Dimensionality reduction for anti-cancer drug synergy prediction

Felipe Peter
Tsinghua University
felipe.peter@tum.de

Abstract

Deep learning approaches to drug combination synergy prediction for cancer treatment have shown to outperform classic machine learning approaches. This paper shows that principal component analysis can be used to greatly reduce the number of features for a dataset used in a recent work. Furthermore, the size and complexity of the used neural network can be reduced as well, without sacrificing model performance. These measures improve training speed while reducing memory consumption of the model.

1. Introduction

For a hundred years the paradigm of "magic bullets" governed the development of new drugs. Nobel laureate Paul Ehrlich stated in 1900 that for each disease there exists a drug, the magic bullet, that specifically targets this disease [7]. Many magic bullets were discovered between 1970 and the 1990's and changed the way doctors practiced medicine. However this period neared its end with the discovery of drugs for breastcancer (1993) and AIDS (1995) treatment. Since the 2000's the pharmaceutical industry struggled to find new magic bullets. Instead, a lack of efficacy and high toxicity of single drugs have become the major reasons for attrition in clinical studies. According to Hopkins [7], advances in systems biology have revealed that selective drugs are not as effective as combinations of drugs or drugs targeting multiple proteins. This implies that there are many more effective drugs or drug combinations to be discovered but also requires new methods in drug development. This work focuses on the discovery of synergistic drug combinations for cancer treatment.

Today's drug development incorporates three main approaches. The most classic approach is to conduct clinical trials to find out whether a drug combination works well. While this method gives a direct feedback, it is time-, labor-, and cost-intensive because it is mainly trial-and-error-based. Furthermore, it can potentially be harmful for the patients. Especially since the shift from classic cytotoxical chemotherapy (targeting cancer cells) to molecularly targeted agents (MTAs, aiming at proteins or enzymes in a cancer cell) the prediction of the relationship between dose, toxicity and efficacy has led to problems.

A less time- and cost-intensive approach is high-throughput screening (HTS). Using robots, data analysis and a high degree of automation, it alows scientists to quickly conduct a large number of pharmacological tests. HTS can be used as a brute force approach to synergistic drug combination discovery by testing a large number of combinations and measuring their synergistic effects. The results can then be used for drug design and investigation of the underlying mechanics. While HTS significantly speeds up the generation of test results for drug combinations, it can not cover the whole combinatorial space [8].

Recently, research has started focusing on developing computational methods for drug combination discovery which can be used to describe the underlying mechanics of drug-target interactions and drug-drug synergies or adverse effects. The necessary models can be trained by incorporating the results from HTS tests. Trained computational models are then used to predict synergistic effects of drug combinations as guidance for in vitro and in vivo tests.

The rest of this work is organized as follows. Chapter 2 gives an overview of recent research in the area of computational models for drug interaction prediction. It then introduces one of the recent approaches that uses deep learning methods to predict synergistic drug combinations for cancer treatment. The utilized dataset is also presented in detail. Chapter 3 explains how the existing method can be improved by using dimensionality reduction techniques. Finally, Chapter 4 evaluates the results in the context of recent research and Chapter 5 wraps up the findings.

2. Related Work

Recently, several approaches for the application of machine learning to drug interaction prediction have been proposed. Pang et al. [11] formulated the drug combination discovery problem as a mixed integer linear program (MILP). Based on the DrugBank [15] database, Pang et al. created a bipartite graph to represent drug-target interactions. Using the MILP approach, they identified drugs that

target a given disease gene set while minimizing the off-target interactions. While this approach showed promising results by predicting approved drug combinations, it lacks expressiveness due to the usage of only binary interaction relations. Furthermore, the framework does not include probabilistic considerations for drug-target interaction.

This limitation was overcome by Peng et al. [8] by introducing a probabilistic framework to predict synergistic drug combinations, which increased predictive expresiveness. However, that approach still only considered binary drug-target interactions.

