Learning using Unselected Features (LUFe)

Joseph G. Taylor[1], **Viktoriia Sharmanska**[1], **Kristian Kersting**[2], **David Weir**[1], **and Novi Quadrianto**[1]
[1]SMiLe CLiNiC and TAG Lab, University of Sussex, Brighton, UK
[2]Technische Universität Dortmund, Dortmund, Germany

Abstract

Feature selection has been studied in machine learning and data mining for many years, and is a valuable way to improve classification accuracy while reducing model complexity. Two main classes of feature selection methods - filter and wrapper - discard those features which are not selected, and do not consider them in the predictive model. We propose that these unselected features may instead be used as an additional source of information at train time. We describe a strategy called Learning using Unselected Features (LUFe) that allows selected and unselected features to serve different functions in classification. In this framework, selected features are used directly to set the decision boundary, and unselected features are utilised in a secondary role, with no additional cost at test time. Our empirical results on 49 textual datasets show that LUFe can improve classification performance in comparison with standard wrapper and filter feature selection.

1 Introduction

Feature selection is a standard procedure in machine learning and data mining, where a subset of the attributes available in a dataset is chosen to build a predictive model, or attributes are re-weighted based on importance to the model.

Current feature selection methods are broadly grouped into three categories. *Filter* methods rank attributes individually, according to some property such as statistical comparison with the labels. *Wrapper* methods iteratively select and evaluate different subsets of attributes in terms of classifier performance. *Embedded* methods perform feature selection concurrently with learning the classifier's parameters.

There is a division between these feature selection methods. Filter and wrapper methods can be summarised as *combinatorial* methods, because they involve the binary decision to either include or exclude each feature, whereas embedded methods are *continuous*, and involve varying the non-discrete weight of each feature. Combinatorial methods produce a 'local view', where only the selected subset is used to build the classifier, whereas continuous methods produce a 'global view', where the predictive model can access all the features. These two different approaches to feature selection have different benefits and drawbacks. Combinatorial methods have the advantage of being less computationally expensive at test time, as they only consider the selected features. However, this local view means that once a feature is unselected, it will not be considered at all in the subsequent classification. Conversely, the entire feature space is taken into account by the global view of continuous methods, but this has the disadvantage of higher computational cost.

In this paper we explore the space between combinatorial and continuous approaches. Specifically, we propose a method to enhance combinatorial feature selection, in order to gain the advantage of a global view, while maintaining lower complexity at test time. This method therefore combines the respective benefits of both combinatorial and continuous feature selection. This is achieved through the inclusion of the unselected features into the predictive model, in a secondary role. We therefore call this technique Learning using Unselected Features (LUFe).

The conceptual contribution of this research is to allow different features to serve different functions in classification, with feature selection used to assign these roles. In this framework, the most informative features are used directly in setting the decision boundary, while the less informative features play an indirect role in guiding the learning process. We conjecture that the unselected features help by constraining the feasible set which is searched for the decision boundary.

We provide exhaustive experimental evaluation on 49 textual datasets; the results demonstrate that LUFe can indeed improve classification performance compared to traditional combinatorial feature selection, without incurring extra costs at test time. We demonstrate that this improvement occurs for both filter and wrapper feature selection methods.

2 Background and Related Work

In the context of data analysis, a number of learning tasks can be performed on a given set of training data. These include classification, regression, ranking, and novelty detection. However, before this learning is performed, the dimensionality of the data (number of features), is typically reduced, for a number of reasons [Kohavi and John, 1997; Molina *et al.*, 2002; Guyon and Elisseeff, 2003; Navot *et al.*, 2005]. An excessive number of features leads to higher

data collection cost, greater difficulty in model interpretation, higher computational cost for the classifier, and in many cases, decreased generalization ability [Song *et al.*, 2012]. Feature selection is therefore an important step in many data mining and machine learning tasks.

Feature selection methods are grouped into three broad categories: filter, wrapper, and embedded approaches. Filter approaches evaluate feature informativeness based on correlation criteria between input data and its labels such as Pearson's Correlation [Van't Veer *et al.*, 2002], mean differences between classes such as t-statistics [Smyth, 2004], or the generalization of those two approaches in terms of non-linear dependency measure between data and labels with Hilbert-Schmidt Independence Criterion (HSIC) [Song *et al.*, 2012].

Wrapper methods assess a feature's value by using a particular learning method of interest. The de facto example of wrapper techniques is recursive feature elimination (RFE) [Guyon *et al.*, 2002], which 'wraps' a support vector machine (this is referred to as RFE-SVM) and removes features with low SVM weights. RFE relies on a backward feature selection process, which starts with a full set of features and iteratively deletes the least informative ones. Forward feature selection methods, which start with an empty set and iteratively add informative features, are also available.

