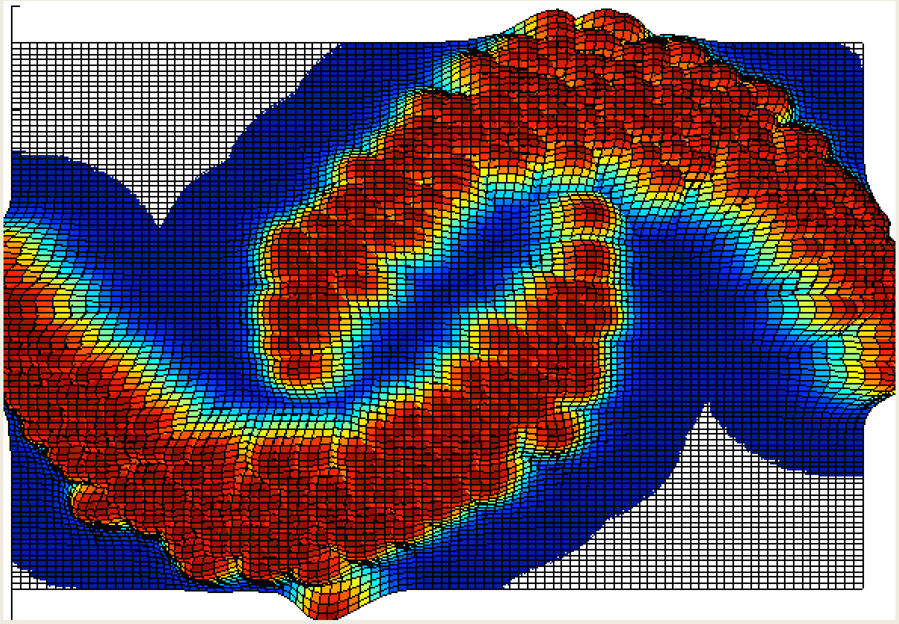
Yottamine Machine Learning Services

Cloud-Based Machine Learning

For Fast Big Data Prediction



A Yottamine Support Vector Machine solving the *Two Moons* problem

Yottamine Analytics

http://www.yottamine.com

contact@yottamine.com

Cloud-based Machine Learning for Fast Big Data Prediction

Yottamine Machine Learning Services provides data scientists and analytical programmers with a family of high performance Support Vector Machines for solving complex machine learning problems across a wide range of industries and applications. YMLS is accurate, fast and easy to use.

YMLS can handle linear or non-linear problems with small or big data in low or high dimensions with sparse or dense features. It is the only commercial SVM offering to provide such versatility.

YMLS is easy to use, offering the data scientist a clean, intuitive UI and the analytical programmer a rich API for building advanced applications using Java now and R soon.

YMLS is powerful, bringing together sophisticated machine learning techniques with highly parallel systems programming to gain unprecedented levels of both speed and accuracy. And, it works with Hadoop to make short work of preparing Big Data for machine learning.

Best of all, YMLS's powerful machine learning algorithms operate in the Amazon AWS computing cloud, a pay-as-you-go computation service available over the web. This means that there is no need to buy hardware or software for doing large scale machine learning.

YMLS includes these powerful Support Vector Machine algorithms:

Linear

A high-performance conventional linear SVM for large scale classification tasks

Gaussian

Easy to use, highly accurate non-linear SVM with automated parameter selection

Polynomial

Fast, compact non-linear SVM for predicting polynomial-ordered behaviors

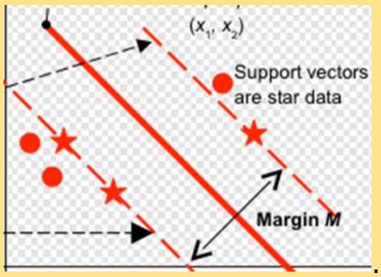
Nearest Neighbor

Highly parallel non-linear SVM for Big Data prediction on Hadoop

What is a Support Vector Machine?

