Exploring Label Dependency in Active Learning for Phenotype Mapping

Shefali Sharma[1], **Leslie Lange**[2], **Jose Luis Ambite**[1], **Yigal Arens**[1], **Chun-Nan Hsu**[1,3]
[1]Information Sciences Institute, University of Southern California, Marina del Rey, CA 90292, USA
[2]Department of Genetics, University of North Carolina, Chaple Hills, NC 27599, USA
[3]Institute of Information Sciences, Academia Sinica, Taipei 115, Taiwan
chunnan@isi.edu

Abstract

Many genetic epidemiological studies of human diseases have multiple variables related to any given phenotype, resulting from different definitions and multiple measurements or subsets of data. Manually mapping and harmonizing these phenotypes is a time-consuming process that may still miss the most appropriate variables. Previously, a supervised learning algorithm was proposed for this problem. That algorithm learns to determine whether a pair of phenotypes is in the same class. Though that algorithm accomplished satisfying F-scores, the need to manually label training examples becomes a bottleneck to improve its coverage. Herein we present a novel active learning solution to solve this challenging phenotype-mapping problem. Active learning will make phenotype mapping more efficient and improve its accuracy.

1 Introduction

Phenotypes are observable traits of an individual organism resulting from the presence and interaction of its genotype with the environment. Phenotypes potentially related to human health are of interest in genetics and epidemiology, including common clinical conditions, inheritance disorders, as well as various risk factors such as diet. Substantial amounts of genomic data, including genome-wide genotyping from GWAS (Genome-Wide Association Studies) (Hardy and Singleton, 2009; Consortium, 2007) and sequencing, are being produced in conjunction with the collection of carefully defined and measured phenotypes to study the role of genetic variations in a wide variety of inherited traits and disorders for many decades.

Recently, there is an emerging need to re-use these valuable phenotype-genotype association data to boost the statistical power and improve sensitivity and specificity of the search of associations between various disorders and genetic variations. New paradigms of genomic studies may be fostered once a map of related phenotypes is easily accessible. In fact, one of such new paradigms, PheWAS (Phenome Wide Association Studies), has been developed and producing interesting findings (Denny et al., 2010; Pendergrass et al., 2011) with the help of phenotype mapping and harmonization. Unlike GWAS, which focus on calculating the association between the variation of hundreds of thousands of genotyped single nucleotide polymorphisms (SNPs) and a single or small number of phenotypes, PheWAS uses an extensive range of detailed phenotypic measurements for comprehensively exploring the association between genetic variations and phenotypes. The investigation of a broad range of phenotypes has the potential to identify pleiotropy, reveal novel mechanistic insights, generate new hypotheses, and define a more complete picture of genetic variations and their impact on human diseases.

To facilitate integration of genomic data sets, the research community needs to categorize comparable phenotype measurements and match them across multiple genomic studies to identify data sets of interest as well as potential future collaborations. While the naming strategy for genetic variants is

Proceedings of the 2012 Workshop on Biomedical Natural Language Processing (BioNLP 2012), pages 146–154,
Montréal, Canada, June 8, 2012.

largely standardized across studies (e.g. *rs* numbers for single nucleotide polymorphisms or SNPs), this is often not the case for phenotype variables. Due to the lack of a standardized terminologies or other controlled vocabularies, it becomes increasingly difficult to find studies with comparable phenotypes as the genomic data accumulate. A researcher searching for the availability of comparable phenotypes across multiple studies is confronted with a veritable mountain of variables to sift through. Even within a study, there are often numerous versions of semantically equivalent phenotypic variables. Manually mapping and harmonizing these phenotypes is a time-consuming process that may still miss the most appropriate variables.

Previously, (Hsu et al., 2011) have developed a supervised learning algorithm that learns to determine whether a pair of phenotypes is semantically related from their descriptors. Though that algorithm accomplished satisfying F-scores, the need to manually label training examples becomes a bottleneck to improve its coverage. Moreover, the algorithm treats each pair independently, but pairs that consist of common phenotypes are not independent. Exploring this dependency may potentially improve its performance. In this paper, we investigate how to apply active learning to solve this challenging phenotype-mapping problem. Application of effective active learning techniques will make phenotype mapping more efficient and improve its accuracy and, along with intuitive phenotype query tools, would provide a major resource for researchers utilizing these genomic data.

