Improving accuracy of microarray classification by a simple multi-task feature selection filter

Liang Lan

Department of Computer and Information Sciences,
Temple University,
321 Wachman Hall, 1805 N. Broad Street,
Philadelphia, PA 19122, USA
E-mail: lanliang@temple.edu

Slobodan Vucetic*

Department of Computer and Information Sciences,
Temple University,
304 Wachman Hall, 1805 N. Broad Street,
Philadelphia, PA 19122, USA
E-mail: vucetic@ist.temple.edu
*Corresponding author

Abstract: Leveraging information from the publicly accessible data repositories can be very useful when training a classifier from a small-sample microarray data. To achieve this, we proposed a multi-task feature selection filter that borrows strength from auxiliary microarray data. It uses Kruskal–Wallis test on auxiliary data and ranks genes based on their aggregated *p*-values. The top-ranked genes are selected as features for the target task classifier. The multi-task filter was evaluated on microarray data related to nine different types of cancers. The results showed that the multi-task feature selection is very successful when applied in conjunction with both single-task and multi-task classifiers.

Keywords: feature filter; microarray classification; multi-task learning; transfer learning; bioinformatics.

Reference to this paper should be made as follows: Lan, L. and Vucetic, S. (2011) 'Improving accuracy of microarray classification by a simple multi-task feature selection filter', *Int. J. Data Mining and Bioinformatics*, Vol. 5, No. 2, pp.189–208.

Biographical notes: Liang Lan is a PhD candidate at Temple University. He got his MS in Computer and Information Sciences from Temple University and BS in Bioinformatics from Huazhong University of Science and Technology. His research focuses on data mining and bioinformatics.

Slobodan Vucetic is an Associate Professor in the Department of Computer and Information Sciences at Temple University. He obtained his PhD from Washington State University, and his MS and BS from the University of Novi Sad. His research interests are in data mining, machine learning and bioinformatics.

1 Introduction

Microarray technology has the 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 labelled 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 feature selection as a pre-processing step and to follow it by applying a classification algorithm that controls model complexity through regularisation. Feature selection filters, which perform a statistical test on every feature to determine its discriminative power (Radivojac et al., 2004), are a reasonable dimensionality reduction strategy in microarray classification. Among the many classification algorithms, SVMs and penalised LR are particularly appropriate in small-sample learning scenarios.

When the number of labelled 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 regularisation term (Li and Li, 2008; Tai and Pan, 2007).

An alternative approach is to utilise information from the external microarray data sets. For example, when learning to discriminate between infiltrative astrocytoma brain tumour and normal brain tissue using labelled microarray data obtained by a single lab, it might be beneficial to explore publicly available microarray data about other brain tumours or even about non-brain tumours. Recent work in multi-task learning that was originally proposed in Caruana (1997) 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 regularisation term (Marx et al., 2005) or by directly using auxiliary data for training (Bickel et al., 2008). However, multi-task learning with regularisation might still not be sufficient to mitigate the negative effects of high dimensionality in microarray data. In this case, feature selection becomes critical for the success of learning.

A straightforward 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 found in 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:

- we propose a new multi-task feature selection filter
- we demonstrate that feature selection is an important pre-processing step in conjunction with several popular single-task and multi-task classification algorithms
- 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 ith example, y_i is its class label, and N is the number of labelled 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 access to the additional K *auxiliary data sets*. Let us denote the kth 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 ith example in D_k, respectively, 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 its similarity to the target data distribution. 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 use of auxiliary data decreases accuracy on the target task. An objective of this paper is to evaluate potential benefits and limitations of multi-task learning in microarray classification.

3 Single-task and multi-task classifiers

The multi-task feature selection filter proposed in Section 4 can be used in conjunction with any single-task or multi-task classification algorithm. To demonstrate the robustness of the proposed multi-task feature selection method, we evaluate its performance on several popular classification algorithms. The results presented in Section 5 are aimed at demonstrating that the proposed multi-task feature filter indeed improves the classification performance no matter which classification algorithm is used. We start this section by an overview of three standard single-task classification algorithms: penalised LR, lasso and SVMs. We also outline several previously proposed extensions of these algorithms for multi-task classification.

3.1 Penalised logistic regression for multi-task learning

3.1.1 Penalised LR

Let us consider for simplicity only binary classification where target variable can have one of two values, $y \in \{0, 1\}$. In LR, the posterior probability of positive class can be written as a logistic sigmoid of a linear function of the feature vector, $p(y = 1|x) = \sigma(w^T x)$, 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 maximising 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

$$w_{\text{MAP}} = \arg\max_w [\log p(D \mid w) + \log p(w)]$$
$$= \arg\max_w [l_D(w) - \frac{1}{2}(w - \mu)^T \Sigma^{-1} (w - \mu)].$$

The second term is the regularisation term that penalises any deviation of weights w from their desired values μ. Thus, the name penalised 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 penalised LR.

3.1.2 Single-task penalised LR (S_LR)

In single-task scenarios, there are typically two approaches to selection of prior hyperparameters μ and Σ. The most common is to use the penalised term solely for the purpose of regularisation. Typically, one would choose $\mu = 0$ and $\Sigma = \sigma^2 I$, where σ is scalar that determines how strongly the weights w are tied to zero and 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 penalised 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 more likely to result in poor performance when the training data are extremely scarce ($N << M$), as is the case with microarray data sets.

The alternative approach is to select hyperparameters based on some prior belief about the weight values. This second approach motivates the multi-task penalised LR described next.

3.1.3 Multi-task penalised LR (M_LR)

Recent work in multi-task learning resulted in many algorithms that define the prior hyperparameters to ensure that the predictors specialised on similar tasks have similar weights (Marx et al., 2005; Raina et al., 2006). One of the simplest, yet effective, approaches of this type is to train an LR model on each of the K auxiliary data sets and to use the resulting weight vectors w^k, $k = 1, \ldots, K$, to determine the hyperparameters μ and Σ. In this paper, we applied the approach of Marx et al. (2005) that calculates μ as

the average weight of auxiliary classifiers and Σ as the diagonal matrix whose elements $\sum_{j,j} = \sigma_j^2$ are weight variances,

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

3.1.4 Multi-task Reweighted LR (RW_LR)

The reweighting approaches for multi-task learning train the target classifier using a mixture of 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 LR model is obtained as

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

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 Bickel et al. (2008) for calculation of weights r_i. Weights of examples are obtained as the outputs of LR model trained to discriminate between target examples (denoted as positive examples) and examples from an 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.

