Fast Query-Optimized Kernel-Machine Classification

Computation is accelerated by an order of magnitude, without loss of accuracy.

NASA's Jet Propulsion Laboratory, Pasadena, California

A recently developed algorithm performs kernel-machine classification via incremental approximate nearest support vectors. The algorithm implements support-vector machines (SVMs) at speeds 10 to 100 times those attainable by use of conventional SVM algorithms. The algorithm offers potential benefits for classification of images, recognition of speech, recognition of handwriting, and diverse other applications in which there are requirements to discern patterns in large sets of data.

SVMs constitute a subset of kernel machines (KMs), which have become popular as models for machine learning and, more specifically, for automated classification of input data on the basis of labeled training data. While similar in many ways to k-nearest-neighbors (k-NN) models and artificial neural networks (ANNs), SVMs tend to be more accurate. Using representations that scale only linearly in the numbers of training examples, while exploring nonlinear (kernelized) feature spaces that are exponentially larger than the original input dimensionality, KMs elegantly and practically overcome the classic "curse of dimensionality." However, the price that one must pay for the power of KMs is that query-time complexity scales linearly with the number of training examples, making KMs often orders of magnitude more computationally expensive than are ANNs, decision trees, and other popular machine learning alternatives.

The present algorithm treats an SVM classifier as a special form of a k-NN. The algorithm is based partly on an empirical observation that one can often achieve the same classification as that of an exact KM by using only small fraction of the nearest support vectors (SVs) of a query.

The exact KM output is a weighted sum over the kernel values between the query and the SVs. In this algorithm, the KM output is approximated with a k-NN classifier, the output of which is a weighted sum only over the kernel values involving k selected SVs. Before query time, there are gathered statistics about how misleading the output of the k-NN model can be, relative to the outputs of the exact KM for a representative set of examples, for each possible k from 1 to the total number of SVs. From these statistics, there are derived upper and lower thresholds for each step k. These thresholds identify output levels for which the particular variant of the k-NN model already leans so strongly positively or negatively that a reversal in sign is unlikely, given the weaker SV neighbors still remaining.

At query time, the partial output of each query is incrementally updated, stopping as soon as it exceeds the predetermined statistical thresholds of the current step. For an easy query, stopping can occur as early as step $k = 1$. For more difficult queries, stopping might not occur until nearly all SVs are touched. A key empirical observation is that this approach can tolerate very approximate nearest-neighbor orderings. In experiments, SVs and queries were projected to a subspace comprising the top few principal-component dimensions and neighbor orderings were computed in that subspace. This approach ensured that the overhead of the nearest-neighbor computations was insignificant, relative to that of the exact KM computation.

This work was done by Dominic Mazzoni and Dennis DeCoste of Caltech for **NASA's Jet Propulsion Laboratory**. *Further information is contained in a TSP (see page 1).*

The software used in this innovation is available for commercial licensing. Please contact Don Hart of the California Institute of Technology at (818) 393-3425. Refer to NPO-40441.

Indentured Parts List Maintenance and Part Assembly Capture Tool — IMPACT

Viewing and maintaining the complex assembly hierarchies of large databases is made easier.

Lyndon B. Johnson Space Center, Houston, Texas

Johnson Space Center's (JSC's) indentured parts list (IPL) maintenance and parts assembly capture tool (IMPACT) is an easy-to-use graphical interface for viewing and maintaining the complex assembly hierarchies of large databases. IMPACT, already in use at JSC to support the International Space Station (ISS), queries, updates, modifies, and views data in IPL and associated resource data, functions that it can also perform, with modification, for any large commercial database. By enabling its users to efficiently view and manipulate IPL hierarchical data, IMPACT performs a function unlike that of any other tool. Through IMPACT, users will achieve results quickly, efficiently, and cost effectively.

Speed, efficiency, and cost are critical issues in maintaining complex assembly hierarchies of large databases. IPLs consist of parts organized into such complex assembly hierarchies. The more complex the hierarchy, the more the associated list grows and the more difficult it becomes to locate a part to modify it. At JSC it was found that existing IPL manipulation methods were too complex, hard to use, and error-prone for time- and cost-sensitive ISS operations. IMPACT was therefore developed to address these drawbacks and to help users achieve results.

IMPACT uses a C++, X-Windows, and Motif application framework. At JSC, it operates with a PRO*C++ interface to an Oracle database. In this way, IMPACT can manipulate the vehicle master data-