

Feature Selection for Classifying High-Dimensional Numerical Data

Yimin Wu and Aidong Zhang

Department of Computer Science and Engineering, SUNY at Buffalo,
{yiminkwu,azhang}@cse.buffalo.edu

Abstract

Classifying high-dimensional numerical data is a very challenging problem. In high dimensional feature spaces, the performance of supervised learning methods suffer from the curse of dimensionality, which degrades both classification accuracy and efficiency. To address this issue, we present an efficient feature selection method to facilitate classifying high-dimensional numerical data. Our method employs balanced information gain to measure the contribution of each feature (for data classification); and it calculates feature correlation with a novel extension of balanced information gain. By integrating feature contribution and correlation, our feature selection approach uses a forward sequential selection algorithm to select uncorrelated features with large balanced information gain. Extensive experiments have been carried out on image and gene microarray datasets to demonstrate the effectiveness and robustness of the presented method.

1 Introduction

Classification is an effective technique for analyzing the patterns of high dimensional numerical data. Nonetheless, it is a very challenging problem to classify high-dimensional numerical data, because, in high-dimensional feature spaces, the performance of supervised learning methods [14, 10] suffer from *the curse of dimensionality* [10]. In other words, in high-dimensional feature space, supervised learning methods (such as support vector machine and tree classifiers) often face the following problems. First, they need large volume of training data to achieve high generalization accuracy [14, 10]. Second, in the full feature space, the time complexity of training the supervised learning methods may become the bottleneck of the system.

However, the following emerging application areas of pattern recognition involve classifying high-dimensional numerical data with small training data:

- **Relevance Feedback for Multimedia Retrieval.** In multimedia retrieval systems, multimedia objects (such as images or video clips) are often indexed by numerical data with hundreds of dimensions. To im-

prove retrieval quality, relevance feedback [18, 16, 17, 19, 20] has been extensively used to learn the user's query concept and/or preference. Relevance feedback is a typical case where the machine has to learn with small training data, because most existing relevance feedback approaches ask the user to label training data online. Recent researches [18, 16, 17, 19] showed that classification methods can be used to improve the performance of relevance feedback. However, with the small training data from relevance feedback, it is always a challenging problem to train a strong classifier in multimedia retrieval.

- **Bioinformatics.** Bioinformatics data [11, 2] are often represented by real-valued feature vectors with thousands of dimensions, but each dataset only contains less than one hundred training samples.

In the above-mentioned applications, supervised learning methods are often trapped by the curse of dimensionality. To alleviate the curse of dimensionality, one of the most extensively used methods is *feature selection*, which selects important features according to some feature importance criterion. Existing feature selection approaches [1] generally belong to the following two categories: *wrapper* and *filter*. *Wrappers* include the target classifier as a part of their performance evaluation, while *filters* employ evaluation functions independent from the target classifier. Since wrappers train a classifier to evaluate each feature subset, they are much more computationally intensive than filters. Hence, filters are more practical than wrappers in high-dimensional feature spaces. Among different filter approaches (such as exponential and sequential [1]), we are interested in the most efficient method, which is the sequential feature selection (SFS). SFS methods add or remove features using a hill-climbing search strategy; and their time complexity is at most quadratic in the dimensions of the feature space.

To select important features for relevance feedback, we proposed a *dynamic feature extraction* method (DFE) [19] in our previous research. Our DFE measures feature contribution (for data classification) with a criterion known as *balanced information gain* [7, 19]; and it selects a low-dimensional feature subset, which comprises features with large balanced information gain. As shown by our exper-

iments [19], DFE can effectively select important features for relevance feedback. However, DFE does not consider feature correlation, so it only performs optimally when features are mutually independent. Moreover, a related approach [12] evaluates feature contribution using symmetrical uncertainty [15]. Since both symmetrical uncertainty and balanced information gain are balanced version of mutual information [3], theoretical analysis and empirical evidence are required to support the choice between the two alternatives.