Active learning queries a user for labels of unlabeled phenotypes that may improve the learning of phenotype mapping the most and thereby reduce the need of labeling efforts. To select the most useful training examples to query, different selection strategies have been proposed in the past (Settles, 2010):

- **Uncertainty Sampling** In this strategy, an active learner chooses an instance that is the most uncertain for the current model to label (Lewis and Catlett, 1994).

- **Query-By-committee** This strategy (Seung et al., 1992) is also known as **maximum disagreement** (Ayache and Quénot, 2007; Di and Crawford, 2011) because the idea is to choose an instance for which a committee of models disagrees the most among its members about its label.

- **Expected Model Change** The general principle of this strategy is to choose an instance to query when if its label is available, the model will be changed the most (Settles and Craven, 2008).

- **Expected Error Reduction** Active learning is useful when the selected instance reduce the error the most and this strategy looks for an instance that can achieve this ultimate goal directly.

- **Variance Reduction** Inspired by the bias-variance analysis of the generalization performance, the variance reduction principle seeks to query for instances that reduce the variance of the model the most. A similar approach is applied in the *optimal experimental design* in statistics (Federov, 1972). However, usually this also requires to solve expensive optimization problems.

- **Density-Weighted Methods** By considering the distribution of the instances, this strategy addresses an issue of uncertainty sampling and query-by-committee where outliers are likely to be selected but contribute limitedly to improving the learning (Fujii et al., 1998; Dasgupta and Hsu, 2008).

The method reported here basically follows the maximum disagreement principle of query-by-committee to select unlabeled pairs of phenotypes to query. A committee must be formed in order for this strategy to be applied, but it has been shown that even a small committee works well in practice. Various approaches can be applied to create committees. For example, co-testing (Muslea et al., 2006) applies this principle by combining forward and backward parsing models for information extraction. A key to the success of this strategy is that member models in the committee complement strengths and weaknesses.

The idea of our method is to compare the match-or-not assignments by the model trained by supervised learning and the class assignments derived

from exploring linkages of the labeled and unlabeled phenotypes. The most useful pairs to query are those whose assignments from the two different sources disagree with the highest confidence.

Exploring linkages may improve classifier learning when the classes of instances depend on each other. This idea has been studied in the context of classification of network data, such as pages on the Web, co-reference resolution, word sense disambiguation, and statistical relational learning (see e.g., (Macskassy, 2007; McCallum and Wellner, 2005; Popescul et al., 2003)).

In this paper, we present an algorithm that implement our idea. This algorithm can be divided into two major steps. The first step of the algorithm explores the linkages and the second step prioritizes pairs of phenotypes to query. By identifying maximum disagreement pair instances between the model classification results and exploring linkages between labeled and unlabeled phenotype variables, our active learner queries users for labels of unlabeled phenotypes that may improve the mapping the most and therefore will reduce the need of labeling efforts. Our experimental results show that exploring linkages can perfectly infer the match-or-not labels for a large number of pairs, and that active learning from maximum disagreement pairs improves the performance faster than from randomly selected pairs, suggesting that active learning by exploring linkages is a promising approach to the problem of phenotype mapping.

2 Phenotype Mapping

2.1 Problem Definition

Phenotype mapping is a task of searching for all databases of participating studies to find a set of phenotype variables that match a requested variable that the researcher is interested in. This is similar to the definition given in (Hsu et al., 2011) where the task is defined as the assignment of every phenotype variable from each participating study to one of a set categories, or classes, which corresponds to the "requested variable."

Table 1 shows a fragment of the phenotype mapping results of the phenotype variables that we matched manually from a consortium of cohort studies for a set of 70 requested variables. In this fragment, we show the phenotype variables assigned to one of the requested variables, the phenotype class `'hypertension'`. The real ID of a phenotype in a **Cohort** is given in column **Variable**. In this example, seven cohort studies have a total of 13 phenotype measurements related to hypertension.

Column **Description** is the main clue for automatic matching. The variable descriptions usually contain less than 10 words. As we can see in Table 1, the description contains abbreviations (e.g., `'HTN'`, `'HBP'`,`dx`), aliases (e.g., `'High Blood Pressure'` vs. `Hypertension`), measurement criteria (e.g., `DBP>90 MMHG`, `sys GE 140`, `per JNC7`, `JNC VI`), and tokens irrelevant to our task. As a result, word-by-word string similarity or sophisticated edit-distance based metrics can only match a small number of them. These examples are phenotypes that share similar semantics and are manually mapped to the same classes but their descriptions contain few or no common words. It is impossible for a model solely using the given descriptions to figure out that they refer to related phenotypes without bringing to bear additional information.

