Maxvol for Machine Learning

Philip Blagoveschensky, Ivan Golovatskikh,
Maria Sindeeva, Mirfarid Musavian

December 20, 2018

Abstract

In this paper we describe applications of of maximum volume submatrices and `rect_maxvol` algorithm in machine learning. We investigate whether `rect_maxvol` method for finding a rectangular submatrix with large volume can be used for feature selection and sample selection.

1 Introduction

In machine learning we often don't have powerful enough computers or large enough memory to perform calculations on the whole dataset which has too many features and/or samples. A dataset is represented with a matrix. If we can find a good subset of features, we'll be able to train the machine learning algorithm using computational resources. We can also try to achieve the same goal by using only the most representative samples. Feature selection and samples selection is equivalent to choosing a submatrix. For both purposes we can try choosing a submatrix with large volume, where volume as the absolute value of determinant for square matrices, is generalized for rectangular matrices.

1.1 Theoretical basics

For square matrix its volume is defined as the absolute value of its determinant. [1] claims that if we want to choose a square submatrix with low condition number, then indeed we need to choose a submatrix with maximum volume. [2] extends the definition of volume for non-square matrices:

$$\mathrm{vol}(A) = \sqrt{\det(A^* A)}$$

Computation of a submatrix of maximum volume is NP-hard [3], hence it is unfeasible to compute. Hence we will compute a submatrix with large but not necessarily maximum volume, and this can be done in polynomial time using **rect_maxvol** algorithm. [2] To be precise, the algorithm lets us choose $\epsilon > 0$ and finds a subset of indices $J \subseteq \{1, \dots, n\}$ such that if we remove one element of J and add a different one, which was not already in J, the volume of $A_{\cdot,J}$ will not increase by more than a factor of $1 + \epsilon$. A submatrix is called quasi-dominant if it has this property.

1.2 Related Work

The problem of dimesionality reduction, or feature selection, in large machine learning datasets is not new. It's a rather well-explored subject, and many methods are being applied to achieve this. For example, in [4] sample matrix **CUR**-decomposition is used for simultaneous sample and feature selection for the task of active learning. The goal is to select a subset of features in such way that further sample selection (active learning) is not influenced by noisy or uninformative features.

The idea of using maximum volume submatrices for dimensionality reduction in large matrices has been explored in [5]. There the rectangular maxvol algorithm is applied in context of antenna selection in massive multiple-input multiple-output (MIMO) systems for maximizing channel capacity. In practice this means that for channel matrix $\mathbf{H}$ N_s columns, corresponding to antennas, should be selected, so that the capacity metric $C(\mathbf{H_s}) = log_2 det(\mathbf{I} + \frac{\rho}{N_s}\mathbf{H_s}\mathbf{H_s^*})$ is maximized. The traditional methods for this task make use of square **maxvol** algorithm to, but it is shown that in such systems `rect_maxvol` achieves similar performance while removing limitations of considering square submatrices only.

Maximum volume submatrices can be used for the "cold start" problem in recommendation systems [6] and for functions approximation, e.g. in the case of linear regression $y = \tilde{A}x + \theta$, maximizing such objective function leads to minimizing noise variance:

$$\mathrm{Var}(x) = (\tilde{A}^* \tilde{A})^{-1} \sigma^2.$$

This is called the *D-optimality* criterion. [7]

1.3 Our hypotheses

We had two main hypotheses to explore the boundaries of `rect_maxvol` algorithm usefulness:

- Maxvol would perform well for feature selection when the number of features is much greater than the number of samples. This hypothesis was tested under two different assumptions:
  - There are some core ('true') features, and the rest are their transformations: either linear or non-linear. Since `rect_maxvol` by definition chooses a submatrix with maximum available volume, intuition behind the algorithm performing well in this case is rather clear.
  - There are some core ('true') features, and the rest are completely uninformative noise, that carries no predictive power. Unlike in the first assumption, it is not immediately clear that `rect_maxvol` would perform well in this case, so it's worth exploring its applicability.
- Maxvol would perform well for subsampling task by selecting the most informative samples of the dataset.

