An Improved Matrix Completion Algorithm For Categorical Variables: Application to Active Learning of Drug Responses

Huangqingbo Sun HUANGQIS@CS.CMU.EDU
Computational Biology Department
Carnegie Mellon University
Pittsburgh, PA, 15213, USA

Robert F. Murphy MURPHY@CMU.EDU
Computational Biology Department
Department of Biological Sciences
Department of Biomedical Engineering
Machine Learning Department
Carnegie Mellon University
Pittsburgh, PA, 15213, USA

Editor:

Abstract

High throughput screening is extensively used to discover potential drugs in the early drug development process. However, screening is typically used to find compounds that have a desired effect but not to identify potential undesirable side effects because of the large size of the search space. Active machine learning has been proposed as a solution to this problem. In this article, we describe an improved imputation method for modeling the effects of many compounds on many targets using latent correlations between compounds and conditions. Using this to drive active learning in well-characterized settings resulted in a reduction of almost 40% in the number of experiments needed to reach a perfect predictive accuracy compared to random selection of experiments. The results were a significant improvement over previous reports and the new algorithm represents the current state-of-the-art for this problem.

Keywords: Matrix Completion, Active Learning, Drug Discovery, Drug Screening

1. Background

Drug development is an expensive and lengthy process. The effects of a potential drug can be learned by screening on a specific biological target. However, drugs often have unfavorable side effects due to the complex network of interactions within cells and tissues, and these are difficult to discover given the number of other targets that could be affected (Lounkine et al. (2012)). A potential approach is to use active learning, which provides dramatic reductions in the number of required experiments for the larger problem of many drugs and many targets (Naik et al. (2013); Kangas et al. (2014)), and for cases in which the possible phenotypes are not known in advance (Naik et al. (2016)).
The underlying computational problem is matrix completion, the prediction of unmeasured elements in a matrix of drugs and targets. There are two settings in which this may take

place. In the first, numerical features are available that are believed to be accurate descriptors of the properties of both the drugs and the targets (the setting explored by Kangas et al. (2014)). There are a number of approaches for this setting (Chiang et al. (2015); Huang et al. (2017); Wang and Elhamifar (2018)). The second, more difficult setting, is when such features are not available. In this case, imputation must be based upon similarities in the observations among drugs or targets. Within this setting, the observations may either be numerical or categorical. For numerical values, a number of methods have been described (Mazumder et al. (2010); Candes and Plan (2010); Candes and Tao (2010)). For categorical values, alternative methods are needed since there are no mathematical relationships between the categorical variables (Davenport et al. (2014); Cao and Xie (2015)). For the drug-target (or condition-target) problem, Naik et al. (2013, 2016) introduced a clustering-based method which groups independent variables to predict consistent categorical phenotypes. Chen et al. (2020) map the categorical matrix to a corresponding real valued-matrix and solve the real valued-matrix by the SOFT-IMPUTE approach (Mazumder et al. (2010)).
In the current work, we introduce an alternative clustering-based, "lazy learning" method. Rather than using the inductive bias that drugs or targets can be grouped to make predictions for each entire group, our algorithm makes distinct imputations from the relevant measured data for each unmeasured experiment. We compare the performance of the new algorithm on both synthetic and experimental datasets with that of prior algorithms and demonstrate significantly improved performance.