Other challenges of the phenotype problem include: not knowing in advance how many classes there are, unavailability of comprehensive categorization of phenotypes, and that the solution should scale well for a large number of phenotypes.

2.2 Supervised Learning for Phenotype Mapping

Here, we review the supervised learning method described in (Hsu et al., 2011), where phenotype mapping was casted as a pair matching problem and applied supervised learning to learn to tag a pair as a match or not. A pair of phenotypes are considered as a match if they are assigned to the same class, otherwise it is not. 13 phenotype variables in Table 1 will yield 78 pairs of positive examples of matched pairs. A maximum entropy classifier (MaxEnt) (Hastie et al., 2009) was used as the model to estimate the probability that a pair is a match. Two types of features were considered. The first type is based on string similarity metrics to combine the strength of a variety of string similarity metrics to measure the edit distance between the descriptions of a pair of phenotypes and use the result to determine if they

Cohort	Requested Variables	Variable	Description
ARIC	Hypertension	HYPERT06	HYPERTENSION, DEFINITION 6
CARDIA	Hypertension	Y01DBP	HYPERTENSION BASED ON DBP> 90 MMHG
CARDIA	Hypertension	Y01HTN	HIGH BLOOD PRESSURE
CARDIA	Hypertension	Y01HTNTP	TYPE OF HYPERTENSION
CFS	Hypertension	htn	HTN: abnormal bp (sys GE 140 or dia GE 90) or meds
CFS	Hypertension	htndx	HTN: self report of MD dx of HTN
CHS	Hypertension	HYPER	CALCULATED HTN STATUS
FHS	Hypertension	A70	HISTORY OF HYPERTENSION
FHS	Hypertension	B373	HYPERTENSION-ON TREAT OR ELEVATED BP
FHS	Hypertension	C332	HBP status
JHS	Hypertension	HTN017	Hypertension Status Per JNC7
MESA	Hypertension	HIGHBP1	HYPERTENSION: SELF-REPORT
MESA	Hypertension	HTN1C	Hypertension by JNC VI (1997) criteria

Table 1: Example variables of phenotype class 'hypertension'

match each other. The other type is the *weighted Jaccard* where appearence of tokens and bi-grams in both or one of the descriptions of a given phenotype pair is used as the features. The training algorithm for MaxEnt will virtually assign to each token or bi-gram a weight when it appears in the descriptions of an input phenotype pair. Weighted Jaccard is superior to string similarity features because string similarity metrics treat all tokens equally and the information provided by these metrics is limited. Therefore weighted jaccard was shown to outperform string similarity features by a large margin in the experimental evaluation.

Before the feature extraction step, descriptions will be augmented with the definitions given in the Merriam-Webster Medical Dictionary (2006)[1]. For example, 'hypertension' will be augmented with its definition in the dictionary 'abnormally high arterial blood pressure' and converted into 'hypertension abnormally high arterial blood pressure'. Augmented 'hypertension' will have many shared tokens with 'high blood pressure'. This augmentation step was proven to be effective in boosting recall, as semantically equivalent pairs described by totally different sets of tokens can be matched.

(Hsu et al., 2011) also reported a transitive inference method to take advantage of the transitive relationship of matched phenotype pairs. The idea is that if v_1 and v_2 are a match, so are v_2 and v_3, then v_1 and v_3 must be a match, too. Applying transitive inference did improve the performance, but when all possible transitive relations are explored, the performance degraded because false positives accumulated. The transitive inference method does not fully explore the dependency between pairs that share common phenotype variables. A more sophisticated approach is required.

[1] www.m-w.com/browse/medical/a.htm

3 Methods

Figure 1 illustrates our active learning idea. The idea is that, given a training set of phenotype variables X manually matched with class labels and a test set of unlabeled phenotype variables, the first step is to infer the class of each unlabeled variable by exploring the pairwise match scores assigned by the model trained by the training set. When we obtain a plausible class assignment to each unlabeled variable, we can classify each pair of unlabeled variables v_1 and v_2 by the trained model again to determine if they are a match or not and compare the result with their plausible class assignments.