In this paper, we present a feature selection filter to facilitate classifying high-dimensional numerical data. We first analyze the advantages and disadvantages of *balanced information gain* and *symmetrical uncertainty*. Our analysis shows that, to evaluate the contribution of numerical features (for data classification), the former is more favorable than the latter. Hence, we employ balanced information gain to measure the contribution of each feature. But unlike DFE, which simply selects features with large balanced information gain, our method takes both feature contribution and correlation into account. To do so, we extend balanced information gain to elicit a feature correlation measurement. We also give a proof for the symmetricity of our feature correlation criterion.

By integrating feature contribution and correlation, our feature importance criterion re-evaluates the importance of features according to their correlation. With our feature importance criterion, we develop a sequential feature selection method termed as *efficient feature selection* (EFS), which selects uncorrelated features with large balanced information gain. Extensive experiments have been carried out on image and gene microarray datasets to demonstrate the effectiveness and robustness of our method as compared against the state-of-the-art approaches [12, 19].

The rest of this paper is organized as follows. We first coin some useful notation and briefly introduce related work in Section 2. Section 3 then presents our feature importance measurement. Our feature selection method is introduced in Section 4. Experimental results are presented in Section 5. Finally, we conclude in Section 6.

2 Related Work

In this section, we introduce some useful notation, related work and the adaptive random forests algorithm [19].

2.1 Useful Notation

We represent the feature space as $F = \{f_1, \dots, f_M\}$, where M is the dimensions of F . We denote the dataset by $db = \{o_1, \dots, o_T\}$, where T is the size of db . To represent each data object $o_t \in db$, we use a real-valued vector (i.e., point) $\vec{o}_t = [o_{t,1}, \dots, o_{t,M}]^T$, where \vec{o}_t is an instance in the feature

space F . From F , we employ our feature selection method to select an M^* dimensional feature subset $F^* \subset F$.

To train a classifier, we obtain the training set $S = \{(s_1, v_1), \dots, (s_N, v_N)\}$, where N is the size of S . In the training set S , each training sample $(s_n, v_n) \in S$ is a labeled object represented by $\vec{s}_n = [s_{n,1}, \dots, s_{n,M}]^T$, where v_n is its class value.

2.2 Related Work

Before presenting the related work, we first introduce two useful definitions from information theory. For a discrete variable X with \mathcal{I} possible values $\{x_i, i = 1, \dots, \mathcal{I}\}$, its *entropy* [14, 3] is defined as follows:

$$H(X) = - \sum_{i=1}^{\mathcal{I}} P(x_i) \log_2(P(x_i)). \quad (1)$$

Given another discrete variable Y , the following *mutual information* $I(X|Y)$ [3] measures information about X carried by Y :

$$I(X|Y) = H(X) + H(Y) - H(X, Y). \quad (2)$$

In machine learning literatures [14, 7], mutual information is often termed as *information gain*

When applied to feature selection, mutual information is biased for excessively multi-valued nominal features and multisplitting numerical feature value ranges [7]. Hence, Hall *et al.* [13, 12] proposed to calculate both feature contribution and correlation using the following *symmetrical uncertainty* [15]:

$$SU(X, Y) = 2 \left[\frac{H(X) + H(Y) - H(X, Y)}{H(X) + H(Y)} \right]. \quad (3)$$

Symmetrical uncertainty not only balances the bias of mutual information, but also gives a symmetrical measurement for feature correlation.

To evaluate the merit of a feature subset $F_S \subset F$, Hall *et al.* [13, 12] suggest the following merit function:

$$Merit(F_S) = \frac{k\bar{r}_{cf}}{\sqrt{k + k(k-1)\bar{r}_{ff}}}, \quad (4)$$

where k is the number of features in F_S ; \bar{r}_{cf} is the mean contribution of features in F_S ; and \bar{r}_{ff} is the average feature correlation. To find the important features, Hall *et al.* [13] suggest using the hill-climbing strategy, which (1) starts with the empty or full feature set; and (2) adds or removes one feature that contributes to the largest increase in the merit value, until no change in merit value can be caused (by adding or removing features). If efficiency is not the key concern, more exhaustive algorithms (such as best first search [12]) can be used.

2.3 Adaptive Random Forests

Given a small training set, most existing supervised learning methods [10, 14] are prone to overfit the training data. To learn with small training data, the most effective method is to train composite classifiers [4, 9, 5], which combine multiple weak classifiers to reduce the risk of overfitting. Random forests [5] is one of the state-of-the-art methods for training composite classifiers. To obtain the composite classifier, random forests employs the bagging [4] technique to combine a collection of randomized classification and regression trees (CART) [6]; and it achieves favorable performance over bagging [4] and AdaBoost [5]. Moreover, random forests is more robust than AdaBoost on noisy data.

