditional probability tables
  - **Step number: 1**

- **Variable selection**
  - Repeat log $KN$ times
    - Process chunks in random order
    - Process variables in random order inside each chunk
  - Physical chunks are very fast to load
  - **Step number: $O(\log KN)$**
Some practical issues

Resource management
- Evaluate required resources
- Evaluate available resources
- Derive chunking strategy

Practical problems
- Resource evaluation is difficult
- Theoretically available resource can be practically unusable
  - Disk space, memory fragmentation…

Be user friendly
- Do not start process in case of lack of resource
- Stop cleanly in case of unexpected missing resource
- Regularly inform about process status
Evaluation on the Large Scale Learning Challenge
Large Scale Learning Challenge

- **Objective:** comparison of learning methods given limited resource

- **Ten datasets**
  - Artificial and real data
  - From hundreds to thousands of variables
  - From hundreds to millions of instances

- **Evaluation**
  - Accuracy criterion: area over the precision recall curve (aoPRC)
  - Accuracy given dataset size
  - Accuracy given training time (excluding loading time)

- **Overall ranking**
  - Complex scalar combination of different performance indicators
Our submission

- **Motivation:** evaluate the scalability of our method
  - Performance of a parameter-free method
  - RAM constraint: 2 Go RAM on a PC, Windows, single processor

- **Three submissions**
  - First one: raw data, no preparation
  - Second one: centered reduced rows for image datasets (OCR and Face)
  - Last one: best of first and second submissions
# Overall challenge ranking (03/07/2008)

<table>
<thead>
<tr>
<th>Rank</th>
<th>Score</th>
<th>Submitter</th>
<th>Title</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3.8</td>
<td>chap - Olivier Chapelle</td>
<td>Newton SVM</td>
<td>02.05.2008</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>antoine - Antoine Bordes</td>
<td>SgdQn</td>
<td>16.06.2008</td>
</tr>
<tr>
<td>3</td>
<td>5</td>
<td>yahoo - Olivier Keerthi</td>
<td>SDM SVM L2</td>
<td>19.06.2008</td>
</tr>
<tr>
<td>4</td>
<td>5.4</td>
<td>yahoo - Olivier Keerthi</td>
<td>SDM SVM L1</td>
<td>18.06.2008</td>
</tr>
<tr>
<td>5</td>
<td>7</td>
<td>kristian - Kristian Woodsend</td>
<td>Interior Point SVM</td>
<td>16.06.2008</td>
</tr>
<tr>
<td>6</td>
<td>7.1</td>
<td>kristian - Kristian Woodsend</td>
<td>IPM SVM 2</td>
<td>23.06.2008</td>
</tr>
<tr>
<td>7</td>
<td>7.2</td>
<td>beaker - Gavin Cawley</td>
<td>LR</td>
<td>19.06.2008</td>
</tr>
<tr>
<td>8</td>
<td>7.4</td>
<td>MB - Marc Boulle</td>
<td>Averaging of Selective Naive Bayes Classifiers final</td>
<td>20.06.2008</td>
</tr>
<tr>
<td>9</td>
<td>8.3</td>
<td>MB - Marc Boulle</td>
<td>Averaging of Selective Naive Bayes Classifiers</td>
<td>27.05.2008</td>
</tr>
<tr>
<td>10</td>
<td>8.8</td>
<td>garcke - Jochen Garcke</td>
<td>AV SVM single</td>
<td>25.06.2008</td>
</tr>
<tr>
<td>11</td>
<td>11.1</td>
<td>garcke - Jochen Garcke</td>
<td>AV SVM</td>
<td>02.06.2008</td>
</tr>
<tr>
<td>12</td>
<td>12.1</td>
<td>rofu - Rofu yu</td>
<td>liblinear</td>
<td>25.06.2008</td>
</tr>
<tr>
<td>13</td>
<td>12.6</td>
<td>ker2 - Porter Chang</td>
<td>CTJ LSVM01</td>
<td>20.06.2008</td>
</tr>
<tr>
<td>13</td>
<td>12.6</td>
<td>rofu - Rofu yu</td>
<td>Coordinate descent dual l1 linear svm</td>
<td>10.06.2008</td>
</tr>
<tr>
<td>15</td>
<td>13.2</td>
<td>beaker - Gavin Cawley</td>
<td>ORRR</td>
<td>12.06.2008</td>
</tr>
<tr>
<td>16</td>
<td>13.6</td>
<td>antoine - Antoine Bordes</td>
<td>LaRankConverged</td>
<td>13.06.2008</td>
</tr>
<tr>
<td>17</td>
<td>14</td>
<td>beaker - Gavin Cawley</td>
<td>WRRR</td>
<td>24.06.2008</td>
</tr>
<tr>
<td>18</td>
<td>14.2</td>
<td>fravoj - Vojtech Franc</td>
<td>Random</td>
<td>18.02.2008</td>
</tr>
<tr>
<td>19</td>
<td>14.5</td>
<td>ker2 - Porter Chang</td>
<td>CTJ LSVM02</td>
<td>25.06.2008</td>
</tr>
<tr>
<td>20</td>
<td>14.7</td>
<td>beaker - Gavin Cawley</td>
<td>ORRR Ensemble</td>
<td>15.06.2008</td>
</tr>
<tr>
<td>20</td>
<td>14.7</td>
<td>fravoj - Vojtech Franc</td>
<td>ocas</td>
<td>25.06.2008</td>
</tr>
<tr>
<td>22</td>
<td>15.8</td>
<td>MB - Marc Boulle</td>
<td>Averaging of Selective Naive Bayes Classifiers CR rows</td>
<td>12.06.2008</td>
</tr>
<tr>
<td>23</td>
<td>16.1</td>
<td>beaker - Gavin Cawley</td>
<td>rr</td>
<td>06.06.2008</td>
</tr>
<tr>
<td>24</td>
<td>16.5</td>
<td>fravoj - Vojtech Franc</td>
<td>random2</td>
<td>25.06.2008</td>
</tr>
</tbody>
</table>
Training time results

