The Effectiveness of Multitask Learning for Phenotyping with Electronic Health Records Data

Daisy Yi Ding[1], Chloé Simpson[1], Stephen Pfohl[1], Dave C. Kale[2], Kenneth Jung[1], Nigam H. Shah[1]

[1]*Stanford Center for Biomedical Informatics Research, Stanford University, Stanford, CA;*
[2]*USC Information Sciences Institute, University of Southern California, Marina del Rey, CA*

Electronic phenotyping is the task of ascertaining whether an individual has a medical condition of interest by analyzing their medical record and is foundational in clinical informatics. Increasingly, electronic phenotyping is performed via supervised learning. We investigate the effectiveness of multitask learning for phenotyping using electronic health records (EHR) data. Multitask learning aims to improve model performance on a target task by jointly learning additional auxiliary tasks and has been used in disparate areas of machine learning. However, its utility when applied to EHR data has not been established, and prior work suggests that its benefits are inconsistent. We present experiments that elucidate when multitask learning with neural nets improves performance for phenotyping using EHR data relative to neural nets trained for a single phenotype and to well-tuned baselines. We find that multitask neural nets consistently outperform single-task neural nets for rare phenotypes but underperform for relatively more common phenotypes. The effect size increases as more auxiliary tasks are added. Moreover, multitask learning reduces the sensitivity of neural nets to hyperparameter settings for rare phenotypes. Last, we quantify phenotype complexity and find that neural nets trained with or without multitask learning do not improve on simple baselines unless the phenotypes are sufficiently complex.

Keywords: Electronic Health Records; Electronic phenotyping algorithms; Deep learning; Multi-task learning.

1. Introduction

The goal of electronic phenotyping is to identify patients with (or without) a specific disease or medical condition using their electronic medical records. Identifying sets of such patients (i.e. a patient cohort) is the first step in a wide range of applications such as comparative effectiveness studies,[1,2] clinical decision support,[3,4] and translational research.[5] Increasingly, such phenotyping is done via supervised machine learning methods.[6–8]

Multitask learning (MTL) is a widely used technique in machine learning that seeks to improve performance on a *target task* by jointly modeling the target task and additional *auxiliary tasks*.[9] MTL has been used to good effect in a wide variety of domains including computer vision,[10] natural language processing,[11,12] speech recognition,[13] and even drug development.[14,15] However, its effectiveness using EHR data is less well established, with prior work providing contradictory evidence regarding its utility.[16,17]

In this work, we investigate the effectiveness of MTL for phenotyping using EHR. Our pre-

© 2018 The Authors. Open Access chapter published by World Scientific Publishing Company and distributed under the terms of the Creative Commons Attribution Non-Commercial (CC BY-NC) 4.0 License.

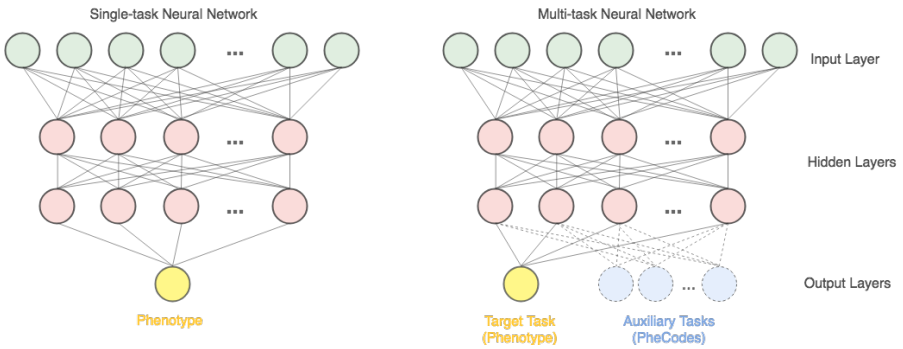

Fig. 1. The architecture of a multitask neural net for electronic phenotyping is shown on the right: the target task (shown in yellow) and the auxiliary tasks (shown in blue) share hidden layers and have distinct output layers; for comparison, we show the corresponding single-task neural net on the left with a single output layer for the target phenotype.

liminary studies recapitulated the inconsistent benefits found in prior work.[16,17] We thus aimed to elucidate the properties of the phenotypes for which MTL helps versus harms performance.

In this paper, we present a systematic exploration of the factors that determine whether or not MTL improves the performance of neural nets for phenotyping with EHR data. Our experiments suggest the following conclusions:

- MTL helps performance for low prevalence (i.e. rare) phenotypes, but harms performance for relatively high prevalence phenotypes. Consistent with some prior work, there is a dose-response relationship with the number of auxiliary tasks, with the magnitude of the benefit or harm generally increasing as auxiliary tasks are added.
- MTL reduces the sensitivity of neural nets to hyperparameter settings. This is of practical importance when one has a limited computational budget for model development.
- Neural nets trained with or without MTL do not improve on simple baselines unless phenotypes are sufficiently complex. However, learning more complex models can be problematic with complex but low prevalence phenotypes. We explore this phenomenon by quantifying phenotype complexity using information theoretic metrics.

