Speeding Algorithms by Shrinking Data

In computer science, the buzzword of the day is "big data [1]." The proliferation of cheap, Internet-connected sensors — such as the GPS receivers, accelerometers and cameras in smartphones — has meant an explosion of information whose potential uses have barely begun to be explored. In large part, that's because processing all that data can be prohibitively time-consuming.

Most computer scientists try to make better sense of big data by developing ever-more-efficient algorithms. But in a paper presented this month at the Association for Computing Machinery's International Conference on Advances in Geographic Information Systems, MIT researchers take the opposite approach, describing a novel way to represent data so that it takes up much less space in memory but can still be processed in conventional ways. While promising significant computational speedups, the approach could be more generally applicable than other big-data techniques, since it can work with existing algorithms.

In the new paper [2], the researchers apply their technique to two-dimensional location data generated by GPS receivers, a very natural application that also demonstrates clearly how the technique works. As Daniela Rus, a professor of computer science and engineering and director of MIT's Computer Science and Artificial Intelligence Laboratory, explains, GPS receivers take position readings every 10 seconds, which adds up to a gigabyte of data each day. A computer system trying to process GPS data from tens of thousands of cars in order to infer traffic patterns could quickly be overwhelmed.

But in analyzing the route traversed by a car, it's generally not necessary to consider the precise coordinates of every point along the route. The essential information is the points at which the car turns; the path between such points can be approximated by a straight line. That's what the new algorithm does.

A key aspect of the algorithm, explains Dan Feldman, a postdoc in Rus' group and lead author on the new paper, is that it can compress data on the fly. For instance, it could compress the first megabyte of data it receives from a car, then wait until another megabyte builds up and compress again, then wait for another megabyte, and so on — and yet the final representation of the data would preserve almost as much information as if the algorithm had waited for all the data to arrive before compressing.

Drawing the line

In some sense, Feldman says, the problem of approximating pathways between points is similar to the problem solved by regression analysis [3], a procedure common in statistics that finds the one line that best fits a scatter of data points. One major difference, however, is that the researchers' algorithm has to find a series of line segments that best fit the data points.

As Feldman explains, choosing the number of line segments involves a trade-off between accuracy and complexity. "If you have N points, k" — the number of line segments — "is a number between 1 and N, and when you increase k, the error will be smaller," Feldman says. "If you just connect every two points, the error will be zero, but then it won't be a good approximation. If you just take k equal to 1, like linear regression, it will be too rough an approximation." So the first task of the algorithm is to find the optimal trade-off between number of line segments and error.

The next step is to calculate the optimal set of k line segments — the ones that best fit the data. The step after that, however, is the crucial one: In addition to storing a mathematical representation of the line segment that best fits each scatter of points, the algorithm also stores the precise coordinates of a random sampling of the points. Points that fall farther from the line have a higher chance of being sampled, but the sampled points are also given a weight that's inversely proportional to their chance of being sampled. That is, points close to the line have a lower chance of being sampled, but if one of them is sampled, it's given more weight, since it stands in for a larger number of unsampled points.

It's this combination of linear approximations and random samples that enables the algorithm to compress data on the fly. On the basis of the samples, the algorithm can recalculate the optimal line segments, if needed, as new data arrives.

Satisfaction guaranteed

During compression, some information is lost, but Feldman, Rus, and graduate student Cynthia Sung, the paper's third author, were able to provide strict

mathematical guarantees that the error introduced will stay beneath a low threshold. In many big-data contexts, a slightly erroneous approximation is much better than a calculation that can't be performed at all.

In principle, the same approach could work with any type of data, in many more dimensions than the two recorded by GPS receivers. For instance, with each GPS reading, a car could also record temperature and air pressure and take a snapshot of the road ahead of it. Each additional measurement would just be another coordinate of a point in multiple dimensions. Then, when the compression was performed, the randomly sampled points would include snapshots and atmospheric data. The data could serve as the basis for a computer system that, for instance, retrieved photos that characterized any stretch of road on a map, in addition to inferring traffic patterns.

The trick in determining new applications of the technique is to find cases in which linear approximations of point scatters have a clear meaning. In the case of GPS data, that's simple: Each line segment represents the approximate path taken between turns. One of the new applications that Feldman is investigating is the analysis of video data, where each line segment represents a scene, and the junctures between line segments represent cuts. There, too, the final representation of the data would automatically include sample frames from each scene.

According to Alexandre Bayen, an associate professor of systems engineering at the University of California, at Berkeley, the MIT researchers' new paper "pioneers the field" of "extracting repeated patterns from a GPS signal and using this data to produce maps for streaming GPS data."

In computer science parlance, Bayen explains, a reduced data set that can be processed as if it were a larger set is called a "coreset." "The coreset is a good solution to big-data problems because they extract efficiently the semantically important parts of the signal and use only this information for processing," Bayen says. "These important parts are selected such that running the algorithm on the coreset data is only a little bit worse than running the algorithm on the entire data set, and this error has guaranteed bounds."

For more information visit www.mit.edu [4].

Source URL (retrieved on *07/30/2014 - 12:42am*):
http://www.wirelessdesignmag.com/news/2012/11/speeding-algorithms-shrinking-data?qt-most_popular=0

Links:
[1] http://web.mit.edu/newsoffice/2012/big-data-csail-intel-center-0531.html
[2] http://people.csail.mit.edu/dannyf/GIS_coreset.pdf
[3] http://web.mit.edu/newsoffice/2010/explained-reg-analysis-0316.html
[4] http://www.mit.edu