Adaptive random forests (ARF) [19] adapts random forests for relevance feedback. In relevance feedback, it is computational intractable to train a regular random forest (RRF) with all training samples, because the number of negative samples increases very quickly. To address this issue, ARF performs a clustering on negative samples. It then trains a random forest with all positive samples and centroids of negative ones. Our extensive experiments [19] showed that ARF achieves comparable retrieval performance against RRF, but runs 2-3 times faster than the latter.

3 Feature Importance

We present our approach for measuring feature importance in this section.

3.1 Feature Contribution

We measure feature contribution (for data classification) and feature correlation with balanced information gain [7], which originates from information theory. Since information theory [3] only deals with discrete variables, we need to discretize numerical features before we can calculate their balanced information gain.

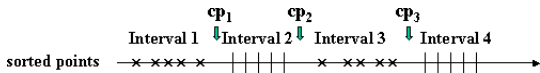


Figure 1: Several useful concepts demonstrated with a two-class situation, where times sign and bar represent samples of two different classes.

Figure 1 demonstrates the common solution [14, 7] for discretizing numerical features. On each feature, we sort all samples in ascending order, and set the mean values of adjacent samples with different classification as potential *cut points* [14, 7] of the discretization. As shown in Figure 1, $t-1$ *cut points* create t continuous *intervals*, which comprise t value ranges on the feature. By considering each value

range (i.e., interval) as an individual value of the given feature, we turn a numerical feature into a discrete variable. To optimize the discretization, we should search the cut points that preserve the maximum class information. This can be achieved by the method presented in [8] and [7].

For each numerical feature, we term the number of value ranges generated by the discretization as the *discretization cardinality*. As shown in Figure 1, if the given three cut points were used to discretize the feature, the discretization cardinality of this feature would be 4. To measure the contribution of numerical feature $f_m \in F$, we employ the following *balanced information gain* [7]:

$$B_g(f_m) = \frac{I(C|f_m)}{\log_2 \kappa}, \quad (5)$$

where C is the class variable and κ is the discretization cardinality of f_m . By considering the class value of each data point as a discrete variable C , $I(C|f_m)$ gives the information gain of feature f_m , while κ places a straightforward penalty on the bias of information gain for excessively multi-valued nominal features and multisplitting numerical feature value ranges [7]. In Formula 5 (as well as the rest of this paper), we simplify the notation by using f_m to denote its discretized counterpart.

So far we have introduced two different criteria for evaluating feature contribution: symmetrical uncertainty (cf. Formula 3) and balanced information gain (cf. Formula 5). To evaluate the contribution of numerical features, balanced information gain is more favorable than symmetrical uncertainty for the following reasons:

- Existing algorithms [7] for searching optimal multi-splitting of numerical features work well with information gain and balanced information gain, but can not directly apply to gain ratio or symmetrical uncertainty. In other words, balanced information gain outperforms symmetrical uncertainty in integrity for evaluating the quality of numerical features.
- Our experimental results indicate that balanced information gain consistently outperforms symmetrical uncertainty for selecting important numerical features on various real-world datasets. These results show that balanced information gain excels symmetrical uncertainty on evaluating the contribution of numerical features (for data classification).

By the above discussions, we measure the contribution of numerical features with balanced information gain, due to its integrity and effectiveness for evaluating the quality of numerical features.

3.2 Feature Correlation and Importance

Previous researches [7] showed that balanced information gain can reliably measure the contribution of each feature

for data classification. However, if a method simply selects features with large contribution, it only performs optimally when features are mutually independent. To optimize the feature selection, we need to re-evaluate the importance of each feature according to its correlation with others.

To calculate feature correlation, we employ balanced information gain, which was previously defined on numerical features [7]. In order to deal with discrete variables, we extend balanced information gain as follows.

