A Multi-Task Feature Selection Filter for Microarray Classification

Liang Lan and Slobodan Vucetic
Department of Computer and Information Sciences, Temple University, Philadelphia, PA, USA
lanliang@temple.edu, vucetic@ist.temple.edu

Abstract

A major challenge in microarray classification and biomarker discovery is dealing with small-sample high-dimensional data where the number of genes used as features is typically orders of magnitude larger than the number of labeled microarrays. One way to address this challenge is by leveraging information from the publicly accessible repositories of microarray data. Following this idea, a multi-task feature selection filter is proposed that borrows strength from the auxiliary microarray classification data sets. The filter uses Kruskal-Wallis test on auxiliary data sets and ranks genes based on their aggregated p-values. Expressions of the top-ranked genes are used as features to build a classifier on the target data set. The proposed approach was evaluated on 9 microarray data sets related to 9 different types of cancers. Comparison of the classification accuracies reveals that the multi-task feature selection is superior to single-task feature selection. Furthermore, the results strongly suggest that multi-task algorithms could improve microarray classification by exploiting auxiliary data during feature selection and learning.

1. Introduction

Microarray technology has ability to simultaneously measure expression levels of thousands of genes for a given biological sample. In microarray classification tasks, samples belong to one of the several classes (e.g., cancer vs. control tissues) and the goal is to classify a new tissue sample based on its microarray measurements. Typical microarray classification data set contains a very limited number of labeled microarrays, ranging from only a few to several hundred. Training classifiers on such small-sample high-dimensional data sets is a challenging problem that has received an increasing attention from the research community. A standard way of addressing the challenge is to perform some kind of feature selection as a preprocessing step and to follow it by applying a classification algorithm that controls model complexity through some form of regularization. Feature selection filters, that perform a statistical test on every feature to determine its discriminative power [6], are reasonable dimensionality reduction strategy in microarray classification. Among the many classification algorithms, support vector machines and penalized logistic regression are particularly appropriate in small-sample learning scenarios.

When the number of labeled microarrays is particularly small (e.g., less than 10), the amount of available information diminishes to the level that even the most carefully designed classification approaches are bound to underperform. The only remedy is to borrow strength from external information sources. One promising approach is to incorporate prior knowledge about features (e.g., information about gene networks and pathways, or functional gene properties) into the regularization term [3,10].

An alternative approach is to utilize information from the external microarray data sets. For example, when learning to discriminate between infiltrative astrocytoma brain tumor and normal brain tissue using labeled microarray data obtained by a single lab, it might be beneficial to explore publicly available microarray data about other brain tumors or even about non-brain tumors. Recent work in multi-task learning [2] indicates that accuracy on the target classification task can be significantly increased if data from the auxiliary tasks are consulted during learning. This can be achieved either by modifying the regularization term [4] or by directly using auxiliary data for training [1]. However, multi-task learning with regularization might not be sufficient to mitigate the negative effects of high dimensionality in microarray data. In this case, feature selection becomes a necessary preprocessing step.

A standard approach is to perform feature selection on training data of the target task. Our hypothesis was that such single-task feature selection suffers from similar problems as single-task

classification. This is why in this paper we propose a multi-task feature selection filter that borrows strength from the auxiliary microarray data sets.

The contributions of this paper are as follows: (1) we propose a new multi-task feature selection filter, (2) we demonstrate that feature selection is an important preprocessing step in conjunction with multi-task classification algorithms, and (3) we demonstrate the potential and limitations of multi-task learning on microarray data.

2. Problem Definition

Let us define the *target data set* as $D = \{(x_i, y_i),\ i = 1 \ldots N\}$, where x_i is an M-dimensional feature vector for i-th example, y_i is its class label, and N is the number of labeled examples. We can assume that D is a random sample from an underlying distribution defined by feature distribution $p(x)$ and conditional distribution $p(y|x)$. The *target task* is defined as training a classifier $f(x)$ that approximates the conditional distribution $p(y|x)$. In *single-task learning*, only the target data D are used for training.