3.2 Lasso

3.2.1 Single-task Lasso (S_Lasso)

Lasso (Tibshirani, 1996) is a popular embedded feature selection technique. It does feature selection automatically by solving the l_1-regularised learning problem. If used in conjunction with the LR, the optimal w is obtained by maximising the regularised log-likelihood

$$w_{S-\mathrm{Lasso}} = \arg\max_{w} [\log p(D \mid w) - \lambda \| w \|_1]$$

where λ is a regularisation coefficient. This optimisation problem can also be explained as regularised LR with Laplacian prior (Krishnapuram et al., 2005; Cawley and Talbot, 2006).

Because of the nature of l_1 penalty, solving this objective problem typically results in a sparse model where most of the weights become zeros. Owing to the sparse solution of Lasso, it becomes a very useful embedded feature selection approach.

3.2.2 Multi-task Lasso (M_Lasso)

A regularisation scheme for multi-task feature selection was proposed by Obozinski et al. (2010). It aims to find a common set of features that are relevant to all available classification tasks. This is achieved by defining a regularisation term as an l_1 sum of

the l_2 norms of the feature-specific vectors. The optimisation problem for multi-task Lasso can be represented as

$$w_{M-\text{Lasso}} = \arg\max_w \left[\sum_{k=1}^{K} \log p(D^k \mid w^k) - \lambda \sum_{j=1}^{M} \left\| w_j \right\|_2 \right]$$

where K is the number of tasks, M is the number of features, w_j is a column vector that represents the coefficients of features j across all K tasks and w^k is a row vector that represents parameters of the learned model for classification task k. The solution is a $K \times M$ matrix containing the learned parameter for all tasks. This l_1/l_2 regularisation scheme selects feature jointly across all considered classification tasks. The l_2 norms measure the overall relevance of a particular feature while the l_1 sum enforces feature selection. It encourages similar sparsity patterns for all task-specific models. Note that this l_1/l_2 regularisation scheme reduces to l_1-regularisation if the number of tasks is reduced to one.

3.3 Support vector machine

3.3.1 Single-task Support Vector Machine (S_SVM)

Support Vector Machines (SVMs) (Vapnik, 1995) are among the most powerful and popular classification algorithms. The SVM is solving the following problem

$$\text{minimise } \frac{1}{2} \| w \|^2 + C \sum_{i=1}^{N} \xi_i$$
$$\text{subject to } \; y_i (w^T x_i + b) \geq 1 - \xi_i, \; \xi_i \geq 0$$

where w is the weight vector, b is the bias term, ξ_i are slack variables and C is the slack parameter. The resulting predictor has the form

$$f(x) = w^T x + b = \sum_{i=1}^{N} \alpha_i y_i (x_i^T \cdot x) + b$$

where $0 \leq \alpha_i \leq C$ (non-zero alphas correspond to the support vectors). If needed, this linear classifier can be easily converted to a non-linear classifier by kernelisation, i.e., by replacing the dot product $x_i^T x$ with an appropriately selected kernel function $K(x_i, x)$.

3.3.2 Multi-task Reweighted Support Vector Machine (RW_SVM)

The reweighting approach introduced in Section 3.1.2 can be used to design a multi-task SVM. Knowing the weight r_i of each example from the target and auxiliary tasks, we use the idea of fuzzy SVM (Lin and Wang, 2002) to train the multi-task SVM through solving the following optimisation problem

$$\text{minimise } \frac{1}{2} \| w \|^2 + C \sum_{i=1}^{N} r_i \xi_i$$
$$\text{subject to } \; y_i (w^T x_i + b) \geq 1 - \xi_i, \xi_i \geq 0.$$

The effect of reweighting is equivalent to changing the value of the slack parameter from C for all examples to $C \cdot r_i$. The resulting predictor has the same form as the original SVM,

with only difference that $0 \leq \alpha_i \leq C \cdot r_i$. Therefore, examples with smaller weight r_i will have smaller influence on the resulting predictor. To determine weights r_i, we can use the approach by Bickel et al. (2008) described in Section 3.1.4.

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 $x_j, j = 1, \ldots, M$, is deemed useful by a filter if its class-conditional distributions $p(x_j|y = 1)$ and $p(x_j|y = 0)$ 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 (Radivojac et al., 2004). We first summarise this standard approach in the single-task scenario.

Single-task feature selection filter (S_FS): Let us consider data set $D = \{(x_i, y_i), i = 1, \ldots, N\}$ and denote $X_j^+ = \{x_{ij}, y_i = 1\}$ and $X_j^- = \{x_{ij}, y_i = 0\}$ as the values of jth 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: $p(x_j|y = 1) = p(x_j|y = 0)$. This is exactly the definition of the p-value of the statistical test. We denote the p-value for jth feature as $p_j = p(Q_j^0 \geq q_j)$, where Q_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. First, all features with p-values below some threshold (e.g., $p_j < 0.05$) are selected. 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 labelled 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 kth 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 jth 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 jth feature from Q_j:

- $q_j = \min(Q_j)$
- $q_j = \text{mean}(Q_j)$
- $q_j = \text{median}(Q_j)$
- $q_j = \max(Q_j)$.

The selected M^* features are those with the smallest q_j values. The proposed strategies have slightly different objectives – $min(Q_j)$ favours 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; $max(Q_j)$ 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 labelled 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 non-parametric Kruskal-Wallis test to determine feature p-values. The test is very appropriate for microarray data because it does not require strong distributional assumptions, it works well on small samples, and its p-values can be calculated analytically. Given positively and negatively labelled values of jth 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 set-up

Data sets used to evaluate the proposed algorithms were published in Ramaswamy et al. (2001) and we downloaded them in the pre-processed form from Statnikov et al. (2005). 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 nine binary

classification data sets by coupling normal and cancer samples from the same tissue type, whenever available. The summary of these nine 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:

- Single-task (S_FS)
- Multi-task (M_FS)
- Random selection (R_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 different classification algorithms (see Section 3):

- Single-task Lasso (S_Lasso)
- Multi-task Lasso (M_Lasso)
- Single-task SVM (S_SVM)
- Multi-task reweighted SVM (RW_SVM)
- Single-task penalised LR (S_LR)
- Multi-task penalised LR (M_LR)
- Multi-task reweighted LR (RW_LR).