2 Experiments

For each hypothesis we carried out at least one experiment. Both synthetic datasets and real-world data were used. Everyone in our team conducted his/her own experiment and wrote a section about it.

2.1 Synthetic datasets

This series of experiments was performed by Philip Blagoveschensky. We modeled a real world dataset by a probability distribution defined as follows.

- Two classes – B and R. Define events B and R to be "the object is of class B" and "the object is of class R" respectively. $P(B) = 0.75, P(R) = 0.25$.
- Introduce k random variables $z_1, \ldots, z_k$ called "true features" . Define $\sigma_{\min}, \sigma_{\max}$ – range of standard deviations for "true features". If class is B, then $\forall i \in \{1, \ldots, k\}\, z_i \sim N(2, \sigma_i)$; if class is R then $\forall i \in \{1, \ldots, k\}\, z_i \sim N(4, \sigma_i)$. Also $\forall i \in \{1, \ldots, k\}\, \sigma_i \in [\sigma_{\min}, \sigma_{\max}]$.

We wanted to check if feature selection by `rect_maxvol` is likely to discard redundant features. For this purpose we performed three experiments, in each of them we defined new random variables $x_1, \ldots, x_{5k}$. These features are stochastic transformations of "true features" and they have redundancy, i.e. for each "true feature" z_i we have generated 5 features $x_i, x_{k+i}, x_{2k+i}, x_{3k+i}, x_{4k+i}$ which are stochastic transformations of z_i.

In each experiment we wanted to choose k features. We sampled k samples from this probability distribution and built a $k \times 5k$ matrix, where each row represents a sample and each column represents a feature. Then we normalized columns (this is a standard data preprocessing step in machine learning) - for each column we subtracted from its elements its empirical mean and divided its elements by its empirical standard deviation. Then we chose ran `rect_maxvol` with target number of columns equal to k. `rect_maxvol` chooses some subset $I \subseteq \{1, \ldots, 5k\}$ of columns, and those are the features we choose, i.e. $\{x_i \mid i \in I\}$.

Our hypothesis was that feature selection by `rect_maxvol` will choose many distinct features, i.e. features which were not generated from the same "true features". To check to what extent this is true for each experiment we calculated the percentage of distinct features among chosen features, which is defined mathematically as

$$D(I) = \frac{|\{i \mid i \in \{1, \ldots, k\} \wedge \exists j \{1, \ldots, k\}\, (j-1)k + i \in I\}|}{k}$$

where I is the chosen subset of columns as described in the previous paragraph.

2.1.1 Experiment: Linearly dependent features

In the first experiment we generated features from true features simply by adding gaussian noise with small standard deviation. That is,

$$\forall i \in \{1, \ldots, k\} \forall j \in \{1, \ldots, 5\} \; x_{(j-1)k+i} = z_i + \eta_{(j-1)k+i}$$

with

$$\eta_{(j-1)k+i} \sim N(0, 0.1).$$

In theory feature selection by `rect_maxvol` is likely to choose a subset I of features with high percentage of distinct features, i.e. $D(I)$, because if I contains two features generated from the same "true feature", then columns of the submatrix will be almost linearly dependent (possibly not actually linearly dependent only because of noise), hence the determinant will be small in magnitude.

Our experiments confirmed this as seen in figure 1.

