Dealing with high-dimensionality in large data sets

Part 1: Foundations and Basics

A QuantUniversity Whitepaper

By

Sri Krishnamurthy, CFA
sri@quantuniversity.com

Introduction:

Many financial data sets are characterized by large number of dimensions. High-dimensional datasets increases the complexity of analysis and requires sophisticated techniques to process these datasets. Whether it is stock data for individual companies or economic data used for macro-economic modeling, high-dimensional data sets present unique challenges. When building predictive models, quants typically have to deploy statistical methods to reduce data complexity and the number of dimensions to make it easier and tractable for processing. Traditional techniques involve choosing important dimensions (Variable Selection methods where a subset of dimensions is chosen) or reducing dimensions (where variables are transformed to a smaller set of new variables) to make analysis feasible and practical. However, these traditional techniques are seeing limits when dealing with today's data sets. Technological innovations in data collection and processing in the last decade has made access to large volumes of data possible. In addition, the data collected has high granularity, frequency and complexity increasing the need to adopt sophisticated data handling techniques. Collectively quants are seeing the 4 'V's of Big data, Volume, Velocity, Variety and Veracity manifest in financial datasets requiring rethinking on approaches to process these datasets(See our prior article from March 2014 for more on this topic). In order to appreciate the nature of the problem high-dimensional data sets, we need to understand both traditional and modern techniques.

In this two-part article, we will cover both traditional and modern techniques to address high-dimensional data sets. In part1 of this article, we will lay the foundation by discussing some of the common traditional techniques to handle high-dimensional data. We begin by discussing the problems in high-dimensional data sets including the famous "Curse of Dimensionality" problem. We then discuss two methods to deal with high-dimensional datasets. The goal of the first method is to reduce the number of variables by variable selection and that of second is to reduce the number of variables by deriving new variables. We will illustrate these methods through sample techniques (regression, decision trees and principal component analysis) and give pointers on implementing these techniques in MATLAB. We will also include sample applications of these techniques in finance and economics as a part of this discussion. In part 2 of this article, we discuss some of the challenges dealing with high-dimensionality in the context of Big Data problems and methodologies and innovations to process these large data sets. We will review some of the proposed methodologies and provide guidance on choosing approaches to handle large high-dimensional data sets.

Curse of Dimensionality and other problems in high-dimensional data sets:

In a seminal lecture in 2000, Donoho[1] discussed the challenges of dealing with high-dimensional data sets. Here, Donoho discussed the "Curse of Dimensionality" problem (coined by Richard Bellman) in the context problems in multivariate data analysis when dealing with high-dimensional data sets. Let's discuss the significance through a simple example. Let's assume there are nine data samples and there are two dimensions for each of these samples. It can be visualized in a 3X3 grid as shown in Figure 1. If there are three dimensions, then we can visualize these nine points in a 3X3X3 grid as shown in Figure 2. Notice that the points are much sparser in the 3-D plot. In order to build a model that ensures statistically sound results, we need to increase the number of samples significantly. In fact, as we keep increasing the dimensions, the number of samples needed would be exponential.