In S_LR, we used hyperparameter $\sigma = 1$. To train an LR model, we used the conjugate gradient ascent method because it was much faster on high-dimensional microarray data than the more common Newton method (Minka, 2003). For S_Lasso and M_Lasso, we used TL_BBLasso algorithm implemented in UC Berkeley Transfer Learning Toolkit (Rakhlin, 2007). We used $\lambda = 1$ in both the S_Lasso and the M_Lasso algorithms. For training the S_SVM and RW_SVM classifiers, we used $C = 1$. For each combination of feature selection and data mining algorithms, we selected one of the nine data sets as target data and the remaining eight 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. Because of lack of normal examples, in breast data set we used $N^+ = \{1, 2, 3, 4\}$ 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 ten 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 nine 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 set-up from Section 5.1, we trained and tested 410 classifiers for each choice of M^*. Table 2 summarises 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	5000	15,009
$M^* = 100$	289 : 121	216 : 194	259 : 141	269 : 141

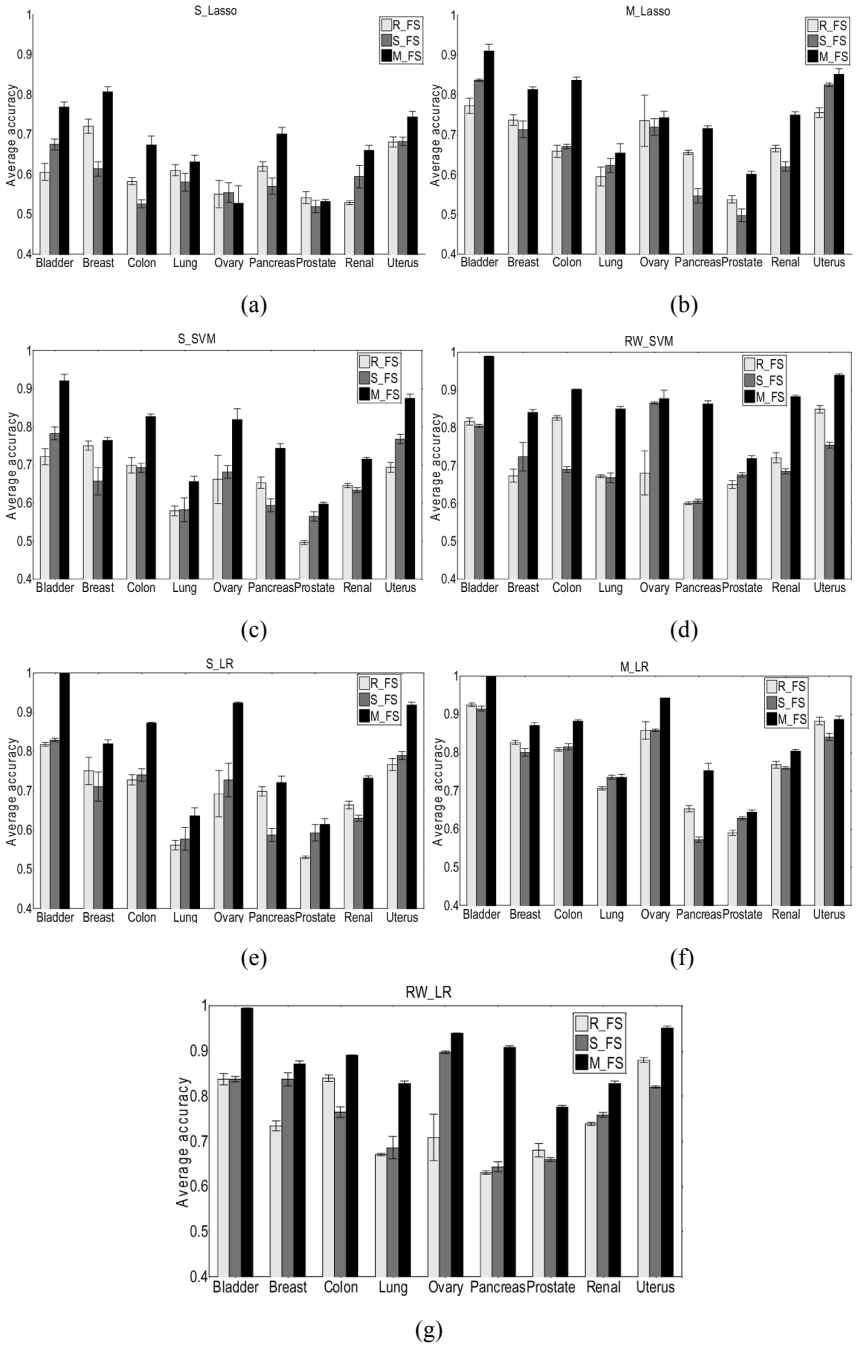
*Choice of number of auxiliary tasks K**: We determined suitable number of auxiliary tasks for feature selection following the approach explained in Section 4. In Table 3, we compare *win* : *loss* statistics of M_FS + S_LR classifier with $K^* = 5$ with the same classifier with $K^* = \{0, 1, 3, 8\}$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

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

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)–(g) shows average accuracies of seven different classifiers trained with $N^- = N^+ = 2$ target examples and using 100 selected features. Each of the nine sets of bars represents a case when a given data set is used as the target task and the remaining data sets as the auxiliary tasks. It could be seen from all seven figures that multi-task feature selection (M_FS) results in superior accuracies. There are only two cases (in S_Lasso experiments) where M_FS was outperformed and several other cases where M_FS performed similar either to S_FS or R_FS. Comparing R_FS and S_FS, it is hard to conclude which one is better. This clearly indicates that four training examples are not sufficient for a successful single-task feature selection. On the other hand, the success of M_FS demonstrates usefulness of the idea of borrowing strength from the auxiliary tasks.