2. Background

2.1. *Multitask nets*

Multitask Learning MTL seeks to improve performance on a given target task by jointly learning additional auxiliary tasks. For instance, if the target task is whether or not a patient has type 2 diabetes, one might jointly learn auxiliary tasks such as whether or not the patient has other diseases such as congestive heart failure or emphysema. MTL is most frequently embodied as a neural net in which the earliest layers of the network are shared among the target and auxiliary tasks, with separate outputs for each task (see Figure 1). MTL was originally proposed to improve performance on risk stratification of pneumonia patients by leveraging information in lab values as auxiliary tasks.[9] It has since been used extensively for health care problems such as predicting illness severity[18] and mortality,[17] and disease risk and progression.[19–23] However, the reported benefits of MTL are inconsistent across problems.

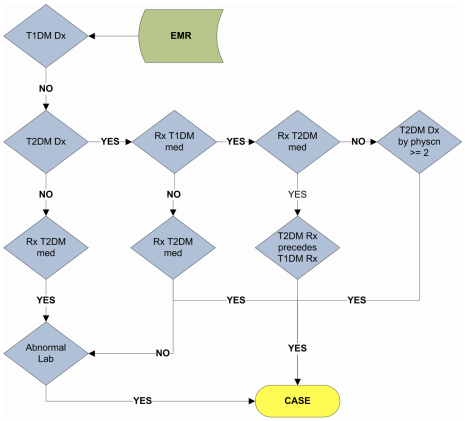

Fig. 2. Rule-based definitions for *Type 2 Diabetes Mellitus* from PheKB.[34]

For example, Che et. al showed that MTL improved performance on identifying physiological markers in clinical time series data,[16] while Nori et. al concluded that MTL failed to improve performance on predicting mortality in an acute care setting.[17] Our aim in this study is to clarify when one might expect MTL to help performance on problems using EHR data. We focus specifically on the foundational problem of phenotyping, which we discuss next.

Electronic Phenotyping In this study, *phenotyping* is simply identifying whether or not a patient has a given disease or disorder. The gold standard for phenotyping remains manual chart review by trained clinicians, which is time-consuming and expensive.[24–26]

This has spurred work on *electronic phenotyping*, which aims to solve the same problem using automated means and EHR data as input. The earliest electronic phenotyping algorithms were rule-based decision criteria created by domain experts.[24–28] Figure 2 shows an example of a rule-based algorithm for type 2 diabetes mellitus. In this approach, identifying patients with the phenotype can be automated once the algorithm is specified, but the latter process is still time consuming and expensive.

More recent work has focused on using statistical learning[6,29–33] to automate the process of specifying the algorithm itself using the methods of machine learning (i.e. models such as logistic regression, random forests, and neural nets). MTL is a particular method for doing this better. Our goal in this work is not to maximize performance for some phenotype but rather to gain insight into when MTL helps versus harms in this approach to phenotyping.

3. Methods

3.1. *Dataset Construction and Design*

Dataset Our data comprises de-identified patient data spanning 2010 through 2016 for 1,221,401 patients from the Stanford Translational Research Integrated Database Environment (STRIDE) database.[35] Each patient's data includes timestamped diagnosis (ICD-9), procedure (CPT), drug (RxNorm) codes, along with demographic information (age, gender, race, and ethnicity). We use a simple multi-hot feature representation whereby each ICD-9, CPT, and RxNorm code is mapped to a binary indicator variable for whether the code occurs in the patient's medical history. We similarly encode gender, race, ethnicity, and each integer

value of age. This process results in a sparse representation of 29,102 features.

Target Task Phenotypes Phenotyping with statistical classifiers is typically framed as a binary classification task, which requires data labeled with whether or not the patient has the phenotype. For this study, we derive the phenotypes using rule-based definitions from PheKB,[36] a compendium of phenotype definitions developed to support genome-wide association studies. We focus on 4 phenotypes, chosen to span a range of prevalences. They are: type 2 diabetes mellitus (T2DM), atrial fibrillation (AF), abdominal aneurysm (AA), and angioedema (AE). The respective prevalences of these phenotypes in our data are 2.95%[a], 2.89%, 0.12%, and 0.08%. We use these rule-based definitions to derive the phenotypes because they are easy to implement, scalable and transparent – later we describe how we take advantage of the rule-based definitions to gain insight into the effectiveness of MTL relative to baselines.

Auxiliary Tasks Our auxiliary tasks are to classify *phecodes*, manually curated groupings of ICD-9 codes originally used to facilitate phenome-wide association studies.[37] We randomly select phecodes with prevalence between 0.08% and 2.95%, i.e. the lowest and the highest target phenotype prevalences, as auxiliary tasks. We conduct binary classification on each phecode and experiment with 5, 10, and 20 randomly selected phecodes as auxiliary tasks.