2. Methods

2.1 Problem Definition

Following the problem description in Naik et al. (2013), we use a finite categorical set T that includes all targets under investigation in a particular study, $T = \{t_i, i = 1, 2, \ldots, m\}$, where m is the total number of targets being considered; similarly, C is a finite categorical set $C = \{c_j, j = 1, 2, \ldots, n\}$, where n is the total number of conditions being considered. We define the collection of all the phenotypes possible from any combination of condition and target as $P = \{p_{i,j}\}$, for condition j and target i. Condition and target are independent variables, and phenotype is the dependent variable. The experimental space is $E = T \times C$, $E = \{e_{i,j}\}$ which includes all possible experiments. Here, $e_{i,j}$ refers to the experiment on t_i under c_j. We define a function $P(e) = p$ which maps an element in E to its phenotype.
The problem is to iteratively improve the predictive model for unmeasured experiments in E by using active learning to choose experiments to measure. We use O to describe the set of measured experiments. The premise of the imputation model, as with all matrix completion methods, is that latent correlations exist in the matrix (in our case, within the experimental space). That is, we assume that there is similarity between targets in T and similarity between conditions in C. The similarity between conditions themselves or targets themselves can be estimated if they have co-observed experiments under the same target or condition. That is, we consider two targets to be similar if the phenotypes of the experiments of these two targets under the same condition are the same. The more co-observed experiments have the same phenotype, the higher the similarity between these two targets. Similarly, we can estimate the similarity of two conditions from co-observed experiments on the same target.

2.2 Imputation Model

Here we introduce a committee voting imputation method for making predictions of unobserved e in E. The committee can be interpreted as a partition of E. With the help of lazy learning, a full use of the observed information is achieved and the model is allowed to be robust to the noise in the observed data.
We use the term "condition vector" to refer to the series of experiments under one particular condition, that is $\hat{c_j} = \{e_{1,j}, \cdots, e_{m,j}\}$. Similarly, we define "target vector" as the series of experiments for one particular target, $\hat{v_i} = \{e_{i,1}, \cdots, e_{i,n}\}$. Here, we define the conflict ζ and consistency ρ between 2 vectors $\hat{v_1}$, $\hat{v_2}$ as:

$$\begin{cases} \zeta(\hat{v_1}, \hat{v_2}) = \sum_i \mathbb{I}(\hat{v_{1,i}} \in O)\mathbb{I}(\hat{v_{2,i}} \in O)\mathbb{I}(\hat{P(v_{1,i})} \neq P(\hat{v_{2,i}})) \\ \rho(\hat{v_1}, \hat{v_2}) = \sum_i \mathbb{I}(\hat{v_{1,i}} \in O)\mathbb{I}(\hat{v_{2,i}} \in O)\mathbb{I}(P(\hat{v_{1,i}}) = P(\hat{v_{2,i}})) \end{cases} \tag{1}$$

Here, $I(*)$ is the indicate function.
The algorithms are shown in Algorithm 1 and 2. First, we consider imputation from conditions by calling *impute missing values*($\hat{conditions}$), returning $predictions\_from\_condition$. Here, $\hat{conditions}$ is the set of condition vectors. For a given vector $\hat{v}$, we define the collection of vectors $\hat{v}$ that have n conflicts to be committee $\mathcal{C}_{\hat{v}}(n)$. Imputation for a given vector is done by constructing a committee with similar vectors and then the prediction for each unobserved element in the given vector is chosen by voting among the observed elements in the corresponding position (elements with the same index) of the member vectors $\hat{m}$ in the committee. We define a function to evaluate the similarity of two given vectors that penalizes predictions that are derived from conflicts

$$score(\hat{v_1}, \hat{v_2}) = \rho(\hat{v_1}, \hat{v_2}) - pnl \times \phi \times \zeta(\hat{v_1}, \hat{v_2}) \tag{2}$$

where ϕ is the fraction of experimental space that has been observed. As this penalty increases, we expect the behavior to approach the Greedy Merge algorithms described by Naik et al. (2013). We explored values of this penalty from 0.5 to 5 and observed only small changes (typically differences of only 1 or 2 rounds required to achieve 90% accuracy). We therefore set it to 2.5 for all studies reported here.