An SVM is a mathematical software framework for machine learning. Using a learning algorithm, the SVM analyzes large amounts of example "training" data to build predictive models for regression and linear or non-linear classification.

The SVM gets its name from the boundaries of the margin between data in one class or another.



(1) Linear SVM

This is Yottamine's proprietary high-performance implementation of a standard linear SVM for large-scale classification tasks. It is implemented to work on large computing clusters to produce models from large volumes of high dimension training data like click path, messaging, or sensor data.

The Standard SVM includes enhancements for performance metrics like *area under the ROC curve* and for handling training data that is highly unbalanced and either very dense or sparse.

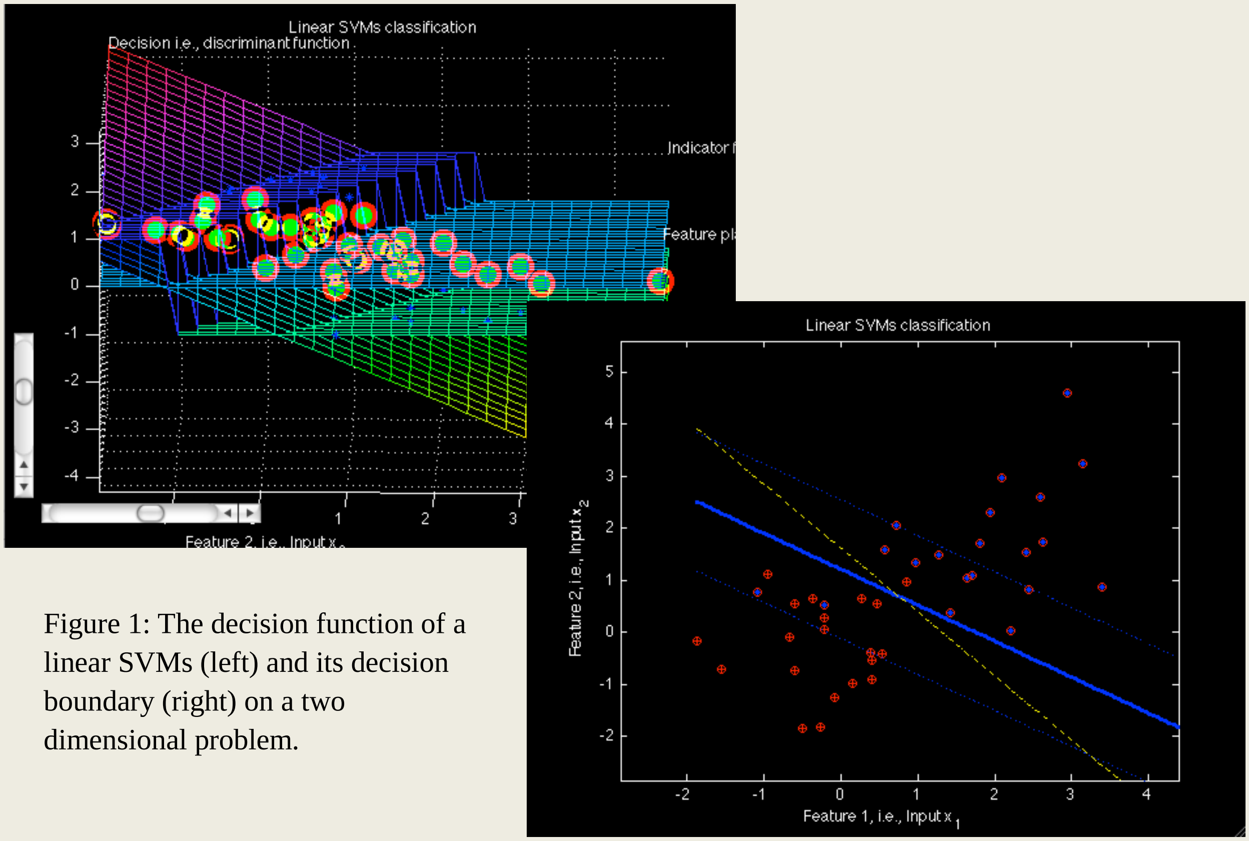


Figure 1: The decision function of a linear SVMs (left) and its decision boundary (right) on a two dimensional problem.