Wildenhain et al. [14] created their own dataset by measuring the pairwise synergistic effects of 128 chemically diverse compounds on non-essential deletion strains of yeast. The gained Bliss independence values for the compound combinations where used to train a naive bayes learner and a random forest classifier in order to predict Bliss independence values for new drug combinations. While this approach incorporated both probabilistic elements and quantifiable synergy values, the authors realized that some of the features they selected did not contribute to the predictive power of the framework.

Deep learning methods can be used to avoid the need for hand-crafted features. They directly learn a mapping from the input data to the desired output values. Preuer et al. [12] where the first to use a deep neural network (named DeepSynergy) for synergy prediction of drug combinations for cancer treatment. The dataset used to train the model was generated in a series of HTS tests in 2016 by O'Neil et al. [10]. The HTS was performed using 39 cancer cell lines derived from 8 different tissue types. A total of 38 experimental and approved anti-cancer drugs where used in pairwise combinations on these cell lines. Only a subset of 22 drugs was exhaustively combined with all other drugs. This led to an overall number of 23,062 samples.

Preuer et al. [12] calculated the Loewe additivity score for all samples from the screening tests. Averaging duplicate samples lead to an overall number of 22,737 samples consisting of drug A, drug B, cell line, and synergy value. The authors calculated chemical features for the drugs, and genomic features for the cell lines. Based on chemical descriptors for the drugs they calculated extended connectivity fingerprints with a radius of 6 using jCompoundMapper [3]. With ChemoPy [2] they computed physico-chemical properties of the drugs. Lastly, toxicophore features were collected from literature [6], leading to an overall number of 4,387 features for each drug. The authors described the cell lines using their gene expression profile fetched from the ArrayExpress database [4], leading to 3,984 genomic features for each cell line. A sample therefore consists of 12,758 input features and one output synergy value.

Preuer et al. normalize the input features and apply the hyperbolic tangent to them before feeding them to a fully-connected neural network with two hidden layers, having 8192 and 4096 neurons respectively. A schematic representation is depicted in Fig. 1. The neural network incorporates a dropout rate of 0.2 for the first layer and 0.5 for the second layer. The layers use ReLU as their activation function. The learning rate is 1e-5 and the mini-batch size is 64. The hyperparameters were optimized using a grid search. To make the prediction independent of the drug input order, every sample is contained twice in the dataset (using drug combination AB and BA).

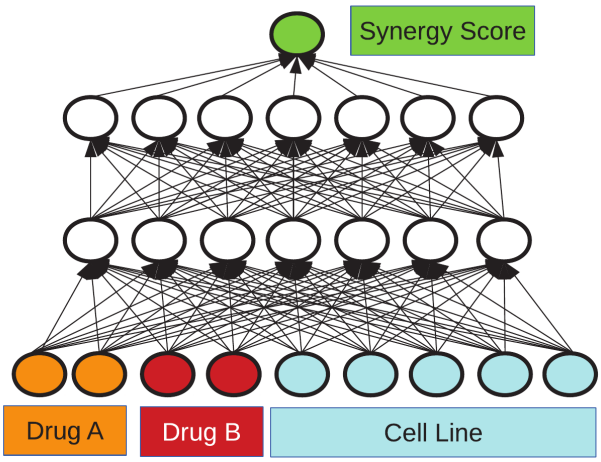

Figure 1. DeepSynergy architecture. Every sample consists of features for drug A, drug B and the cell line. The network uses 8192 neurons in the first layer and 4096 neurons in the second layer. [12]

DeepSynergy prediction results are reported using stratified cross-validation. That means that test data is extracted from the dataset in way that it only contains drug combinations, single drugs, or cell lines that are not present in the training dataset. This cross-validation strategy gives a good indication of how well the trained model generalizes to new data that has not been seen before. The results show that the model has poor performance on new drugs or new cell lines, which indicates that the dataset does not have enough samples. The predictive performance for new drug combinations is higher than for classic machine learning approaches like random forests, SVMs or elastic nets, leading to an overall MSE of 255.49 with a corresponding confidence interval of [239.93; 271.06].