3.2. *Experimental Design*

We aim to investigate whether and under what circumstances MTL improves performance upon baselines. Recent work suggests that we need to be careful in order to draw robust conclusions on the relative merits of machine learning, especially neural net based methods.[38–41]

First, one typically randomly partitions data into training, validation and test sets. We fit models to the training set, select or tune models using the validation set, and estimate performance on new data using the test set. All three steps use finite samples and are thus subject to noise due to sampling. This is especially true when data exhibit extreme class imbalance, as is the case with our phenotypes. Second, the performance of even simple feed-forward neural nets is known to be sensitive to hyperparameters such as the number of hidden layers and their sizes. Finally, fitting neural nets is inherently stochastic due to random initialization of model parameters and training by some variation of stochastic gradient descent. This, combined with the highly non-convex nature of neural nets, implies that different training runs of a neural net with fixed hyperparameters and dataset splits can still result in widely varying performance.[42]

We thus designed our experiments to mitigate noise due to these factors. First, for each phenotype, we perform ten random splits of the data into training (80%), validation (10%), and test sets (10%). We use stratified sampling to fix the prevalence of the targets to the overall sample prevalence in each of the training, validation and test sets. Second, for each of these splits, we perform a grid search over these hyperparameters for the MTNN and STNN models: we vary the number of hidden layers (1 or 2), their size (128, 256, 512, 1024, and 2048), and the initial learning rate for the algorithm (1e-4 and 5e-5). Moreover, we performed experiments

[a]The prevalence is low compared to the population prevalence of approximately 9% because the rule-based definitions from PheKB are tuned for high precision at the cost of lower recall.

varying the number of auxiliary tasks (in the form of 5, 10, and 20 nested, randomly selected phecodes) for MTNNs by conducting the above grid search for each scenario. For each split, we also fit an L1 regularized logistic regression model, tuned on the validation set. We use the area under the Precision-Recall curve (AUPRC) as our evaluation metric since it can be more informative than the area under the receiver operator characteristic curve (AUROC) in problems with extreme class imbalance.[43]

Phenotype Complexity Our experiments suggested that the complexity of the phenotype is important in whether MTNNs and STNNs outperform well-tuned logistic regression. We quantified the phenotype complexity with regard to a subset of the features upon which the classifiers are built[b]. If we had access to an oracle that told us which features of the patient representation are important in determining a patient's phenotype, we could characterize the complexity of the phenotype with regard to the observed combinations of these features in the positive cases. We could also compare the distributions of the positive and negative cases to examine how difficult it is to discriminate positive and negative cases given the relevant features.

Our phenotypes are derived from the rule-based definitions, which we use as such an oracle: for each phenotype, we extract the features involved in its rule-based definitions (the *oracle features*) and count occurrences of each distinct combination of these features observed in the positive and negative cases. Each unique combination is represented as a binary string with each digit indicating the presence or absence of an oracle feature. Since some of the phenotype definitions involve very many combinations, we hash the combinations into a lower-dimensional space, i.e. a fixed number buckets. Specifically, we use a hash function to map the combinations (the variable-length binary strings) to a fixed number of hash codes (the buckets). We obtain the counts in each bucket for the positive and negative cases and analyze the resulting histograms using two information theoretic metrics.

Let $\mathbf{x}_i$ be the vector of oracle features for bucket i. We summarize the phenotype complexity of positive cases by treating the histogram as a discrete probability distribution and calculate its information entropy,[44] defined as:

$$H(\mathbf{X}) = \mathbb{E}_{\mathbf{x}\sim P}\left[\log(\mathbf{x})\right] = \sum_{i=1}^{n} p(\mathbf{x}_i)\log(\mathbf{x}_i),$$

where n is the number of buckets. This metric summarizes the diversity of positive cases with respect to the oracle features and is higher for more complex phenotypes.

We compare the distributions of the positive and negative cases using the Kullback-Leibler (KL) divergence.[45] For discrete probability distributions P^+ and P^-, the KL divergence from P^- to P^+ is defined as:

$$D_{KL}(P^+ \parallel P^-) = \sum_{i=1}^{n} P^+(\mathbf{x}_i)\frac{P^-(\mathbf{x}_i)}{P^+(\mathbf{x}_i)},$$

where n is the number of buckets[c]. $P^+(\mathbf{x}_i)$ and $P^-(\mathbf{x}_i)$ are the normalized frequencies of bucket

[b]There is no direct way to quantify the complexity of the rule-based definitions shown in Figure 2.
[c]KL divergence does not admit zero probabilities so we use Laplace smoothing on the distributions to deal with combinations that do not have mutual support.

i for cases and controls respectively. KL divergence measures the dissimilarity between the case and control distributions and is lower for the phenotypes that are harder to discriminate.[d]

Neural Net Details All neural nets used ReLU activations[46] for the hidden layers and Xavier initialization[47] and were trained using Adam[48] with standard parameters ($\beta_1 = 0.9$ and $\beta_2 = 0.99$) for 6 epochs[e]. We controlled overfitting with batch normalization and early stopping on the validation set.