Figure 1 Accuracies of three FS algorithms coupled with: (a) S_Lasso; (b) RW_Lasso; (c) S_SVM; (d) RW_SVM; (e) S_LR; (f) M_LR and (g) RW_LR

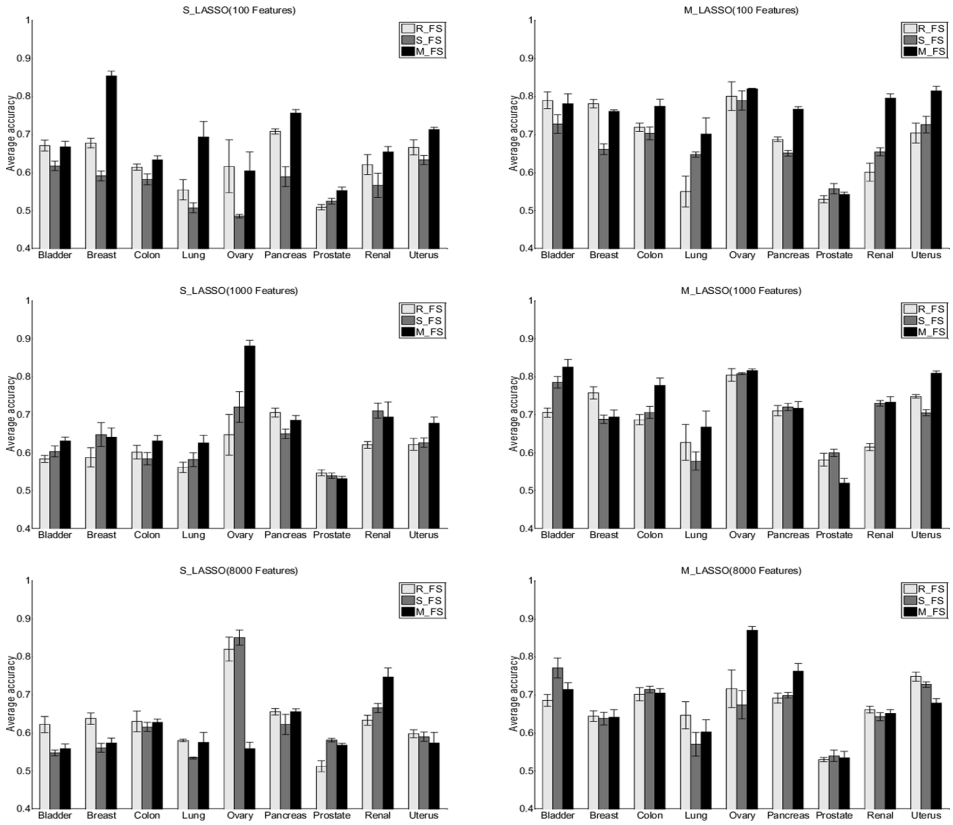


It is worth observing that results in Figure 1(b), (d), (f) and (g) correspond to different multi-task learning algorithms. Comparing the obtained accuracies with the corresponding single-task algorithms, it can be seen that they indeed increase the

classification accuracy. While multi-task algorithms are superior to their single-task counterparts, an important conclusion is that the proposed multi-task feature selection filter can further improve their performance. Therefore, the multi-task filter was a very useful pre-processing step for both single and multi-task learning algorithms.

Lasso accuracy vs. number of input features: The l_1 regularisation used in the Lasso model plays the role of an embedded feature selection. Figure 1(a) and (b) shows results of applying Lasso on 100 selected features. To better test the usefulness of the proposed feature selection filter to Lasso, we also evaluated its performance (on $N^- = N^+ = 2$ target training examples) using $M^* = \{8000, 1000, 100\}$ selected features. The results are shown in Figure 2. We can observe that in the S_Lasso case (left column of Figure 2) 100 M_FS selected features achieved the best results overall. It demonstrates that our multi-task feature filter is in fact beneficial to S_Lasso. Similar behaviour occurred in M_Lasso experiments, although the difference between selecting 100 features using M_FS and other methods is slightly smaller. Comparing M_Lasso with S_Lasso, we could conclude that the M_Lasso is more accurate. Overall, the M_Lasso with 100 M_FS (top right panel in Figure 2) achieves the highest accuracy.

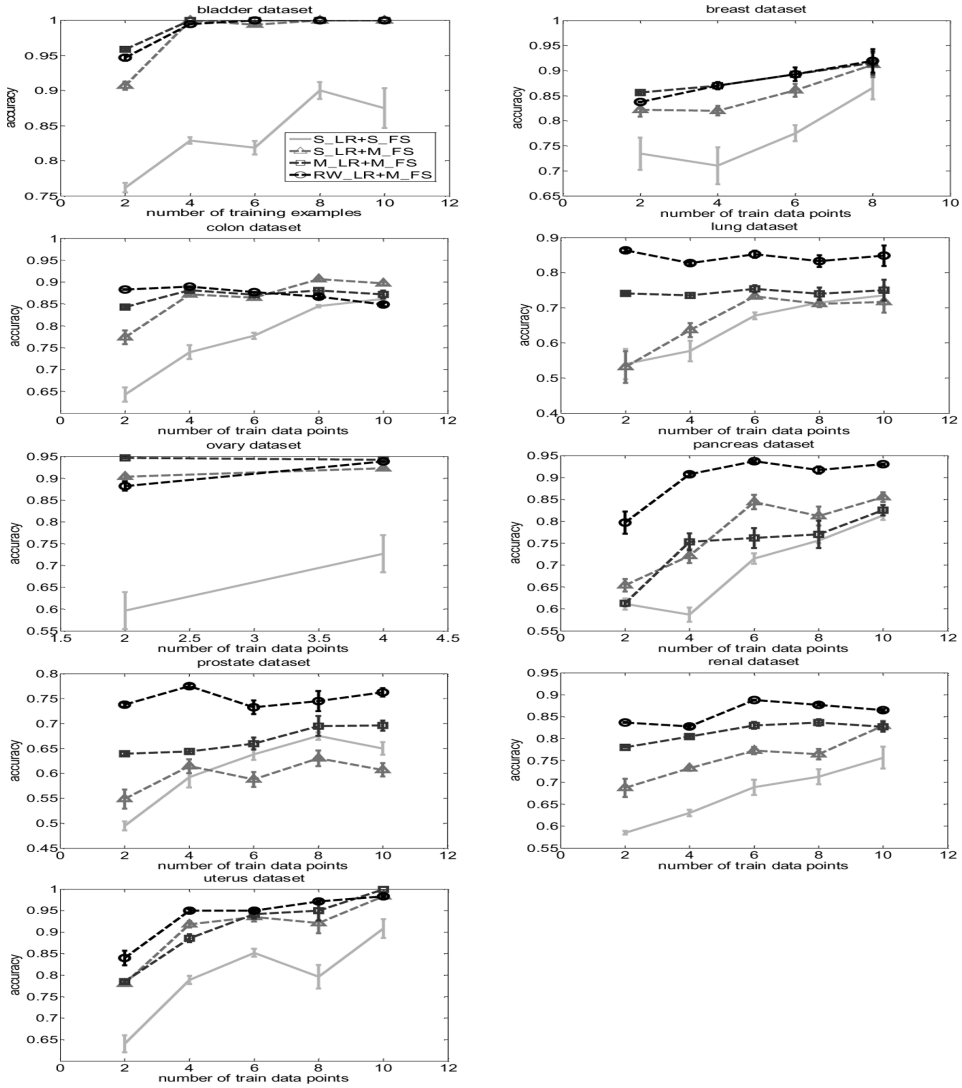
Figure 2 Accuracies of three FS algorithms with different number of selected features coupled with (left) S_Lasso and (right) M_Lasso