3. Methods and Results

This work is based on the observation that the number of input features (12,758) for DeepSynergy is very high compared to the number of samples (45,474 with both orders of drugs). Furthermore, the model has a high number of 138 million tunable parameters. It is therefore questionable whether this configuration actually leads to the best possible performance, as the optimization of such a big number of parameters on such a sparsely covered input space is an ill-

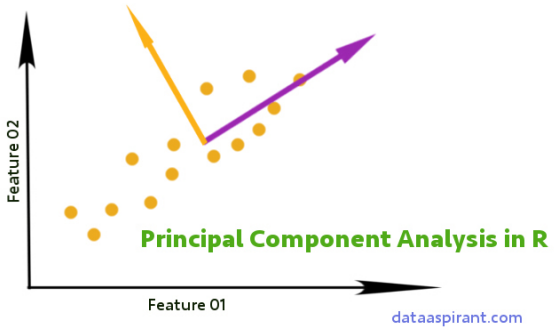

Figure 2. Depiction of a two-dimensional principal component analysis. The two arrows represent the two principal components of the dataset [1].

posed problem. This work examines whether a feature reduction of the input space allows for using a smaller model, therefore improving time- and memory-consumption during training, while retaining or improving the predictive performance.

3.1. Principle Component Analysis

Principle component analysis (PCA) is a statistical procedure that converts a set of possibly correlated features into a set of orthogonal (uncorrelated) features given a set of samples [9]. The orthogonal features are determined one by one, in a way that each new feature is orthogonal to all previous ones while covering the largest possible variance of the dataset. These new features are alled principal components. Fig. 2 shows the result of a PCA on a two-dimensional dataset.

In order to maximize the variance for the dataset X consisting of samples x_i, the first principal component w_1 must satisfy

$$w_1 = \underset{||w||=1}{\arg\max} \left\{ \sum_i (x_i w)^2 \right\} = \underset{||w||=1}{\arg\max} \left\{ w^T X^T X w \right\}$$

with $X^T X$ being the covariance matrix of the dataset. Solving that equation leads to the eigendecomposition of the covariance matrix with eigenvalue λ and eigenvector w

$$X^T X w = \lambda w.$$

So finding the principal component covering the greatest variance is equivalent to finding the eigenvector with the highest corresponding eigenvalue. The principal components can subsequently be used to transform the input data to a lower dimensional feature space while keeping as much variance of the original dataset as possible. This can be done by computing XW using a truncated W in which the columns consist of the eigenvectors ordered by decreasing corresponding eigenvalue.

PCA is usually not computed in the naive way due to problems with numerical stability. Instead singular value decomposition (SVD) is applied. The basic idea of SVD is that any real matrix $A \in \mathbb{R}^{n \times d}$ can be decomposed into

$$A = U \Sigma V^T,$$

where $U \in \mathbb{R}^{n \times r}$, $\Sigma \in \mathbb{R}^{r \times r}$, $V \in \mathbb{R}^{d \times r}$, with r being the number of desired reduced dimensions. The existence of a solution for any real matrix A can be proven [5]. To project data to the low dimensional space we can compute AV.

The relationship between PCA and SVD can be shown as follows, keeping in mind that U and V are column orthonormal and that Σ is a diagonal matrix:

$$\begin{aligned} X^T X &= (U \Sigma V^T)^T U \Sigma V^T \\ &= V \Sigma^T U^T (U \Sigma V^T) \\ &= V \Sigma \Sigma^T V^T \\ &= V \Sigma^2 V^T. \end{aligned}$$

The last line is equivalent to the eigendecomposition of $X^T X$ with Σ^2 containing the eigenvalues and V containing the eigenvectors.

To determine how many eigenvectors should be used, the variance var_i covered by each eigenvector v_i can be calculated by using its corresponding eigenvalue σ_i:

$$var_i = \frac{\sigma_i}{\sum_j \sigma_j}$$

The cumulated covered variance of the eigenvectors in decreasing order of corresponding eigenvalues gives an indication of the overall variance that is covered by these eigenvectors.