In *multi-task learning,* we have an access to the additional K *auxiliary data sets*. Let us denote the k-th auxiliary data set as $D_k = \{(x_i^{(k)}, y_i^{(k)}),\ i = 1 \ldots N_k\}$, where $x_i^{(k)}$ and $y_i^{(k)}$ are feature vector and target label for i-th example in D_k, and N_k is the size of D_k. We can assume that D_k is a random sample from feature distribution $p_k(x)$ and conditional distribution $p_k(y|x)$. The goal of multi-task learning is to improve accuracy of the target classifier by exploiting the auxiliary data.

Auxiliary data can be particularly useful when the target data set is small. Clearly, the usefulness of an auxiliary data set depends on the similarity between its and the target data distributions. Interestingly, even when the similarity is only moderate, it has been demonstrated that the auxiliary tasks can boost the classification accuracy on the target task. It is also worth noting that multi-task learning can sometimes lead to negative transfer, where accuracy on the target task is decreased due to the use of auxiliary data. An objective of this paper is to evaluate potential benefits and limitations of multi-task learning in microarray classification.

3. Penalized Logistic Regression for Multi-Task Learning

The multi-task feature selection filter proposed in Section 4 could in principle be used in conjunction with any single-task or multi-task classification algorithm. However, in this paper we focus attention to the penalized Logistic Regression (LR) classifiers due to their popularity and ability to easily introduce regularization. In this section, we give an overview of penalized LR classifiers and their use in single-task and multi-task scenarios.

Penalized LR. Let us consider for simplicity only binary classification where target variable can have one of two values, $y \in \{0, 1\}$. In logistic regression, the posterior probability of positive class can be written as a logistic sigmoid of a linear function of the feature vector, $p(y = 1|x) = \sigma(w^{\mathrm{T}}x)$, where w is the weight vector to be learned, and the logistic sigmoid is defined as $\sigma(z) = 1/(1+exp(-z))$.

The weight vector w can be estimated by maximizing the log-likelihood l_D of the training data set expressed as

$$l_D(w) = \log p(D \mid w) = \sum_{i=1}^{N} \log p(y_i \mid x_i, w) =$$

$$= \sum_{i=1}^{N} y_i \log \sigma(w^T x_i) + (1 - y_i) \log(1 - \sigma(w^T x_i)).$$

Therefore, the maximum likelihood estimate for w is given as

$$w_{ML} = \arg\max_w l_D(w).$$

To reduce overfitting, we can introduce a prior distribution $p(w)$ for the weight vector w. By conveniently setting the prior to be multivariate Gaussian $p(w) = N(w|\mu, \Sigma)$, we could obtain the maximum a posteriori (MAP) weight estimate as

$$w_{MAP} = \arg\max_w [\log p(D \mid x) + \log p(w)] =$$

$$= \arg\max_w [l_D(w) - \frac{1}{2}(w - \mu)^T \mathbf{\Sigma}^{-1}(w - \mu)].$$

The second term is the regularization term that penalizes any deviation of weights w from their desired values μ. Thus the name penalized LR for any training algorithm that estimates w_{MAP}. Evidently, the choice of weight prior hyperparameters μ and Σ is very important for the success of penalized LR.

Single-Task Penalized LR (S_LR). In single-task scenarios, there are typically two approaches in selection of prior hyperparameters μ and Σ. The most common is to use the penalized term solely for the purpose of regularization. Typically, one would choose $\mu = 0$ and $\Sigma = \sigma^2 \mathrm{I}$, where σ is scalar that determines how strongly the weights w are tied to zero and where I is the identity matrix. As for the choice of σ, it could be determined using cross-validation or by the more sophisticated empirical Bayes procedure. We call the resulting approach *the single-task penalized LR* and denote it as S_LR.

