Big Data Analytics – Many Variables, Many Observations, and the SVM

A. Big Data Analytics vs. Classical Data Science

Prof. Dr. Claus Weihs
TU Dortmund

Colchester, 07.07.2014
Big Data Analytics vs. Data Science

What is Big Data?

- Depends on whom you ask!
- Machine Learning (ML) Benchmarks in the 90s (e.g. UCI): hundreds to thousands of data points
- Modern benchmarks: often $> 10^6$ data points
- Google & Co.: (much to) big for your computer storage, only streaming possible from a cloud, distributed analytics, ...
- **Definition.** We have a “**Big Data**” problem, if it is impossible by computer time reasons to exactly solve the learning problem.
- → Information in the data is not optimally utilizable.
- **Question:** Which information brings us as fast as possible as near as possible to the solution?
- **Aim:** „Perfect“ approximation algorithm
- **Project:** Support-Vector-Machines for extremely big data sets: Glasmachers und Weihs, 2013-2015, funded by Mercator Research Center Ruhr: 2nd part of this talk
Big Data Analytics vs. Data Science

What is Data Science?

- Study of **generalizable extraction of knowledge from data**
- Data Science builds on **techniques and theories from many fields**, including mathematics, machine learning, statistics, visualization, data warehousing, and high performance computing.
- Data Science is **not restricted to big data**, although methods scaling up with data get more and more important.
- “Classical” Data Science methods in Data Analytics are **Classification and Regression** together with Dimension Reduction, Variables Selection, and Hyperparameter Tuning.
- **NEW:** European Association for Data Science in foundation
Big Data needs special methods

We will demonstrate this by means of

Classification and Regression methods

We will particularly discuss the cases of

I. Many Variables

II. Many Observations
I. Many Variables

- In many actual classification problems: Dimension $p$ of influential factor vector $>>$ size $n$ of available training data set
- Most often, only a small part of the $p$ factors or a small number of directions important for classification.
- Classical analysis methods adequate for $p < n$ (not too big), but problematic
  - in high dimensions \textit{(curse of dimensionality)} and
  - for very large $n$
I. Many Variables

**Classification: Projection based**  (Bickel, Levina, 2004; Fan et al., 2011)

- **Example: Fisher discrimination** between 2 classes with Gauss distributions
  - **Assumption:** \( \pi_1 = \pi_2 = 0.5 \) and \( n_1 = n_2 \)
  - **Classification function:** \( \delta_F(x) = (x - \mu)^T \Sigma^{-1}(\mu_1 - \mu_2) = (x - \mu)^Ta_{opt} \) with \( a_{opt} \) linear combination of means of all variables and \( \mu = (\mu_1 + \mu_2)/2 \)
  - **Sample version:** Assign class 1, iff \( d_F(x) > \log(\pi_2 / \pi_1) = 0 \).
  - If \( p > n \) inverse of estimated covariance matrix \( S \) does not exist!
  - Moore-Penrose generalized inverse used instead!
- **Property:** If \( p / n \to \infty \), then \( \text{err} \to 0.5 \) (= random!) **Generalization:** Analogous results for all methods using projections on linear combinations
- Projection directions constructed with probability 1 using essentially all variables
- \( \to \) misclassification error big because of error accumulation.
I. Many Variables

Classification: Projections by principal components analysis

Example: Effect of high dimensions

- 2 classes in 4500-D space
- Only 90 of the 4500 variables influence the class in data set
- Idea: Use only those \( m \) variables for dimension reduction with the highest effect on class separation (\( \rightarrow \) variables selection)

- Figure (a)-(c): Projections on first 2 principal components with \( m = 2, 100, 500 \)
- Figure (d): Result of a projection on 2 random directions

\( \rightarrow \) Usage of noise variables disturb class separation
Classification Example for the effect of high dimensions on projections
I. Many Variables

Classification: Distance based (Hall et al., 2008; Fan et al., 2011)

• Are there distance based classifiers better suited in high dimensions?

• Example: kNN-Method (e.g. with Euclidean distance)
  • Any plausible distance-based classifier \( g \) has 2 properties:
    (a) \( g \) assigns \( X \) to class 1 if it is closer to each of the \( X_i \)'s in class 1 than it is to any of the \( X_j \)'s in class 2.
    (b) If \( g \) assigns \( X \) to class 1 then \( X \) is closer to at least one of the \( X_i \)'s in class 1 than to the most distant \( X_j \)'s in class 2.

• Consider the model \( X_{ij} = \mu_{kj} + \varepsilon_{ij} \), for \( i \in G_k, k = 1, 2 \), where \( X_{ij} \) denotes the \( j \)-th component of \( X_i \), \( \mu_{kj} \) represents the \( j \)-th component of mean vector \( \mu_k \), and \( \varepsilon_{ij} \)'s are independent and identically distributed with mean 0 and finite 4th moment.

• Property: Probability(distance based classifier classifies new observation correctly) \( \to 1 \), iff \( p = o(|| \mu_2 - \mu_1 ||^4) \) for \( p \to \infty \).

• For distance based classifiers perfect class prediction possible, but only if distance of class means grows very quickly with no.(influential factors)

• Result independent of sample size \( n \).

• All distance based classifiers are questionable in high dimensions.
I. Many Variables

Classification: Variables selection

Simple Filters as fast alternatives

- **Filter methods**: Construction of numerical scores $s_i$ for description of the influence of feature $i$ on the dependent class variable.
- Filters are generally independent of classification models.
- **Example Filters**
  - $\chi^2$-statistics for the evaluation of independence between (discretized) feature $i$ and the class variable
  - P-value of a *t*-test, indicating whether the distribution of feature $i$ is different for 2 classes.
  - **Mutual Information** in feature $i$ and the class variable

- **Combination of a filter with a model**
  - Calculate filter values (scores).
  - Sort features according to scores.
  - Choose the best $k$ features: **But how big is $k$?**
  - Train the model on these $k$ best features.