4. Experiments and Results

In this section, we present results that provide insight into the following questions:

- When does MTL improve performance relative to single-task models for phenotyping?
- How do the effects of MTL change with the number of phecodes as auxiliary tasks?
- How do the neural net methods compare with strong baseline methods, and what are the characteristics of the tasks for which they provide some benefit?

4.1. *When Does Multitask Learning Improve Performance?*

We investigate the performance of MTNNs over a range of hyperparameter settings and over multiple random splits of the data. MTNN performance is compared to the performance of STNNs over the same hyperparameter settings and data splits. Figure 3 shows the optimal MTNN and STNN performance achieved on each split for the four phenotypes. We find that MTNNs consistently outperform STNNs for the low prevalence phenotypes, i.e. angioedema and abdominal aneurysm. In contrast, MTL harms performance for the relatively high-prevalence phenotypes, i.e. T2DM and atrial fibrillation. The left plot in Figure 4 shows the pairwise differences between MTNN and STNN optimal performance across the splits.

Moreover, the performance of STNNs is very sensitive to hyperparameter settings for the low prevalence phenotypes, as illustrated by the large spread in AUPRC values (see Figure 3). In contrast, MTNNs are more robust to hyperparameter settings for these phenotypes. In practice, tuning neural nets is time-consuming and finding an ideal model demands extensive computation. MTL may increase our chance of finding a reasonable model, which is of practical value when one has a limited computational budget on model space exploration.

4.2. *Relationship Between Performance and Number of Tasks*

We investigate how MTL is influenced by the number of auxiliary tasks as defined in the form of phecodes. We trained MTNNs with nested sets of 5, 10, and 20 randomly selected phecodes (i.e. the 5-phecode set is a subset of the 10-phecode set, and so on), and reported the performance with the optimal hyperparameter setting for each split. The right plot in Figure 4 shows pairwise differences in AUPRC values between MTNNs and STNNs. For the low prevalence phenotypes, more phecodes increases performance gains. Similarly, more phecodes

[d]Please refer to https://arxiv.org/abs/1808.03331 for a more detailed description of our method.
[e]We found 6 epochs was sufficient for all models to converge.

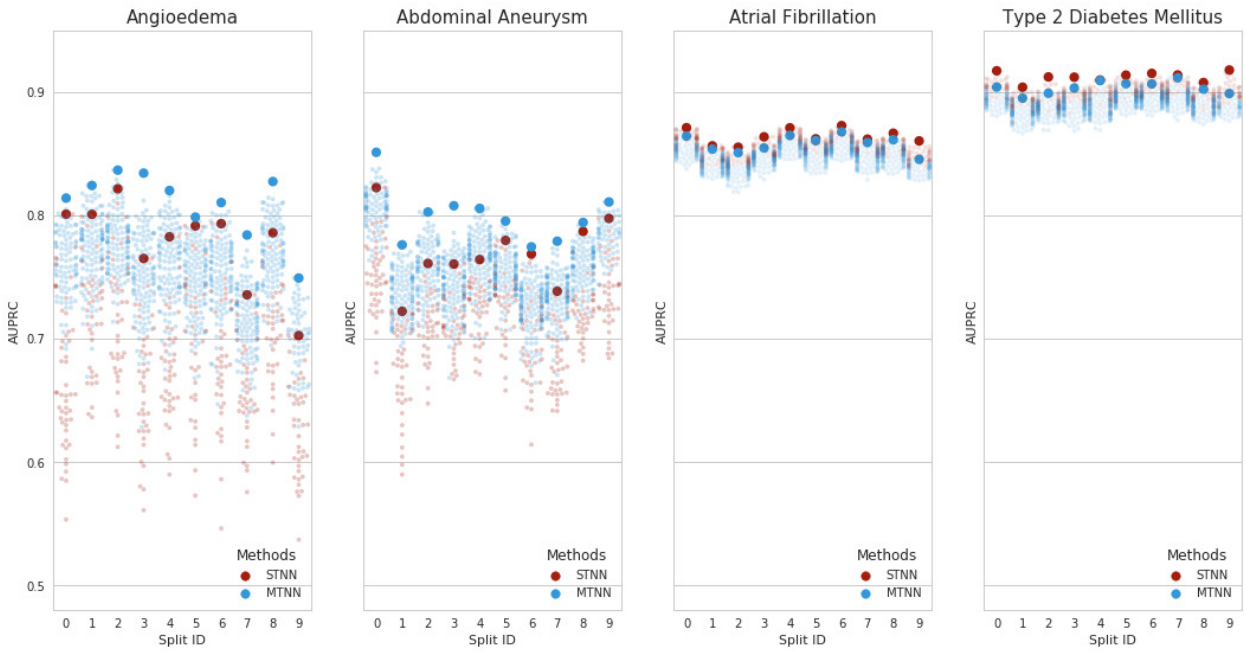

Fig. 3. MTNN and STNN performance for Angioedema, Abdominal Aneurysm, Atrial Fibrillation, and Type 2 Diabetes Mellitus with various hyperparameter settings across the ten splits; the best case MTNN and STNN performance is emphasized by the solid dots: the blue and red dots correspond to MTNNs and STNNs respectively.