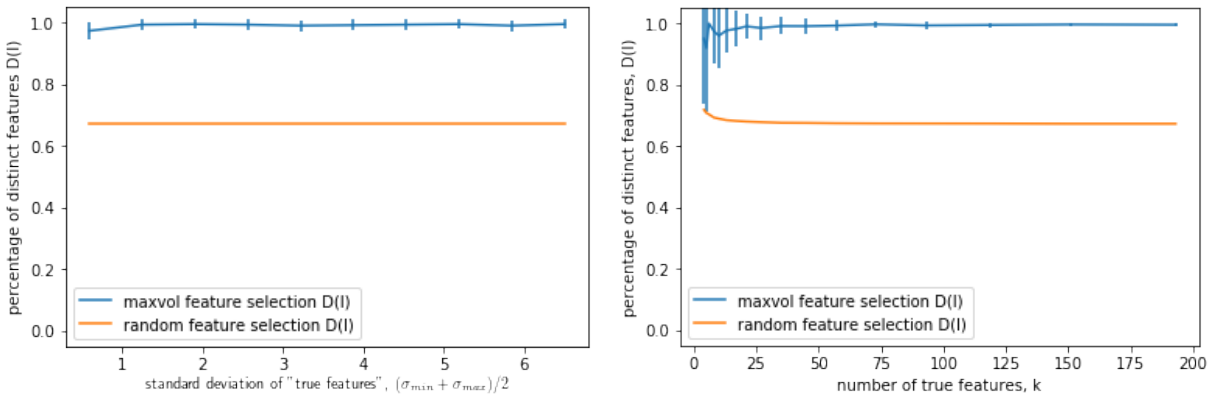

Figure 1: Features are generated from "true features" simply by adding noise, hence features are almost linearly dependent. In the left plot the number of "true features" $k = 60$ is constant but standard deviation of "true features" is changing with $\sigma_{\max} - \sigma_{\min} = 1$. In the right plot k is changing while standard deviation of "true features" is constant with $\sigma_{\min} = 3, \sigma_{\max} = 4$. In both plots $D(I)$ (larger is better) of `rect_maxvol` feature selection is compared to the expected percentage of distinct features $D(I)$ if features were selected randomly. For each value of k and $\frac{\sigma_{\min}+\sigma_{\max}}{2}$ we have performed 10 experiments. Error bars represent range of experimental results no more than two standard deviations above and below mean.

2.1.2 Experiment: transformations by continuous functions

This experiment's purpose was to model a more realistic dataset. We generated features from true features by adding small noise and then applying a continuous function from the following collection of functions:

1. $f_1(z) = z + c_1$
2. $f_2(z) = e^(z + c_2)$
3. $f_3(z) = \sqrt{(|z + c_3|)}$
4. $f_4(z) = (z + c_4)^2$
5. $f_5(z) = (z + c_5)^3$

Here $c_1, \ldots, c_5$ are some randomly generated constants (not random variables of our probability distribution). They remain the same for all samples sampled from the distribution.

Mathematically feature generation for this experiment is described as

$$\forall i \in \{1, \ldots, k\} \forall j \in \{1, \ldots, 5\} \; x_{(j-1)k+i} = f_j(z_i + \eta_{(j-1)k+i})$$

with

$$\eta_{(j-1)k+i} \sim N(0, 0.1).$$

As you can see in figure 2, the percentage of distinct features used `rect_maxvol` is greater than would be expected with random feature selection, but significantly worse than with linearly dependent feature generation.