- **Training time, excluding loading time**
  - Between 2 and 3 orders of magnitude slower than online methods
  - Typically: one hour wall-clock time per processed gigabyte of data

- **Scalability: super-linear \( O(N \log N) \)**
  - Dataset fits into the RAM
    - Process time \( \sim \) IO time
  - Dataset does not fit into the RAM
    - Overhead: only a factor two

---

Face dataset

- 5.5 millions instances
- 900 variables
- 50 Go file, 2 Go RAM
- Two days wall-clock time
Training time: what matters?

- Different training times
  - Theoretical time complexity
    - Interesting, necessary but not sufficient
  - Training time, excluding loading time
    - Can be dominated by loading time
  - Wall-clock time
    - Heavily depend on programmer's skill, choice of language, compiler…
    - Can be dominated by data preparation time
  - Whole process time, including data preparation and modeling
    - Conform to business constraints

- Our focus: whole process time
  - Our submission  (27/05/2008): First challenge results available on six datasets
    - Webspam, DNA, Face, OCR, Epsilon, and Zeta
Accuracy results

- Best accuracy on three datasets
  - Face
  - Gamma
  - Delta

- Second best accuracy on two datasets
  - DNA
  - Beta

- Good accuracy on four datasets
  - Webspam
  - OCR
  - Epsilon
  - Zeta

- Poor accuracy on one dataset
  - Alpha
Illustration: nonparametric evaluation of univariate conditional probabilities

Gamma

Size 100

Size 1000

Size 10000

Size 100000

Size 500000

Beta

Size 100

Size 1000

Size 10000

Size 100000

Size 500000
Importance of data representation

- For image datasets, two evaluated representations
  - Raw data format
  - Centered reduced rows

- Results: large impact on predictive performance
  - Best representation for OCR: raw data format
  - Best representation for Face: centered reduced rows

- There is room for much better results!
Conclusion
Summary

Motivation
- Most limited resource: lack of data analysts
- Main challenge: automation of data mining process

Our method: Averaging of Selective Naive Bayes Classifiers
- Exploit the Naive Bayes assumption
- Preprocessing of variables using the MODL method
- Regularization of variable selection
- Model averaging using compression weights

Algorithms
- Offline method: exploit all the train data
- Super-linear time complexity: $O(KN \log KN)$
- Efficient chunking strategy when datasets does not fit into the RAM
Conclusion and future work

- **Method results on the challenge**
  - Highly scalable, in a fully automated way
  - Slow for CPU training time only, fast for the whole learning process
  - State of the art accuracy results

- **Future work**
  - Improve the regularization scheme in variable selection
    - Account for the complexity of the univariate density estimators
  - Extends applicability of the method
    - Regression
    - Classification with many target values
References

- **Discretization**

- **Value grouping**

- **Averaging of Selective Naive Bayes classifiers**

- **Data preparation & scoring tool for supervised learning**
  Download shareware at http://perso.rd.francetelecom.fr/boulle/