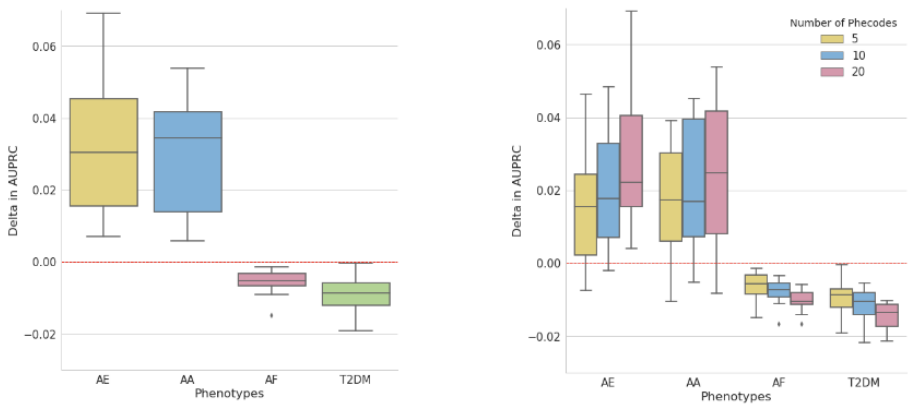

Fig. 4. The left plot shows the pairwise differences in AUPRC values of the optimal MTNNs and STNNs for Angioedema, Abdominal Aneurysm, Atrial Fibrillation, and Type 2 Diabetes Mellitus across the ten splits. The right plot shows the pairwise differences in AUPRC values of the optimal STNNs and MTNNs with different number of phecodes as auxillary tasks.

for high prevalence phenotypes leads to more severe negative effects, though the scale of the negative effects is smaller than the positive effects for low prevalence phenotypes[f].

[f]This dose-response relationship with the number of auxiliary tasks recapitulates the findings of Ramsundar et al,[14] but we find the relationship holds for both the benefit and harm of MTL.

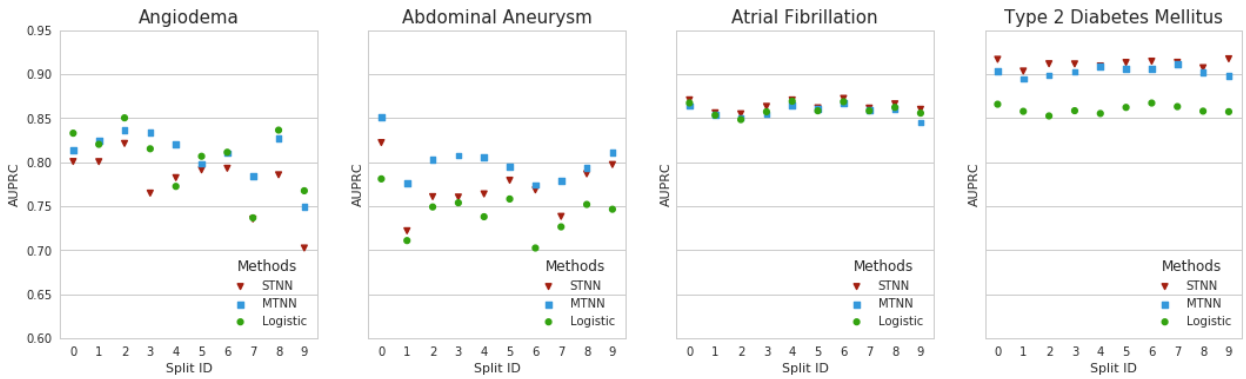

Fig. 5. MTNN, STNN, and LR optimal performance for Angioedema, Abdominal Aneurysm, Atrial Fibrillation, and Type 2 Diabetes Mellitus across splits: the blue squares, the red triangles, and the green dots correspond to MTNN, STNN, and LR respectively.

4.3. *Comparison with Logistic Regression Baseline*

In discussing the merits of MTL, it is important to also compare the performance against simpler baseline methods in addition to single-task neural nets. We compare the performance of the neural nets with L1 regularized logistic regression (LR), a consistently strong baseline for EHR data[49,50] (see Figure 5). LR is consistently outperformed by the neural nets for abdominal aneurysm and type 2 diabetes mellitus, which are low and high prevalence respectively. For angioedema, a low prevalence phenotype, performance relative to LR is inconsistent across the splits, although MTNNs consistently beat STNNs. And for atrial fibrillation, a high prevalence phenotype, MTNNs and STNNs provide little or no benefit over LR. Prevalence alone is insufficient to account for the relative performance between both MTNN and STNN and LR.

4.4. *Interaction between Phenotype Prevalence and Complexity*

Our comparison of MTNNs and STNNs versus LR suggests that phenotype prevalence alone cannot explain when neural nets outperform simpler linear models. We hypothesized that phenotype complexity also plays a role since neural nets with or without MTL can automatically model non-linearities and interactions, while LR must have non-linearities and interactions explicitly encoded in features. We leveraged the rule-based phenotype definitions to explore this hypothesis and found evidence of an interaction between phenotype prevalence and complexity.