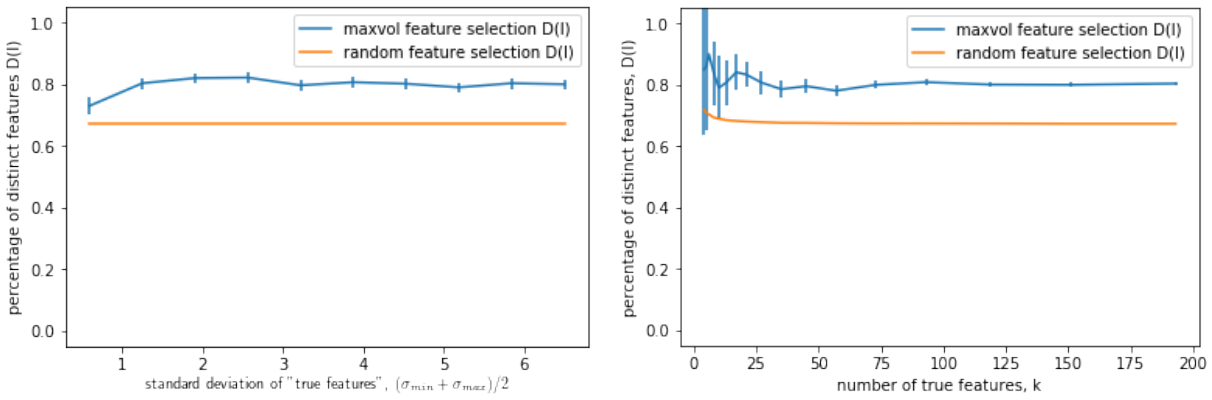

Figure 2: Features are generated from "true features" by stochastic continuous transformations. In the left plot the number of "true features" $k = 60$ is constant but standard deviation of "true features" is changing with $\sigma_{\max} - \sigma_{\min} = 1$. In the right plot k is changing while standard deviation of "true features" is constant with $\sigma_{\min} = 3, \sigma_{\max} = 4$. In both plots $D(I)$ (larger is better) of `rect_maxvol` feature selection is compared to the expected percentage of distinct features $D(I)$ if features were selected randomly. For each value of k and $\frac{\sigma_{\min}+\sigma_{\max}}{2}$ we have performed 10 experiments. Error bars represent range of experimental results no more than two standard deviations above and below mean.

2.1.3 Experiment: one class

We wanted to check if it matters that we have two classes B and R with different "true feature" means. We decided to see what will happen if there is only one class. Does `rect_maxvol` select distinct features because we have two classes which have different means of distributions of features? Hence we've repeated the previous two experiments, except with $P(B) = 1, P(R) = 0$.

In figure 3 you can see the plots for this experiment. On average `rect_maxvol` selection performed as well as with two classes, but variance of $D(I)$ was much larger as can be seen from the size of error bars. We don't know what to infer from this result.

2.2 ARCENE dataset

This series of experiments was carried out by Maria Sindeeva.

ARCENE dataset consists of mass-spectrometry data for healthy and cancer patients, and target to be predicted is whether the patient is healthy or not. This dataset is from 2003 NIPS feature selection contest, and it features 7 000 real features and 3 000 uninformative ones with no predictive power. It has data for only 100 patients for training, which makes it perfect for testing the first hypothesis under the second assumption.

To evaluate how well Maxvol performs feature selection, we trained a perceptron on the transformed dataset.

The dataset is a full-rank matrix, which means that the `rect_maxvol` algorithm can be applied directly to it. However, we also tried to apply it to the dataset after it's features have been scaled (via MinMax scaling), since column-wise scaling, often used to speed up the training of a perceptron, does change the choice of submatrix in `maxvol` and `rect_maxvol` algorithms.

In both scaled and unscaled settings of the experiment, the accuracy of the four following models was compared:

- Trained on `rect_maxvol`-selected features
- Trained on a randomly selected subset of features. In this case accuracy of the model is averaged over models trained on several random feature choices from the original dataset
- Trained on all features
- Trained on PCA transformation of the data. Since there are only 100 samples available for training, we take maximum available number of components for PCA dimensionality reduction, which is 100

On figure 4 we can see that there is no obvious dependency of accuracy from the number of features selected by maxvol – it doesn't give us any advantage to use it in this case. We can not even say that it performs better than random selection of the features.

