From: AAAI-00 Proceedings. Copyright © 2000, AAAI (www.aaai.org). All rights reserved.

Incremental and Distributed Learning with Support Vector Machines

Doina Caragea, Adrian Silvescu, and Vasant Honavar
Artificial Intelligence Research Laboratory
Department of Computer Science
Iowa State University, Ames, IA 50011
{dcaragea|silvescu|honavar}@cs.iastate.edu

Due to the increase in the amount of data gathered every day in the real world problems (e.g., bioinformatics), there is a need for inductive learning algorithms that can incrementally process large amounts of data that is being accumulated over time in physically distributed, autonomous data repositories. In the incremental setting, the learner gradually refines a hypothesis (or a set of hypotheses) as new data become available. Because of the large volume of data involved, it may not be practical to store and access the entire dataset during learning. Thus, the learner does not have access to data that has been encountered at a previous time. Learning in the distributed setting can be defined in a similar fashion. An incremental or distributed learning algorithm is said to be exact if it gives the same results as those obtained by batch learning (i.e., when the entire dataset is accessible to the learning algorithm during learning). We explore exact distributed and incremental learning algorithms that are variants and extensions of the support vector machine (SVM) family of learning algorithms.

For the sake of simplicity, suppose that we have two data sets D_1 and D_2, and we want to learn from them in an incremental setting using SVM. A naive approach (Syed, Liu & Sung, 1999) works as follows:

1. Apply the SVM algorithm to D_1 and generate a set of support vectors SV_1
2. Add SV_1 to D_2 to get a data set D_2'
3. Apply the SVM algorithm to D_2' and generate a set of support vectors SV_2

One can envision a similar approach in the distributed setting. The naive approach works reasonably well in practice if the two data sets D_1 and D_2 each individually are representative of the entire training set $D_1 \cup D_2$, so that the *maximal margin* separating hyperplane determined by the support vectors derived from either one of them doesn't differ very much from that derived from the entire data set. In general, however, we can prove that the hyperplane obtained using such a naive approach can have arbitrarily high error with respect to the hyperplane obtained by applying the SVM algorithm directly to $D_1 \cup D_2$.

We have explored a more sophisticated approach to distributed and incremental learning of SVM (Caragea, Silvescu & Honavar, 2000). Let L be an inductive learning algorithm for pattern classification, which outputs hypotheses that are encoded directly in terms of training examples. SVM has this property because the maximal margin hyperplane is completely specified by a linear combination of a subset of training examples (the so-called support vectors). Given such a learning algorithm L and data sets $D_1, D_2, \cdots, D_N$, a sufficient condition for exact learning, i.e. $(L...L(L(D_1) \cup D_2)... \cup D_N) = L(D_1 \cup ... \cup D_N)$ (incremental case), is the following (*u-closure*) property: $L(L(D) \cup D') = L(D \cup D')$, for any arbitrary sets D and D'. We can state a similar property for distributed learning.

It is easy to show that the naive approach to incremental learning using SVM violate the *u-closure* property. However, the subset of the the positive (and negative) examples that form the vertices of the convex hulls of the positive (and negative) examples in the respective data sets do satisfy the *u-closure* property. So exact incremental and distributed learning algorithms can be obtained by combining the vertices of the respective two convex hulls (one for the positive examples, and another for the negative examples) and then applying SVM to generate a hyperplane that maximizes the margin of separation between the two classes. Our experiments using carefully constructed artificial data sets verify the soundness of this approach. However, since complexity of convex hull computation has a linear dependence on the number of facets of the convex hull (and the number of facets can be exponential in dimension of the space), this approach is likely to be practical only when the convex hulls are simple (i.e., have relatively few facets). We have a characterization of the necessary and sufficient subset of each of the two convex hulls that guarantee exact incremental and distributed learning. Work in progress seeks to precisely characterize hypotheses classes that lend themselves to efficient learning in exact or approximate incremental and distributed settings.

Copyright © 2000, American Association for Artificial Intelligence (www.aaai.org). All rights reserved.

References

Caragea, D., Silvescu, A., Honavar, V. (2000). *Distributed and Incremental Learning with Support Vector Machines*, Tech. Rep.ISU-CS-TR 2000-04.

Syed, N.A., Liu, H., Sung, K.K. (1999). *Incremental Learning with Support Vector Machines*, In: *KDD'99*, San Diego, CA.