Phenotype Complexity For each phenotype, we generated histograms of the observed combinations of the oracle features for the positive and negative cases (see Figure 6) and calculated the information entropy of the positive cases and the KL divergence between the positive and negative cases (see Table 1) as described in Methods 3.2.

We find that atrial fibrillation, a high-prevalence phenotype, has low entropy and high KL divergence. With respect to the oracle features, all the positive cases are similar to each other, while the positive and negative cases are very dissimilar to each other. A relatively simple model should be able to capture this, explaining the observation that LR achieves comparable

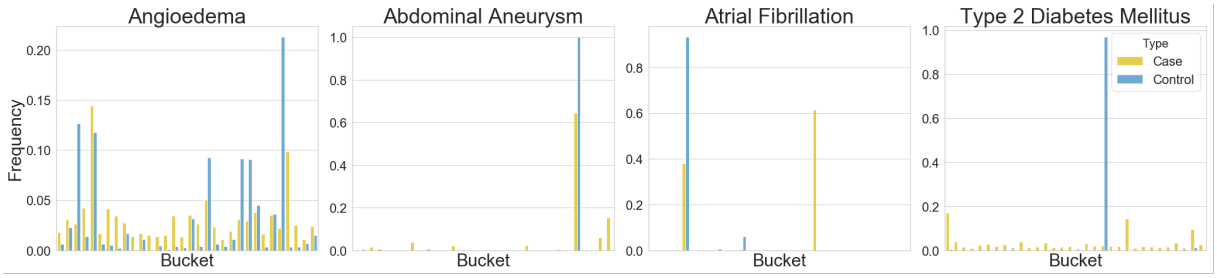

Fig. 6. Distributions of the combinations of the oracle features involved in the rule-based definitions for Angioedema, Abdominal Aneurysm, Atrial Fibrillation, and Type 2 Diabetes Mellitus. The yellow and blue bars correspond to the positive and negative cases respectively. The x-axes represent the buckets of unique combinations of the oracle features: in our study, we use 32 buckets. Note that the choice of 32 buckets was arbitrary and not tuned in any way.

Table 1. Phenotype Complexity

Phenotype	Prevalence	Entropy	KL Divergence
Angioedema	0.08 %	3.233	0.930
Abdominal Aneurysm	0.12%	1.396	2.414
Atrial Fibrillation	2.89%	0.709	5.383
Type 2 Diabetes Mellitus	2.95 %	3.012	3.806

performance to MTNNs and STNNs for this phenotype.

Abdominal aneurysm, a low prevalence phenotype, and T2DM, a high prevalence phenotype, have higher information entropy and lower KL divergence values than atrial fibrillation. Thus, the positive cases are more diverse and discrimination is more difficult than atrial fibrillation with respect to each phenotype's oracle features. For these phenotypes, both MTNNs and STNNs outperform LR – we benefit from more expressive models. However, whether MTNNs beat STNNs depends on prevalence.

Finally, angioedema has the highest entropy and lowest KL divergence – it is both the most complex and hardest to discriminate of the four phenotypes. Complex phenotypes should benefit from more expressive models. However, we observe that while MTNNs consistently outperform STNNs, their performance relative to LR is inconsistent across splits. One possible explanation for this behavior is that relative performance is sensitive to the assignment of patients to training, validation and test sets: with such diverse cases and common support with respect to the oracle features, it is much more likely for the test set to contain patients unlike any seen in the training set.

5. Limitations

We have set out to investigate MTL and its effectiveness for electronic phenotyping. However, our work has important limitations. First, we randomly select phecodes for auxiliary tasks, but it has been argued that auxiliary tasks should be directly related to the target task.[51] It is possible that better auxiliary tasks would improve the benefit of MTL. Specifically, more related phecodes might mitigate or eliminate the performance degradation observed for the

high-prevalence phenotypes or inconsistent relative performance between MTNN and LR for angioedema. However, the notion of task relatedness is underspecified so it is problematic to compute in order to select auxiliary tasks. Indeed, in preliminary work we explored various formulations of relatedness to select auxiliary tasks but found that none performed better than random selection. One could ask domain experts to manually construct or pick auxiliary tasks for specific phenotypes, but this is beyond the scope of this work. Moreover, it has also been shown that the task relatedness is unnecessary for MTL to provide benefits.[52] However, we acknowledge that it is an interesting line of inquiry for future work to further explore how to improve multitask learning for electronic phenotyping. Second, to address the unavailability of large-scale ground truth phenotypes, we use rule-based definitions because they are transparent and available, but we recognize that the phenomenon we observe may be artifacts of the rule-based definitions. We also acknowledge the possibility that the observed phenomenon might not generalize to other phenotypes; we focused on four phenotypes to conduct an in-depth examination, sacrificing breadth. Finally, the rule-based phenotype definitions contain predicates encoding temporal relationships, e.g., a drug code followed by a diagnosis code. Our simple multi-hot feature representation does not encode temporal information. As a result, there is an upper bound on the performance of any statistical classifier using this feature representation.