If it turns out that the results agree with each other, we will move the pair to a set called *sure pairs*, otherwise, we will move the pair to a queue which will be sorted in descreasing order by how much the results disagree. Then we can query for true labels of the pairs in the queue to add to the training set the most useful examples and thus accomplish the active learning.

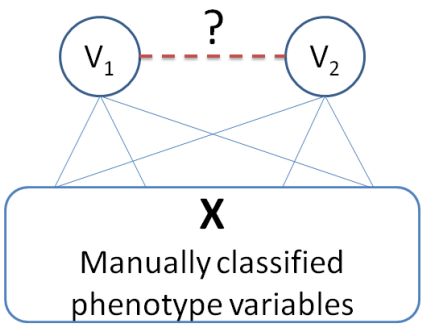

Figure 1: Inferernce of match between unlabeled phenotype variables by exploring their linkages to labeled pairs

3.1 Assigning Phenotype Categories

Procedure `LabelA` is to assign a class label to each unlabeled test variable by matching them to labeled training variables. Let A denote the set of all pairs between a test variable and a training variable. For each variable, the output contains an element of the variable, its assigned class label (may be null) and a score (log-likelihood). Function $I(.)$ in line 2 is the indicator function that returns 1 if its parameter is true and 0 otherwise. H is the model learned by calling the supervised training procedure. In line 7, P^H_{vx} is the probability that variables v and x are a match estimated by H. In line 8, `LabelA` assigns v to a class c, which is the class of the training variable x that maximizes P^H_{vx}. That is to assign the class of x as that of v if P^H_{vx} is the largest. Other selection can be used. For example, for each class c, we can estimate P^H_{vx} for all training variables x in c, and select c as the class of v if $\frac{1}{n}\sum \log P^H_{vx}$, the geometric mean of the probabilities, is the largest. These selection criteria are based on different assumptions and we will empirically compare which one is a better choice. In fact, any type of average can potentially be considered here.

3.2 Prioritizing Unlabeled Pairs

Procedure `LabelB` orders pairs of test variables to query for match-or-not and class labels. Let B be the set of all pairs of test variables. `LabelB` also generates a set called `SurePairs`. For each pair in B, `LabelB` checks if the model H considers the pair as a match ($P^H_{vx} \geq 0.5$) or not, and then checks if the pair is assigned by `LabelA` to the same class

Algorithm 1 Procedure LabelA

1: **Initialization**
- Training variables X with their class annotated $class(x) = c \in C, \forall x \in X$
- Test variables V with unknown class $class(v), \forall v \in V$

2: $H \leftarrow Train(\{(x_1, x_2, m) | x_1, x_2 \in X, m = I(class(x_1) = class(x_2))\})$
3: $A \leftarrow \{(v, x) | v \in V \wedge x \in X\}$
4: **procedure** LABELA(A, H)
5: Output $\leftarrow \emptyset$
6: **for** $v \in V$ **do**
7: $\forall x \in X, P^H_{vx} \leftarrow H(v, x)$
8: $c \leftarrow \arg\max_c(P^H_{vx})$
9: $L^H_{vx} \leftarrow \max_C(logP^H_{vx})$
10: **if** $L^H_{vx} < -2$ **then**
11: $c \leftarrow$ null,
12: $s \leftarrow \log(1 - 2^{L^H_{vx}})$
13: **else**
14: $s \leftarrow L^H_{vx}$
15: **end if**
16: Add (v, c, s) to Output
17: **end for**
18: Return Output
19: **end procedure**

or not. If it is a match and assigned to the same class, or not a match and assigned to different classes, that is, if H and `LabelA` agree, then the pair will be moved to `SurePairs`, otherwise, the pair will be moved to `Queue`. For a disagreed pair, `LabelB` also estimate the degree of disagreement by the sum of the log-probabilities of the class assignments ($L^H_{c_1}$ and $L^H_{c_2}$) and the match-or-not by the model ($P^H_{v_1 v_2}$). `SurePairs` can then be used for training.

We can then query for true labels of pairs in `Queue`. We can either query whether a pair is a match or not or query for their class label. After a certain number of queries, we can repeat the procedure to compute a new set of `SurePairs` and `Queue`, until all phenotypes are correctly assigned to a class.