Accuracy vs. training size: In Figure 3, we compare accuracies of four representative LR algorithms as a function of the training size on each target task using 100 selected

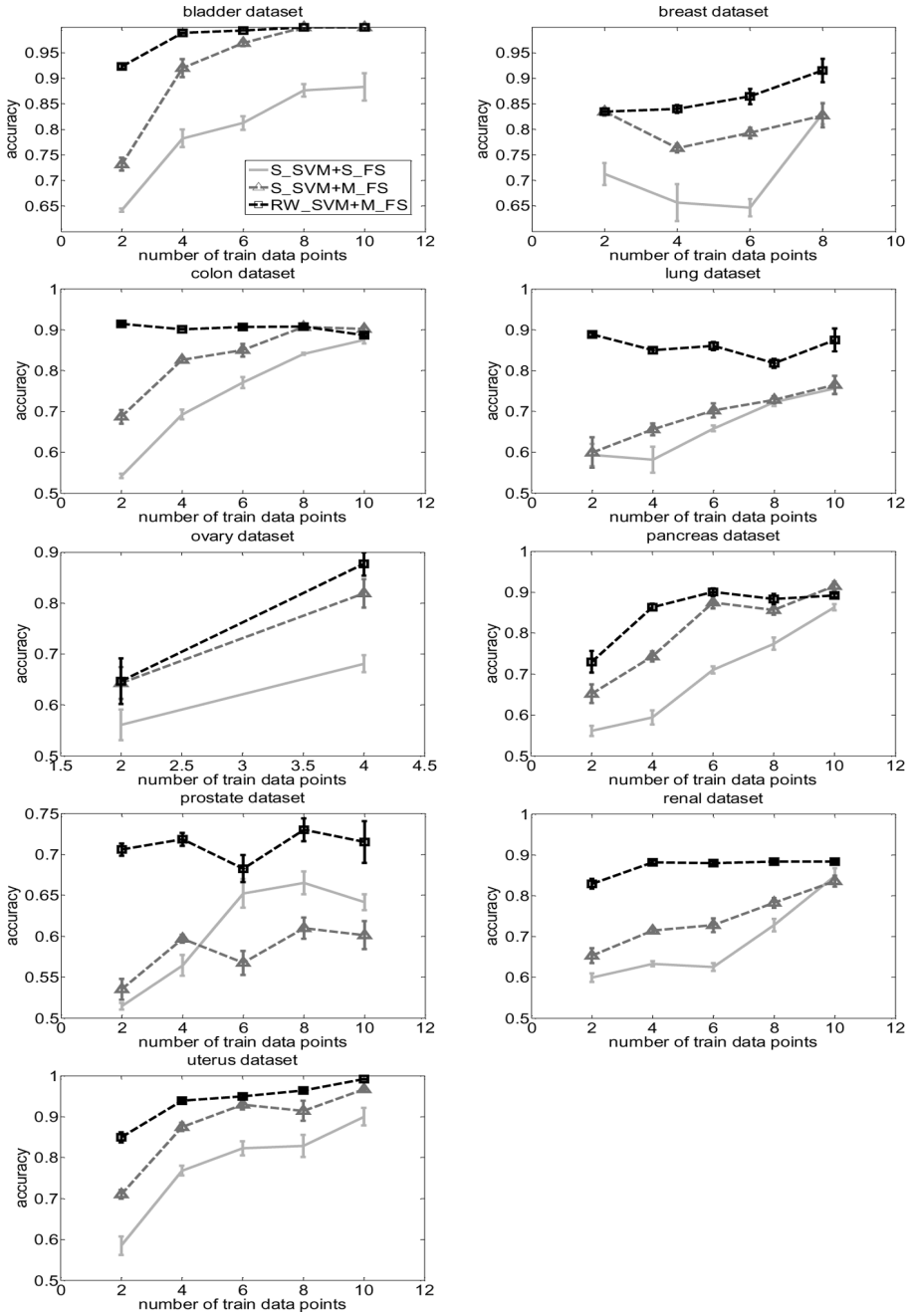
features. S_LR + S_FS is purely single-task algorithm, S_LR + M_FS is combination of single-task classifier and multi-task feature selection, and the remaining two are purely multi-task. It can be seen that S_LR + S_FS was inferior to the other algorithms and that the difference in accuracy was particularly large for small target training sizes. The difference between the three multi-task algorithms is rather small, with RW_LR + M_FS being most accurate overall. Interestingly, S_LR + M_FS is a very competitive combination of single-task classifier and multi-task feature selection. When $N^- = N^+ = 5$, the difference between the four algorithms decreased and, in few cases, we could even observe slight negative transfer for the multi-task algorithms. In one particular case (prostate cancer as the target task), S_LR + S_FS was better than S_LR + M_FS. The reason might be that the biological mechanism and biomarkers of prostate cancer are very different from other eight cancers.