6. Conclusion

We have investigated the effectiveness of multitask learning on electronic phenotyping with EHR data, aiming to elucidate the properties of situations for which MTL improves or harms performance. We trained multitask neural networks to classify a target phenotype jointly with auxiliary tasks drawn from phecodes. We found that MTL provided consistent performance improvements over single-task neural networks on extremely rare phenotypes. However, for relatively higher prevalence phenotypes, MTL actually reduced performance. In both cases, the effect scaled with the number of auxiliary tasks as defined in the form of phecodes. Moreover, we found that MTL improved the robustness of neural networks to hyperparameter settings for the extremely rare phenotypes, which is of practical value in situations when one has a limited computational budget for model exploration. Finally, we analyzed phenotype complexity to shed light on the relative performance of both MTNN and STNN versus well-tuned L1 regularized logistic regression baselines and found evidence of an interaction between phenotype prevalence and complexity. We showed that simple linear models are sufficient for non-complex phenotyping tasks. More expressive models can substantially improve performance for more complex phenotypes, but only if the data support learning them well, which may be problematic for rare phenotypes.

Acknowledgments

This work was supported by NLM R01-LM011369-05 and a grant supporting the Observational Health Data Science and Informatics (OHDSI) by Janssen Research and Development LLC. Internal funding by the School of Medicine at Stanford also supported part of this work. We gratefully acknowledge Jason Fries for many helpful discussions about this work.

References

1. D. C. Crawford, D. R. Crosslin, G. Tromp, I. J. Kullo, H. Kuivaniemi, M. G. Hayes, J. C. Denny, W. S. Bush, J. L. Haines, D. M. Roden *et al.*, *Frontiers in Genetics* **5**, p. 184 (2014).
2. F. J. Manion, M. R. Harris, A. G. Buyuktur, P. M. Clark, L. C. An and D. A. Hanauer, *Current Oncology Reports* **14**, 494 (2012).
3. C. A. Longhurst, R. A. Harrington and N. H. Shah, *Health Affairs* **33**, 1229 (2014).
4. W.-Q. Wei and J. C. Denny, *Genome Medicine* **7**, p. 41 (2015).
5. N. H. Shah, *Nature Biotechnology* **31**, p. 1095 (2013).
6. V. Agarwal, T. Podchiyska, J. M. Banda, V. Goel, T. I. Leung, E. P. Minty, T. E. Sweeney, E. Gyang and N. H. Shah, *Journal of the American Medical Informatics Association* **23**, 1166 (2016).
7. Y. Halpern, S. Horng and D. Sontag, *Proceedings of the 1st Machine Learning in Health Care (MLHC)* , p. 209 (2016).
8. J. Banda, Y. Halpern, D. Sontag and N. Shah, *AMIA Summits on Translational Science Proceedings* , p. 48 (2017).
9. R. Caruana, S. Baluja and T. M. Mitchell, *Advances in Neural Information Processing Systems* , 959 (1995).
10. R. Girshick, J. Donahue, T. Darrell and J. Malik, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* , 580 (2014).
11. B. Plank, A. Søgaard and Y. Goldberg, *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics* (2016).
12. P. Liu, X. Qiu and X. Huang, *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics* , 1 (2017).
13. S. Toshniwal, H. Tang, L. Lu and K. Livescu, *18th Annual Conference of the International Speech Communication Association* , 3532 (2017).
14. B. Ramsundar, S. Kearnes, P. Riley, D. Webster, D. Konerding and V. Pande, *arXiv preprint arXiv:1502.02072* (2015).
15. P. Zhang, F. Wang and J. Hu, *AMIA Annual Symposium Proceedings* **2014**, p. 1258 (2014).
16. Z. Che, D. Kale, W. Li, M. T. Bahadori and Y. Liu, *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* , 507 (2015).
17. N. Nori, H. Kashima, K. Yamashita, H. Ikai and Y. Imanaka, *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* , 855 (2015).
18. M. Ghassemi, M. A. F. Pimentel, T. Naumann, T. Brennan, D. A. Clifton, P. Szolovits and M. Feng, *Proceedings of the 29th Conference on Artificial Intelligence* , 446 (2015).
19. C. Ngufor, S. Upadhyaya, D. Murphree, D. Kor and J. Pathak, *IEEE International Conference on Data Science and Advanced Analytics* , 1 (2015).
20. X. Wang, F. Wang, J. Hu and R. Sorrentino, *AMIA Annual Symposium Proceedings* **2014**, p. 1180 (2014).
21. N. Razavian, J. Marcus and D. Sontag, *Machine Learning for Healthcare Conference* , 73 (2016).
22. J. Zhou, L. Yuan, J. Liu and J. Ye, *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* , 814 (2011).
23. Z. C. Lipton, D. C. Kale, C. Elkan and R. Wetzel, *arXiv preprint arXiv:1511.03677* (2015).
24. K. M. Newton, P. L. Peissig, A. N. Kho, S. J. Bielinski, R. L. Berg, V. Choudhary, M. Basford, C. G. Chute, I. J. Kullo, R. Li *et al.*, *Journal of the American Medical Informatics Association* **20**, e147 (2013).
25. C. L. Overby, J. Pathak, O. Gottesman, K. Haerian, A. Perotte, S. Murphy, K. Bruce, S. Johnson, J. Talwalkar, Y. Shen *et al.*, *Journal of the American Medical Informatics Association* **20**, e243 (2013).