Regardless of its simplicity, the S_LR is regarded as quite successful in overfitting prevention. However, it is also worth noting that its prior oversimplifies the rich structure among the features. It assumes that the weights are independent of each other and have equal uncertainty. This approach is likely to result in poor performance when the training data are extremely scarce ($N<<M$), as is the case with microarray datasets. The alternative approach is to select hyper-parameters based on some prior belief about the weight values. This second approach motivates the multi-task penalized LR described next.

Multi-Task Penalized LR (M_LR). Recent work in multi-task learning resulted in many algorithms that define the prior hyperparameters to ensure that the predictors specialized on similar tasks have similar weights [4,7]. One of the simplest, yet effective, approaches of this type is to train a logistic regression model on each of the K auxiliary data sets and to use the resulting weight vectors w^k, $k = 1…K$, to determine the hyperparameters μ and Σ. In this paper, we applied the approach of [4] that calculates μ as the average weight of auxiliary classifiers and Σ as the diagonal matrix whose elements $\Sigma_{j,j} = \sigma_j^2$ are weight variances,

$$\mu = \frac{1}{K}\sum_{k=1}^{K} w^k, \qquad \sigma_j{}^2 = \frac{1}{K-1}\sum_{k=1}^{K}(w_j^k - \mu_j)^2.$$

Multi-Task Reweighted LR (RW_LR). The reweighting approaches for multi-task learning train the target classifier on both target examples and examples from the auxiliary data sets. Typically, examples from the target data set are given larger weights than those from the auxiliary data sets. The weights for each example can be calculated in various ways. Given the example weights, the logistic regression model is obtained as

$$w_{RW} = \arg\max_{w} [\sum_{i=1}^{N_{tot}} r_i \log p(y_i \mid x_i, w) + \log p(w)],$$

where N_{tot} denotes all examples from the target and the auxiliary data sets and r_i the example weights.

In this paper, we applied the approach of [1] for calculation of weights r_i. Weights of the target examples are set to 1. Weights of examples from the k-th auxiliary data set are obtained as the outputs of logistic regression model trained to discriminate between target examples (denoted as positive examples) and examples from the k-th auxiliary task (denoted as negative examples). Therefore, large overlap between the distributions of the target and the auxiliary data implies the large weights of the auxiliary examples.

4. Multi-Task Feature Selection

In this section we propose a multi-task feature selection filter suitable for microarray data. The main property of filters is that they are very efficient and do not require training a classifier. In general, feature x_j, $j = 1…M$, is deemed useful by a filter if its class-conditional distributions $p(x_j \mid y = 1)$ and $p(x_j \mid y = 0)$ are different. A standard approach to measure the difference is to run a statistical test of the null hypothesis that the two distributions are identical [6]. We first summarize this standard approach in the single-task scenario.

Single-task feature selection filter (S_FS). Let us consider data set $D = \{(x_i, y_i), i = 1…N\}$ and denote $X_j^+ = \{x_{ij}, y_i = 1\}$ and $X_j^- = \{x_{ij}, y_i = 0\}$ as the values of j-th feature in positive and negative examples, respectively. Let us denote $\theta_j = \theta(X_j^+, X_j^-)$ as a statistic measuring the difference between the two samples. Some examples of possible θ statistics are the difference in sample means, information gain, and χ^2 statistics. In the statistical filter, we ask the question of how likely it is that value θ_j or bigger occurs under the null hypothesis H_0: $p(x_j \mid y = 1) = p(x_j \mid y = 0)$. This is exactly the definition of the p-value of the statistical test. We denote the p-value for j-th feature as $p_j = p(\theta_j^0 \geq \theta_j)$, where θ_j^0 is the statistics under the null hypothesis. The p-values can be obtained analytically for some standard statistics such as the difference of means, while the permutation test can be used for other statistics such as the information gain.

Given the p-values, features can be selected in two ways. In first, all features with p-values below some threshold (e.g. $p_j < 0.05$) are selected. In second, M^* features with the smallest p-values are selected. In both cases, it is not clear what threshold is appropriate with respect to the resulting classification accuracy and a common practice is to use validation to determine it.