(2) Gaussian SVM

Gaussian SVMs and other radial basis machine learning techniques are widely recognized for their accuracy in predicting highly non-linear behavior, as in facial recognition or signal processing. But, Gaussian SVMs typically require a lot of tweaking for various training and test parameters to get accurate models, and more tweaking of the storage, networking and computing environment to get cost-effective solution performance.

Yottamine's Gaussian SVM is engineered to address those challenges by replacing guesswork on things like optimal sigma parameter range with automated calculations and recommendations done by the SVM software. In both classification and regressions, the Yottamine Gaussian SVM delivers both high accuracy and the shortest time to solution. And, it offers better out-of-the-box learning performance on highly unbalanced training data than any other Gaussian SVM solution.

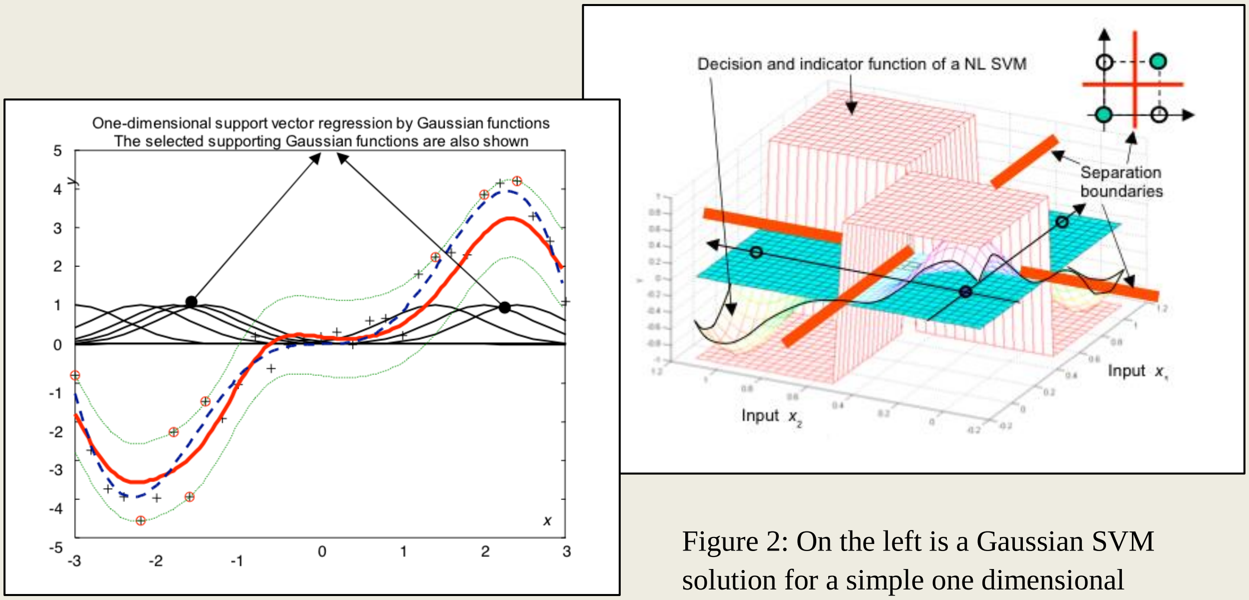


Figure 2: On the left is a Gaussian SVM solution for a simple one dimensional problem. On the right, the decision function of a Gaussian SVM for the XOR problem.

(3) Polynomial SVM

For fast, compact, and accurate modeling of non-linear behaviors that follow known polynomial orders, a polynomial SVM can provide a solution. And, for some problems, like image recognition, trading, or failure prediction, where Gaussian SVMs might also be used, polynomial SVMs can often do the job more simply, without a lot of parameter tweaking and experimentation.