26. H. Mo, W. K. Thompson, L. V. Rasmussen, J. A. Pacheco, G. Jiang, R. Kiefer, Q. Zhu, J. Xu, E. Montague, D. S. Carrell *et al.*, *Journal of the American Medical Informatics Association* **22**, 1220 (2015).
27. A. N. Kho, J. A. Pacheco, P. L. Peissig, L. Rasmussen, K. M. Newton, N. Weston, P. K. Crane, J. Pathak, C. G. Chute, S. J. Bielinski *et al.*, *Science Translational Medicine* **3**, 79re1 (2011).
28. M. Conway, R. L. Berg, D. Carrell, J. C. Denny, A. N. Kho, I. J. Kullo, J. G. Linneman, J. A. Pacheco, P. Peissig, L. Rasmussen *et al.*, *AMIA Annual Symposium Proceedings* **2011**, p. 274 (2011).
29. Y. Huang, P. McCullagh, N. Black and R. Harper, *Artificial Intelligence in Medicine* **41**, 251 (2007).
30. Y. Chen, J. Ghosh, C. A. Bejan, C. A. Gunter, S. Gupta, A. Kho, D. Liebovitz, J. Sun, J. Denny and B. Malin, *Journal of Biomedical Informatics* **55**, 82 (2015).
31. J. Zhou, F. Wang, J. Hu and J. Ye, *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* , 135 (2014).
32. J. C. Ho, J. Ghosh and J. Sun, *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* , 115 (2014).
33. Y. Halpern, S. Horng, Y. Choi and D. Sontag, *Journal of the American Medical Informatics Association* **23**, 731 (2016).
34. Type 2 diabetes mellitus. `https://phekb.org/phenotype/18`, Accessed: 2018-07-23.
35. H. J. Lowe, T. A. Ferris, P. M. Hernandez and S. C. Weber, *American Medical Informatics Association Annual Symposium* (2009).
36. J. C. Kirby, P. Speltz, L. V. Rasmussen, M. Basford, O. Gottesman, P. L. Peissig, J. A. Pacheco, G. Tromp, J. Pathak, D. S. Carrell *et al.*, *Journal of the American Medical Informatics Association* **23**, 1046 (2016).
37. W.-Q. Wei, L. A. Bastarache, R. J. Carroll, J. E. Marlo, T. J. Osterman, E. R. Gamazon, N. J. Cox, D. M. Roden and J. C. Denny, *PloS One* **12**, p. e0175508 (2017).
38. Y. Li, N. Du and S. Bengio, *arXiv preprint arXiv:1708.00065* (2017).
39. G. Melis, C. Dyer and P. Blunsom, *arXiv preprint arXiv:1707.05589* (2017).
40. M. Lucic, K. Kurach, M. Michalski, S. Gelly and O. Bousquet, *arXiv preprint arXiv:1711.10337* (2017).
41. A. Oliver, A. Odena, C. Raffel, E. D. Cubuk and I. J. Goodfellow, *arXiv preprint arXiv:1804.09170* , 1 (2018).
42. N. S. Keskar, D. Mudigere, J. Nocedal, M. Smelyanskiy and P. T. P. Tang, *arXiv preprint arXiv:1609.04836* (2016).
43. T. Saito and M. Rehmsmeier, *PloS One* **10**, p. e0118432 (2015).
44. C. E. Shannon, *ACM SIGMOBILE Mobile Computing and Communications Review* **5**, 3 (2001).
45. S. Kullback and R. A. Leibler, *The Annals of Mathematical Statistics* **22**, 79 (1951).
46. V. Nair and G. E. Hinton, 807 (2010).
47. X. Glorot and Y. Bengio, *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics* , 249 (2010).
48. D. P. Kingma and J. Ba, *arXiv preprint arXiv:1412.6980* (2014).
49. A. Rajkomar, E. Oren, K. Chen, A. M. Dai, N. Hajaj, M. Hardt, P. J. Liu, X. Liu, J. Marcus, M. Sun *et al.*, *Digital Medicine* **1**, p. 18 (2018).
50. N. Razavian, S. Blecker, A. M. Schmidt, A. Smith-McLallen, S. Nigam and D. Sontag, *Big Data* **3**, 277 (2015).
51. R. Caruana, *Machine learning* **28**, 41 (1997).
52. B. Romera-Paredes, A. Argyriou, N. Berthouze and M. Pontil, *Proceedings of the 15th International Conference on Artificial Intelligence and Statistics* , 951 (2012).