Presentation to

SAS Data Mining Forum

Re-Evaluation of Computer Resource Usage in Data Mining

May 2012

Masoud Charkhabi
Senior Consultant
CIBC

Disclaimer

All values and figures in the following presentation are intended solely for the purpose of demonstration. The findings do not necessarily reflect any literal data from historical studies. Any views or opinions presented are solely those of the presenter and do not necessarily represent those of CIBC.

Everything is an Optimization Problem

"… Of course everything is an optimization problem … What matters is, what type of optimization problem it is, because basically, most optimization problems you can't solve."

-Stephen Boyd (Convex Optimization, Stanford)

It is useful to position data mining as a mathematical optimization problem:

[Value] = [Insight Benefit] - [Program Cost] - [Programmer Cost]

Insight Benefit: The benefits lie in the changes to the business as a result of any insight.

Program Cost: CPU, memory, I/O, etc. on desktop, SAS Server and Database Server (HW and SW).

Programmer Cost: Cost of modelers, developers, consultants, etc.

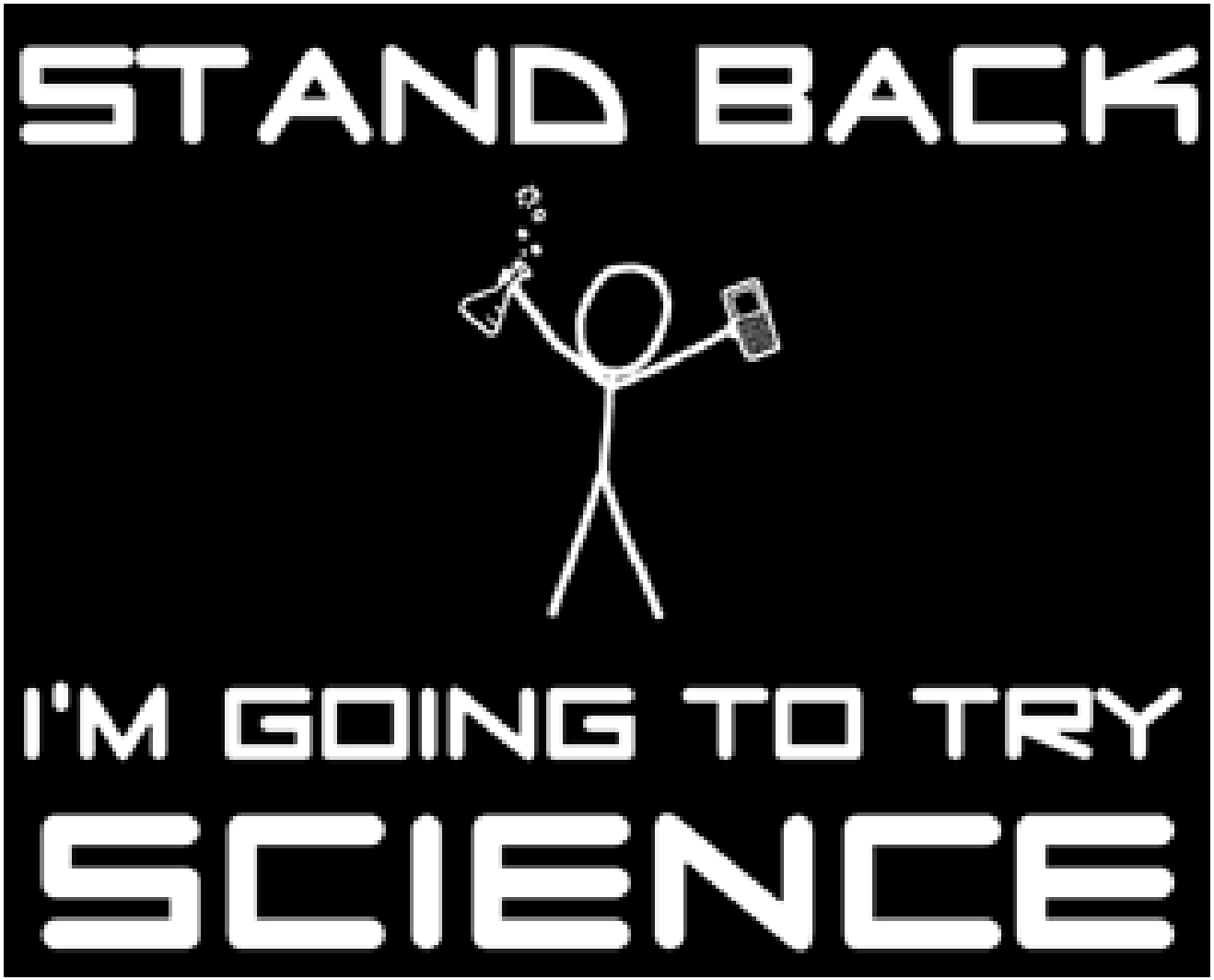
Objective: Maximize Value

Variables: Some efficiencies can be created by using less hardware resources for the same modeling or development work or by running the programs faster. However, the benefits of extracting new insight dwarf any value impact from program efficiency or programmer efficiency, given the insights are applied.