Figure 3 Comparison of four LR based classifiers on nine target tasks for varying training sizes



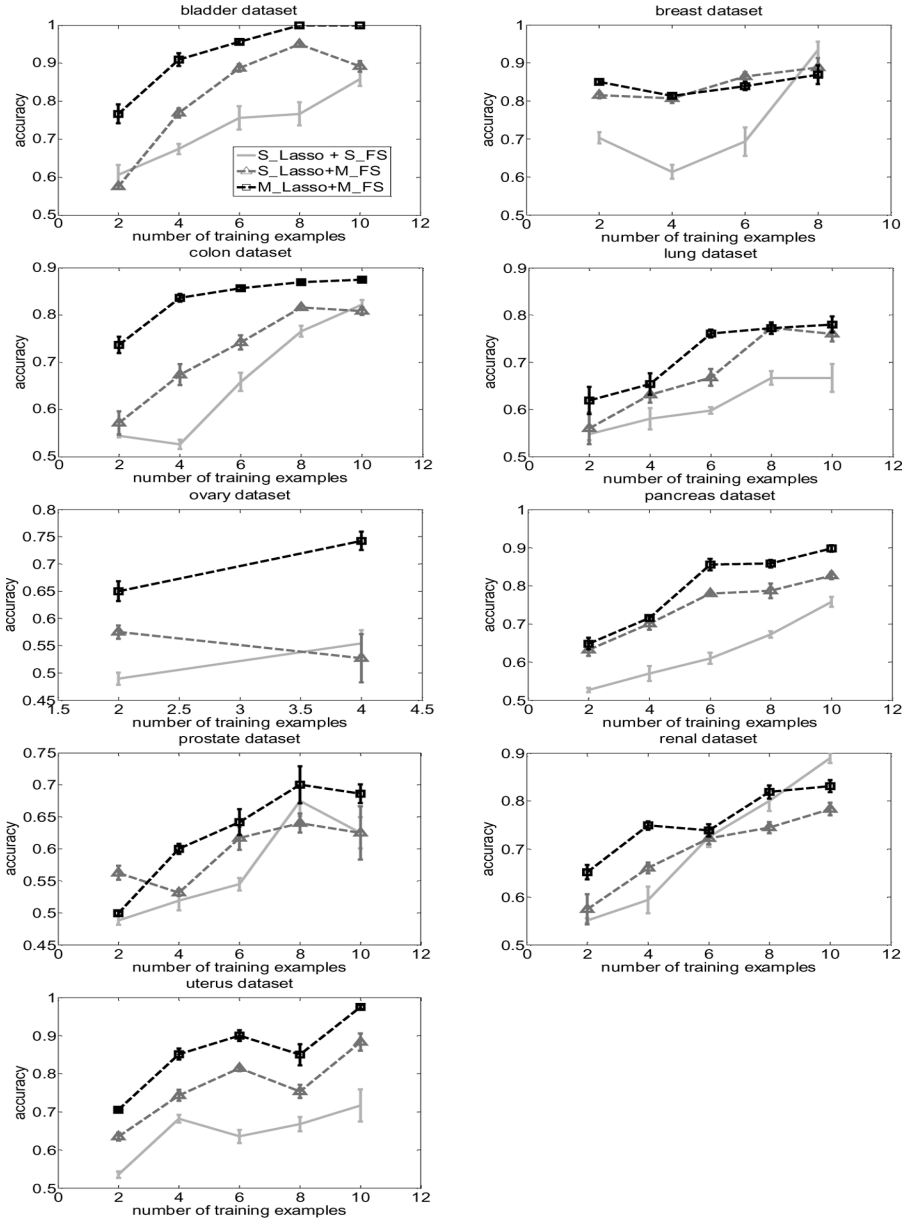
In Figure 4, we compared three different linear SVM-based classifiers where the number of selected features was 100. The results are similar to those in Figure 3. Overall, RW_SVM + M_FS was the most accurate model while S_SVM + S_FS was the least accurate. The negative transfer can also be seen when prostate cancer is the target task.

Figure 4 Comparison of three SVM based classifiers on nine target tasks for varying training sizes



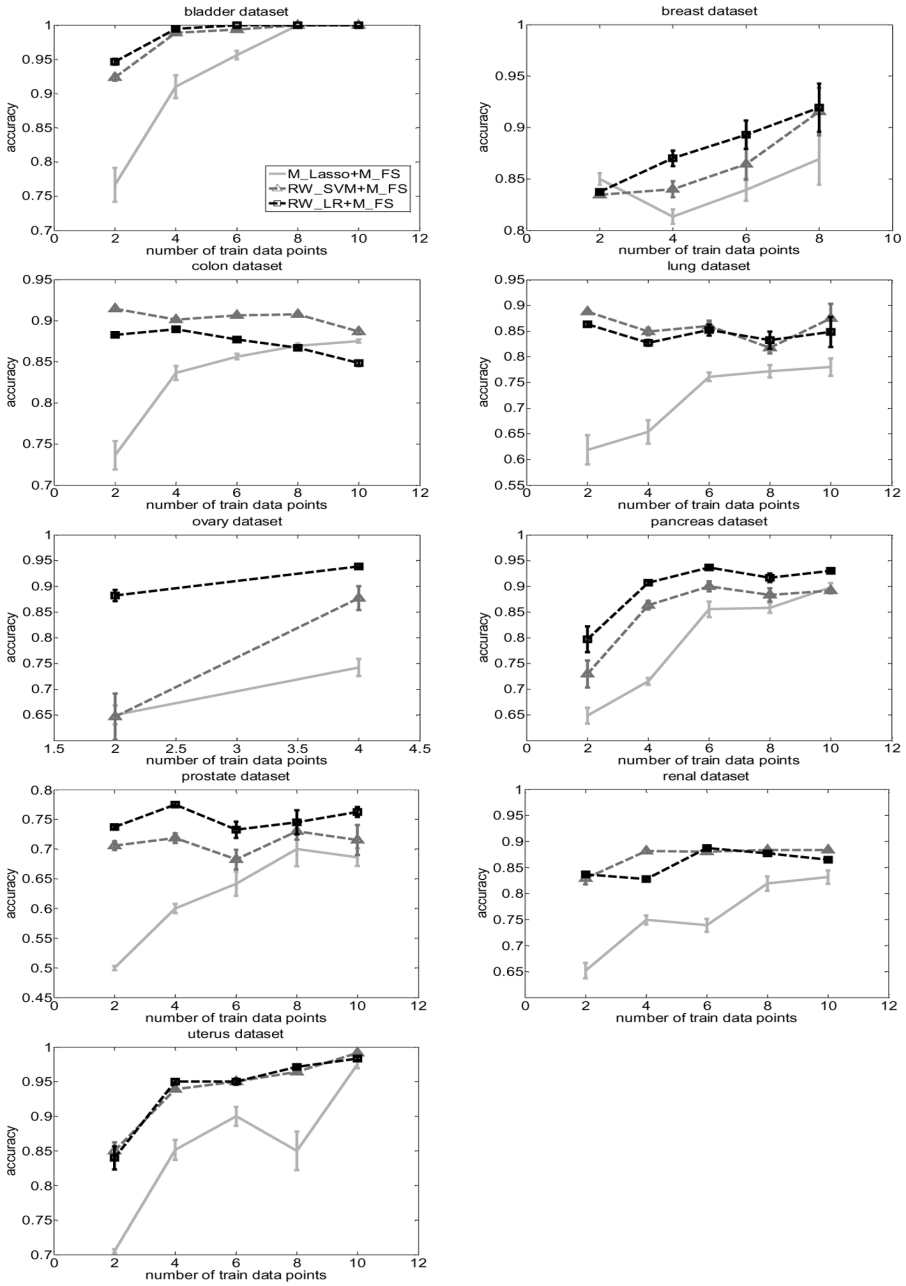
In Figure 5, we evaluated three algorithms based on Lasso, where the number of selected features was 100. It can be concluded that the M_FS filter increases the accuracy of both S_Lasso and M_Lasso. Overall, the M_Lasso + M_FS achieved the best results. Because Lasso results in sparse predictors, the number of final features used in the LR model is actually less than 100. We found that the number of features used in S_Lasso model was in the range between 1 and 15 and that the number of features used in M_Lasso model was in the range between 20 and 40.