Algorithm 2 Procedure LabelB

Initialization
H, A as in LabelA
$B \leftarrow \{(v_1, v_2) | v_1, v_2 \in V\}$
SurePairs $\leftarrow \emptyset$; Queue $\leftarrow \emptyset$
$\forall v_1, v_2 \in V, P^H_{v_1 v_2} \leftarrow H(v_1, v_2)$
$(v, class(v), L^H_c), \forall v \in V \leftarrow$ LabelA(A,H)
procedure LABELB(B, A, H)
  for $(v_1, v_2) \in B$ **do**
    if $P^H_{v_1,v_2} \geq 0.5$ **then**
      if $c_1 = c_2$ **then**
        Add $(v_1, v_2, 1)$ to SurePairs
      else
        $s \leftarrow L^H_{c_1} + L^H_{c_2} + \log(1 - P^H_{v_1 v_2})$
        Add (v_1, v_2, s) to Queue
      end if
    else
      if $c_1 = c_2$ **then**
        $s \leftarrow L^H_{c_1} + L^H_{c_2} + \log P^H_{v_1 v_2}$
        Add (v_1, v_2, s) to Queue
      else
        Add $(v_1, v_2, 0)$ to SurePairs
      end if
    end if
  end for
  Sort (v_1, v_2, m) in Queue by m
  Return Queue and SurePairs
end procedure

4 Results

4.1 Data

We manually selected 1,177 phenotype variables from a total of 35,041 in the databases of seven cohort studies as shown in Table 1 and assigned them to one of 70 requested variables that are common trait classes related to a large consortium study of cardiovascular disorders. These seven cohorts include ARIC (the Atherosclerosis Risk In Communities study `www.cscc.unc.edu/aric/`), CARDIA (the Coronary Artery Risk In Young Adults study `www.cardia.dopm.uab.edu`), CFS (the Cleveland Family study `dceweb1.case.edu/serc/collab/project_family.shtml`), CHS (the Cardiovascular Heart Study `www.chs-nhlbi.org/`), FHS (Framingham Heart Study `www.framinghamheartstudy.org/`), JHS (Jackson Heart Study `jhs.jsums.edu/jhsinfo/`), and MEC (the Multi-Ethnic Cohort `www.crch.org/multiethniccohort/`, `www.uscnorris.com/mecgenetics/`).

From these 1,177 phenotypes, 21,886 pairs are considered matches, that is, they are positive pairs with both phenotype variables in the same class. 670,190 pairs are negatives.

Method / Model	Precision	Recall	F-score
String similarity			
MaxEnt	0.5557	0.0660	0.1179
Weighted Jaccard			
MaxEnt	0.8791	0.4848	0.6250
w/ dictionary	0.9200	0.6104	0.7339
w/ transitive infer.	0.7735	0.6612	0.7129
w/ both	0.7728	0.8402	0.8051

Table 2: Performance results of supervised learning

4.2 Result of Supervised Learning

We divided all pairs in our data set by half into training and test sets and evaluate different options of the supervised learning algorithm with different options as described in (Hsu et al., 2011). The results as shown in Table 2 are consistent with the conclusions given in (Hsu et al., 2011). That is, weighted Jaccard features with dictionary augmentation plus transitive inference yields the best performance.

We also performed a split-by-variable test, where the set of all variables is divided into three equal parts. Two of them are used for training and the other for testing. This is closer to the realistic application scenario and provides a better estimation of the generalization performance of a trained model. The results are given as the first two rows in Table 3.

4.3 Result of Active Learning

We implemented the two algorithms and evaluate the performance. We still applied split-by-variable to divide the data with $\frac{1}{3}$ for testing and $\frac{2}{3}$ for training. We measured the performance when `SurePairs` produced by procedure `LabelB` was added to the training set, and then increasingly add more pairs in `Queue`, also produced by `LabelB`, to the training set, and measured the performance of the trained models to simulate an active learning

Method/Model	Precision	Recall	F-score
w/o dictionary	0.8344	0.4106	0.5504
w/ dictionary	0.6310	0.5287	0.5753
Test on A	0.7956	0.5243	0.6321
GM SurePairs (62622)	0.8772	0.5909	0.7061
Model (62622)	0.9577	0.2936	0.4494
MP SurePairs (74229)	0.8845	0.6196	0.7287
Model (74229)	0.9660	0.2875	0.4431

Table 3: Performance results of splitting by variables. Numbers in the parentheses show the number of pairs in `SurePairs`.

query sequence.