The Yottamine Polynomial SVM is a high performance implementation of a standard polynomial SVM with a growing array of unique features. Like the other Yottamine SVMs, it uses parallel programming to get top speed on clustered systems and it has additional optimizations for highly unbalanced data.

Yottamine is also doing pioneering R&D in polynomial SVM algorithm development. This work includes Compact Modeling, which produces a very small, very fast run-time model and optimizations for low order polynomials. These and other features are being added to the Polynomial SVM kernel as they reach appropriate levels of performance and accuracy for production use. We welcome inquiries from organizations interested in testing these innovations on large real-world data sets.

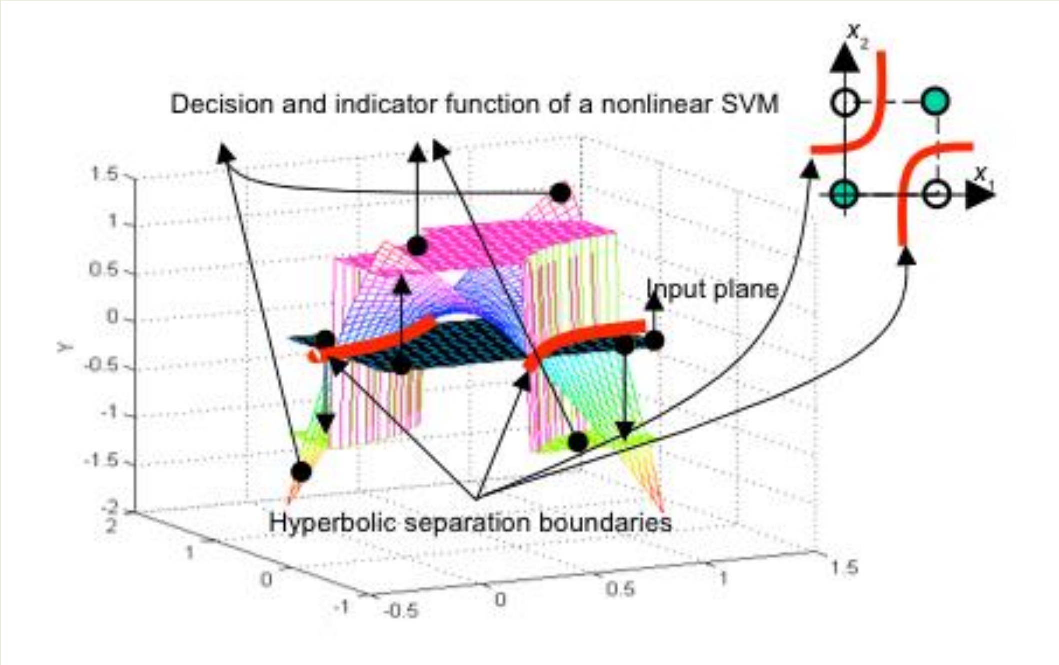


Figure 3: The decision surface of a polynomial SVMs for the XOR problem.