The cumulated variance coverage of the first 200 eigenvectors for the input dataset used by DeepSynergy can be seen in Fig. 3. The plot shows that only 107 eigenvectors are required to cover the whole variance of the dataset. The complexity of the synergy prediction task can be greatly simplified by transforming input data to this lower dimensional input space. In fact, using the same network architecture, but reducing the input space from 12,758 to 107 features reduces the number of tunable model parameters from 138 million to 34.5 million.

3.2. Neural Network

Reducing the number of input features without loss of information suggests that the size of the neural network can also be reduced. In this work a hyperparameter search using Bayesian optimization [13] was performed. The hyperparameter configuration that was used for the final evaluation used a batch size of 2048, a learning rate of 0.001, 512 neurons in the first hidden layer, 256 neurons in the second hidden layer, and a dropout rate of 0.3 for both layers. Furthermore batch normalization layers were included after the

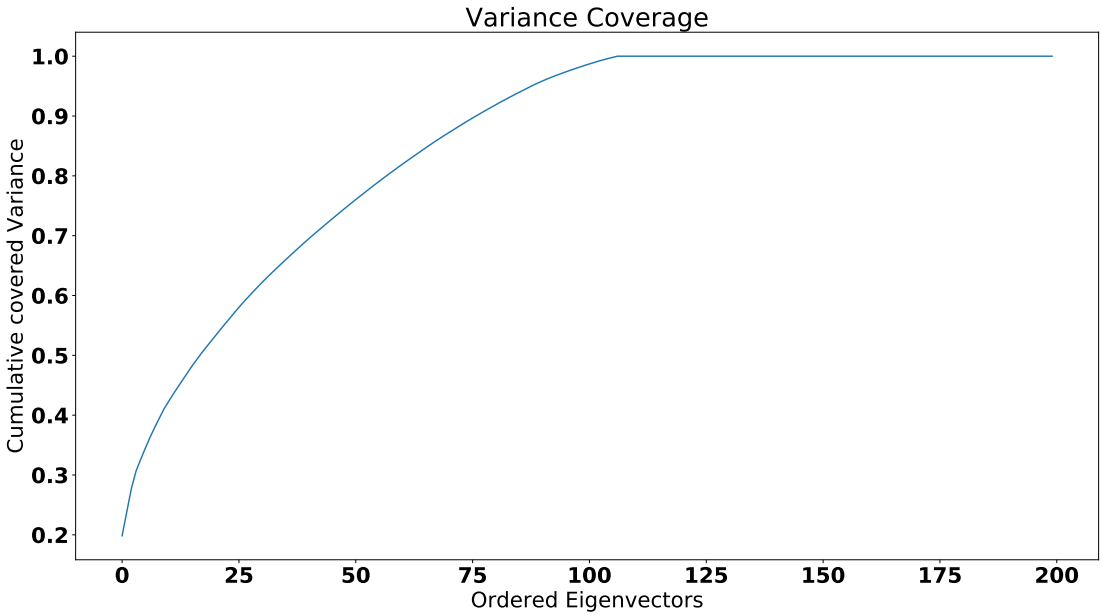

Figure 3. Variance coverage of the 200 eigenvectors of the DeepSynergy dataset with the biggest corresponding eigenvalues. The first 107 eigenvectors cover 100% of the datasets variance.

hidden layers and as the input layer of the network, as the low dimensional projection of the input data denormalizes the data. The reduction of the number of neurons further reduces the number of tunable parameters to 187K. This allows for bigger batch sizes during training, as the original configuration can lead to GPU memory being the restricting factor.

Preuer et al. provide a 5-fold split of the dataset according to the stratified cross-validation scheme, in which each fold contains drug combinations that are not present in any of the other schemes. After separating one fold as test data, Preuer et al. use 5-fold cross-validation on the remaining training data to determine the best hyperparameters. They then train a model on the whole training data for 1000 epochs. It is not clear from the provided source code how the authors determined this number of epochs and as there is no validation data to indicate an overfitting of the model, the question could be raised whether the number of epochs is optimized towards reducing the test error. The final error is reported by averaging the performance on the five test folds.