Algorithm 1 impute missing values(*vector_set*)

predictions $\leftarrow \emptyset$
for $\hat{v}$ in *vector_set* **do** # *vector_set* can be the set of target vectors or condition vectors in E
  $n_{conflict} \leftarrow 0$
  $U = \{e | e \in \hat{v}, e \notin O\}$
  while $O \neq \emptyset$ **do**
    $\mathcal{C}(n_{conflict}) \leftarrow \{\hat{m_1}, \hat{m_2}, \cdots\}$
    $pred_eF \leftarrow \text{infer}(e, \hat{v}, \mathcal{C}(n_{conflict}))$
    if $pred_e \neq \emptyset$ **then**
      remove e out of U
      $predictions \cup pred_e$
    end if
    $n_{conflict} \leftarrow n_{conflict} + 1$
  end while
end for
return *predictions*

Algorithm 2 infer(e,$\hat{v}$,$\mathcal{C}$)

$pred_e \leftarrow \emptyset$
for $\hat{m}$ in $\mathcal{C}$ **do**
  if $m_* \in O$ **then** # here, $m_*$is the element in $\hat{m}$ with the same index as e in $\hat{v}$
    if (($P(m_*)$,anyscore)$\notin pred_e$ **then** # P(*) is the function mapping e to its phenotype
      $pred_e \cup \{(P(m_*), \text{score}(\hat{v}, \hat{m}))\}$ # "anyscore" can be any value, we only care about $P(m_*)$
    else
      replace ($P(m_*)$,anyscore) with ($P(m_*)$, anyscore + score($\hat{v}, \hat{m}$))
    end if
  end if
end for
return $pred_e$

Symmetrically, imputation can also be done by predicting unobserved experiments from the targets by calling $imputation\ missing\ values(\hat{targets})$, to yield $predictions\_from\_target$. Here, $\hat{targets}$ is the set of target vectors. We then merge the set $predictions\_from\_condition$ and the set $predictions\_from\_target$; the predictive candidate phenotypes for some experiments may appear in both imputation sets, the predictive score for these phenotypes is the sum of the corresponding scores from the two sets. We define Ps_e as the set of all resulting candidate phenotypes for e. For each possible phenotype for each unobserved e, we assign a predictive confidence as

$$prob_{e,p_i} = \frac{score_e(p_i) + (1 - score_e(p_e*))}{\sum_{p \in Ps_e} (score_e(p) + (1 - score_e(p_e*)))} \quad (3)$$

where $p_e* = \arg\min_{p \in Ps_e} score_e(p)$. For each unobserved e, we choose the predicted phenotype to be the candidate phenotype with the highest score.

2.3 Active Learning

For choosing experiments, we used uncertainty sampling. Each predicted phenotype is accompanied by a confidence score. We define two criteria for ranking predictions for unobserved e by their confidence: ranking by the predictive confidence score itself or by the information entropy of the prediction score. We used three uncertainty querying strategies: querying by lowest confidence score, querying by highest entropy, and querying by a hybrid of 50% of selections by uncertainty and 50% by entropy.

3. Results and Discussion

3.1 Numerical Simulation

We performed testing with simulated data following the approach in Naik et al. (2013). This involves using two variables to control the generation process for E. One, "uniqueness" (u), refers to the fraction of unique combinations of phenotypes for a given target or condition; the other, "responsiveness" (r), describes the probability that the phenotype for a given target under a given condition is changed from its unperturbed phenotype. We tested the performance of the model on simulated E with a range of u and r for 100 targets $\times$ 100 conditions and 60 possible phenotypes. We measured differences in the percentage of E need to reach 90% and 100% accuracy between active and random learning. As shown in Figure 1A, the active learning strategy reaches 100% more rapidly than random learning

across most combinations of uniqueness and responsiveness. The same behavior is seen for 90% predictive accuracy. Our new algorithm also performs better than previous algorithms when comparing the number of rounds to reach 90% predictive accuracy (Figure 1C).