Definition 1 Let X and Y be two different discrete variables. Denote the value sets of X and Y by $\{x_1, \dots, x_{\mathcal{I}}\}$ and $\{y_1, \dots, y_{\mathcal{J}}\}$, where \mathcal{I} and \mathcal{J} are their cardinality, respectively. We define the partition of X given by Y as $P(X||Y) = \{P_{i,j}, i = 1, \dots, \mathcal{I}, j = 1, \dots, \mathcal{J}\}$, where $P_{i,j}$ is the nonempty set of events that $X = x_i$ and $Y = y_j$. Let $\kappa_{X||Y} = |P(X||Y)|$ be the cardinality of the partition $P(X||Y)$. We define the extended balanced information gain of X given by Y as:

$$E_{b_g}(X||Y) = \frac{I(X|Y)}{\log_2 \kappa_{X||Y}} \quad (6)$$

In Definition 1, $\kappa_{X||Y}$ does not necessary equal to $\mathcal{I} * \mathcal{J}$, because each member of $P(X||Y)$ must be a non-empty set. Let X and Y (given in the above definition) be the attributes of training samples. $P_{i,j} \in P(X||Y)$ is then nothing but the set of samples with attributes $X = x_i$ and $Y = y_j$.

We now prove the symmetricity of extended balanced information gain.

Theorem 1 (Symmetricity of Extended Balanced Information Gain) For any two discrete variables X and Y , $E_{b_g}(X||Y) \equiv E_{b_g}(Y||X)$.

Proof: From Formula 2, we have the following symmetricity of mutual information:

$$I(X|Y) = I(Y|X), \quad (7)$$

where X and Y are any two discrete variables. From the definition of $P(X||Y)$, for any $P_{i,j} \in P(X||Y)$, we have at least one event with $X = x_i$ and $Y = y_j$. This implies $P_{j,i} \in P(Y||X)$. Similarly, $P_{j,i} \in P(Y||X)$ implies $P_{i,j} \in P(X||Y)$. The 1-to-1 map (i.e., $P_{i,j} \leftrightarrow P_{j,i}$) between elements of $P(X||Y)$ and $P(Y||X)$ indicates

$$\kappa_{X||Y} = \kappa_{Y||X}. \quad (8)$$

By Formulas 6, 7 and 8, $E_{b_g}(X||Y) \equiv E_{b_g}(Y||X)$. \square

From the above discussions, we can see that extended balanced information gain is a balanced and symmetrical measurement for mutual information between discrete variables. Hence, we measure the correlation between any two

features $f_m, f_n \in F$ as follows:

$$\begin{aligned} Corr(f_m, f_n) &= E_{b_g}(f_m||f_n) \\ &= \frac{I(f_m|f_n)}{\log_2 \kappa_{m||n}}, \end{aligned} \quad (9)$$

where $\kappa_{m||n}$ is the cardinality of the partition $P(f_m||f_n)$.

We define the normalized correlation value of any two features $f_m, f_n \in F$ as follows:

$$Corrn(f_m, f_n) = \frac{Corr(f_m, f_n)}{\text{Max}(Corr(f_k, f_l))}, \text{ for all } k \neq l. \quad (10)$$

The normalization ensures $Corrn(f_m, f_n) \in [0, 1]$ for all $f_m, f_n \in F$ and $f_m \neq f_n$. Let $F_s \subseteq F$ be a feature subset, we calculate the correlation between feature $f_m \notin F_s$ and F_s as follows:

$$Corr(f_m, F_s) = \text{Max}(Corrn(f_m, f_k)), \text{ for } \forall f_k \in F_s. \quad (11)$$

Denote the selected feature subset by $F^* \subset F$, we term any feature $f_m \notin F^*$ (and $f_m \in F$) an *undetermined feature*. For an undetermined feature $f_m \notin F^*$, we define its importance as follows:

$$\text{Imp}(f_m) = B_g(f_m) \times (1 - Corr(f_m, F^*)). \quad (12)$$

By Formula 12, an undetermined feature f_m will have a large importance value if and only if it contributes significant information for data classification and does not highly correlate with any selected feature.