Sparse regularization methods, such as those based on an L1 regularization term, are an example of embedded techniques. For these methods, feature selection is performed as part of the statistical estimation procedure. They are generally less computationally expensive and less prone to over-fitting [Guyon and Elisseeff, 2003] than wrapper methods. Embedded methods are not without computational challenges; this is mostly caused by the non-smooth nature of the regularization term. However, due to growing interest, efficient solvers have been proposed for logistic regression with L1 regularization [Lee *et al.*, 2006; Koh *et al.*, 2007], support vector machine with L1 regularization [Zhu *et al.*, 2004], and recently using proximal algorithms [Parikh and Boyd, 2014].

Our work expands upon an earlier proof of concept for the usage of discarded attributes [Caruana and de Sa, 2003], wherein these attributes were used as extra outputs in a multi-task learning setting [Caruana, 1997]. Conversely, our approach incorporates this information as a secondary input at train time. We provide more extensive and robust evidence of the benefit of incorporating discarded features, beyond the usage in multi-task learning shown in that paper.

The LUFe model described in this paper is designed to allow the classifier to employ a global view of the entire feature space, while using combinatorial feature selection. Wu et al. followed similar motivation to propose Bayesian model averaging (BMA) as an alternative to feature selection [Wu *et al.*, 2015]. This model relies on averaging over multiple classifiers, which each uses a different feature subset. The resultant averaged model considers the exponential-sized 'power set' of all possible feature subsets, which effectively enables access to a global view. However, this approach is restricted to a naïve Bayes classifier, in order to exploit the assumption of conditional independence between features, to make the problem computationally tractable. In contrast, our proposed framework is general in principle, as it works with any suitable choice of classifier. Furthermore, it does not involve extra computational cost at test time.

Learning Using Privileged Information (LUPI) is a learning paradigm that allows extra information, that is available only for training data, to be incorporated into the classifier [Vapnik and Vashist, 2009]. This 'privileged information' is portrayed as highly informative data, comparable to the assistance of a teacher during human learning. It is of a different modality to the standard information, and is unavailable at test time. The LUFe technique described in this paper takes inspiration from LUPI. However, our interpretation is novel, as the secondary data source consists of features which are of the same modality, and have been designated as *less* informative, in contrast to the privileged information described by [Vapnik and Vashist, 2009]. We will further discuss the relations between LUFe and LUPI in the remark at the conclusion of Section 3.

3 Learning using Unselected Features (LUFe)

Let input $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ and output $Y = \{y_1, \ldots, y_N\}$. In a typical learning setting, we are given a set of N input-output data points $(X, Y) = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)\} \subset \mathcal{X} \times \mathcal{Y}$. In this paper, we focus on a binary classification task, that is $\mathcal{Y} = \{+1, -1\}$ and D dimensional feature representation of the data, that is $\mathcal{X} = \mathbb{R}^D$. We seek to infer a latent binary classification function $f : \mathcal{X} \to \{+1, -1\}$ that is taken from a particular function space $\mathcal{F}$. The inferred classifier f will then be used to attach a label y_{new} for a new input $\mathbf{x}_{\text{new}}$. We assume that it is necessary to reduce data dimensionality for the purpose of reducing storage and computational requirements at deployment time. To achieve this, we will focus on filter and wrapper approaches.

Trading off between the number of features used and the (regularised) training error has usually been solved in a two-stage approach [Weston *et al.*, 2003]: feature selection via filter or wrapper, followed by minimizing the training error in a regularised risk functional framework [Vapnik, 1995]. Here feature selection can be understood as a combinatorial optimization problem. We denote a full set of features as $\mathcal{T}$. Each element in the set $\mathcal{T}$ is corresponds to one data dimension, therefore, we have $|\mathcal{T}| = D$. The goal of feature selection techniques is to select a subset of features $\mathcal{S} \subseteq \mathcal{T}$ such that this subset contains the most non-redundant information of X. Suppose we have defined a feature quality functional $\mathcal{Q}(\mathcal{S})$ that captures the informativeness of a feature subset; this is computed by restricting the data X to have only features that are contained in $\mathcal{S}$. The combinatorial problem of feature selection can then be formulated as follows:

$$\widehat{\mathcal{S}} = \underset{\mathcal{S} \in \mathcal{T}}{\arg\max}\ \mathcal{Q}(\mathcal{S}) \tag{1a}$$

$$\text{subject to } |\mathcal{S}| \leq K, \tag{1b}$$

where K is an upper bound on the desired number of features. We will subsequently denote the selected feature subset as $\widehat{\mathcal{S}}$ and the unselected feature subset as $\widehat{\mathcal{U}} := \mathcal{T} \setminus \widehat{\mathcal{S}}$. It is generally an NP-hard problem to find a global solution to the formulation in (1) [Weston *et al.*, 2003]. In practice, a greedy approach such as forward feature selection, backward

feature selection, or a mixed approach is adopted [Guyon *et al.*, 2002]. There are numerous candidates for the feature criterion $\mathcal{Q}(\cdot)$ in the data mining and machine learning literature (vide section 2).