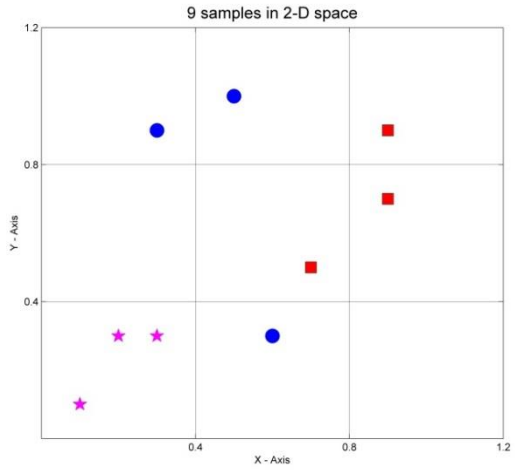

Figure 1: Samples in 2-D space

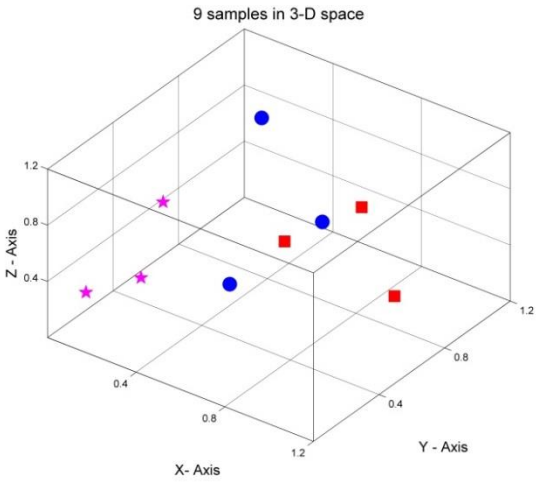

Figure 2: The same samples in 3-D space

Collecting large amounts of data is expensive and processing them brings both computational, model design and implementation challenges. Fitting models that factor the entire variable space can be an expensive computational exercise when using algorithms that span the variable space in search of optimal solutions. Sometimes, data availability can be an issue. Data may not be available or may be noisy or missing in pockets restricting modeling choices to build statistically sound models. Even if data is available, variables may be irrelevant and may not add significant value requiring the modeler to exclude such variables when building models. In other cases, there may be variables which are dependent and exhibit correlation. When building models such as regression models, multicollinearity becomes an issue that makes the coefficient estimates subject to huge changes. When building predictive models, the goal of a modeler should be to build a parsimonious model that incorporates the fewest possible variables, has been tested for predictive power and is generic enough to be deployed for new data sets. Building predictive models that incorporate large number of variables lead to statistically unstable models which may not have predictive power since they overfit to samples and aren't generic enough for prediction. This becomes an issue for applications such as forecasting. In techniques such as regression, adding additional variables could increase the R-squared value indicating better fit but such models typically don't add predictive power when tested with new data.

Now that we understand the importance of reducing variables in large data sets, the question is how do we reduce the number of dimensions? A naïve approach would be to manually choose variables for a model. This may work when working with few variables and when variables are chosen by domain experts but may not be optimal when dealing with very large number of variables. In this approach, the modeler has to make assumptions about the importance of variables that may or may not be valid. In addition, the model may not have variables that add to the predictive power of a model. On the other end of the spectrum, a kitchen sink approach is to do an exhaustive search of all possible combinations of variables and choose the best model. This is a computationally intensive exercise since testing two variables would mean trying three models (Example: If and b are predictor variables that are used to model a target response variable, we can build three models Y = f(a,b), Y = f(a), Y = f(b)). But testing ten variables means testing more than a thousand possible models which becomes impractical very soon. Let us consider some techniques that are commonly applied to handle high-dimensional data sets.

Techniques for handling high-dimensionality in finance and sample applications

Two methods are predominantly used when dealing with high-dimension datasets.

- **Variable Selection** involves using techniques to select the best features that add to the predictive power of the model.
- **Variable Reduction** involves generating new sets of variables that are derived from the original variable set.

Variable Selection

The goal of variable selection is to select features from the universe of variables based on a specific criterion. Typically, these methods are iterative and computationally intensive since these methods searches the best subset of predictors. Variable selection is preferred when the final model needs to preserve the original variables to understand the contributions of each variable to the model. Criterion based methods use metrics such as Akaike Information Criterion (AIC) and the Bayes Information Criterion (BIC), Adjusted R-squared, Mallows C_p to compare and evaluate models. For classification, misclassification rates are typically used. We will illustrate variable selection using subset section methods in regression and variable importance measures for decision trees in this section and touch upon using criterion based methods with regression and decision trees.

Regression

When building regression models, subset selection methods provide an effective way to retain the influential variables in a model. Various software packages provide options to implement feature selection. Typically, there are four possibilities:

- **Exhaustive search:** As previously explained, this method evaluates all possible combinations of variables and chooses the best model based on the chosen criterion.
- **Forward selection:** Here the model adds one predictor at a time and continues till adding another predictor is no longer statistically significant

- **Backward selection:** It is the opposite of Forward selection and all variables are included in the model to start with and variables are dropped one at a time till only the statistically significant variables remain.
- **Stepwise regression:** It combines both Forward and Backward eliminations and drops/adds variables based on their statistical significance.

It is to be noted exhaustive searches give the best/optimal model but is computationally intensive. Forward, backward and stepwise regressions optimize on computational intensity but can miss the best possible model. MATLAB's implementation of these methods and examples on using criterion based methods for regression can be found at [2].