4 Efficient Feature Selection

We present our feature selection method in Algorithm 1. In this algorithm, we directly discard non-important features (whose importance values are less than a constant ϵ on some step). To prove that the importance value of a non-important feature will always be less than ϵ , we first present the following lemma:

Lemma 1 (Non-decreasing Property of Feature Correlation) In Algorithm 1, the correlation $Corr(f_m, F^*)$ between any undetermined feature $f_m \in F'$ and the selected feature subset F^* is non-decreasing during feature selection (cf. the steps 9-13 of Algorithm 1).

Proof: For any $i (=1, \dots, M^*)$, we denote the selected feature subset of size i as F_i^* , and we let $F_0^* = \{\}$. Suppose f_i is the feature selected on the i th step, we can deduce from Formulas 11 and 12 that, for any undetermined feature $f_m \in F'$:

$$\begin{aligned} corr(f_m, F_i^*) &= corr(f_m, F_{i-1}^* \cup \{f_i\}) \\ &= \text{Max}(Corr(f_m, F_{i-1}^*), Corrn(f_m, f_i)). \end{aligned} \quad (13)$$

Algorithm 1 *Efficient Feature Selection*

- 1: **Input** 1) training set $S = \{(s_1, v_1), \dots, (s_N, v_N)\}$; 2) feature space $F = \{f_1, \dots, f_M\}$; 3) M^* : dimensions of selected feature subset F^* ; 4) ϵ : a small constant.
 - 2: **Initialize**: the selected feature subset $F^* \leftarrow \{\}$.
 - 3: **Set** the undetermined feature set $F' \leftarrow F$.
 - 4: **for** each feature $f_m \in F'$ **do**
 - 5: **Calculate** the balanced information gain $B_g(f_m)$ of f_m using the greedy algorithm (cf. Section 3.1);
 - 6: **Set** the importance value $Imp(f_m) \leftarrow B_g(f_m)$;
 - 7: **Set** $F' \leftarrow F' - \{f_m\}$ if $Imp(f_m) < \epsilon$.
 - 8: **end for**
 - 9: **for** $i = 1$ to M^* **do**
 - 10: **Find** feature $f_i \in F'$, whose importance value $Imp(f_i)$ is the maximum in F' ;
 - 11: **Set** $F^* \leftarrow F^* \cup \{f_i\}$ and $F' \leftarrow F' - \{f_i\}$;
 - 12: For all $f_m \in F'$, **update** $Imp(f_m)$ with Formula 12; if $Imp(f_m) < \epsilon$, set $F' \leftarrow F' - \{f_m\}$.
 - 13: **end for**
 - 14: **Output**: the selected feature subset F^* .
-

The proceeding formula indicates that $corr(f_m, F_i^*) \geq corr(f_m, F_{i-1}^*)$ for $i = 1, \dots, M^*$. Hence, for any undetermined feature $f_m \in F'$, the value $Corr(f_m, F^*)$ is non-decreasing during feature selection. \square

By Lemma 1, we have the following non-increasing property of feature importance:

Lemma 2 (*Non-increasing Property of Feature Importance*) *In Algorithm 1, the importance $Imp(f_m)$ of any undetermined feature $f_m \in F'$ is non-increasing during feature selection.*

Proof: By Lemma 1, $1 - Corr(f_m, F^*)$ is non-increasing in feature selection. Since $B_g(f_m)$ is unchanged in the steps 9-13 of Algorithm 1, the feature importance $Imp(f_m)$ ($= B_g(f_m) \times (1 - Corr(f_m, F^*))$) (see Formula 12) is non-increasing during feature selection. \square

Lemma 2 suggests the following theorem:

Theorem 2 (*Integrity of Algorithm 1*) *Algorithm 1 only discards features whose importance values are always less than the threshold ϵ .*

Proof: Lemma 2 suggests that, for any $i = 1, \dots, M^*$, if $Imp(f_m) < \epsilon$ on the i th step of feature selection, we will always have $Imp(f_m) < \epsilon$ on any step larger than i . \square

The time complexity $T(e)$ of our feature selection method is dominated by the sort operations (for calculating the balanced information gain) and the feature re-evaluation

(during feature selection). The time complexity of sort operations is $O(MN \log N)$. Since each re-evaluation requires to calculate the importance of $O(M)$ features, the time complexity of the feature selection is $O(M^*M)$. So, we have $T(e) = O(MN \log N + M^*M)$.

