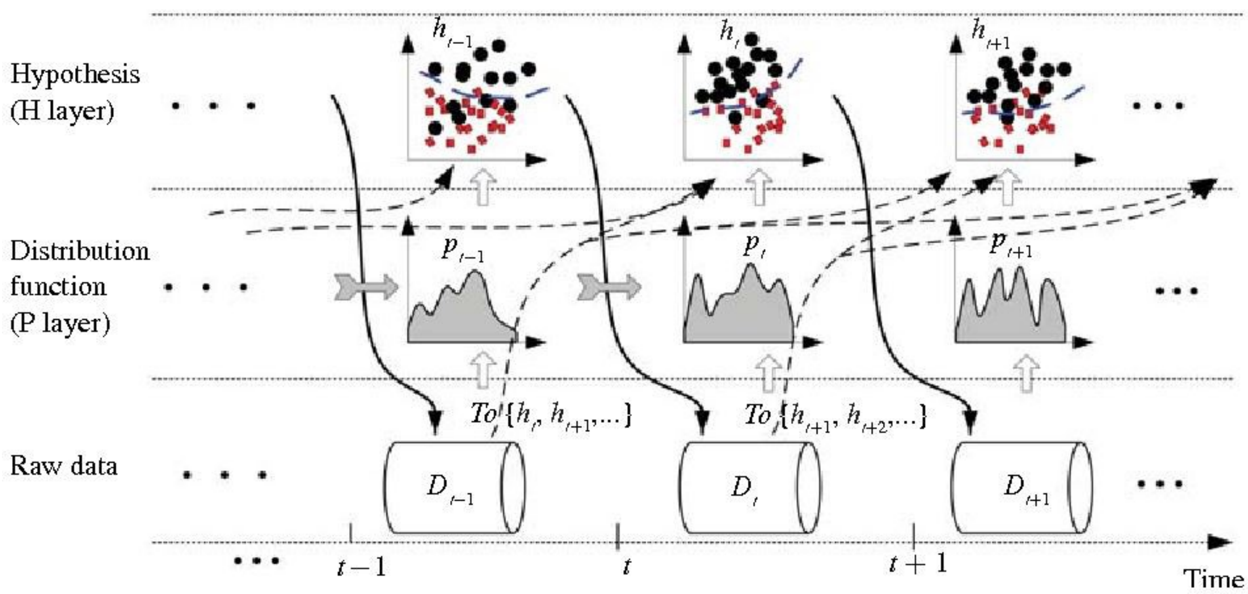
NICE ACTIMIZE

Online Machine Learning – Incremental Learning with Support Vector Machines (ISVM)

Part 3

Insights Article

Danny Butvinik
Chief Data Scientist | NICE Actimize



Support Vector Machines (SVMs) is a popular tool for learning with large amounts of high dimensional data. However, it may sometimes be preferable to learn incrementally from previous SVM results, because computing SVMs is very costly in terms of time and memory consumption or because the SVM may be used in an online learning setting. This paper presents an approach for incremental learning with Support Vector Machines that improves existing approaches. Empirical evidence is given to prove that this approach can effectively deal with changes in the target concept, which result from the incremental learning setting.

As mentioned in part one of my Online Machine Learning series, the ability to incrementally learn from batches of data is an important feature that makes a learning algorithm more applicable to real-world problems. Incremental learning may be used to keep memory and time consumption of the learning algorithm at a manageable level or because one needs to make predictions at a time when all the data are not yet available (online setting). The most important question of incremental learning is whether the target concept may change between the learning steps or is assumed to be constant. The first case is called concept drift, while the second case is true incremental learning. The practical difference between these learning is that in the concept drift setting, old examples may be misleading, as they represent an old target concept that may be quite different from the concept being learned. In the case of true incremental learning, all examples contain the same information about the target concept. Consequently, one can judge the performance of an incremental learning algorithm simply by comparing its results to the results of the learning algorithm trained all data simultaneously as the gold standard. This article will focus on incremental learning.

Support Vector Machines have been successfully used for learning with large and high dimensional data sets. This is because the generalization property of an SVM does not depend on all the training data, but only a subset thereof, the so-called “Support Vectors." Unfortunately, the training of SVMs itself can very time-consuming, especially when dealing with noisy data. As the number of Support Vectors typically is very small compared to the number of training examples, SVMs promise to be an effective tool for incremental learning by compressing the data of the previous batches to their Support Vectors. This approach to incremental learning with Support Vector Machines has been investigated and been shown to compare very well to their non-incrementally trained equivalent.