It is now worth highlighting the following two important observations regarding the trade off between the number of features used and the (regularised) training error in the context of filter and wrapper feature selection methods.

Observation 1: Once a subset of features is unselected, they will not be considered in the subsequent regularized risk minimization. Standard feature selection methods simply discard those features $\widehat{\mathcal{U}}$ that were not chosen for subset $\widehat{\mathcal{S}}$. This standard practice has been only rarely disputed, such as by [Caruana and de Sa, 2003] (as discussed in section 2).

Observation 2: Removing unselected features directly translates to reducing the complexity of the classification function f. Less complex models will produce lower prediction variability (less variance), but at the expense of higher bias with respect to the correct value. Bias-variance trade off translates directly into the generalization performance of the classifier. Hence, it is desirable to reduce the bias in the context of filter and wrapper feature selection methods.

The main conceptual contribution of this paper is to *rethink* the practice of filter and wrapper feature selection approaches that perform a local view of regularized risk minimization. Triggered by the two observations above, we ask the following question: *can we take both selected and unselected features into account in different roles during the regularized risk minimization, to bias the learning process of a classifier towards better generalization performance?* It is important to note that the unselected features will not be available *at deployment time*, therefore, they can not be used as a direct input to the latent function f.

The main intuition of our proposed Learning using Unselected Features (LUFe) method is that the feature criterion computed for unselected features, i.e. $\mathcal{Q}(\widehat{\mathcal{U}})$, can be used to define a *data-dependent* upper bound on the classifier's loss function incurred using selected features. In effect, the unselected features will constrain the feasible set which is searched for the classifier's decision boundary. This 'upper bound' is a heuristic, which depends on the choice of feature evaluation metric.

Definition 1: for each data point $\mathbf{x}_i$, the data-dependent upper bound on the classifier's loss incurred when using selected features $\mathbf{x}_i^{\widehat{\mathcal{S}}}$ is defined as $\mathcal{Q}^i(\widehat{\mathcal{U}}) = \left\langle \mathbf{x}_i^{\widehat{\mathcal{U}}}, \mathcal{Q}(\widehat{\mathcal{U}}) \right\rangle$, where $\mathbf{x}_i^{\widehat{\mathcal{U}}}$ is the restriction of a data point $\mathbf{x}_i$ to have only unselected features and we have abused the notation $\mathcal{Q}(\widehat{\mathcal{U}})$ to denote an array of computed feature criterion on all singletons of $\widehat{\mathcal{U}}$.

The following assumption is needed for LUFe.

Assumption 1: The feature criterion $\mathcal{Q}(\cdot)$ is non-negative.

Assumption 1 is rather a weak assumption and is fulfilled in *almost all* criteria of feature selection methods, such as, RFE [Guyon *et al.*, 2002], HSIC [Song *et al.*, 2012], mutual information [Lefakis and Fleuret, 2014], and leverage score [Paul *et al.*, 2015].

Consider now the linear form of a classifier function $f(\mathbf{x}) := \langle \mathbf{w}, \mathbf{x} \rangle + b$, the LUFe optimization problem can then be described as follows:

$$\underset{\mathbf{w},b}{\text{minimize}} \ \|\mathbf{w}\|_{\ell_2}^2 \tag{2a}$$

subject to, for all $i = 1, \ldots, N$,

$$\underbrace{1 - y_i[\langle \mathbf{w}, \mathbf{x}_i^{\widehat{\mathcal{S}}} \rangle + b]}_{\substack{\text{classifier's loss} \\ \text{based on selected features}}} \leq \underbrace{\left\langle \mathbf{x}_i^{\widehat{\mathcal{U}}}, \mathcal{Q}(\widehat{\mathcal{U}}) \right\rangle}_{\substack{\text{data-dependent upper bound} \\ \text{based on unselected features}}}. \tag{2b}$$

The above problem formulation resembles hard-margin SVMs with a *data-dependent margin* for i-th data point defined as $1 - \mathcal{Q}^i(\widehat{\mathcal{U}})$. The constraint in (2b) enforces the following:

- A *small* value of data-dependent upper bound $\mathcal{Q}^i(\widehat{\mathcal{U}})$ means the informativeness of unselected features is *relatively low* and of selected features is *relatively high*, therefore we expect the classifier based on selected features to *perform well*. This is reflected by a small upper bound value on the classifier's loss.
- On the contrary, a *large* value of data-dependent upper bound $\mathcal{Q}^i(\widehat{\mathcal{U}})$ means the informativeness of unselected features is *relatively high* and of selected features is *relatively low* (albeit it is of course still higher than the informativeness of unselected features), therefore we should not expect the classifier to perform well. This is reflected by a large upper bound value on the classifier's loss.

