High-Performing Machine Learning Models for Phenotype Development

Victor A. Rodriguez*, MPhil, **Tony Y. Sun***, BS, **Phyllis M. Thangaraj***, MPhil, **Karthik Natarajan, PhD, Patrick Ryan, PhD** (* Contributed equally)

Department of Biomedical Informatics, Columbia University

1 Introduction

- Phenotyping algorithms are essential tools for conducting clinical research on observational data, as high-thoroughput phenotype development remains an open challenge.
- **eMERGE (electronic Medical Records and Genomics) phenotypes** are manually defined by the eMERGE Network of clinicians and informaticians in terms of multiple clinical concepts, or **concept sets**.
- **We propose a framework for learning from the structure of eMERGE phenotype concept sets** to aid **construction of novel concept set phenotype definitions.**

1.1 Concept Sets → Concept Graph

- We study the structure of eMERGE phenotype concept sets by considering **all possible concept-concept pairs within a concept set**.
- If we treat **concepts as nodes**, and **concept-concept pairs as edges**, then a **concept set is a fully connected subgraph, or clique**, within a larger **concept graph**.
- From this perspective, constructing a phenotype is equivalent to constructing a concept clique.
- We learn how to build concept set cliques by first learning how to predict concept-concept pair edges using rich edge features.

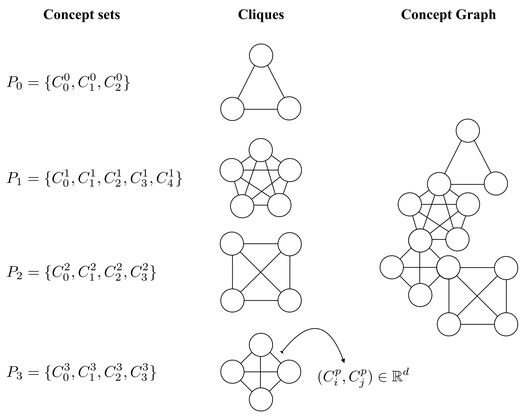

Figure 1: eMERGE phenotypes form cliques within a concept graph. We study concept set construction by learning how to predict concept-concept pair edges using rich edge features. Here P_i indicates the i^{th} eMERGE phenotype. C_j^i indicates the j^{th} concept in the i^{th} concept set. (C_i^p, C_j^p) is a concept-concept pair represented by features in $\mathbb{R}^d$.

2 Methods

- We train several models to perform edge prediction within our concept graph.
- Our dataset is comprised of concept-concept pairs (edges), each described by a rich feature vector (See Figures 1 and 2)
- **All edges appearing within at least one concept set** are treated as **positive instances**; **all other edges** are considered **possible negative instances**.

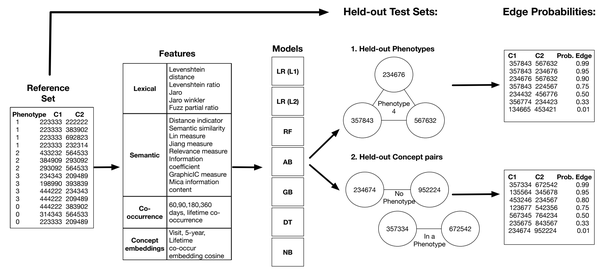

Figure 2: Methods visual summary. C1 and C2 represent concept 1 and concept 2 pairs, Model abbreviations: LR (L1)- Logistic Regression with L1 penalty, LR (L2)- Logistic Regression with L2 penalty, RF- Random Forest, AB- Adaboost, GB- Gradient Boosting, DT- Decision Tree, and NB- Naive Bayes.

2.1 Concept-Concept Pair Features

- A set of rich features was developed for each possible concept-concept pair.
- These include **lexical**, **semantic**, **co-occurence**, and **concept embedding** features.

2.2 Training & Test Set Splits

- We generate training and test partitions in two ways: **random** & **phenotype aware**.
  - **Random**:
    1. Randomly select 90% positive edges for training; remaining 10% used for testing.
    2. Sample negative edges to equal number of positive edges in training and test sets.
  - **Phenotype Aware**:
    1. Select concept sets comprising $\sim 10\%$ of total positive edges; use for testing; remaining positive edges used for training.
    2. Sample negative edges to equal number of positive edges in training and test sets.

2.3 Models

- We train several simple and ensemble binary classifiers for the purpose of edge prediction (See Figure 2 for a full list; all models implemented in `scikit-learn`.).

3 Results

- Concept-concept pair prediction results for all models are show in Table 1
- For each model we evaluate the areas under the ROC & PR curves (AUROC, AUPRC), the maximum F1 (Max. F1) and the Precision at 50% of the test set (Prec.@50%).