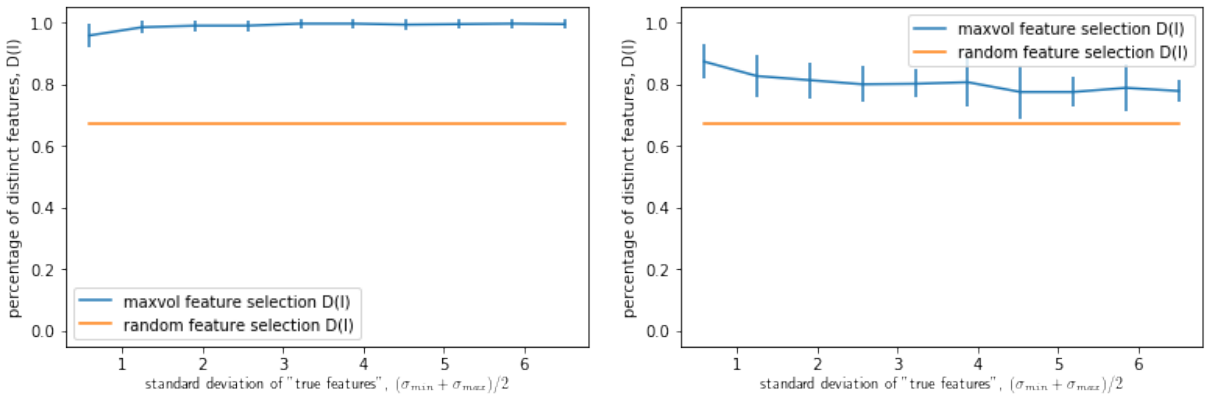

Figure 3: There is only one class, i.e. $P(R) = 0$. Features are generated from "true features" (in the left plot) by adding noise, hence features are almost linearly dependent; (in the right plot) by stochastic continuous transformations. The number of "true features" $k = 60$ is constant but standard deviation of "true features" is changing with $\sigma_{\max} - \sigma_{\min} = 1$. In both plots $D(I)$ (larger is better) of `rect_maxvol` feature selection is compared to the expected percentage of distinct features $D(I)$ if features were selected randomly. For each value of $\frac{\sigma_{\min}+\sigma_{\max}}{2}$ we have performed 10 experiments. Error bars represent range of experimental results no more than two standard deviations above and below mean. Note that error bars are much larger than on analagous plots with two classes B and R.

When applied to the scaled dataset, the obtained results seem to be much better, as can be seen on figure 5. Staring from 3 000 features, maxvol even gives better accuracy than random feature selection, and gives peak performance at 7 000 selected features - exactly the number of truly useful features in the original dataset. However, if we take a more detailed look into what actually happens around that peak, we can see (figure 6) that these great results we saw on figure 5 are just a coincidence, and there is in fact no real improvement.

Thus, there is no reason to believe that maxvol would perform well for feature selection under the second assumption.

2.3 MNIST dataset

MNIST is a well-known machine learning dataset consisting of handwritten digits. Each entry of the dataset is represented by vector of 784 numbers (binray 28×28-pixels image), 60000 training samples and 10000 test samples in total. Features in this dataset are represented by interger numbers from 0 to 255, so they are distributed equally and there are no need to scale them to the same interval.

For this dataset we checked both of our hypotheses — sampling and feature selection capabilities of Maxvol algorithm.

The dataset is sparse and contains many redundant features (first and last columns of dgits images consist of zeros), so on the one hand this gives us a natural reason to test feature selection approach, but on the other hand we can't directly apply Maxvol algorith to dataset matrix because this matrix is column rank deficient. To overcome this property of the dataset we used rank-k SVD approximation of dataset matrix:

$$Y_k = U\Sigma_k V^T$$

2.3.1 Feature selection

For the feature selection Maxvol algorithm was applied to the V matrix. Obtained indices were used to select subset of features. After that we trained linear perceptron model to classify vectors to 10 digits classes. We compared this approach with random feature selection and PCA (figure 7). Maxvol significantly outperformed both random and PCA feature selection for small amount of features: $k = \{20, 50, 100, 20\}$. Indeed, MNIST dataset is represented by sparse matrix and Maxvol algorithm efficiently skips zero rows/columns. For some points Maxvol-based approach outperforms training on the whole features set.