Based on Definition 1, we can further generalize the LUFe formulation in (2) by replacing the array of computed feature criterion on all singletons $\mathcal{Q}(\widehat{\mathcal{U}})$ with an unknown weight vector $\mathbf{w}^\star$. The general formulation of LUFe is now:

$$\underset{\mathbf{w},b,\mathbf{w}^\star}{\text{minimize}} \ \|\mathbf{w}\|_{\ell_2}^2 + \lambda_1 \|\mathbf{w}^\star\|_{\ell_2}^2 + \lambda_2 \sum_{i=1}^{N} \left\langle \mathbf{x}_i^{\widehat{\mathcal{U}}}, \mathbf{w}^\star \right\rangle \tag{3a}$$

subject to, for all $i = 1, \ldots, N$,

$$1 - y_i[\langle \mathbf{w}, \mathbf{x}_i^{\widehat{\mathcal{S}}} \rangle + b] \leq \left\langle \mathbf{x}_i^{\widehat{\mathcal{U}}}, \mathbf{w}^\star \right\rangle \tag{3b}$$

$$\left\langle \mathbf{x}_i^{\widehat{\mathcal{U}}}, \mathbf{w}^\star \right\rangle \geq 0. \tag{3c}$$

In the above, the weight vector $\mathbf{w}$ is a K-dimensional vector and $\mathbf{w}^\star$ is a $(D\text{-}K)$-dimensional vector. The scalar parameters λ_1 and λ_2 are the trade-off parameters. The second and third terms of the objective function in (3a) are added to ensure that the *pseudo* feature criterion values on all singletons do not take unreasonably large values. Assumption 1 is enforced by the constraint in (3c). The optimization problem in (3) can be solved in the dual representation using a standard quadratic programming (QP) solver. For the pseudocode of LUFe, please refer to algorithm 1.

Remark The LUFe formulation in (3) coincides with the SVM+ [Vapnik and Vashist, 2009] algorithm for LUPI when the unselected feature representation in LUFe is interpreted as an additional data modality in SVM+. There are two clear distinctions between LUFe and LUPI: first, LUPI considers

Algorithm 1 Learning using Unselected Features (LUFe)

Input a set of data points $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$, $\mathbf{x}_i \in \mathbb{R}^D$, an upper bound on the number of selected features K, and trade-off parameters λ_1 and λ_2
Apply a feature selection method to produce a feature subset $\widehat{\mathcal{S}}$ with $\widehat{\mathcal{S}} \subseteq \mathcal{T}$ and $|\widehat{\mathcal{S}}| \leq K$
Solve an optimization problem in (3)
Return a classifier f that works on selected feature subset $\widehat{\mathcal{S}}$ but does not discard the $\widehat{\mathcal{U}}$ features during training.

two different data modalities, $\mathcal{X}$ and $\mathcal{X}^\star$, for example images and texts, whereas selected $\widehat{\mathcal{S}}$ and unselected $\widehat{\mathcal{U}}$ features in LUFe express the same modality. Secondly, the interpretation differs where LUPI uses privileged information to discriminate between easy and difficult examples in the privileged space $\mathcal{X}^\star$ and subsequently transfer this information to the original data space $\mathcal{X}$ [Vapnik and Vashist, 2009; Sharmanska *et al.*, 2013; Hernández-Lobato *et al.*, 2014; Vapnik and Izmailov, 2015]. Conversely, LUFe uses features which have been designated as *less* informative as the secondary data source, therefore the interpretation of easy and hard transfer between privileged and original data does not explain LUFe's behaviour. We instead consider the unselected features as a *data-dependent* upper bound on the classifier's loss function incurred using selected features.

4 Experiments

This paper aims to address the following questions:

1. Following combinatorial feature selection, can classifier performance be improved by Learning using Unselected Features?
2. Is a performance improvement consistent across different combinatorial feature selection methods?
3. Can the performance be improved further by using only a subset of unselected features?

This initial experimentation will focus on the application of LUFe to SVM - one of the most widely-used learning methods. However, our notion of using unselected features to upper bound the loss is, in principle, applicable to any classifier.

Dataset We follow the protocol of [Paul *et al.*, 2015], in using a subset of the TechTC-300 collection consisting of 49 datasets, pre-processed to remove all features corresponding to any word of less than 5 characters. The TechTC-300 collection consists of 300 textual datasets, which have baseline SVM error rate uniformly distributed between 0.4 and 0.0.[1]

Model selection A consistent model selection procedure was carried out in all experiments. All experimental settings were tested over 100 repeats, and each repeat, stratified 5-fold cross-validation was used to estimate the λ parameters for each setting. All parameters were selected from seven log-spaced values in the range $\{10^{-3} \ldots 10^{3}\}$). The two SVM+ parameters for *LUFe* (λ_1 and λ_2) were jointly optimised through grid search; that is, 49 combinations were assessed. This LUFe model selection over two parameters added a small computational overhead but the main bottleneck for our method is RFE feature selection.