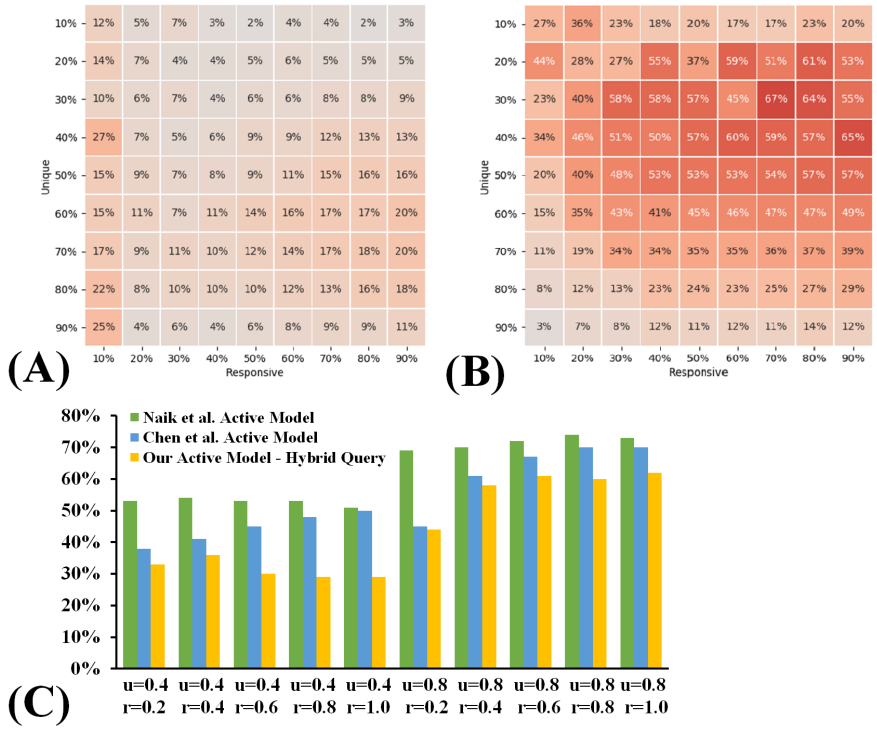

Figure 1: Comparisons of active learning performance. The number and color in the heatmaps indicate the difference between active and random learners in the percentage of experiments needed to reach 90% (A) or 100% (B) predictive accuracy. Both learners used the imputation model described above, and the active leaner used the hybrid query strategy. (C) A comparison of number of rounds to reach 90% accuracy for algorithms from Chen et al. (2020), Naik et al. (2013), and our algorithm on various combinations of uniqueness (u) and responsiveness (r) and 32 phenotypes.

3.2 Protein Subcellular Patterns Screening Experiment

We also evaluated the performance of the various model designs on a high throughput screening image dataset which contains features from fluorescent microscopy images. The experimental setting consists of 47 sub-cellular targets and 46 different conditions. For every unique experiment, we silently make 3 replications to introduce at least some similarities between targets themselves and conditions themselves. The final size of E is therefore 94×92. The experimental procedure is shown in Figure 2A. In order to determine the phenotypes, we use hierarchical clustering with a stop distance threshold of 10 to group the measured experiments, and assign those phenotypes to unmeasured experiments by

a 5-nearest neighbor classifier. After each round of data collection, we first compare the predictions of the model made for unobserved experiments with the ground truth of the E generated for that round (this is necessary because we cannot penalize a learner for not predicting a phenotype that has not yet been observed in any experiment.) Learning curves are shown in Figure 2 for our improved imputation method and the imputation method used by Naik et al. (2016). It is important to note however that our tests were done with only the average feature values for each combination of target and condition, and therefore the learning rates are not directly comparable to those reported previously. The results are also not directly comparable to those of Chen et al. (2020) because they used the phenotypes learned in each round by the original study rather than learning the phenotypes anew only from the data observed up to that round (thus taking advantage of information from experiments that were not observed). Obviously, models with an active learning strategy perform better than random models. Note that the curves of active learners with the hybrid query strategy and entropy query strategy learn more rapidly compared to the previous imputation method (Naik et al. (2016)) in our tests.