The training procedure in this work uses the same 5-fold split to separate test data from the dataset. Cross-validation is performed on the remaining four folds and early stopping is implemented to determine how long to train the models. Therefore, an ensemble of four models is trained for each test set. The trained models on each of the four folds are then averaged for performance evaluation on the test data. Using this approach, every model is only trained on three folds (60%) of the data. The early stopping function uses a patience of 50 epochs on an exponentially filtered validation loss with window size 10. That means that training is finished if the filtered validation loss does not drop for 50 consecutive epochs.

The MSE results on the five test sets are [228.9, 347.2, 282.5, 234.6, 219.5] which leads to an average MSE of 262.5 and a confidence interval for one standard deviation of [214.9, 310.1]. The result can be seen in Fig. 4. The model performance could probably be improved by using a random split of training and validation data instead of the provided folds. That way a higher number folds could be used, leading to more training data available per model. While the average result is competitive with the original DeepSynergy work, the standard deviation is higher in this work. This is mostly due to the bad performance on one particular test dataset. Unfortunately, Preuer et al. did not publish the results for all five test datasets, so it can not be evaluated whether their approach generalized better overall. On the other hand the results suggest that the simplified approach in this work performs better on the remaining test datasets. Overall the results are competitive and it is questionable whether the available data actually allows for more precise predictions without incorporating supplementary information.

4. Discussion

The results of this work show that good knowledge of the available dataset is crucial to implementing an efficient

Figure 4. The results for the five test data sets. The model generalizes well on four test data sets but significantly worse on one. Created with Microsoft Bing Calculator.

model. Preuer et al. [12] aggregated features for the drugs and cell lines of the dataset but apparently did not analyze the statistics of the resulting dataset. While the number of features might be of value for bigger datasets, many of them do not add to the expresiveness of the dataset in its current form. Only leaving out the features that are constant for all samples of the dataset would have already lead to a decrease of the imput dimensions by 3920 features. The PCA discussed in Chapter 3.1 shows that most of the remaining features are highly correlated and can therefore be projected to a lower dimensional space.

A recently proposed paper by Zhang et al. [16] improves the DeepSynergy approach by integrating multi-omics data. The authors focus on incorporating additional information about the used cell lines. While their approach improves the predictive performance of the model, they increase the number of input features for a sample of drug A, drug B, and cell line to 93,019. Such a high number of features makes the application of a neural network with as many parameters as the original DeepSynergy network infeasible. While the authors removed features with zero variance, they did not further investigate the independence of input features. This suggests that researchers currently try to incorporate as many features as possible without considering the significance of these features.

Because many features used in the DeepSynergy are computed from the chemical structure of the drugs, future work could try to find a general low-dimensional representation of the features of a chemical structure. Possible approaches are recurrent neural networks in combination with convolutional neural networks. These models could be trained on a large database of chemical compounds and afterwards be used to extract the fundamental features of the drugs of interest.

5. Conclusion

The results of this work show that the performance of the DeepSynergy approach by Preuer et al. [12] can be improved by using principal component analysis on the input dataset. The feature dimensions can be reduced from 12,758 to 107 without losing any information from the dataset. Furthermore, the number of hidden neurons in the neural network can be greatly reduced without sacrificing its predictive performance.

As a result, the number of tunable parameters and therefore the memory consumption of the neural network can be reduced from 138 million to 187K. This leads to faster convergence of the network parameters during training and a shorter training time due to bigger batch sizes.

References

[1] https://opendatascience.com/wp-content/uploads/2017/10/principal_component_analysis_in_r.jpg.

[2] D.-S. Cao, Q. Xu, Q.-N. Hu, and Y.-Z. Liang. Chemopy: Freely available python package for computational biology and chemoinformatics. *Bioinformatics (Oxford, England)*, 29, 03 2013.

