XLSTAT > CUDA Accelerated Cross Validated Best Subset Selection with XLSTAT

A. Bellétoile - Addinsoft, 40 rue Damrémont, 75018 Paris, France

1. Context

XLSTAT is the most complete and widely used statistical and data analysis add-on to Microsoft Excel. It offers a wide variety of statistical tools directly accessible from the MS Excel environment. With XLSTAT, statistics are made simple and our users can benefit from the most advanced methods via a simple, intuitive and effective interface.

Today, as a response to an increasing demand for mobility, XLSTAT is moving to the cloud with the MS Office 365 Suite and its Azure cloud services. However, operating computationally intensive algorithms via the cloud for thousands of users is a major challenge. It requires that we reconsider the way our algorithms are thought (or implemented) and embrace the new era of massive parallel computing to push them further.

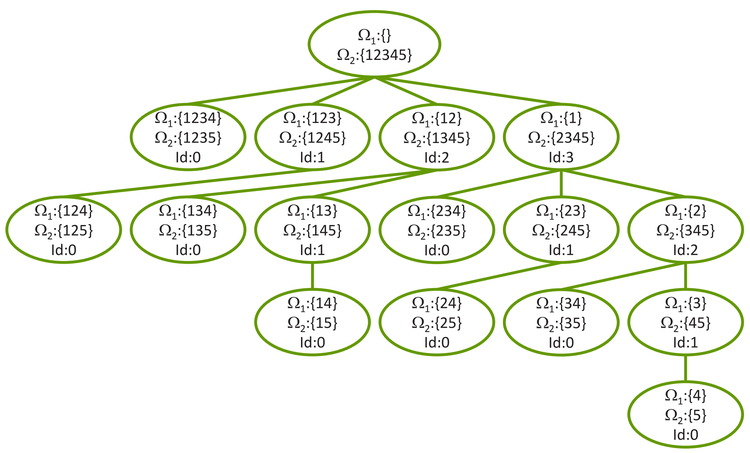
As our next GPU-enabled feature, we present here the new XLSTAT Cross Validated Best Subset Selection algorithm accelerated with CUDA.

2. Best Subset Selection

Based on the Leaps and Bounds algorithm proposed by Furnival & Wilson[1], our implementation identifies the best subset of predictors in the least squares sense in a model of the form:

$$Y = X.\beta$$

Where Y [Nx1] contains the observations, X [NxP] the explanatory variables and β [Px1] contains the model coefficients to be determined.
Our approach follows the one described in [2] where a tree, as the one just below, is grown from an initial regression on all possible covariates.



In our example, a maximum of P = 5 predictors are considered. The tree will be traversed from top to bottom and from left to right.

The initial node left apart, each node is made up of two distinct regressions named Ω_1 and Ω_2. They are obtained either by removing one covariate from the Ω_2 regression of the mother node or by removing the last covariate to the Ω_1 regression of its older sister, immediately on its left, if there are at least 2 siblings.

Every possible subset of covariates appears once and only once in this tree. It is an exhaustive enumeration that will grow exponentially with the number of covariates.

While growing, the tree will be pruned from branches of no interest by using the following basic inequality on each node:

$$RSS(A) \geq RSS(B)$$

Where A and B are two subsets of possible covariates with B ⊆ A and RSS(.) designate the Residual Sums of Squares from the regression on the corresponding subset.
Three possibilities may arise when considering a node:

1. $RSS(|\Omega_1|) \leq RSS(\Omega_2)$
   every child of this node is disregarded
2. $RSS(|\Omega_2|-k) < RSS(\Omega_2) \leq RSS(|\Omega_2|-k-1)$
   the k first children are disregarded
3. $RSS(\Omega_2) < RSS(|\Omega_2| - 1)$
   no child can be disregarded

When completed, the algorithm has isolated P best subsets, one for each number of covariates.

3. Removing predictors

Our implementation uses the Dense LAPACK functions available in the CUDA cuSolver. First, the full regression with all ordered covariates is perform via a QR factorization of the X matrix:

$$X = Q.R$$

Where R is an upper triangular matrix and Q an orthogonal matrix.
The R matrix is obtained from the geqrf function, then the ormqr function is used to compute B defined as follows:

$$B = Q^T.Y$$

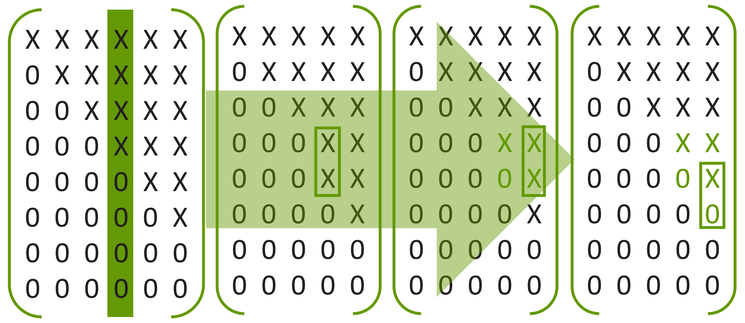
Finally, residuals are computed from B:

$$RSS = sum(B(P+1:end)^2)$$

From this point, each successive node regression is obtained from the removal of one covariate at a time of a previous regression. Numerous computations are saved by avoiding the complete computation of the next QR factorization [3,4,5].

Instead the R matrix resulting from the precedent regression is simply updated by removing the corresponding column. Then, if the removed predictor was located on the rightmost column (Ω_1 series), the update is over.

Otherwise, the trapezoidal submatrix of R resulting from the column removal is made triangular again by the means of Givens rotations as shown below. The same transformation is applied to B in order to get the RSS.



4. Cross Validation test

After propagating through the tree, we are left with P best models, one for each possible number of predictors. We now wish to select the best possible model among those P possible subsets.

One of the major issue with the RSS estimator used in our algorithm is that its value decreases systematically as the number of predictors increases, potentially leading to the undesirable overfit. To avoid this situation, we introduce a leave-one-out cross-validation technique as a last step towards our selection of the best model.

First, a test-data set is isolated at the beginning of the algorithm until the P best models have been identified. Then, the P sets of model coefficients are computed by solving the R and B system with the trsm back substitution function of the cuSolver.

Finally, the prediction power of each model is estimated via the PRESS (or PRedictive Error Sum of Squares) with the test-data set and the model coefficients.

5. Conclusion

Our new Cross Validated Best Subset Selection under development on GPU has been presented. From the first measurements, the speed gain resulting from the new CUDA implementation is expected to exceed 10x for the largest datasets in the final version.

Now, with the NVIDIA grid accessible from the MS Azure cloud services, we can leverage GPU benefits to offer short time responses to our users even during heavy load periods while controlling both costs and energy consumption.

References

[1] Furnival & Wilson, 1974
[2] Ni & Huo, 2005
[3] Hammarling & Lucas, 2008
[4] Andrew & Dingle, 2014
[5] Gatu & Kontoghiorghes, 2006