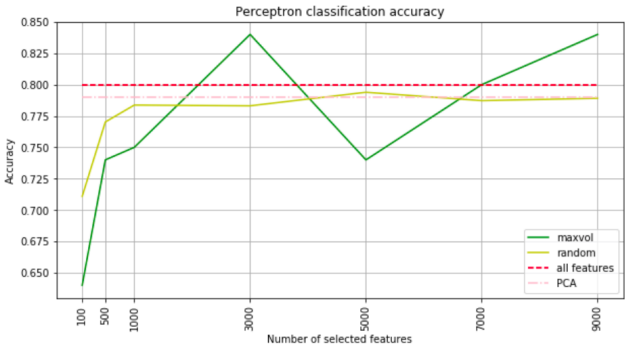

Figure 4: No scaling results

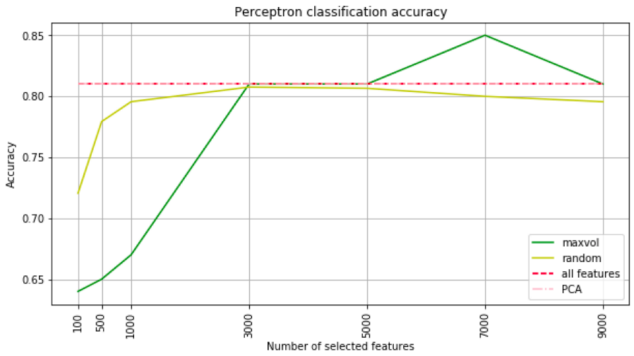

Figure 5: MinMax scaling results

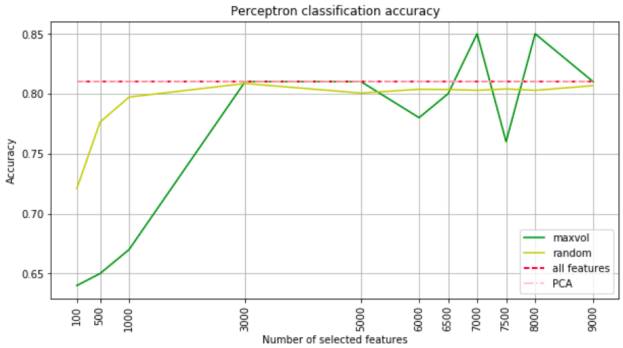

Figure 6: MinMax scaling results in more detail

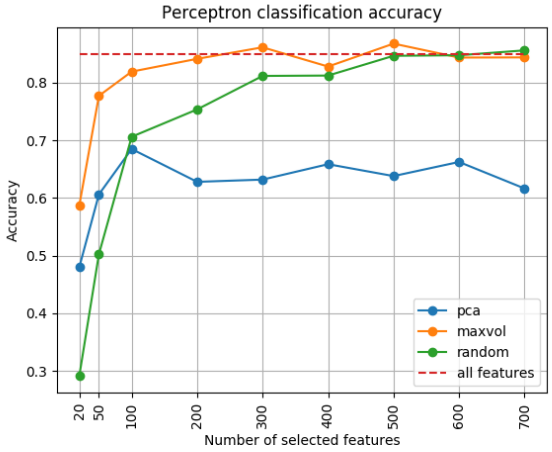

Figure 7: MNIST feature selection comparison. Maxvol shows good result even for 20-50 features. It can be explained by the fact that Maxvol efficiently skips zero columns of sparse MNIST dataset

2.3.2 Sampling

For the sampling problem we tried different approaches. The most meaningful way to find representative samples for each class is to try a similar to feature selection approach: a submatrix with the largest determinant was selected from matrix U in the rank-k SVD approximation, after that obtained indices gave us desired samples, which we used to train perceptron. However that approach didn't work: even random examples sampling showed better results (figure 8). Thus, this topic needs further research.

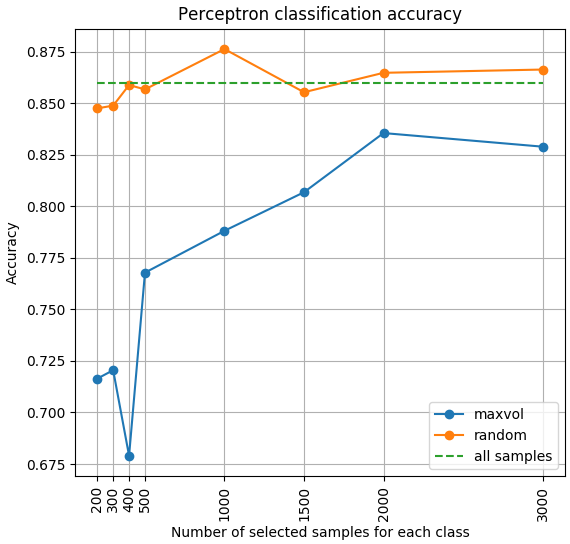

Figure 8: MNIST dataset sampling results. Maxvol works even worse than random sampling

2.4 Housing Data sets

One of the most popularly used techniques for predicting continuous values is linear regression. In these following experiments we have checked the second hypothesis, optimal sub-sampling capabilities of Maxvol algorithm.