(4) Nearest Neighbor SVM

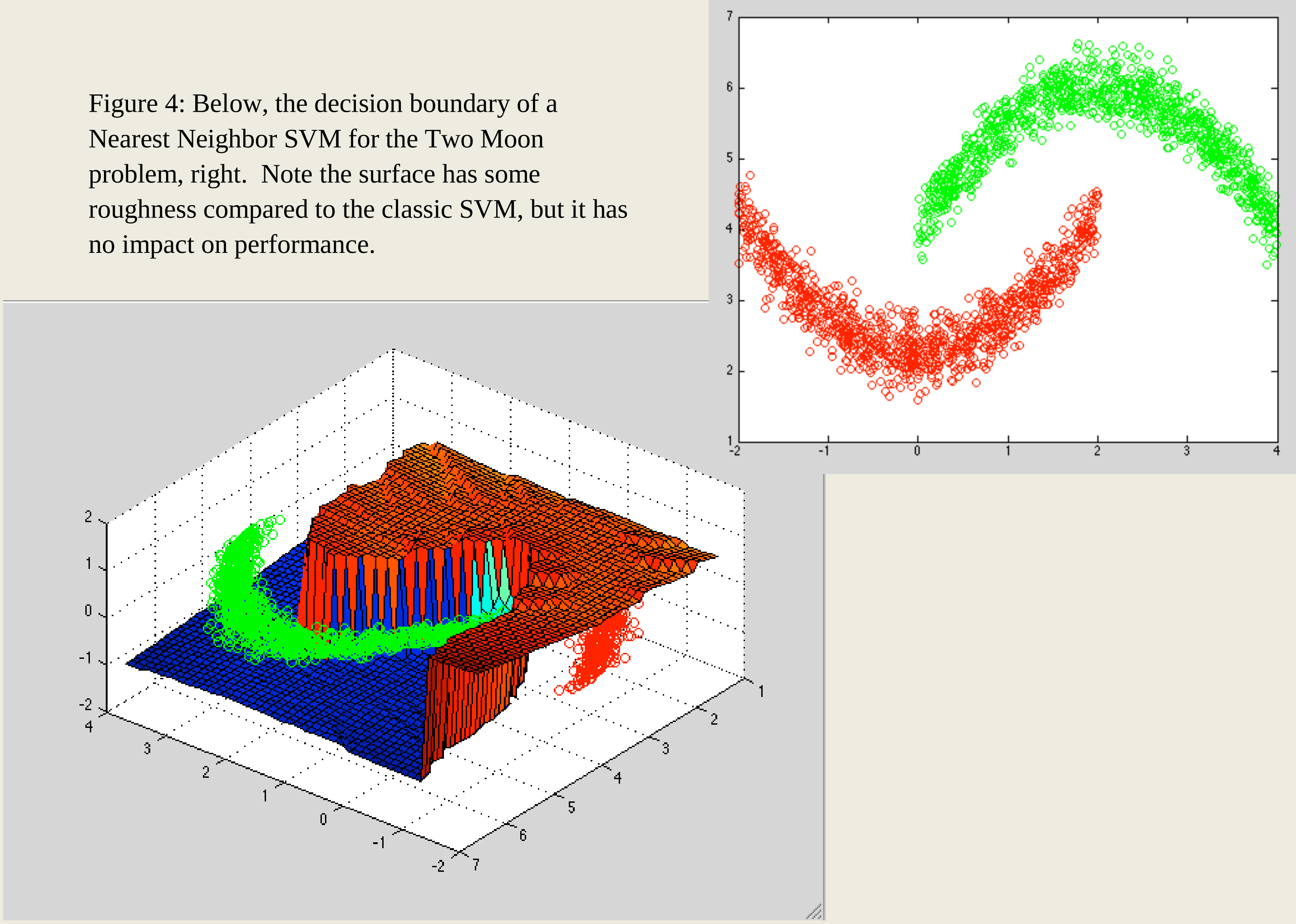
Complex non-linear behaviors can only be captured and analyzed by mapping huge amount of data into a high dimension space, as demonstrated in digital marketing, network analysis, text processing, facial recognition, and other diverse domains.

Developing highly efficient classification models from large amount of training data mapped to a high dimension space is a hard problem. Traditional techniques relying on sequential observation by a single learner become highly inefficient and impractical in high dimension space, requiring too much processing and memory.

Yottamine solves this problem in a unique way through the parallel generation of thousands or even millions of models for nearest neighbor classification on many dimensions at once. Learning and testing happen quickly on a massive scale and yield highly accurate models.

The Yottamine Nearest Neighbor SVM is based on a proprietary algorithm designed to produce the fastest, most accurate non-linear classifications on large volumes of unstructured and machine data. Built on top of Hadoop storage and operating on Amazon's AWS cloud infrastructure, this SVM can tackle the biggest Big Data for machine learning and prediction.

Figure 4: Below, the decision boundary of a Nearest Neighbor SVM for the Two Moon problem, right. Note the surface has some roughness compared to the classic SVM, but it has no impact on performance.