[1] http://techtc.cs.technion.ac.il/techtc300/techtc300.html

4.1 Experiment 1: Assessing LUFe performance on 49 datasets

Motivation The purpose of this experiment was to assess whether LUFe can provide a boost in classification accuracy compared to standard combinatorial feature selection, by allowing the classifier to have a global view of the data at training. The *RFE* baseline setting consisted of an SVM trained and tested using just the top K features, chosen using feature selection. The *LUFe-RFE* setting was trained with the same top K features used as normal information, but supplemented with the remaining D-K features. A further baseline setting, *ALL*, was a standard SVM trained and tested using all features.

Experimental Protocol We follow the protocol of [Paul *et al.*, 2015] in using RFE to select the top 300 and top 500 most informative features, and in using 10x10-fold cross-validation experiments to compare our technique with the baselines. This protocol was observed in all experiments.

Results The results demonstrate an improvement by LUFe over standard feature selection. While RFE does reduce error rate compared to the *ALL* baseline, LUFe produces a bigger reduction. Results for the 3 settings (*RFE, LUFe-RFE* and *ALL*) across 49 datasests are shown in Figure 1, and summarised in Table 1.

For the $K = 300$ setting, LUFe was better than RFE in 44 of 49 (89.8%) cases, with mean improvement of 1.97%. LUFe performance surpassed all-features performance in 41 of 49 cases (83.67%), with a mean improvement of 4.4%. In comparison, RFE performance bettered all-features in just 31 cases, with mean improvement of 2.5%.

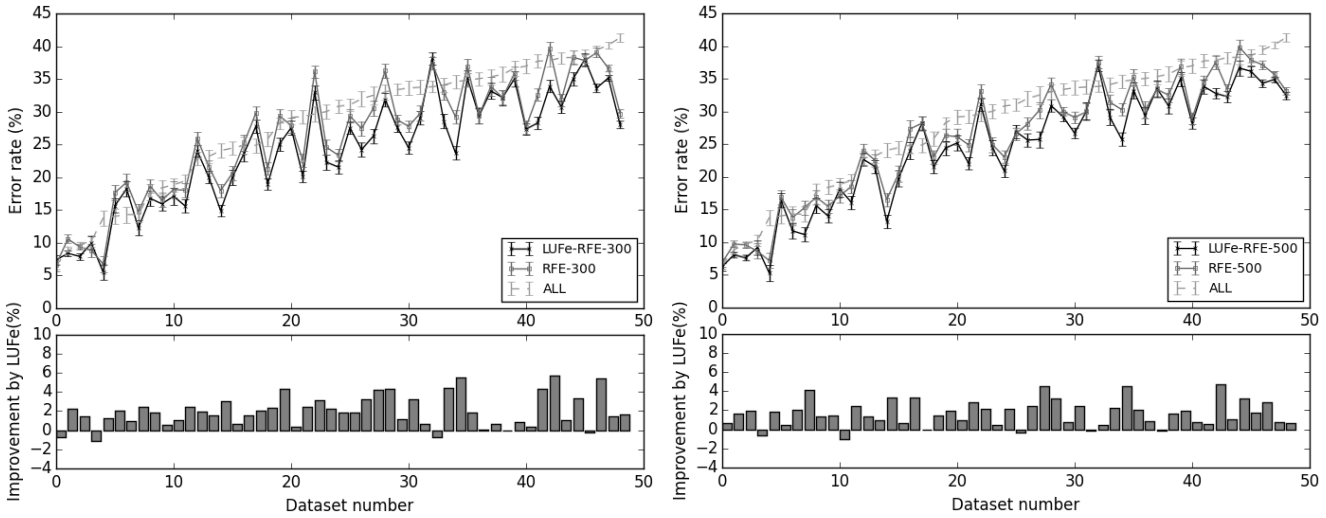
Similar results were observed for $K = 500$; LUFe outperformed RFE in 43 cases (87.76%), with mean improvement of 1.65%. LUFe achieved 3.72% improvement over *ALL*, improving in 45 cases, whereas RFE achieved only 2.07% mean improvement, in 36 cases.

To verify that the improved performance by *LUFe-RFE* was not simply a result of the higher capacity afforded by its additional weight vector $\mathbf{w}^\star$, performance was further compared with an additional setting, *LUFe-RFE-random*. This selected the top K features as before, but replaced the supplementary information with random Gaussian features of the same dimensionality. *LUFe-RFE-random* produced small improvements over *RFE* but was outperformed by *LUFe-RFE* in 37 of 49 datasets.

Concluding remarks These initial findings demonstrate the ability of LUFe to improve classification accuracy beyond RFE performance, on a wide variety of datasets.