2.4.1 Experiment no.1 : USA Housing

This data set is a relatively small data set 5000×7, which even simple out of the box linear regression gives good results. This following plot shows the obvious correlation between the predicted prices using sklearn Linear Regression, with default parameters.

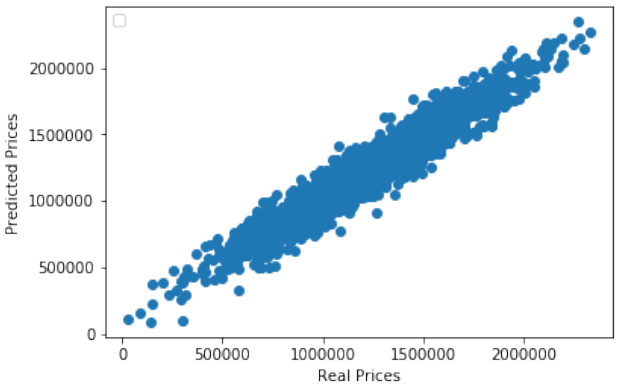

Figure 9: Showing the correlation between predicted and actual prices

So we did tried sampling different number of sub samples of the inputs via rectangular Maxvol algorithm vs simple random selection to check how much better it would perform. The following plots are comparing the result of this experiment on the basis of how well each of them has performed regarding r^2 score and mean squared errors.

As presented in these plots rect_maxvol is capable of showing very promising results with very low sampling sizes, mainly because the accuracy of the random sample selection method has a lot of variability and unreliability.

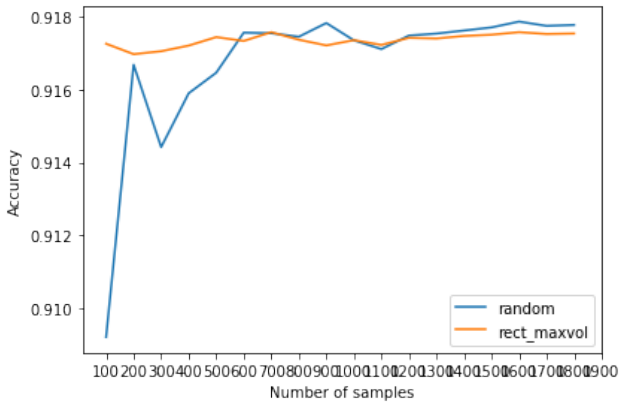

Figure 10: the r^2 accuracy of the logistic regression performed on various subsamples with random selection and Maxvol

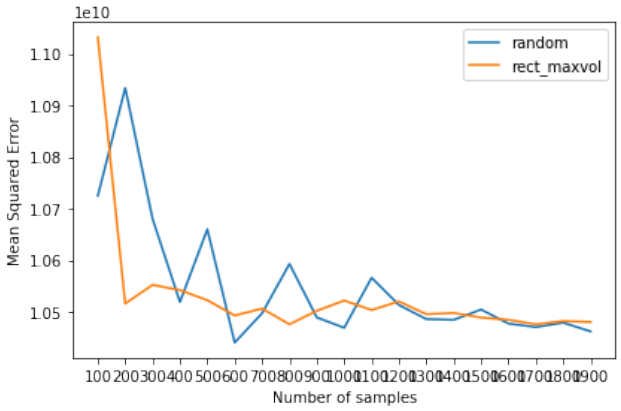

Figure 11: the Mean Squared Errors of the logistic regression performed on various subsamples with random selection and Maxvol

2.4.2 Experiment no.2 : Housing data-set with more features

To experiment the potentials of Maxvol algorithm in regression modeling further, we tried the same procedure on a data set with relatively much more features on the same subject, housing price prediction. This is a data set for a ongoing kaggle competition, the aim being getting high scores in cross validation The original size of this data set is 2919×73, which after some standard pre processing steps we its final size became 2919×131. However, in this experiment we have divided the trainging data itself into train and test sets to be more compatible with the results of the experiment mentioned the section above. After trying different regression models with little to no success, the following resluts were obtained for accuracy scores for different number of sub sampling with Maxvol vs random sampling using Ridge Regression (figure 12).