Regression has many applications in finance from macro-economic modeling to understanding which factors are best predictors of stock returns. When dealing with large sets of variables, variable selection methods are key to get sound results. For example, Northfield's macroeconomic equity risk model [3] uses 5000 securities and the security return is explained by 12 factors whose exposures are inferred through step-wise regression. Jank[4] demonstrates how Stepwise regression can be used to predict a company's stock price using 25 variables. An example illustrating using of stepwise regression to select a basket of securities using MATLAB is available at [5] and addition a case study is available at [6].

Decision Trees

Decision trees are binary trees predominantly used for classifying data into subgroups. It is built using splitting rules to recursively generate sub-trees at the leaf notes. The leaves are typically the classification groups of interest. In classification tasks, having trees with large number of variables complicates the decision trees and typically trees are pruned to incorporate the most significant variables that are required for the classification task. In addition, techniques such as random forests and bagging are used to improve the stability of the algorithm. Here, the variable importance measures [7] are used to estimate the importance of each predictor. The Treebagger algorithm (See [8] for implementation in MATLAB) uses bagging which is an ensemble of decision trees to build a decision tree. Here, one of the outputs is the variable importance measure that can be used for variable selection. Criterion based methods for decision trees [2] can also be used for classification using misclassification rates as the criterion.

Decision trees are used in varied applications like bankruptcy prediction and fraud detection. Cho et.al [9] uses variable selection using decision trees for bankruptcy prediction. Genuer et.al [7] discusses variable selection using random forests. An example illustrating using of the Treebagger algorithm to select a basket of securities using MATLAB is available at [8] and a case study elaborating use of decision trees for variable selection is available at [6].

Variable Reduction

Variable reduction methods create new variables that are transformed from the original variables and try to capture the predictive power of a large set of variables in a smaller set of variables. Since variable transformation is involved, the new variables don't provide the intuitive attribution of contributions of individual variables in the derived variables. Principal Component Analysis, Independent Component Analysis, Factor analysis and Singular Value Decomposition are example variable reduction methods. We will review Principal component analysis as an example and discuss some sample applications.

Principal Component Analysis (PCA)

When variables are highly correlated, PCA provides a method to transform a large set of variables into a smaller set of variables that have the predictive power of the original variable set. The new variables are a weighted linear combination of the original variables and are uncorrelated. These new variables, called principal components are ordered in a way to ensure that the highest variance is captured in the first principal component, the second component capturing the second highest variance and the subsequent components capturing the rest of the variability in decreasing order. Therefore the first few components capture most of the variability observed in the original dataset. Note that PCA works only for numeric variables. Discussion on using PCA in MATLAB is available at [10] and a case study elaborating using of PCA is available at [6].

PCA has found multiple applications in finance. Forecasting economic time series is a classic example where high-dimensionality is evident. Typically economic time series data has 100s of potential variables that could be used for forecasting. Stock and Watson [11] provide a comprehensive survey of the problems with forecasting with large number of variables and various methods including PCA to handle such problems. Fifield et.al [13] discusses how PCA can be used to identify relevant factors from the pool of macroeconomic data. Portfolio optimization is another high-dimensional problem. Here, the covariance matrix for returns of assets needs to be estimated which becomes challenging for large portfolios. As we discussed in the curse of dimensionality section, for a universe of 500 stocks, 125,500 parameters need to be estimated. Jorion[12] discusses ways of simplifying the covariance matrix estimation process including how PCA can be used to compute the covariance matrix when correlations in the asset return series are high. Tsay[14] provides a detailed example on how to implement PCA for a sample stock series. Yield curve modeling is another well-known application of PCA. Litterman and Scheinkman[15] discuss their three-factor approach(level, steepness, curvature) to explain the variation of returns in fixed-income securities using PCA. Simulations are another application area for PCA. When large number of simulations is required, running PCA helps reduce the number of factors thus reducing computational intensity. Huynh et.al [16] discusses how apply PCA and Monte Carlo (MC) and Quasi Monte Carlo (QMC) simulations to compute the VAR of a bond portfolio. Jamshidan et.al [17] discusses how they use PCA to reduce the number of factors when analyzing large multi-currency portfolios. There are many other interesting applications that have been published recently. For example, Ambrusa et.al [18] describes using PCA to reduce the number of variables when modeling interest rate risk with 13 risk-factors per currency for the Swiss Solvency Test Standard Model. With these diverse applications, we are seeing adoption of dimension reduction techniques like PCA increase and quants should consider using these techniques when dealing with high-dimensional problems.