Machine Learning in the Cloud

Classification and regression analysis on very large amounts of training data can require a great deal of computer memory and processing power. Especially with data representing complex non-linear behaviors, as with text, speech, handwriting, face recognition, stock price prediction, and financial forecasting, the computing bill can be quite large.

Indeed, the first SVM was developed many years ago and since then technique has shown great promise, but its adoption has been slowed by the high cost of the required computing resources needed to actually use SVMs for solving routine problems. Few organizations could afford to dedicate the needed computing power to machine learning, despite its great promise.

However, the emergence in recent years of cloud computing changes everything. Infrastructure service providers like Amazon Web Services now offer access to virtually unlimited computing power on demand, in the form of clustered parallel servers that can be used for an hourly fee.

Combining the economics of cloud computing with the performance of parallel systems programming and the accuracy of sophisticated SVM algorithms, as Yottamine has done, promises to energize and democratize machine learning as never before. With the YMLS running on Amazon Web Services, even small organizations can perform analytics on a scale that was previously the province of only the world’s largest companies and well-funded institutions.

With the Yottamine YMLS, all the software for machine learning and model development runs in the Amazon cloud, so there is no need for in-house hardware or software. Customers can pay as they go and there is no long-term lock-in.

Some companies are understandably reluctant to upload sensitive private business data to a public cloud like AWS, due to security, governance, or regulatory concerns. With Yottamine, there is no need to upload business data to the cloud. The customer can extract the training data values from that business data behind the firewall and upload a matrix of encoded numerical values from which no source information can be derived or reconstructed.

The Yottamine SVMs operate on the uploaded encoded training data to produce a model that can be downloaded for use with applications back behind the firewall, or wherever the user chooses.

Balanced and Unbalanced Data

Real-world data used for many kinds of classification analysis is often highly unbalanced, where the number of items of one type is significantly different than for another. In spam filtering, fraud detection, product failure prediction, and many other applications, the goal is to predict something important that doesn't happen a lot. But, despite the prevalence of unbalanced data in the real world, machine learning researchers have often used artificially balanced data in developing new techniques and algorithms.

Why? Because, especially before SVMs, creating a classifier that gives good performance and accuracy on highly unbalanced data was very difficult and in many cases may have seemed to be more trouble than it was worth. So, balancing tricks like removing a dominant class or re-proportioning the training data have not been uncommon in many machine learning approaches, even though it means losing information, almost always an undesirable thing.

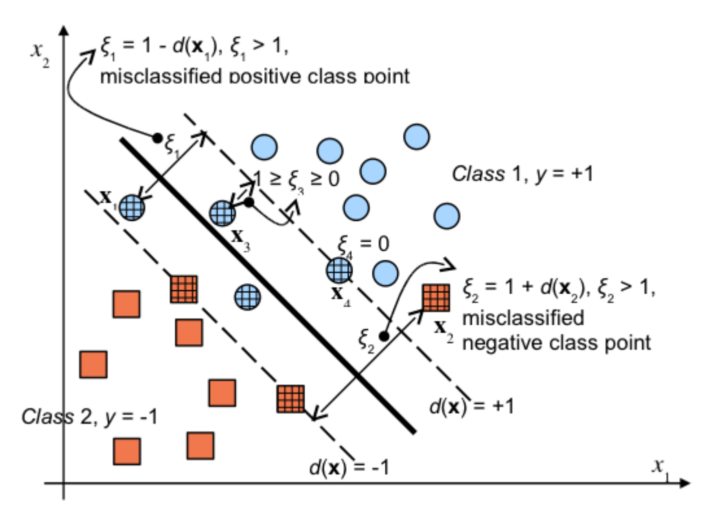
SVMs have improved the ability to use training data that is both real and realistic, because, unlike some other techniques, they “mind the gap”. That is, SVMs focus more on finding the largest margins between items in one class and items in another. They do not need to account for the size of either class in the solution and, theoretically, they therefore can be indifferent to ratio of the class sizes.