[3] Hinselmann Georg, Rosenbaum Lars, Jahn Andreas, Fechner Nikolas, and Zell Andreas. jcompoundmapper: An open source java library and command-line tool for chemical fingerprints. *Journal of cheminformatics*, 3(1):3–3, Jan. 2011.

[4] F. Iorio, T. Knijnenburg, D. Vis, G. R. Bignell, M. Menden, M. Schubert, N. Aben, E. Gonçalves, S. Barthorpe, H. Lightfoot, T. Cokelaer, P. Greninger, E. van Dyk, H. Chang, H. de Silva, H. Heyn, X. Deng, R. K. Egan, Q. Liu, and M. J. Garnett. A landscape of pharmacogenomic interactions in cancer. *Cell*, 166, 07 2016.

[5] D. James. Lecture 4. the singular value decomposition. http://www.cs.cornell.edu/courses/cs322/2008sp/stuff/TrefethenBau_Lec4_SVD.pdf, 2008.

[6] R. Kumar, P. Singh, A. Negi, D. P. Gupta, and M. Chauhan. Toxicophore exploration as a screening technology for drug design and discovery: Techniques, scope and limitations. *Archives of Toxicology*, 90, 08 2015.

[7] A. L Hopkins. Network pharmacology: The next paradigm in drug discovery. *Nature chemical biology*, 4:682–90, 11 2008.

[8] P. Li, C. Huang, Y. Fu, J. Wang, Z. Wu, J. Ru, C. Zheng, Z. Guo, X. Chen, W. Zhou, W. Zhang, Y. Li, J. Chen, A. Lu, and Y. Wang. Large-scale exploration and analysis of drug combinations. *Bioinformatics (Oxford, England)*, 31, 02 2015.

[9] K. P. Murphy. *Machine Learning: A Probabilistic Perspective*. The MIT Press, 2012.

[10] J. O'Neil, Y. Benita, I. Feldman, M. Chenard, B. Roberts, Y. Liu, J. Li, A. Kral, S. Lejnine, A. Loboda, W. Arthur, R. Cristescu, B. B Haines, C. Winter, T. Zhang, A. Bloecher, and S. D Shumway. An unbiased oncology compound screen to identify novel combination strategies. *Molecular Cancer Therapeutics*, 15, 03 2016.

[11] K. Pang, Y.-W. Wan, W. T. Choi, L. A. Donehower, J. Sun, D. Pant, and Z. Liu. Combinatorial therapy discovery using mixed integer linear programming. *Bioinformatics*, 30(10):1456–1463, 2014.

[12] K. Preuer, R. Lewis, S. Hochreiter, A. Bender, K. Bulusu, and G. Klambauer. Deepsynergy: Predicting anti-cancer drug synergy with deep learning. *Bioinformatics (Oxford, England)*, 34, 12 2017.

[13] J. Snoek, H. Larochelle, and R. P. Adams. Practical bayesian optimization of machine learning algorithms.

[14] J. Wildenhain, M. Spitzer, S. Dolma, N. Jarvik, R. White, M. Roy, E. Griffiths, D. S. Bellows, G. D. Wright, and M. Tyers. Prediction of synergism from chemical-genetic interactions by machine learning. *Cell Systems*, 1:383–395, 12 2015.

[15] D. S. Wishart, Y. D. Feunang, A. C. Guo, E. J. Lo, A. Marcu, J. R. Grant, T. Sajed, D. Johnson, C. Li, Z. Sayeeda, N. Assempour, I. Iynkkaran, Y. Liu, A. Maciejewski, N. Gale, A. Wilson, L. Chin, R. Cummings, D. Le, A. Pon, C. Knox, and M. Wilson. Drugbank 5.0: a major update to the drugbank database for 2018. *Nucleic Acids Research*, 46(D1):D1074–D1082, 2018.

[16] T. Zhang, L. Zhang, P. R. Payne, and F. Li. Synergistic drug combination prediction by integrating multi-omics data in deep learning models. https://arxiv.org/abs/1811.07054, November 2018.