4.2 Experiment 2: Assessing the use of LUFe with different feature selection metrics

Motivation Experiment 1 demonstrated that LUFe can exploit the information from features which are not selected by RFE, which would otherwise be lost when these features were discarded. The purpose of this experiment is to extend the application of LUFe to other feature selection methods, beyond



(a) $K = 300$ (b) $K = 500$

Figure 1: Top: Error rates (%) for *ALL*, *RFE* and *LUFe-RFE* settings, across 49 datasets (sorted by performance of *ALL* setting). Bottom: Improvement in accuracy score by *LUFe-RFE* over *RFE* (%)

'wrapper' methods such as RFE. A common sub-class of 'filter' feature selection methods is univariate methods; these assess each attribute individually in terms of some statistical measure. The ANOVA F-value was chosen as an exemplar metric for univariate feature selection.

Experimental Protocol Experimental protocol followed experiment 1, except that features were ranked by ANOVA F-scores, instead of using RFE. The top K features were used to train a baseline classifier; this setting is referred to as *ANOVA*. The same K features were used as primary information in the *LUFe-ANOVA* setting, with the remaining D-K features used as supplementary information.

Results The benefit of LUFe was maintained while using this different feature selection procedure; *LUFe-ANOVA* improved on *ALL* by a larger margin than *ANOVA*. *LUFe-ANOVA-300* was shown to perform better than *ANOVA* in 44 cases, with a mean improvement of 2.65%. *LUFe-ANOVA-300* accuracy exceeded that of the *ALL* in 45 cases, with mean improvement 5.38%. This compares favourably with *ANOVA-300* helped in 37 cases vs baseline with mean improvement=2.73%. Results are shown in Figure 2 and Table 1.

Concluding remarks These findings show that LUFe is still beneficial to classifier performance when using filter, as well as wrapper, feature selection methods.

4.3 Experiment 3: Improving performance with a subset of unselected features

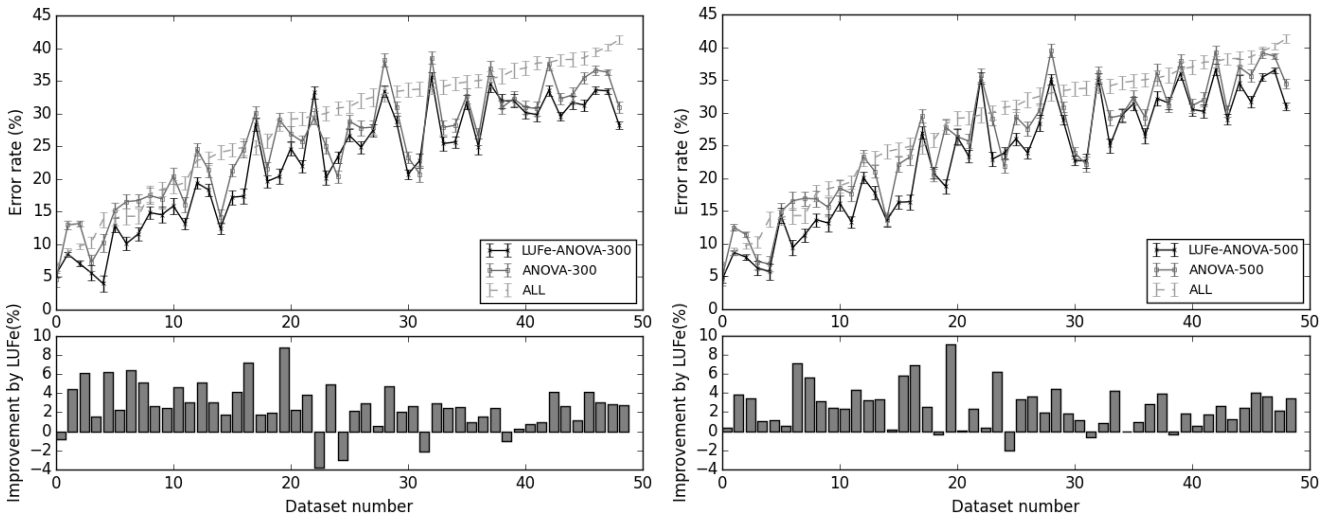
Motivation The final experimental procedure sought to further improve LUFe by incorporating only a specific subset of the unselected features. Whereas Experiments 1 and 2 used the entire set of unselected features as secondary information, it was hypothesised that the performance enhancement may be greater if some selectivity was applied. In much the same way that feature elimination in the domain of standard information can improve classifier performance, discarding some unselected features may also allow a more generalizable model to be constructed, improving classification performance.

Table 1: **Summary Results**
Accuracy scores for different settings, with improvements relative to *ALL*, and to corresponding feature selection setting, using the number of datasets where performance improved (out of 49 datasets), and the difference in mean accuracy score. In the setting column, 300 and 500 refer to the number of selected features.