Figure 5 Comparison of three Lasso based classifiers on nine target tasks for varying training sizes



Comparing different classification algorithms: In Figure 6, we compare performance of three different multi-task learning algorithms with 100 M_FS selected features. RW_LR is the most accurate on six data sets, while RW_SVM is the best on the remaining three data sets. Both algorithms are significantly more accurate than M_Lasso. The difference between the three algorithms decreases with the training size.

Figure 6 Comparison of multi-task version of Lasso, SVM, LR classifiers on nine target tasks for varying training sizes



6 Related work

Transfer or Multi-Task Learning has attracted significant attention in recent years. The objective of transfer learning is to improve the accuracy on target classification task by borrowing strength from similar auxiliary tasks. Previous work (Caruana, 1997; Ando and Zhang, 2005; Bickel et al., 2008) showed that the transfer learning technology can be very useful in many real-life applications.

One approach for transfer learning attempts to transfer knowledge from auxiliary data sets on the example level. Most often, this entails using examples from auxiliary data sets for training the target task classifier. For example, Bickel et al. (2008, 2009) proposed a reweighting approach, which compares the distributions of auxiliary and target tasks. Different types of reweighting approaches were proposed by Wu and Dietterich (2004) and Liao et al. (2005). Another approach attempts to transfer knowledge from auxiliary data sets at the feature level. These methods try to learn the common feature representation across all tasks (Ando and Zhang, 2005; Argyriou et al., 2008; Obozinski et al., 2010). The underlying assumption is that multiple classification tasks share a common predictive feature structure and that the feature structure can be more reliably estimated by considering all tasks together. Obozinski et al. (2010) proposed an approach that encourages similar sparsity models for all tasks by penalising the sum of l_2 norms of the blocks of coefficients associated with each feature across different classification tasks. Similar idea of using a block l_1/l_2 norm for feature selection in multi-task learning setting was also independently proposed by Argyriou et al. (2008). A third type of transfer learning is attempting to incorporate prior information by assuming the models of target and auxiliary tasks share the same prior distribution (Marx et al., 2005; Raina et al., 2006). The prior information could be learned from the auxiliary tasks or from the domain knowledge. For example, in the microarray classification problem, the prior information about the genes (features) can be mined from the Gene Ontology (Harris et al., 2004) or KEGG (Kanehisa et al., 2004) database.

Feature selection is one of the main issues in microarray classification. Feature selection filters are among the most successful algorithms in practice despite their simplicity. Wrapper or embedded feature selection methods consult the prediction model and offer an opportunity to construct more accurate classifiers. Lasso (Tibshirani, 1996) and Elastic Net (Zou and Hastie, 2005) are two popular embedded feature selection methods for microarray classification. Both of these two methods use the l_1 norm to encourage a sparse solution of the model. The Lasso and Elastic Net can be efficiently solved by the recently proposed algorithm LARS (Efron et al., 2004). Li and Li (2008) extend the idea of the elastic net by incorporating the prior information about the genes (e.g., information about gene networks and pathways, or functional gene properties) into the regularisation term. The prior information about the genes could be easily obtained from Gene Ontology (Harris et al., 2004) or KEGG (Kanehisa et al., 2004). Another type of embedded methods uses the weight of each feature in the model. The Recursive Feature Elimination (RFE) proposed by Guyon et al. (2002) involves an iterative procedure where a linear SVM is trained at each iteration and features corresponding to the smallest absolute weights are discarded. Weston et al. (2003) proposed a method for feature selection by minimising the zero-norm of the parameters of linear model.

The l_2-AROM method gives a simple but practical solution to the original NP-hard problem of minimising the zero-norm. So, the zero-norm minimising problem can be solved efficiently by iteratively calling the standard SVM solver on rescaled inputs. Helleputte and Dupont (2009) extend the AROM method such that the prior knowledge about the domain can be more easily incorporated.

7 Conclusion

Transfer learning is a very attractive technology for microarray classification because of the availability of rich public repositories of microarray and related data. In this paper, we proposed a multi-task feature selection filter suitable for microarray classification. The filter can be used as a pre-processing 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.

Acknowledgement

This study was supported by NSF grant IIS-0546155.

References

Ando, R. and Zhang, T. (2005) 'A framework for learning predictive structures from multiple tasks and unlabeled data', *Journal of Machine Learning Research*, Vol. 6, pp.1817–1853.

Argyriou, A., Evgeniou, T. and Pontil, M. (2008) 'Convex multi-task feature learning', *Machine Learning*, Vol. 73, No. 3, pp.243–272.