5 Empirical Results

5.1 Experimental Configuration

We evaluate the performance of our feature selection method with the following 3 numerical datasets:

- **Corel** [19] comprises a collection of 31,438 Corel images, which contain general-purpose images of various content, such as plants, animals, buildings and human society, etc. Each image is represented by a 179-dimensional feature vector [19], which covers information about color coherence vector, wavelet texture feature and statistical shape descriptors, etc. We test the performance of our method with 44 semantic classes, such as rose, tiger, building and falls, etc. To initialize online learning, we construct a training set of 220 samples, with 5 samples from each class.
- **Leukemia** [11] consists of 72 samples of leukemia patients, where each sample is measured over 7,129 genes [11]. These samples include two different classes of leukemia patients: 47 acute myeloid leukemia (AML) and 25 acute lymphoblastic leukemia (ALL). The samples are split into a training set of 38 samples (ALL/AML = 27/11) and a test set of 34 samples (20/14) by the provider.
- **Lymphoma** [2] contains 96 samples from normal and cancerous populations of human lymphocytes, where each sample is measured over 4,026 genes [11]. This dataset includes 72 cancerous samples and 24 normal ones. We randomly split the samples into a training set of 60 samples (CANCEROUS/NORMAL = 45/15) and a test set of 36 samples (27/9).

For Corel dataset, we employ the ground truth based on the above-mentioned 44 high-level semantic classes for performance evaluation. For each semantic class, we count every training sample as positive or negative – according to whether that sample belongs to the current semantic class; and we train a binary classifier to classify the dataset. We then employ the relevance feedback method presented in [19] to iteratively improve the initial retrieval result, until 10 feedback iterations are run. We measure retrieval performance with *precision*, which is the number of returned images belonging to the current semantic class over the total number of images returned to the user. To calculate the precision, only images from the current semantic class are counted as positive. In our experiments, we run each query on the whole Corel dataset (31,438 images); and we present

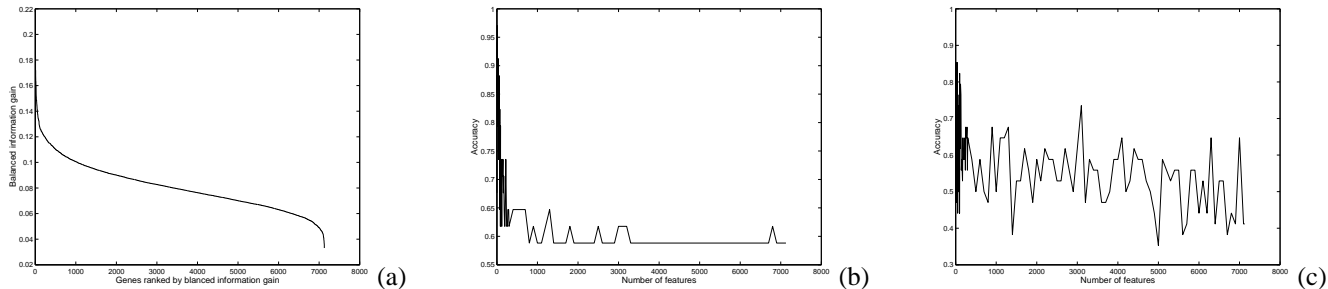


Figure 2: Accuracy of random forests and CART on various number of genes from Leukemia dataset. (a) genes ranked by balanced information gain; (b)/(c) accuracy of random forests/CART when 1-7129 ranked genes are used.

the precision when 20 images are returned.

With respect to gene datasets, the goal is to train a classifier based on the gene expressions. We evaluate the performance of the classifier with its *accuracy*, which is the ratio of correctly classified test samples.

To demonstrate the effectiveness of our method, we train different classifiers on the testing datasets. On the gene datasets, we train both regular random forests (RRF) and CART [6]. To improve the efficiency of relevance feedback, we train adaptive random forests (ARF) and CART on the Corel dataset. We compare our efficient feature selection (EFS) with the following state-of-the-art approaches: (1) DFE: the simple feature selection approach that directly selects features with large balanced information gain [19, 7]; and (2) CFS: the correlation-based method proposed by Hall *et al.* [13]. In our experiments, we employ cross-validation to choose the dimensions of the feature subsets extracted by EFS and DFE.