Multi-task feature selection filter (M_FS). When labeled data set D is very small, even feature selection filters can become highly inaccurate. This is because the power of statistical tests to discriminate between useful and irrelevant features drops with the sample size. The consequence is that a large number of irrelevant features would be selected and that sizeable number of relevant features would not, regardless of threshold choice. The remedy proposed here is to use multi-task feature selection filters. The idea is to first determine p-values of features in each auxiliary data set and to integrate those values for target task feature selection.

Let us consider k-th auxiliary data set $D_k = \{(x_i^{(k)}, y_i^{(k)}), i = 1 \ldots N_k\}$. We propose to calculate p_{jk} as p-value of j-th feature in D_k. By repeating the procedure for each auxiliary task, each feature could be represented by a K-dimensional vector $Q_j = \{p_{jk}, k = 1 \ldots K\}$. The following are four simple strategies to determine significance q_j of j-th feature from Q_j: (1) $q_j = min(Q_j)$; (2) $q_j = mean(Q_j)$; (3) $q_j = median(Q_j)$; (4) $q_j = max(Q_j)$. The selected M^* features are those with the smallest q_j values. The proposed strategies have slightly different objectives – $min(Q_j)$ favors features deemed very significant by any of the auxiliary tasks; $mean(Q_j)$ and $median(Q_j)$ prefer features that are most significant in average; and $max(Q_j)$ prefers features significant on all auxiliary tasks.

Related to the choice of M^*, the multi-task filter determines it by validation of single-task classifiers on auxiliary tasks that use single-task filters. This approach is expected to be more powerful than an arbitrary selection, or doing validation on the target data set with only a few labeled examples.

The proposed filter does not consider the similarity between auxiliary and target tasks and treats each auxiliary task as equally useful for feature selection. In practice, some auxiliary tasks are more similar to the target task than others. Therefore, our multi-task filter provides an option to use only a subset $K^* < K$ of the auxiliary data sets in feature selection. There, only p_{jk} values from the K^* auxiliary data sets are used in Q_j. The practical issue is that the target data set can be too small to aid in detection of unrelated auxiliary tasks.

To resolve this issue, we propose to measure accuracy of single-task auxiliary classifiers on target task examples and to select K^* auxiliary tasks as those with the most accurate classifiers. To measure accuracy, we use the log-likelihood as defined in Section 3. To select the best value of K^* parameter, while avoiding the danger of over-learning from the target data set, we validate K^* on auxiliary data sets.

Kruskal-Wallis test. Among many alternatives, we used the nonparametric Kruskal-Wallis test to determine feature p-values. The test is very appropriate for microarray data because it does not require strong distribution assumptions, it works well on small samples, and its p-values can be calculated analytically. Given positively and negatively labeled values of j-th feature, X_j^+ and X_j^-, Kruskal-Wallis sorts the values and calculates the average rank of positive and negative labels. Then, it calculates test statistic kw_j that becomes large when the average ranks deviate from the expected rank $(N+1)/2$. The p-value of the statistics is calculated easily because kw_j follows the standard χ^2 distribution.

5. Results

5.1 Data Description and Experimental Setup

Data sets used to evaluate the proposed algorithms were published in [8] and we downloaded them in the pre-processed form from [9]. They were obtained using Affymetrix microarrays that measured expression of $M = 15{,}009$ genes. The original data corresponded to multiple samples from 14 human cancer tissues and 12 normal tissues. From that data, we extracted 9 binary classification data sets by coupling normal and cancer samples from the same tissue type, whenever available. The summary of these 9 data sets in Table 1 indicates that all of them are small and that some of them are very imbalanced with just a few samples from normal tissues.