To ensure a fair comparison, we always use the set A, the pairs between a labeled and unlabeled phenotype variables, as the hold-out set for testing in all performance evaluations. Note that pairs in the set A never appear in either `SurePairs` or `Queue`, because pairs in `SurePairs` or `Queue` are selected from the set B, which contains the pairs between unlabeled phenotype variables. The third row of Table 3 shows the performance of the model tested only on A.

We implemented two versions of procedure `LabelA` that are different in the methods they used to assign a class to an unlabeled variable. The first, `MP`, is to use the maximum probability and the other, `GM`, is to use the maximum geometric mean of the probabilities (see Section 3.1).

We start by evaluating the quality of `SurePairs`. `GM` produced 62,622 pairs (1,642 positives) while `MP` had 74,229 pairs (1,816 positives). The match-or-not labels assigned by `LabelB` for both methods turn out to be perfectly correct, suggesting that combining model training and linkage exploration can effectively infer the match-or-not labels.

Adding `SurePairs` to the training set boosts F-scores, as shown in Table 3, which also shows that, in contrast, if we add the same number of pairs to the training set, but assign them match-or-not labels with the trained model, they will degrade F-scores.

Next, we added pairs in `Queue` to the training set, 280 pairs at a time, and measured the F-scores achieved by the resulting model. Figure 2 shows the learning curves of three different ways to order `Queue` produced with `GM`: descreasing, increasing, and random scores. The decreasing-score one performed the best by improving F-scores the fastest, confirming that higher-scored pairs are more useful. The end points of the three curves do not meet because we have not exhausted all training examples.

Similarly, we evaluated decreasing and random ordering of `Queue` produced by applying `MP`. We note that `MP` already produced a large set of `SurePairs`. As a result, less pairs are in `Queue` compared to that by `GM`. Therefore, after 9 passes, all pairs are exhausted and no obvious difference can be observed between decreasing and random ordering in the end.

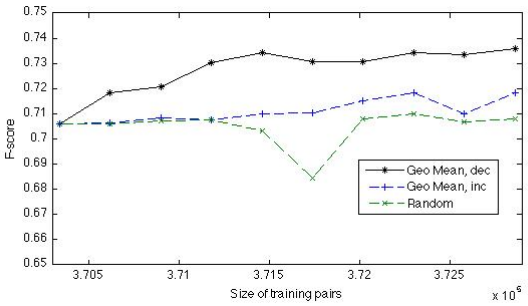

Figure 2: Learning curves of active learning: class assignment by maximum geometric mean of probabilities

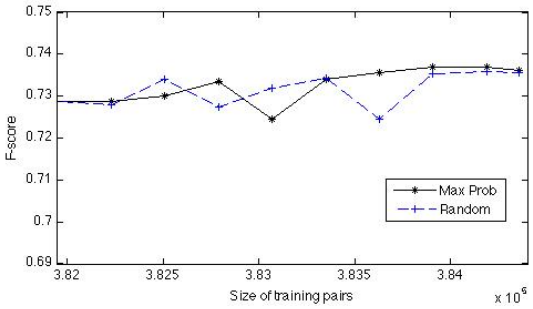

Figure 3: Learning curves of active learning: class assignment by maximum probabilities

5 Conclusions and Future Works

Despite the vast amounts of genomic data available in repositories, identification of relevant datasets can be challenging for researchers interested in specific

phenotypic measures. This paper presents our active learning approach that will be implemented as a component of new informatics tools for the research community to categorize phenotype measurements from genomic studies.

We show that comparing class assignment by exploring linkages and by the model can be effective in both improving the match-or-not assignments and ordering unlabeled pairs as queries for active learning. It is interesting that when two sources of class assignment agree, the pairs' match-or-not assignments are perfectly correct. How generalizable for this result deserves further investigation. We note that in order to perform a fair comparison, no pair between labeled and unlabeled phenotype variables are used for training. In a real application, they can be added to either `SurePairs` or `Queue` by extending procedure `LabelB` to include them.

Acknowledgments

We thank Cheng-Ju Kuo and Congxing Cai for their help in producing the results reported in Section 4.2. This research is supported by NHLBI-NIH grant 1UH2HL108780-01.

References

Stéphane Ayache and Georges Quénot. 2007. Evaluation of active learning strategies for video indexing. *Signal Processing: Image Communication*, 22(78):692 – 704. Special Issue on Content-Based Multimedia Indexing and Retrieval.

The Wellcome Trust Case Control Consortium. 2007. Genome-wide association study of 14,000 cases of seven common diseases and 3,000 shared controls. *Nature*, 447(7145):661–678, June.