Constraints: Thy shall not speak of programmer cost with programmers!

[Insight Benefit] >> anything else

Solve By Applying Scientific Methods



Diagnostics and Debugging

Begin with separating the recommendation system from the automation system.

Andrew Ng suggests evaluating the recommendation system or learning component by:

1. Statistical Measure Evaluation -> Consistent weak performance -> High Bias -> Try adding features

2. Cross Validation -> Inconsistent performance -> High Variance -> Try adding observations

3. Evaluate the Algorithm

Evaluation of the automation system:

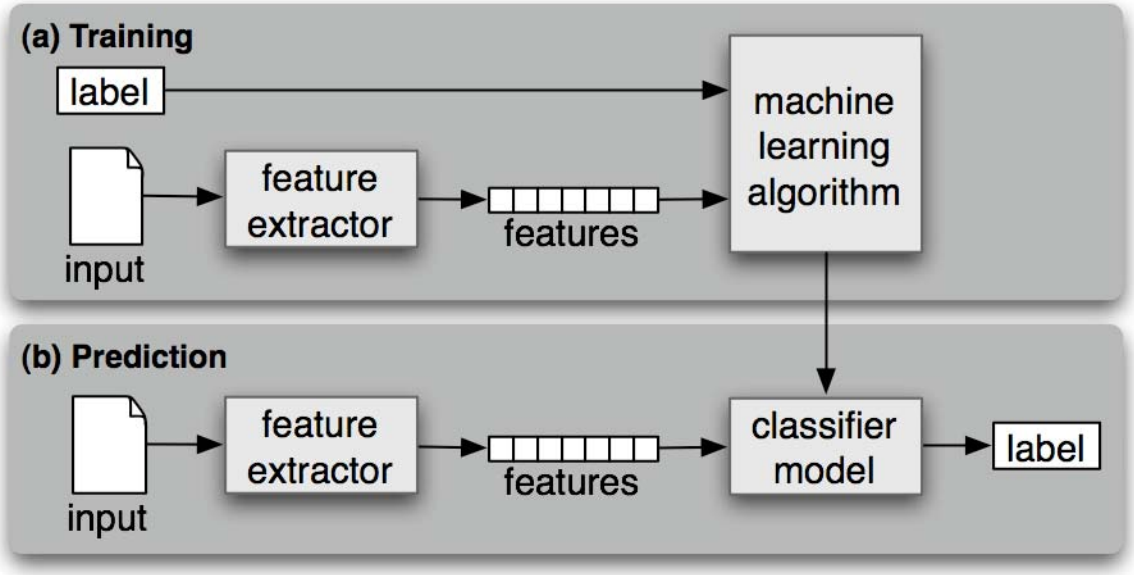
1. Use log parsing programs (Raithel, 2008) and query plans to understand the program.

2. To avoid eviction, estimate computer resource usage before you try science!

According to Amdahl's law, Symmetric Multi-Processor (SMP) servers can only accommodate a moderate increase in computer resource usage and convergence occurs quickly (Moler, 1987). With the addition of in-database, in-memory and Massively Parallel Processing (MPP) databases, the boundaries have been extended. Feature expansion tends to put more pressure on CPU and memory and observation expansion mostly on I/O (Cohen, 2002).

SMP servers may not accommodate your feature and observation expansion aspirations.

Text Classification



Source: Bird et al., Natural Language Processing with Python, 2009, Chapter 6

Application to Classification

- Supplement an attrition model with text based customer data.
- Start with an initial model with simple regular expression based features.
- Thy shall not speak of inputs to customer models with customers!