Table 1. Data set (cancer:normal cases)

Bladder (11:7)	Lung (20:7)	Prostate (14:9)
Breast (17:5)	Ovary (15:3)	Renal (11:13)
Colon (15:11)	Pancreas (11:10)	Uterus (11:6)

We evaluated several feature selection methods (see Section 4) in our experiments: (1) Single-task (S_FS), (2) Multi-task (M_FS), (3) Random selection (R_FS), and (4) Select all features (A_FS). In M_FS algorithm, we used the p-value aggregation based on minimum p-value, $q_j = min(Q_j)$. This approach proved slightly better than median and significantly better than maximum in our preliminary studies. The feature selection algorithms were evaluated in conjunction with 3 logistic regression algorithms (see Section 3): (1) Single task penalized LR (S_LR), (2) Multi-task penalized LR (M_LR), (3) Multi-task reweighted LR (RW_LR). In S_LR we used hyperparameter $\sigma = 1$. To train a logistic regression model, we used the conjugate gradient ascent method because it was much faster on high-dimensional microarray data than the more common Newton method [5].

For each combination of feature selection and logistic regression algorithms, we selected one of the 9 data sets as target data and the remaining 8 as the auxiliary data. To explore the influence of training size on transfer learning algorithms, we used $N^+ = \{1, 2, 3, 4, 5\}$ positive and the same number of negative target task examples for training. We used balanced training data to facilitate result interpretation. The remaining examples were used for testing. Due to lack of normal examples, in breast data set we used $N^+ = \{1, 2, 3, 4\}$

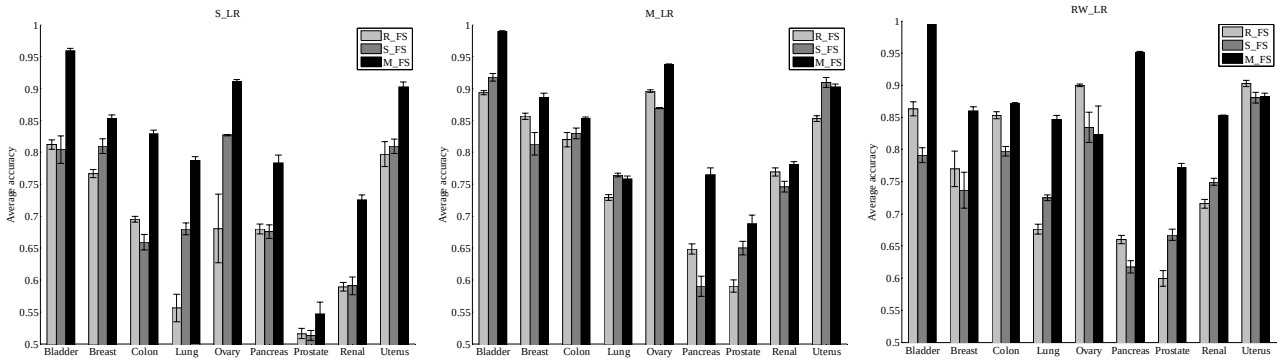

Figure 1.a-c Accuracies of 3 FS algorithms coupled with a) S_LR, b) M_LR, and c) RW_LR

and for ovary data set $N^+ = \{1, 2\}$. The reported accuracy was calculated as average between accuracy on positive and negative test examples, $Acc = Acc^+/2 + Acc^-/2$, where $Acc^+ = TP/(TP+FN)$, $Acc^- = TN/(TN+FP)$. For each choice of N^+ we repeated experiments 10 times, each with different randomly selected N^+ positive and $N^- = N^+$ negative examples from the target data. Thus, we built and tested 50 (except for breast and ovary data) classifiers for each feature selection-classifier combination. All experiments were repeated 9 times so that each cancer data set was used as target data set in one set of experiments.

5.2 Experimental Results

Choice of number of features M^*. Following the approach proposed in Section 4, we trained S_LR classifiers using $M^* = \{50, 100, 500, 5000, M\}$ features selected by S_FS. Following the setup from Section 5.1, we trained and tested 410 classifiers for each choice of M^*. Table 2 summarizes how many times S_FS + S_LR classifier with $M^* = 100$ was more accurate than the same classifier with different choice of M^*. $M^* = 100$ was better than the alternative values.