In Summary:

High dimensionality in datasets poses modeling and processing challenges and must be dealt with to build effective statistical models. With the volume and variety of data increasing, renewed focus has been placed on addressing high-dimensionality in data sets. In this article, we provide the foundation for high-dimensional data analysis and discuss some of the problems posed by high dimensional data sets. We discussed some of the traditional techniques used in quantitative finance to handle datasets with large number of variables with particular focus on variable selection and variable reduction. We also illustrated how these techniques are implemented in practice when building models using regression, decision trees and PCA and discussed sample financial applications. In a sequel to this article, we will focus on discuss high-dimensionality in the context of Big Data problems and methodologies and innovations to process these large data sets. The Curse of dimensionality has been recognized as one of the most challenging problems in statistics. With the knowledge and tools to deal with high dimensionality, quants can effectively leverage computational power and appropriate algorithms to mine the nuggets of information hidden in large datasets.

References:

1. Donoho DL. High-dimensional data analysis: The curses and blessings of dimensionality. Aide- Memoire of a Lecture at AMS Conference on Math Challenges of the 21st Century. 2000
2. http://www.mathworks.com/help/stats/feature-selection.html
3. http://www.northinfo.com/documents/7.pdf
4. Jank, W. (2011). Business Analytics for Managers. Springer.
5. http://www.mathworks.com/machine-learning/examples.html?file=/products/demos/machine-learning/basket_selection/basket_selection.html
6. http://www.quantuniversity.com/variableReduction.html
7. Genuer, R., Poggi, J. M., & Tuleau-Malot, C. (2010). Variable selection using random forests. Pattern Recognition Letters, 31(14), 2225-2236.
8. http://www.mathworks.com/help/finance/examples/credit-rating-by-bagging-decision-trees.html
9. Cho, S., Hong, H., & Ha, B. C. (2010). A hybrid approach based on the combination of variable selection using decision trees and case-based reasoning using the Mahalanobis distance: For bankruptcy prediction. Expert Systems with Applications, 37(4), 3482-3488.
10. http://www.mathworks.com/help/stats/feature-transformation.html
11. Stock, JH.; Watson, MW. Forecasting with many predictors. In: Elliott, G.; Granger, C.; Timmermann, A., editors. Handbook of Economic Forecasting. Vol. 1. 2006. p. 515-554.Chapter 10
12. Jorion,P. Value at Risk: The New Benchmark for Managing Financial Risk, 3rd edition
13. Fifield P, S.G.M. , D.M. Power and C.D. Sinclair (2002), 'Macroeconomic Factors and Share Returns: An Analysis using Emerging Market Data' , International Journal of Finance and Economics, 7: 51-62 .
14. Tsay, R. S. (2005). Analysis of financial time series (Vol. 543). John Wiley & Sons.
15. Litterman, R. B., & Scheinkman, J. (1991). Common factors affecting bond returns. The Journal of Fixed Income, 1(1), 54-61.
16. Huynh, H. T., & Soumare, I. (2011). Stochastic simulation and applications in finance with MATLAB programs (Vol. 633). John Wiley & Sons
17. Jamshidian, F., & Zhu, Y. (1996). Scenario simulation: Theory and methodology. Finance and stochastics, 1(1), 43-67.
18. Ambrus, M., Crugnola-Humbert, J., & Schmid, M. (2011). Interest rate risk: dimension reduction in the Swiss Solvency Test. European Actuarial Journal,1(2), 159-172.

About:

QuantUniversity offers quantitative modeling and consulting services to financial institutions. QuantUniversity offers quantitative modeling and consulting services to financial institutions and specializes in analytics, optimization and big data solutions.

Sri Krishnamurthy, CFA, CAP is the founder of www.QuantUniversity.com , a data and quantitative analysis company. Sri has significant experience in designing quantitative finance applications for some of the world's largest asset management and financial companies. He teaches quantitative methods and analytics for MBA students at Babson College and is the author of the forthcoming book published by Wiley titled "Financial Application Development: A case study approach". Sri can be reached at sri@quantuniversity.com