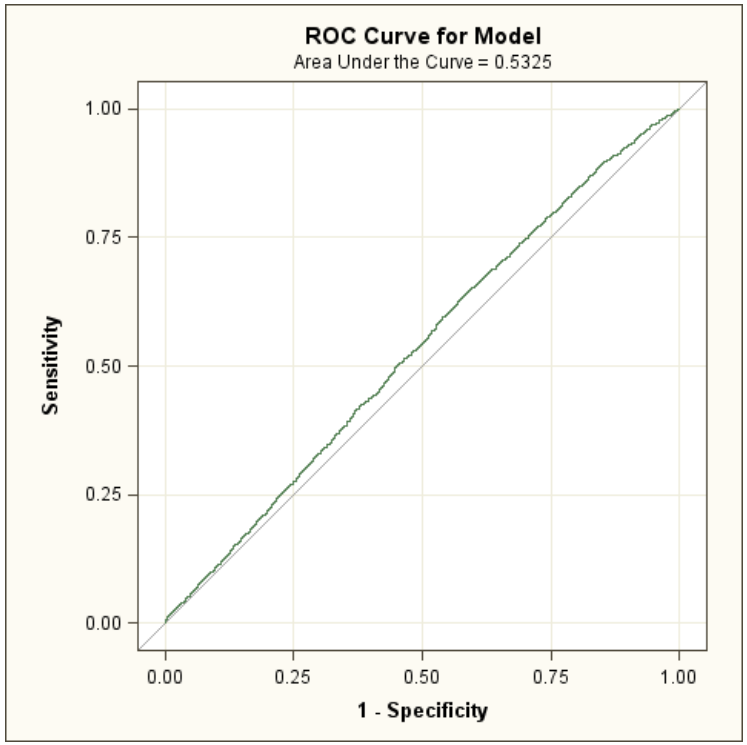
Text	RegEx1(ador.|lov.)	RegEx2(feel.*|friend.*)	Label A	Label B	Score
I adore CIBCx.	1	0	1	0	100
I have feelings for CIBCx.	0	1	1	0	80
CIBCx and I are really just friends.	0	1	0	0	60
I … CIBCx.	0	0	0	0	40
I've been seeing Scotiax.	0	0	0	1	0
Bienvenue au Canada!	0	0	0	0	

1. Find French
2. Classifier for Label A (great indicator)
3. Classifier for Label B (leave indicator)
4. Regression for Score

In many text mining applications, multiple classifiers are required.

Initial Performance

Performance with two features suggested by the requestor:



Case 1: Adding Features

High Bias may suggest that the true parameters required to estimate the problem are not present in the training set, but extracting and selecting more features adds significantly to computer resource usage, especially in text mining!

1. Use log parsing to project your feature expansion plans

2. Push feature creation and selection in-database:

   a. Add new features (use SAS macro and DO loops to write feature creating syntax)

   b. Feature selection in-database (SVD, clustering, Association in Netezza)

Unsupervised methods help expose long distance relationships in text:

CIBCx and **I** are really just **friends** over the web.

CIBCx is **nice**.

• 30,000+ words, phrases, grammar, relation and other mixed language dimensions

• After collapsing ~ 5,000 dimensions

• Before attempting to analyze the 870,000 x 5,000 dataset, project the computer resource usage.

Case 1: Adding Features Cont.

Reduce observations significantly and vary the number of features to forecast feature expansion:

Obs	Step Name	Data Prep. Time	Analysis Time	Observations	Features	Memory Used
1	IML	0.15	2.46	500	500	169022
2	IML	0.2	9.95	500	1000	169022
3	IML	0.21	49.13	500	2000	169022
4	IML	0.26	204.52	500	4000	169022

The feature expansion process comes at high multiple computational cost.

Push the feature creation and selection process in-database.

Case 1: Adding Features Cont.

```
proc iml; /* make larger matrices incrementally, this is much easier than adding column names */
use temp05; /* reduced obs */
read all var _num_ into A;
close temp05;
B500 = A[1:500,3:503]; /* 500 obs and 500 dims, start from 3 to exclude ID and target var */
B1000 = A[1:500,3:1003];
B2000 = A[1:500,3:2003];
B4000 = A[1:500,3:4003];
create B500 from B500;
append from B500; create B1000 from B1000; /* create tables */
append from B1000; create B2000 from B2000;
append from B2000; create B4000 from B4000;
append from B4000;
proc iml; /* repeat this for larger matrices */
use B500;
read all var _num_ into B500;
close B500;
call svd(U,Q,V,B500);
create U500 from U; append from U;
create Q500 from Q; append from Q;
create V500 from V; append from V;
```

Case 2: Adding Observations

High variance suggests the learning method may benefit from reviewing more observations.

1. Cross Validation for variance testing requires the building of multiple datasets and the scoring of multiple datasets. SAS Scoring Accelerator processes in-database and reduces run time to speed-of-thought.

2. In the case of text classification, more labeled data for training is usually not present so adding observations is not that simple. In this case boot strapping can help to make more training data. Again, boot strapping means an iterative build and score process and can become far more efficient with either SAS Scoring Accelerator or using vendor specific analytic packages in-database.

3. For classical data mining there are always more observations in the DBMS.

Use in-database to find more features.

Train a base model in-database only to evaluate the impact of additional observations.

Case 3: Evaluate the Algorithm

1. SAS High Performance Analytics allows the modeler to attempt computationally intensive methods through SAS HP nodes. Unfortunately, you need Teradata or Greenplum.

2. Alternatively, one can use the packages provided by the database vendors only to evaluate the impact of using an alternative learning algorithm on the same data that would otherwise not be technically feasible. The Netezza Analytics package offers native Decision Tree, Naïve Bayes, etc. classifiers that can be used in-database (and there's always R).