Bickel, S., Bogojeska, J., Lengauer, T. and Scheffer, T. (2008) 'Multi-task learning for HIV therapy screening', *Proceedings of 25th International Conference on Machine Learning*, Helsinki, Finland, pp.56–63.

Bickel, S., Sawade, C. and Scheffer, T. (2009) 'Transfer learning by distribution matching for targeted advertising', *Advances in Neural Information Processing Systems*, pp.145–152.

Caruana, R. (1997) 'Multitask learning', *Machine Learning*, Vol. 28, pp.41–75.

Cawley, G.C. and Talbot, N.L.C. (2006) 'Gene selection in cancer classification using sparse logistic regression with Bayesian regularization', *Bioinformatics*, Vol. 22, No. 19, pp.2348–2355.

Efron, B., Hastie, T., Johnstone, I. and Tibshirani, R. (2004) 'Least angle regression', *Annual Statistics*, Vol. 32, pp.407–499.

Guyon, I., Weston, J., Barnhill, S. and Vapnik, V. (2002) 'Gene selection for cancer classification using support vector machines', *Machine Learning*, Vol. 46, pp.389–422.

Harris, M.A., Clark, J., Ireland, A., Lomax, J., Ashburner, M., Foulger, R., Eilbeck, K., Lewis, S., Marshall, B., Mungall, C., Richter, J., Rubin, G.M., Blake, J.A., Bult, C., Dolan, M., Drabkin, H., Eppig, J.T., Hill, D.P., Ni, L., Ringwald, M., Balakrishnan, R., Cherry, J.M., Christie, K.R., Costanzo, M.C., Dwight, S.S., Engel, S., Fisk, D.G., Hirschman, J.E., Hong, E.L., Nash, R.S., Sethuraman, A., Theesfeld, C.L., Botstein, D., Dolinski, K., Feierbach, B., Berardini, T., Mundodi, S., Rhee, S.Y., Apweiler, R., Barrell, D., Camon, E., Dimmer, E., Lee, V., Chisholm, R., Gaudet, P., Kibbe, W., Kishore, R., Schwarz, E.M., Sternberg, P., Gwinn, M., Hannick, L., Wortman, J., Berriman, M., Wood, V., de la Cruz, N., Tonellato, P., Jaiswal, P., Seigfried, T. and White, R. (2004) 'Gene Ontology Consortium (2004) 'The Gene Ontology (GO) database and informatics resource'', *Nucleic Acids Res.*, Vol. 32 (database issue), pp.D258–D261.

Helleputte, T. and Dupont, P. (2009) 'Partially supervised feature selection with regularized linear models', *Proceedings of the 26th International Conference on Machine Learning*, Quebec, Canada, pp.409–416.

Kanehisa, M., Goto, S., Kawashima, S., Okuno, Y. and Hattori, M. (2004) 'The KEGG resource for deciphering the genome', *Nucleic Acids Res*., Vol. 32, pp.D277–D280.

Krishnapuram, B., Carin, L., Figueiredo, M. and Hartemink, A. (2005) 'Sparse multinomial logistic regression: fast algorithms and generalization bounds', *Machine Intelligence*, Vol. 27, No. 6, pp.957–968.

Li, C. and Li, H. (2008) 'Network-constrained regularization and variable selection for analysis of genomic data', *Bioinformatics*, Vol. 24, pp.1175–1182.

Liao, X., Xue, Y. and Carin, L. (2005) 'Logistic regression with an auxiliary data source', *Proceedings of the 22nd International Conference on Machine Learning*, Bonn, Germany, pp.505–512.

Lin, C.F. and Wang, S.D. (2002) 'Fuzzy support vector machine', *IEEE Transactions on Neural Networks*, Vol. 13, pp.464–471.

Marx, Z., Rosenstein, M.T., Kaelbling, L.P. and Dietterich, T.G. (2005) 'Transfer learning with an ensemble of background tasks', *NIPS 2005 Workshop on Transfer Learning*, Whistler, BC.

Minka, T.P. (2003) *A Comparison of Numerical Optimizers for Logistic Regression*, Technical Report 758, Carnegie Mellon University, Department of Statistics.

Obozinski, G., Taskar, B. and Jordan. M. (2010) 'Joint covariate selection and joint subspace selection for multiple classification problems', *Statistics and Computing*, Vol. 20, No. 2, pp.231–252.

Radivojac, P., Obradovic, Z., Dunker, A.K. and Vucetic, S. (2004) 'Characterization of permutation tests for feature selection', *Proc. 15th European Conference on Machine Learning*, Pisa, Italy, pp.334–346.

Raina, R., Ng, A.Y. and Koller, D. (2006) 'Constructing informative priors using transfer learning', *Proc. 23rd International Conference on Machine Learning*, Pittsburgh, PA, pp.713–720.

Rakhlin, A. (2007) *Transfer Learning Toolkit*, http://multitask.cs.berkeley.edu

Ramaswamy, S., Tamayo, P., Rifkin, R., Mukherjee, S., Yeang, C-H., Angelo, M., Ladd, C., Reich, M., Latulippe, E., Mesirov, J.P., Poggio, T., Gerald, W., Loda, M., Lander, E.S. and Golub, T.R. (2001) 'Multi-class cancer diagnosis using tumor gene expression signatures', *Proceedings of the National Academy of Sciences*, Vol. 98, pp.15149–15154.

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

Tai, F. and Pan, W. (2007) 'Incorporating prior knowledge of predictors into penalized classifiers with multiple penalty terms', *Bioinformatics*, Vol. 23, pp.1775–1782.

Tibshirani, R. (1996) 'Regression shrinkage and selection via the lasso', *Journal of Royal Statistical Society: Series B*, Vol. 58, pp.267–288.

Vapnik, V. (1995) *The Nature of Statistical Learning Theory*, Springer, New York.

Weston, J., Elisseef, A., Schlkopf, B. and Tipping, M. (2003) 'Use of the zero-norm with linear models and kernel methods', *Journal of Machine Learning Research*, Vol. 3, pp.1439–1461.

Wu, P. and Dietterich, T. (2004) 'Improving SVM accuracy by training on auxiliary data sources', *Proceedings of the 21st International Conference on Machine Learning*, Morgan Kaufmann, pp.871–878.

Zou, H. and Hastie, T. (2005) 'Regularization and variable selection via the elastic net', *Journal of the Royal Statistical Society: Series B*, Vol. 67, pp.301–320.