But a more careful look in this result shows some abnormal results on mean squared errors in this trainings, which is evident in this following plot 13.

3 Conclusion

Our experiments suggest that submatrix volume can be used for feature selection, but the evidence is not conclusive. It works especially well when some features are linearly dependent. We don't know when submatrix volume can be used for subsampling, because it showed good results in one experiment and worse than random results in another. And finally, don't forget to normalize your features!

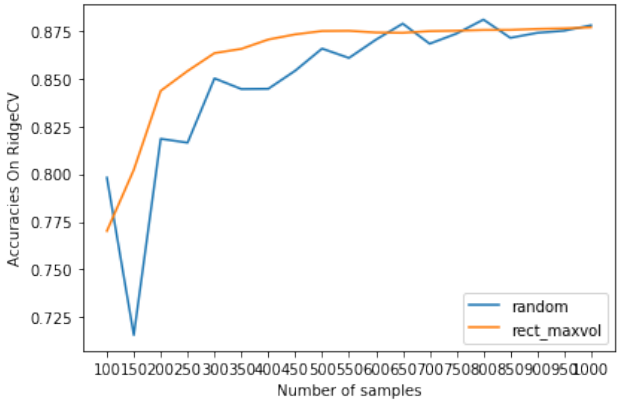

Figure 12: the r^2 accuracy of the Ridge regression performed on various subsamples with random selection and Maxvol

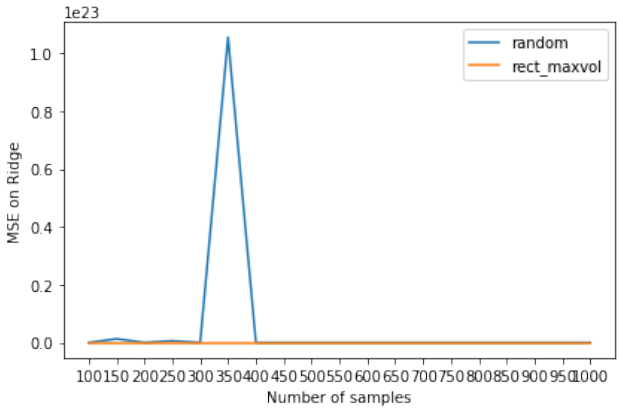

Figure 13: the Mean Squared Errors of the Ridge regression performed on various subsamples with random selection and Maxvol

4 Source code

All code for our experiments can be found here: `https://github.com/Vicontek/maxvol`

References

[1] S. A. Goreinov et al. "How to find a good submatrix". In: *Research Report 08-10, ICM HKBU, Kowloon Tong, Hong Kong* (2008), pp. 08–10.

[2] A. Mikhalev and I. V. Oseledets. "Rectangular maximum-volume submatrices and their applications". In: *ArXiv e-prints*, arXiv:1502.07838 (Feb. 2015), arXiv:1502.07838. arXiv: `1502.07838 [math.NA]`.

[3] John J. Bartholdi III. "A Good Submatrix is Hard to Find". In: *Oper. Res. Lett.* 1.5 (Nov. 1982), pp. 190–193. ISSN: 0167-6377. DOI: `10.1016/0167-6377(82)90038-4`. URL: `http://dx.doi.org/10.1016/0167-6377(82)90038-4`.

[4] C. Li et al. "Joint Active Learning with Feature Selection via CUR Matrix Decomposition". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2018), pp. 1–1. ISSN: 0162-8828. DOI: `10.1109/TPAMI.2018.2840980`.

[5] H. Tang and Z. Nie. "RMV Antenna Selection Algorithm for Massive MIMO". In: *IEEE Signal Processing Letter* 25.2 (Feb. 2018), pp. 239–242. DOI: `10.1109/LSP.2017.2783350`.

[6] Nathan N Liu et al. "Wisdom of the better few: cold start recommendation via representative based rating elicitation". In: *Proceedings of the fifth ACM conference on Recommender systems*. ACM. 2011, pp. 37–44.

[7] J Kiefer. "Optimum experimental designs V, with applications to systematic and rotatable designs". In: *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability*. Vol. 1. Univ of California Press. 1961, pp. 381–405.