Understand the mechanics of your algorithm.

Data is the new oil.

We would all rather keep the S in SAS. So prepare fuel in-database and produce energy in SAS. Once the right features are determined and a solid set of training and testing data sets are built then more time can be allocated towards examining the learning function.

Poll: What is the most accurate data prep. to analysis ratio (Refining/Releasing) for a typical analytics project?

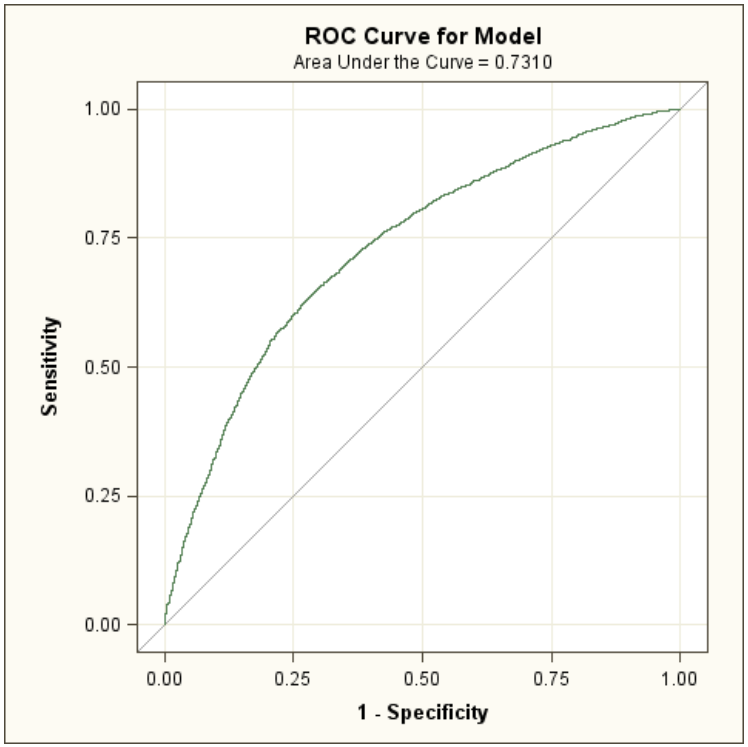
1) ¼ 2) 1 **3) 4** (hint hint …) 4) Other

Software Utilization

$ sas sas1.sas & sas2.sas & sas3.sas	Initial build in SAS
$ sed 's/"//g' sas_output.csv > temp1.csv	Clean SAS output with UNIX tools
$ python py_1.py	Create a Python program for feature extraction
$ ./clean_python.sh	Remove problematic symbols
$ sas sas4.sas & sas5.sas & sas6.sas	Loaded language (Netezza & Oracle pass-through)
$ sas sas7.sas	Train model in SAS

Fully utilize software resources: SAS, UNIX, Python, Netezza SQL & analytics package (NZA), etc.

Improved Performance



Obviously, this performance lift compared to the two feature model is irregular. The focus however, is on the process of increasing lift while carefully consuming computer resources.

The Catch: No Risk No Reward

Maximize Value by Re-Evaluating Computer Resource Usage

By using more hardware intelligently and leveraging more languages and functions we increase programmer cost slightly, reduced programming time drastically but most importantly we add insight to the business.

And yes ETL and reports ran faster too!

The approach to focus computer resources on extracting insight from information resources can be seen as a form of re-evaluation of computer resource usage in data mining.

Of course success is not guaranteed. You may process your universe of data with strong computers and a full set of functions only to conclude that improvements are negligible, but you will keep programmers employed longer (maximize programmer value!).

References

Albright, Russell. Taming Text with the SVD. SAS Institute. 2004. Web.

Bird, Steven, Ewan Klein, and Edward Loper. Natural Language Processing with Python. O'Reilly Media Inc., 2009. Web. <nltk.org/book>.

CEBR (Center for Economics and Business Research) “Data Equity, Unlocking the Value of Big Data." Center For Economics And Business Research. CEBR, Apr. 2012. Web.

Cohen, Robert A. SAS Meets Big Iron: High Performance Computing in SAS Analytic Procedures. SAS Institute. NESUG 15. 2002. Web.

Jurafsky, Daniel, and James Martin. Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition. Prentice Hall, 2000. Web. <http://www.cs.colorado.edu/~martin/slp.html>.

Moler, C. A Closer Look at Amdahl’s Law,” Intel Technical Report, TN-02-687. 1987.

Raithel, Michael, Westat. "Programmatically Measure SAS® Application Performance on Any Computer Platform with the New LOGPARSE SAS Macro." SUGI 219.30 (2007): Print.

TDWI (The Data Warehousing Institute, 2011). Big Data Analytics. Seven Keys to High-Performance Data Management for Advanced Analytics. Sep 2011. Web. <tdwi.org/research>