5.2 Threshold ϵ for Different Datasets

We show how to choose appropriate threshold ϵ (used in Algorithm 1) for the testing datasets in this section.

Dataset	Leukemia	Lymphoma	Corel
ϵ	0.1021	0.0748	10^{-6}

Table 1: Threshold ϵ for different datasets.

Due to the extremely high dimensionality of gene datasets, most features from these datasets are redundant for data classification. This is demonstrated in Figure 2. We first rank the genes from Leukemia dataset according to their balanced information gain. And then, we use 1 to 7,129 ranked genes to train random forests (with 120 trees) and CART. From Figure 2, we can see that both random forests and CART achieve best performance when less than 100 genes are used; and their accuracy approaches mini-

um when trained with more than 900 genes¹. Hence, we fasten our feature selection method by setting ϵ to the top-900 largest balanced information gain on gene datasets. As for Corel dataset, we simply set ϵ to a small positive constant, so we only discard features whose balanced information gains are close to zero. Based on the above discussions, we set threshold ϵ for different datasets as shown in Table 1.

5.3 Relevance Feedback for Image Retrieval

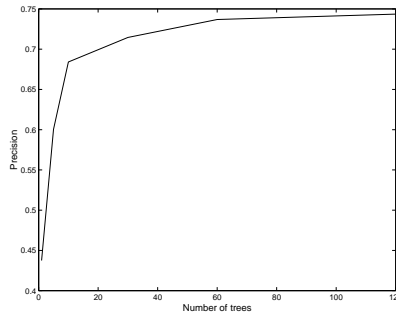


Figure 3: Precision of ARF with 1-120 trees.

In image retrieval, the time complexity of training ARF is the bottleneck of system efficiency. Since time complexity of ARF increases linearly with its tree number [5], we need to minimize tree number, while maintaining the (nearly) optimal precision. Figure 3 presents the precision of variously sized random forests in the full feature space. We can see from this figure that a random forest with 60 randomized trees performs nearly optimally: it noticeably outperforms random forests with less trees; and it performs as good as random forests with substantially more trees (such as 120 trees). Thus, we set the tree number (of ARF) to 60.

Table 2 compares the performance of different methods on Corel dataset. We can draw the following observations

¹To save presentation space, we abbreviate similar curves for Lymphoma dataset.

COREL	Adaptive Random Forests			CART		
	Precision(%)	Dim	Time (in s)	Precision(%)	Dim	Time (in s)
Full Space	73.69	179	4.43	43.75	179	1.44
EFS	74.04	20	2.34	45.62	20	1.06
DFE	73.09	60	2.97	45.27	30	0.85
CFS	74.02	60	3.35	45.23	60	1.36

Table 2: Performance of different methods on Corel dataset. Dim shows the dimensions of the feature subset extracted by each method; time is the average echo time (in seconds) of each iteration.

from this table:

- Our method (EFS) noticeably outperforms all compared approaches; and it dramatically boosts the efficiency of ARF and CART: both ARF and CART achieves optimal precision on 20 features. As a result, our method doubles the efficiency of ARF, while fastening CART by 40%. Moreover, our approach also improves the precision of ARF and CART by about 1 – 2%.
- Although CFS and DFE achieve comparable precision against EFS, both of them selects three times as many features as EFS does. As the only exception, DFE only outputs 50% more features for CART. For random forests and CART, EFS achieves a gain in efficiency over CFS by about 30 – 40%.
- To select the same number of features, DFE is less computationally intensive than the compared methods. This is because it does not consider feature correlation. However, the performance of DFE suffers from its negligence of feature correlation. For example, DFE selects much more features than EFS does. To train ARF, DFE degrades the efficiency against EFS by about 20%, while deteriorating the precision by 1%.
- ARF noticeably outperforms CART. It achieves a gain in precision over CART by about 30%.

In image retrieval retrieval, composite classifiers like ARF are more favorable than traditional classifiers (such as CART), because they do not overfit small training data (from relevance feedback). By extracting a low-dimensional feature subset, our feature selection method significantly boosts the efficiency of ARF, so it noticeably increases the feasibility of applying ARF to relevance feedback. Moreover, we can see that the precision of ARF is nearly optimal in the full feature space. This justifies our method to choose the tree number for ARF in the full feature space (cf. Figure 3).