Train/Test Split	Model	AUROC	AUPRC	Max. F1	Prec.@50%
Random	LR (L1)	0.9417 ± 0.0008	0.9337 ± 0.0014	0.8701 ± 0.0011	0.9700 ± 0.0205
	LR (L2)	0.9402 ± 0.0009	0.9321 ± 0.0015	0.8700 ± 0.0010	0.9560 ± 0.0377
	Naive Bayes	0.9318 ± 0.0009	0.9202 ± 0.0014	0.8436 ± 0.0024	0.9620 ± 0.0316
	Decision Tree	0.9448 ± 0.0009	0.9224 ± 0.0030	0.8838 ± 0.0008	0.9700 ± 0.0241
	Random Forest	0.9507 ± 0.0007	0.9424 ± 0.001	0.8848 ± 0.0018	0.9720 ± 0.0204
	Gradient Boosting	$\mathbf{0.9540 \pm 0.0008}$	0.9430 ± 0.0017	$\mathbf{0.8909 \pm 0.0013}$	0.9780 ± 0.0166
	AdaBoost	0.9516 ± 0.0007	$\mathbf{0.9431 \pm 0.0013}$	0.8840 ± 0.0010	$\mathbf{0.9800 \pm 0.0179}$
Phen. Aware	LR (L1)	0.8635 ± 0.0293	0.8694 ± 0.0254	0.8030 ± 0.0223	0.9700 ± 0.0257
	LR (L2)	0.8632 ± 0.0331	0.8771 ± 0.0262	0.8005 ± 0.0252	0.9580 ± 0.0166
	Naive Bayes	0.8691 ± 0.0290	0.8731 ± 0.0232	0.7659 ± 0.0509	0.9460 ± 0.0156
	Decision Tree	0.8853 ± 0.0261	0.8698 ± 0.0257	0.8179 ± 0.0133	0.9340 ± 0.0559
	Random Forest	$\mathbf{0.8953 \pm 0.0234}$	0.8833 ± 0.0213	$\mathbf{0.8314 \pm 0.0101}$	0.8340 ± 0.2057
	Gradient Boosting	0.8924 ± 0.0241	$\mathbf{0.8856 \pm 0.0232}$	0.8204 ± 0.0116	0.9440 ± 0.0280
	AdaBoost	0.8704 ± 0.0273	0.8651 ± 0.0259	0.8024 ± 0.0189	$\mathbf{0.9780 \pm 0.0140}$

Table 1: Concept-concept pair prediction evaluation

4 Ongoing Work – Recovering Concept Sets

- Our overall goal is to leverage concept-concept pair prediction models to inform construction of phenotype concept sets.
- We experiment with two algorithms to grow a "seed" of concepts within our concept graph to recover held-out eMERGE phenotypes in our phenotype-aware test sets (See Figures 3 & 4 and Table 2).

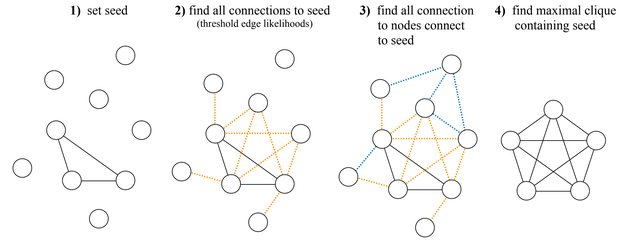

Figure 3: Phenotype recovery algorithm 1.

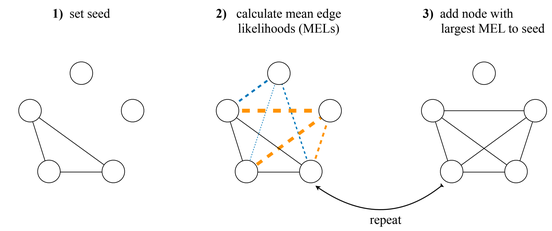

Figure 4: Phenotype recovery algorithm 2.

Seed size	Precision @ 100%	Proportion @ 100%
10%	0.631 ± 0.291	15.82 ± 30.36
20%	0.664 ± 0.278	9.599 ± 22.46
30%	0.694 ± 0.269	7.117 ± 19.15
40%	0.712 ± 0.270	5.289 ± 17.89
50%	0.721 ± 0.276	4.784 ± 17.70
60%	0.745 ± 0.277	4.057 ± 14.81
70%	0.768 ± 0.268	3.661 ± 12.75
80%	0.806 ± 0.253	2.147 ± 8.179
90%	0.842 ± 0.282	1.056 ± 0.297

Table 2: Concept set recovery using algorithm 2 and LR (L1) edge likelihoods

5 Conclusion

- Simple and ensemble binary classifiers are capable of faithfully predicting held-out concept-concept pairs.
- This is true even when using phenotype aware train-test splits, suggesting utility in predicting concept-concept pairs for novel concept set construction.
- A simple, greedy algorithm for recovering held-out concept sets performs reasonably well, providing a path forward to high-throughput concept set construction.