Table 2. Win:loss for $M^* = 100$ vs. other M^*

$M^* =$	50	500	5,000	15,009
$M^* = 100$	289:121	216:194	259:141	269:141

Table 3. Win:loss for $K^* = 5$ vs. other K^*

$K^* =$	0	1	3	8
$K^* = 5$	241:169	251:159	219:191	216:194

Choice of number of auxiliary tasks K^*. We determined suitable number of auxiliary tasks for feature selection following the approach explained in Section 4. In Table 3 we compare win:loss statistics of M_FS + S_LR classifier with $K^* = 5$ to the same classifier with $K^* = \{0, 1, 3, 8\}$ ($K^* = 0$ corresponds to S_FS + S_LR classifier). Although the differences were rather small, $K^* = 5$ seemed to be the best overall choice.

Comparing feature selection algorithms. In Figure 1 we compare performance of three different feature selection methods, R_FS, S_FS, M_FS, all using 100 selected features. Figure 1.a shows average accuracies of S_LR classifier trained with $N^- = N^+ = 2$ target examples and using 100 selected features. Each of the 9 sets of bars represents a case when a given data set is used as the target data and the remaining as the auxiliary data. Figures 1.b and 1.c show accuracies of multi-task classifiers M_LR and RW_LR. It could be seen from all three figures that multi-task feature selection (M_FS) results in superior accuracies. There are only few cases when it is not the best performing approach. Comparing R_FS and S_FS, S_FS is slightly better when coupled with S_LR and slightly worse with RW_LR. This clearly indicates that 4 training examples are not sufficient for a successful single-task feature selection. On the other hand, borrowing strength from the auxiliary tasks appears very helpful.

Accuracy vs. training size. In Figure 2 we compare accuracies of four representative algorithms on each target task as a function of the training size. S_LR + S_FS is purely single-task algorithm, S_LR + M_FS is combination of single-task classifier and multi-task feature selection, and the remaining two are purely multi-task. It can be seen that S_LR + S_FS is inferior to the other algorithms and that the difference in accuracy is particularly large for small target training sizes. Interestingly, S_LR + M_FS is very competitive to purely multi-task algorithms. When $N^- = N^+ = 5$, the difference decreases and, in few cases, we could even observe slight negative transfer. The difference between the 3 remaining algorithms is rather small, with RW_LR + M_FS being most accurate overall.