- **Extension: MRMR** (Minimum Redundancy – Maximum Relevance feature selection): Penalize a feature's relevancy by its redundancy in the presence of the other selected features
I. Many Variables


- **Estimation** of unknown model coefficients in linear model:
  \[ Y = X\beta + \varepsilon . \]

- **Least Absolute Shrinkage and Selection Operator (LASSO):**
  \[ b(c) = \text{argmin}_\beta (||Y - X\beta||^2_2/n + c||\beta||_1), \quad c > 0 \text{ penalty term.} \]

- **Alternative:** Ridge-Regression:
  \[ b(c) = \text{argmin}_\beta (||Y - X\beta||^2_2/n + c||\beta||_2^2), \quad c > 0 \text{ penalty term.} \]

- **Properties:** In both cases \( ||Y - X\beta||^2_2/n \) is minimized,
  with LASSO under the side condition \( ||\beta||_1 \leq R(c) \) and
  with Ridge Regression for \( ||\beta||_2 \leq R(c) \).

- LASSO leads to very different results for variables selection.

- **Properties of LASSO** (➔ name of estimator)
  - 1-norm penalty (Least Absolute),
  - \( b(c) \) can be seen as Shrinked Least-Squares estimator, and
  - Variables selection realized in the sense that \( b(c)_j = 0 \) for certain \( j \).
I. Many Variables

Regression: Loss function based

Comparison LASSO - Ridge-Regression: LASSO encloses variables selection.

- Usage of the LASSO for **variables selection** is often much more successful than classical forward selection
- **However**: LASSO is time consuming  → parallel version, utilization of sparseness
II. Many Observations: Streaming / Online Methods

- Data arrive in batches and cannot be stored together
- Analysis of data which change incrementally or dynamically over time
- **Existing computation methods** often not practical or unusable
- **Alternative**: One-pass methods using only little storage for “sufficient statistics” (aggregation)

**Example**: Variance computation
- **Standard** (2 pass): \( S_n = \sum_{j=1}^{n} (x_j - \bar{x})^2 \) vs.
- **Youngs and Cramer**: \( T_j = T_{j-1} + x_j, \quad S_j = S_{j-1} + (jx_j - T_j)^2 / (j(j-1)), \quad T_1 = x_1, \quad S_1 = 0 \)

**Example**: Incremental Linear Discriminant analysis
- Introduction of new classes
- Calculation of \( (S_b, S_w, \bar{x})_j \) from \( (S_b, S_w, \bar{x})_{j-1} \) or in blocks

**Example**: Control Charts
- Calculation of Center Line and Control Limits after new block from corresponding values from former blocks (Mejri et al., 2014)
II. Many Observations: Observations selection (Meyer et al., 2013)

Example: Cascade SVM: Partition of data set (bagging),
Identification of important observations

1. Partition the data into $k$ disjoint subsets of possibly the same size (bagging)
2. Train SVM independently on each subset
3. Generate new data sets by combination of the support vectors (SVs) of pairs (or triplets, …) of such analyses
4. Multiple repetition of steps 2 and 3
5. Train an SVM on all SVs in the last group → SVM model

Example:

Analogue for other methods:
- Instead of SVs with SVM use, e.g.,
- Distance to decision border
Results

Examples: 67000 to 581000 observations

Methods:
SVM: linear standard full SVM
B-9: bagged SVM, 9 bags
SWB-9: stepwise B-9
C-9-9: 9-1 SVMs
C-9-3: 9-3-1 SVMs
C-27-27: 27-1 SVMs
C-27-3: 27-9-3-1 SVMs

Note: Bagging = 1st step of Cascade + Voting

Experiment: ¾-¼ partition, 10 repetitions

Cascade

- Sometimes much better than Bagging
- Much faster and only a little worse than full SVM
II. Many Observations: Observations selection

Example: SVM with subsampling (thanks to Daniel Horn)

Method

− Choose parameter \textit{subsamplingRate} controlling the relative number of observations used for training.
− Randomly subsample the dataset with this rate before kernel SVM.

First Results

− Optimize penalty parameter \textit{cost}, kernel-width $\gamma$, duality gap $\varepsilon$, and \textit{subsamplingRate} with respect to misclassification error and required training time
− Randomly split dataset once: 50% training, 25% test, 25% validation
− Analyse the trade-off between the two measures by means of model-based multi-criteria optimization to compute the \textit{Pareto-front}
− We received very promising results on test sets for several large datasets.

Work in progress: Comparison of new method with more complex full-sample approximate SVM solvers to find „the best“ approximate solver.
The paretofront of subsampling libSVM on the dataset mnist

- 2.5% error
- 93% subsampling

mnist:
- 70,000 obs.
- 780 features

The paretofront of subsampling libSVM on the dataset vehicle

- 13.7% error
- 59% subsampling

vehicle:
- 98,528 obs.
- 100 features
Big Data Analytics vs. Classical Data Science

Summary

Differences to classical data analysis methods

- **Many Variables**: projections, distances, losses, variables selection
- **Many Observations**: streaming, observations selection
Big Data Analytics vs. Classical Data Science

Literature

- Bickel, P. J. and Levina, E. (2004). Some theory for Fisher's linear discriminant function, “naive Bayes”, and some alternatives when there are many more variables than observations; Bernoulli 10, 989-1010
- Tibshirani, R. (1996): Regression analysis and selection via the LASSO, JRSS B 58, 267-288