		Improvements			
		vs ALL		vs RFE/ANOVA	
Setting	Accuracy	Wins (/49)	Mean (%)	Wins (/49)	Mean (%)
ALL	72.04	-	-	-	-
RFE-300	73.85	33	1.81	-	-
LUFe-RFE-300	75.82	41	3.78	44	1.97
RFE-500	74.11	36	2.07	-	-
LUFe-RFE-500	75.76	45	3.72	43	1.65
ANOVA-300	74.77	37	2.73	-	-
LUFe-ANOVA-300	77.42	45	5.38	44	2.65
ANOVA-500	74.26	36	2.22	-	-
LUFe-ANOVA-500	76.81	44	4.77	44	2.55

Experimental Protocol Features were ranked according to RFE, with the top K features used as standard information, as before. The same RFE ranking was then used again, to select the top-ranking $t\%$ of unselected features and only this top $t\%$ of unselected features were used as secondary information in this protocol. The remaining unselected features were not used in training the classifier.

The t parameter was varied over the set $\{10,20,...,100\}$; for t=100, all unselected features are used as secondary information, recovering the *LUFe-RFE* setting from Experiment 1. It was expected that the accuracy score would (a) initially



(a) $K = 300$ (b) $K = 500$

Figure 2: Top: Error rates (%) for *ALL*, *ANOVA* and *LUFe-ANOVA* settings, across 49 datasets (sorted by performance of *ALL* setting). Bottom: Improvement in accuracy score by *LUFe-ANOVA* over *ANOVA* (%)

increase with $t\%$, as more useful features were incorporated into the secondary information and (b) then decrease and converge with the performance seen in Experiment 1, when less useful features were taken into consideration by the classifier.

LUFe-RFE was also compared with an '*RFE-augmented*' setting, wherein the same top *t* unselected features were concatenated with the selected features, and this expanded feature set used to train and test a standard SVM.

Results Figure 3 depicts the variation in classifier performance on all 49 datasets, when trained on a subset of secondary data, according to two metrics: the mean improvement over the *ALL*, *RFE*, *RFE-augmented* baselines, and the number of datasets where performance exceeded *ALL*, *RFE* and *RFE-augmented*.

All settings led to a mean improvement over the *ALL* baseline, but the amount of improvement grew as more secondary information was provided. This suggests that the LUFe paradigm depends strongly on the size, and not just the quality of the secondary information. This growth in performance levelled out, with little difference between using 70% and 100% of secondary data - perhaps as the lowest-ranked 30% of features convey little benefit to the classiifer.

The number of datasets where *LUFe-RFE* outperformed both *ALL* and *RFE-augmented* remained constant at 41. The number where *LUFe-RFE* beat *RFE* grew with increasing amounts of secondary data, peaking at 46 of 49 datasets, when 60 to 80% of unselected features were used, and dropping to 44 when all were used. This suggests that the informativeness of the unselected features may play a part in borderline cases where LUFe can slightly improve over RFE.

Concluding remarks This work demonstrates that the size of performance increase gained by LUFe is dependent on the amount of secondary data provided. This leads to broader research questions, concerning which attributes of secondary data are necessary to gain peak benefit from LUFe.

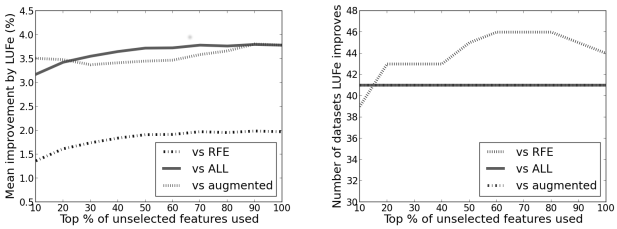


Figure 3: Variation in *LUFe* performance, compared to *RFE* and *ALL* baselines in terms of mean improvement and number of improvements

5 Conclusions and Future Work

We have outlined a promising new approach to feature selection called Learning using Unselected Features, where attributes that are discarded in feature selection are utilised as secondary input to a classifier. This allows a 'global view' of the feature space, without increasing testing time. We have shown that LUFe can outperform standard feature selection practices on a wide range of datasets, and that this enhancement is consistent across both filter and wrapper types of feature selection. Finally, we have carried out introductory research suggesting that LUFe can be improved further by using only a subset of discarded features. Future work will seek to replicate the performance boost due to LUFe on a wider range of datasets, extend our method of using unselected features to upper bound the loss to other classifiers, and formulate a joint optimization problem of LUFe and feature selection by taking a regularization viewpoint [Schölkopf *et al.*, 2001; Kimeldorf and Wahba, 1971]. These research ideas will invigorate research into re-thinking how feature selection is used.

References

[Caruana and de Sa, 2003] Rich Caruana and Virginia R. de Sa. Benefitting from the variables that variable selection discards. *Journal of Machine Learning Research (JMLR)*, 3:1245–1264, 2003.

[Caruana, 1997] Rich Caruana. Multitask learning. *Machine Learning*, 28:41–75, 1997.

[Guyon and Elisseeff, 2003] Isabelle Guyon and André Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research (JMLR)*, 3:1157–1182, 2003.