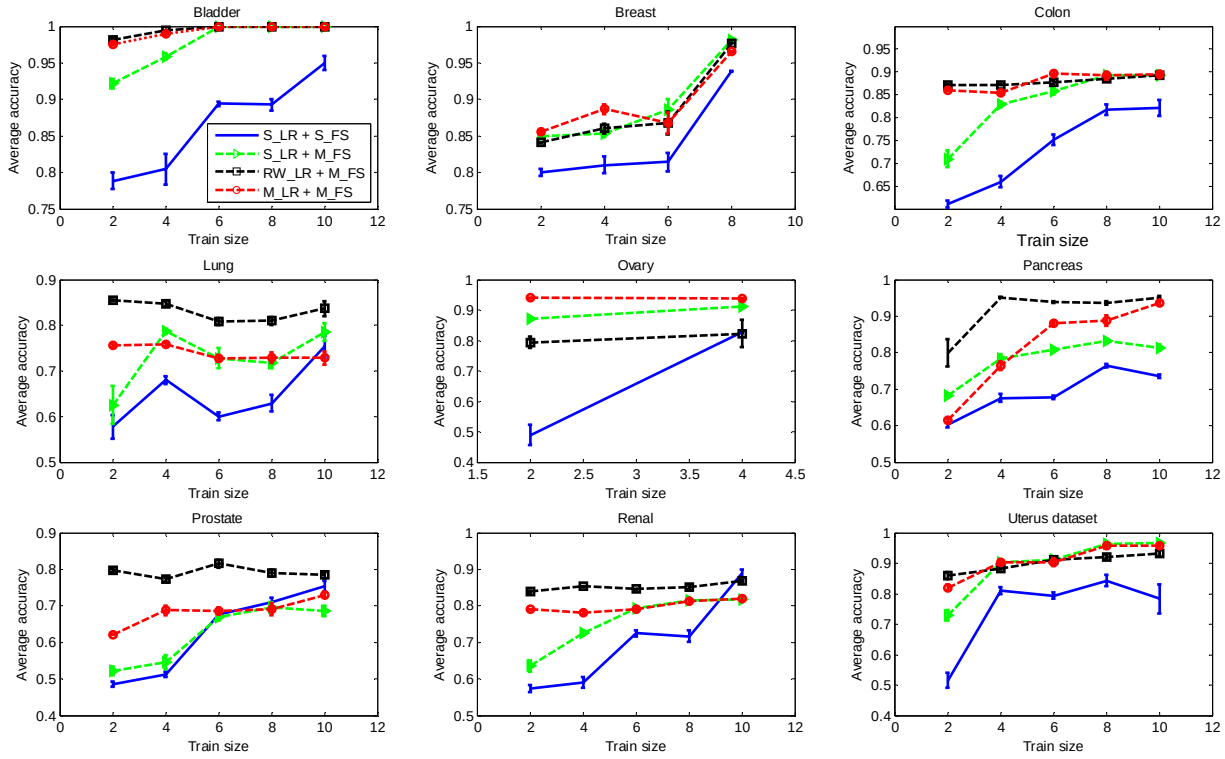

Figure 2 Comparison of 4 classifiers on 9 target tasks for varying training sizes.

6. Conclusion

In this paper we proposed a multi-task feature selection filter suitable for microarray classification. The filter is widely applicable because of the availability of rich public repositories of microarray data. The filter can be used as a preprocessing step for an arbitrary classification algorithm. Experimental results indicate that the proposed filter boosts accuracy of both single-task and multi-task classifiers. Combination of multi-task feature selection and classification appears particularly successful. We observed that multi-task learning is the most useful when target examples are scarce. Its benefits decrease with data size and could even lead to negative transfer. Despite its limitations, it is evident that multi-task learning can be a useful bioinformatics tool in many biological problems.

7. Acknowledgement

This study was supported by NSF grant IIS-0546155.

8. References

[1] Bickel, S., Bogojeska, J., Lengauer, T., Scheffer, T., Multi-task learning for HIV therapy screening, *Proc. 25th International Conference on Machine Learning*, pp. 56-63, Helsinki, Finland, 2008.

[2] Caruana, R., Multitask learning, *Machine Learning*, 28, pp. 41–75, 1997.

[3] Li, C., Li, H., Network-constrained regularization and variable selection for analysis of genomic data, *Bioinformatics*, 24, pp. 1175–1182, 2008.

[4] Marx, Z., Rosenstein, M.T., Dietterich,T.G., Kaelbling, L.K., Two algorithms for transfer learning, In *Inductive Transfer: 10 years later*, 2008.

[5] Minka, T.P, A comparison of numerical optimizers for logistic regression, *Technical report*, 2003.

[6] Radivojac, P., Obradovic, Z., Dunker, AK, Vucetic, S., Characterization of permutation tests for feature selection, *Proc. 15th European Conference on Machine Learning*, pp. 334-346, Pisa, Italy, 2004.

[7] Raina, R., Ng A.Y., Koller, D., Constructing informative priors using transfer learning, *Proc. 23rd International Conference on Machine Learning*, pp.713-720, Pittsburgh, PA, 2006.

[8] Ramaswamy, et al., Multiclass cancer diagnosis using tumor gene expression signatures, *Proc. of the National Academy of Sciences*, 98, pp. 15149–54, 2001.

[9] Statnikov, A., Aliferis, C., Tsamardinos, I., Hardin, D., Levy, S., A comprehensive evaluation of multicategory classification methods for microarray gene expression cancer diagnosis, *Bioinformatics*, 21, pp. 631–643, 2005.

[10] Tai, F., Pan, W., Incorporating prior knowledge of predictors into penalized classifiers with multiple penalty terms, *Bioinformatics*, 23, pp. 1775–82, 2007.