Sanjoy Dasgupta and Daniel Hsu. 2008. Hierarchical sampling for active learning. In *Machine Learning, Proceedings of the Twenty-Fifth International Conference on Machine Learning (ICML-08)*, pages 208–215.

Joshua C. Denny, Marylyn D. Ritchie, Melissa A. Basford, Jill M. Pulley, Lisa Bastarache, Kristin Brown-Gentry, Deede Wang, Dan R. Masys, Dan M. Roden, and Dana C. Crawford. 2010. Phewas: demonstrating the feasibility of a phenome-wide scan to discover gene-disease associations. *Bioinformatics*, 26(9):1205–1210.

Wei Di and Melba M. Crawford. 2011. View generation for multiview maximum disagreement based active learning for hyperspectral image classification. *Geoscience and Remote Sensing, IEEE Transactions on*, PP(99):1 –13.

Valeriĭ V. Federov. 1972. *Theory of Optimal Experiments*. Academic Press.

Atsushi Fujii, Takenobu Tokunaga, Kentaro Inui, and Hozumi Tanaka. 1998. Selective sampling for example-based word sense disambiguation. *Computational Linguistics*, 24(4):573–597.

John Hardy and Andrew Singleton. 2009. Genomewide association studies and human disease. *New England Journal of Medicine*, 360(17):1759–1768.

Trevor Hastie, Robert Tibshirani, and Jerome H. Friedmann. 2009. *The Elements of Statistical Learning (2nd Edition)*. Springer-Verlag, New York, NY, USA.

Chun-Nan Hsu, Cheng-Ju Kuo, Congxing Cai, Sarah A. Pendergrass, Marylyn D. Ritchie, and Jose Luis Ambite. 2011. Learning phenotype mapping for integrating large genetic data. In *Proceedings of BioNLP 2011 Workshop*, BioNLP '11, pages 19–27, Portland, OR, USA. Association for Computational Linguistics.

David D. Lewis and Jason Catlett. 1994. Heterogeneous uncertainty sampling for supervised-learning. In *Proceedings of the International Conference on Machine Learning (ICML-94)*, pages 148–156.

Sofus A. Macskassy. 2007. Improving learning in networked data by combinng explicit and mined links. In *Proceedings of the National Conference on Artificial Intelligence (AAAI-07)*, page 590.

Andrew McCallum and Ben Wellner. 2005. Conditional models of identity uncertainty with application to noun coreference. In Lawrence K. Saul, Yair Weiss, and Léon Bottou, editors, *Advances in Neural Information Processing Systems 17*, pages 905–912. MIT Press, Cambridge, MA.

Merriam-Webster. 2006. *Medical Dictionary*. Merriam-Webster, Springfield, MA, USA.

Ion Muslea, Steve Minton, and Craig A. Knoblock. 2006. Active learning with multiple views. *Journal of Artificial Intelligence Research*, 27:203–233.

S. A. Pendergrass, K. Brown-Gentry, S. M. Dudek, E. S. Torstenson, J. L. Ambite, C. L. Avery, S. Buyske, C. Cai, M. D. Fesinmeyer, C. Haiman, G. Heiss, L. A. Hindorff, C. N. Hsu, R. D. Jackson, C. Kooperberg, L. Le Marchand, Y. Lin, T. C. Matise, L. Moreland, K. Monroe, A. P. Reiner, R. Wallace, L. R. Wilkens, D. C. Crawford, and M. D. Ritchie. 2011. The use of phenome-wide association studies (phewas) for exploration of novel genotype-phenotype relationships and pleiotropy discovery. *Genetic Epidemiology*, 35(5):410–422.

Alexandrin Popescul, Rin Popescul, and Lyle H. Ungar. 2003. Statistical relational learning for link prediction.

In *In Workshop on Learning Statistical Models from Relational Data at the International Joint Conference on Articial Intelligence (IJCAI-2003)*.

Burr Settles and Mark Craven. 2008. An analysis of active learning strategies for sequence labeling tasks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1069–1078. ACL Press.

Burr Settles. 2010. Active learning literature survey. Computer Science Technical Report 1648, University of Wisconsin-Madison, January.

H. Sebastian Seung, Manfred Opper, and Haim Sompolinsky. 1992. Query by committee. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory (COLT'92)*, pages 278–294.