The problem of drifting concepts in incremental Support Vector Machine learning has been addressed. It's also been experimentally validated that SVMs handle drifting concepts well with respect to the criteria of stability of the result during the learning steps, improvement of the prediction accuracy during the progress of the training, and recoverability from errors resulting from the drifting concepts. Another approach to handling the drifting concepts performance estimator has been used to detect whether a drift in the underlying concept did occur, at which point the old data was being discarded, with training taking place only on the new data.

Support Vector Machines

Support Vector Machines (SVMs) are based on the work of Vladimir Vapnik in statistical learning theory. Statistical learning theory deals with how a function from a class of functions can be found that minimizes the expected risk

$$R[f] = \int \int L(y, f(x)) dP(y|x) dP(x)$$

with respect to a loss function L, when the distribution of the examples P(x) and their classifications P(y|x) is unknown and must be estimated from finitely many examples.

SVM algorithm solves this problem by minimizing the regularized risk, which is a weighted sum of the empirical risk concerning the data and a complexity term ||w||²

$$R\text{reg}[f] = R\text{emp}[f] + \lambda ||w||^2$$

In their basic formulation, SVMs find a linear decision function y = f(x) = sign(w*x+b) that minimizes the prediction error on the training set, promises the best generalization, minimizes the prediction error on the training set, and promises the best generalization performance. Given the examples (x1, y1), … ,(xn, yn) this is done by solving the following optimization problem:

$$\begin{aligned} \Psi(w, \xi, \xi^*) &= \frac{1}{2}(w^T w) + C \sum_{i=0}^{n} \xi_i \\ &\rightarrow \min \end{aligned}$$

subject to

$$\begin{aligned} y_i(w^T x_i + b) &\leq 1 - \xi_i, i = 1, \ldots, n \\ \xi_i &\geq 0, i = 1, \ldots, n \end{aligned}$$

The hyperplane vector w is represented in terms of the training examples (xi, Yi) and their lagrangian multipliers (alpha_i), which is calculated during the optimization process:

$$w = \sum_{i \in I} \alpha_i y_i x_i.$$

The optimal constant *C* for the learning problem at hand is usually determined by some model selection technique, e.g. cross-validation.

New Incremental Learning Algorithm

As stated above, the SV-incremental algorithm suffers from the problem that Support Vectors do not describe the whole set of data, but only the decision function by which it is induced. To compensate for this problem in the incremental learning algorithm, one needs to make an error on the old Support Vectors (which represent the old learning set) more costly than an error on a new example. Fortunately, this can easily be accomplished in the Support Vector algorithm. Let (xi, Yi) be the old Support Vectors and (x'i, y'i) be the new examples. Then

$$\Psi(w, \xi, \xi^*) = \frac{1}{2}(w^T w) + C\left(\sum_{i \in I} \xi_i + L \sum_{i \in S} \xi_i\right)$$

This modification of the SVM problem can be viewed as training the SVM concerning a new loss function. A natural choice for *L* is to let *L* be the number of examples in the previous batch divided by the number of Support Vectors.

This comes from the concept of approximating the average error of an arbitrary decision function for all examples by the average error for just the Support Vectors. In other words: Every Support Vector stands for a constant fraction of all examples. This algorithm will be called the SV-L-incremental algorithm.

Learn more about NICE Actimize Financial Crime Analytics research here.

ABOUT NICE ACTIMIZE

NICE Actimize is the largest and broadest provider of financial crime, risk and compliance solutions for regional and global financial institutions, as well as government regulators. Consistently ranked as number one in the space, NICE Actimize experts apply innovative technology to protect institutions and safeguard consumers and investors assets by identifying financial crime, preventing fraud and providing regulatory compliance. The company provides real-time, cross-channel fraud prevention, anti-money laundering detection, and trading surveillance solutions that address such concerns as payment fraud, cybercrime, sanctions monitoring, market abuse, customer due diligence and insider trading.

info@niceactimize.com | www.niceactimize.com/blog | @nice_actimize | /company/actimize | NICEactimize