[Guyon *et al.*, 2002] Isabelle Guyon, Jason Weston, Stephen Barnhill, and Vladimir Vapnik. Gene selection for cancer classification using support vector machines. *Machine Learning*, 46(1-3):389–422, 2002.

[Hernández-Lobato *et al.*, 2014] Daniel Hernández-Lobato, Viktoriia Sharmanska, Kristian Kersting, Christoph H. Lampert, and Novi Quadrianto. Mind the nuisance: Gaussian process classification using privileged noise. In *Neural Information Processing Systems (NIPS)*, 2014.

[Kimeldorf and Wahba, 1971] George Kimeldorf and Grace Wahba. Some results on tchebycheffian spline functions. *Journal of mathematical analysis and applications*, 33(1):82–95, 1971.

[Koh *et al.*, 2007] Kwangmoo Koh, Seung-Jean Kim, Stephen Boyd, and Yi Lin. An interior-point method for large-scale l1-regularized logistic regression. *Journal of Machine Learning Research (JMLR)*, 8:1519–1555, 2007.

[Kohavi and John, 1997] Ron Kohavi and George H. John. Wrappers for feature subset selection. *Artificial Intelligence*, 97(12):273 – 324, 1997.

[Lee *et al.*, 2006] Su-In Lee, Honglak Lee, Pieter Abbeel, and Andrew Y. Ng. Efficient l1 regularized logistic regression. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2006.

[Lefakis and Fleuret, 2014] Leonidas Lefakis and François Fleuret. Jointly informative feature selection. In *Artificial Intelligence and Statistics (AISTATS)*, 2014.

[Molina *et al.*, 2002] L.C. Molina, L. Belanche, and A. Nebot. Feature selection algorithms: a survey and experimental evaluation. In *IEEE International Conference on Data Mining (ICDM)*, 2002.

[Navot *et al.*, 2005] Amir Navot, Ran Gilad-Bachrach, Yiftah Navot, and Naftali Tishby. Is feature selection still necessary? In *Subspace, Latent Structure and Feature Selection, Statistical and Optimization*, 2005.

[Parikh and Boyd, 2014] Neal Parikh and Stephen Boyd. Proximal algorithms. *Foundations and Trends in Optimization*, 1:127–239, 2014.

[Paul *et al.*, 2015] Saurabh Paul, Malik Magdon-Ismail, and Petros Drineas. Feature selection for linear SVM with provable guarantees. In *Artificial Intelligence and Statistics (AISTATS)*, 2015.

[Schölkopf *et al.*, 2001] Bernhard Schölkopf, Ralf Herbrich, and Alex J Smola. A generalized representer theorem. In *Computational Learning Theory (COLT)*, pages 416–426, 2001.

[Sharmanska *et al.*, 2013] Viktoriia Sharmanska, Novi Quadrianto, and Christoph H. Lampert. Learning to rank using privileged information. In *International Conference on Computer Vision (ICCV)*, 2013.

[Smyth, 2004] Gordon K. Smyth. Linear models and empirical bayes methods for assessing differential expression in microarray experiments. *Statistical Applications in Genetics and Molecular Biology*, 3(1), 2004.

[Song *et al.*, 2012] Le Song, Alex Smola, Arthur Gretton, Justin Bedo, and Karsten Borgwardt. Feature selection via dependence maximization. *Journal of Machine Learning Research (JMLR)*, 13(1):1393–1434, 2012.

[Van't Veer *et al.*, 2002] Laura J Van't Veer, Hongyue Dai, Marc J Van De Vijver, Yudong D He, Augustinus AM Hart, Mao Mao, Hans L Peterse, Karin van der Kooy, Matthew J Marton, Anke T Witteveen, et al. Gene expression profiling predicts clinical outcome of breast cancer. *Nature*, 415(6871):530–536, 2002.

[Vapnik and Izmailov, 2015] Vladimir Vapnik and Rauf Izmailov. Learning using privileged information: Similarity control and knowledge transfer. *JMLR*, pages 2023–2049, 2015.

[Vapnik and Vashist, 2009] Vladimir Vapnik and Akshay Vashist. A new learning paradigm: Learning using privileged information. *Neural Networks*, pages 544–557, 2009.

[Vapnik, 1995] Vladimir N. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag New York, Inc., New York, NY, USA, 1995.

[Weston *et al.*, 2003] Jason Weston, André Elisseeff, Bernhard Schölkopf, and Mike Tipping. Use of the zero norm with linear models and kernel methods. *Journal of Machine Learning Research (JMLR)*, 3:1439–1461, 2003.

[Wu *et al.*, 2015] Ga Wu, Scott Sanner, and Rodrigo F.S.C. Oliveira. Bayesian model averaging naive bayes (bma-nb): Averaging over an exponential number of feature models in linear time. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2015.

[Zhu *et al.*, 2004] Ji Zhu, Saharon Rosset, Trevor Hastie, and Rob Tibshirani. 1-norm support vector machines. In *Neural Information Processing Systems (NIPS)*, 2004.