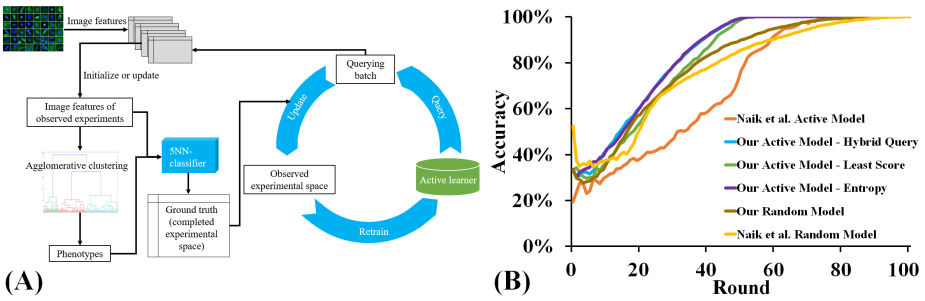

Figure 2: (A). Experiment workflow of the learning process on the protein sub-cellular patterns screening dataset. (B). Performances of learners using our model and that of Naik et al. (2013) are shown for the real image dataset.

4. Conclusion

We have introduced an improved imputation method suitable for learning the latent correlations in a target-condition combination system; further, when used with active learning, it shows superior performance over previous methods both on learning an accurate predictive model while performing fewer experiments and on making more accurate predictions with the same number of performed experiments. Active learning will play an increasingly important role in drug development as the need to consider larger and larger experimental spaces eliminates the possibility of exhaustive experimentation. Effective methods for imputation from limited experimental data are a key component of these efforts.

5. Reproducible Research Archive

The code, test data and results are available at `http://murphylab.cbd.cmu.edu/software/2020_categorical/`.

References

E. J. Candes and T. Tao. The power of convex relaxation: Near-optimal matrix completion. *IEEE Transactions on Information Theory*, 56(5):2053–2080, 2010.

Emmanuel J Candes and Yaniv Plan. Matrix completion with noise. *Proceedings of the IEEE*, 98(6):925–936, 2010.

Y. Cao and Y. Xie. Categorical matrix completion. In *2015 IEEE 6th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, pages 369–372, 2015.

J. Chen, J. Hou, and K. Wong. Categorical matrix completion with active learning for high-throughput screening. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, pages 1–1, 2020.

Kai-Yang Chiang, Cho-Jui Hsieh, and Inderjit S Dhillon. Matrix completion with noisy side information. In *Advances in Neural Information Processing Systems*, pages 3447–3455, 2015.

Mark A Davenport, Yaniv Plan, Ewout Van Den Berg, and Mary Wootters. 1-bit matrix completion. *Information and Inference: A Journal of the IMA*, 3(3):189–223, 2014.

Li Huang, Xianhong Li, Pengfei Guo, Yuhua Yao, Bo Liao, Weiwei Zhang, Fayou Wang, Jiasheng Yang, Yulong Zhao, Hailiang Sun, et al. Matrix completion with side information and its applications in predicting the antigenicity of influenza viruses. *Bioinformatics*, 33 (20):3195–3201, 2017.

Joshua D Kangas, Armaghan W Naik, and Robert F Murphy. Efficient discovery of responses of proteins to compounds using active learning. *BMC bioinformatics*, 15(1):143, 2014.

Eugen Lounkine, Michael J Keiser, Steven Whitebread, Dmitri Mikhailov, Jacques Hamon, Jeremy L Jenkins, Paul Lavan, Eckhard Weber, Allison K Doak, Serge Côté, et al. Large-scale prediction and testing of drug activity on side-effect targets. *Nature*, 486(7403): 361–367, 2012.

Rahul Mazumder, Trevor Hastie, and Robert Tibshirani. Spectral regularization algorithms for learning large incomplete matrices. *Journal of machine learning research*, 11(Aug): 2287–2322, 2010.

Armaghan W Naik, Joshua D Kangas, Christopher J Langmead, and Robert F Murphy. Efficient modeling and active learning discovery of biological responses. *PLoS One*, 8(12), 2013.

Armaghan W Naik, Joshua D Kangas, Devin P Sullivan, and Robert F Murphy. Active machine learning-driven experimentation to determine compound effects on protein patterns. *Elife*, 5:e10047, 2016.

Yugang Wang and Ehsan Elhamifar. High rank matrix completion with side information. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.