In practice though, especially with very large and highly unbalanced training data sets like web clicks, for example, conventional SVMs still often struggle to deliver good performance along with acceptably accurate results. Predicting the presence of one buyer in a million clickers is a heavy computational load.

To lighten that load, some still employ “lossy” training data and most do one other thing: they ignore the outlier values within or on the other side of the support vector margins , despite their possible importance.

Yottamine’s SVM algorithms are optimized for unbalanced data and include unique capabilities for automatically classifying marginal data correctly.

Figure 5: The Soft Margin innovation enables reliable prediction even with noisy data.



High Performance Machine Learning

Even if the software itself costs nothing, as is the case with open-source *Libsvm,* the most widely used SVM, using support vector machines has these other major costs:

1. Processing and memory for learning and testing computation
2. Data scientist hours for building and testing predictive models
3. Additional processing and analysis for optimizing parameters

These costs for calculating complex non-linear functions from Big Data training sets can be quite high. Data scientists are an even costlier resource; and developing highly accurate predictive models requires large amounts of computer memory and processing power, and lots of data scientists' time.

Cloud computing eliminates the need for dedicated high-powered in-house and hosted systems by making it possible to purchase processing power only as needed, and quite cost-effectively so. But, with SVMs that are not scalable or efficient, cloud computing charges can add up to very high cloud service fees.

Libsvm and many other SVMs use single-threaded software, meaning they cannot take advantage of multi-core processors, multi-processor servers, and multi-server clusters. They can process one program instruction at a time.

Yottamine YMLS is built on highly parallel software that can use multiple cores, processors, and servers at the same time, dramatically reducing the time to solution and solution cost.

This difference is dramatically demonstrated in a side-by-side comparison of Libsvm and Yottamine's Nearest Neighbor SVM using the Covtype data set, a popular machine learning training data set used for testing and optimizing new algorithms. The data set consists of 581,012 data points with 54 features and two classes.

Libsvm used one 3.3GHz core on one CPU and processed 0.002 parameters per minute or 1 parameter in 8 hours.

Yottamine's Nearest Neighbor SVM was able to use 40 2.93GHz cores to process 1.35 parameters per minute, or 270 parameters in 3.3 hours.

On this test, Yottamine produced a 4.31% error rate with no tuning, while Libsvm, with expert tuning and numerous trial iterations produces an error rate just below 4%.

With only a .3% accuracy penalty, Yottamine was 675 times faster and required no expert time or trial iterations. What took Yottamine 3.3 hours would take Libsvm 90 days of machine time.

Worse though, a Libsvm user won't know in advance what the best C and Sigma parameter values should be and both parameters range from 0 to infinity. So it might even take a machine learning expert 10-20 iterations or more to find the best settings.

So, it is important to note that even if Libsvm were faster, it would still require the additional scientist time for tuning to get that .3% better accuracy. The cost of data scientist talent is rising rapidly. Such expensive resources should really only be used for high-value tasks. And yet, a great deal of that precious resource is routinely wasted on trial-and-error optimizing only to produce marginally better predictive accuracy.

To do cost-effective high performance machine learning and prediction on Big Data requires three things that Yottamine YMLS provides:

- Parallel programming on multiprocessor systems
- Innovative algorithms like Yottamine's KNN SVM
- Automated parameter selection to replace guesswork

Summary

Yottamine specializes in high performance Big Data Support Vector Machine technology, marrying sophisticated mathematics with advanced software engineering to deliver a platform solution that is fast, accurate, and very easy to use.

The Yottamine YMLS implements a combination of algorithms to solve a wide assortment of machine learning and prediction problems across many different industries.

Yottamine YMLS

- Delivers great performance, even on very large unbalanced training data
- Automates parameter setting, eliminating the need for guesswork
- Highly parallel for cost-effective on-demand cloud-based computation
- Advanced API for easy development of enterprise machine learning applications
- Web-based analyst workspace UI can be accessed from anywhere
- Can use all available training data for the greatest possible accuracy
- Subscription/usage-based pricing – no hardware or software to buy
- Predictive modeling platform for data scientists deployed in weeks