5.4 Performance on Gene Datasets

In this experiment, we choose tree number of random forests with cross-validation. Table 3 presents the performance of different methods on gene datasets. From this table, we can make the following observations:

- EFS dramatically improves the accuracy and efficiency for classifying gene data.
 - On Leukemia dataset, EFS achieves a gain in accuracy of about 42% for random forests and CART; on lymphoma dataset, it boosts the accuracy of random forests and CART by 19% and 23%, respectively.
 - By selecting important features, our method improves the efficiency of random forests and CART by at least two orders of magnitude.
- EFS noticeably outperforms all compared methods; and DFE constantly outperforms CFS. In most cases, EFS noticeably improves the accuracy over DFE by about 3%; and DFE improves the accuracy over CFS by 3 – 6%. These results indicate that: (1) it is critical to integrate feature correlation into feature importance criterion; and (2) balanced information gain noticeably outperforms symmetrical uncertainty for evaluating the importance of numerical features.
- Random forests noticeably outperforms CART. It improves the accuracy over CART by about 9 – 15% on gene datasets.

Although random forests dramatically outperforms CART, it still suffers from the curse of dimensionality on gene datasets. By extracting a low-dimensional feature subset, our method remarkably enhances both accuracy and efficiency of random forests on gene datasets.

6 Conclusion

In this paper, we presented a feature selection filter to facilitate classifying high-dimensional numerical data. Our method uses *balanced information gain* [7] to measure the contribution of each feature (for data classification); and it calculates the correlation between features with a novel extension of balanced information gain. To search the important feature subset, our approach employs a forward sequential selection algorithm to select uncorrelated features with large balanced information gain.

Extensive experiments have been carried out to demonstrate the effectiveness and robustness of our method. At first, on a Corel image dataset (with 31,438 images), our

Datasets		Random Forests				CART		
		Accuracy(%)	Dim	Tree #	Time (in s)	Accuracy(%)	Dim	Time (in s)
Leuk- emia	Full Space	58.82	7129	30	10.21	41.18	7129	1.11
	EFS	100	19	20	0.02	85.29	17	0.006
	DFE	97.06	24	30	0.008	85.29	19	0.003
	CFS	91.18	22	30	0.04	82.35	22	0.009
Lymph- homa	Full Space	75	4026	40	9.01	63.89	4026	0.98
	EFS	94.44	27	8	0.02	86.11	30	0.009
	DFE	91.69	50	11	0.05	83.33	83	0.02
	CFS	86.11	23	14	0.04	80.56	23	0.008

Table 3: Performance of different methods on Gene datasets. Time shows the number of seconds used for feature selection and training the classifiers.

approach doubles the efficiency for training adaptive random forests, while achieving a gain in precision by 1 – 2%. Secondly, the experiments over gene microarray datasets showed that our method can significantly improve the accuracy and efficiency for classifying high-dimensional numerical data. Our approach not only improves the classification accuracy by 19 – 42%, but also boosts the efficiency by at least two orders of magnitude. Finally, extensive experiments indicated that our feature selection method noticeably outperforms the state-of-the-art approaches [12, 19].

References

- [1] D. W. Aha and R. L. Banker. *Learning from Data, Chapter 4*, chapter A comparative evaluation of sequential feature selection algorithms, pages 199–206. New York, USA, 1996.
- [2] A. A. Alizadeh and M. B. E. *et al.* Distinct types of diffuse large b-cell lymphoma identified by gene expression profiling. *Nature*, 403:503 – 511, February 2000.
- [3] R. Ash. *Information Theory*. John Wiley and Sons, 1965.
- [4] L. Breiman. Bagging predictors. *Machine Learning*, 24:123 – 140, 1997.
- [5] L. Breiman. Random forests–random features. Technical Report 567, Department of Statistics, University of California, Berkeley, September 1999.
- [6] L. Breiman, J. Friedman, R. Olshen, and C. Stone. *Classification and Regression Trees*. Wadsworth International Group, 1984.
- [7] T. Elomaa and J. Rousu. General and efficient multisplitting of numerical attributes. *Machine Learning*, 36(3):1 – 49, 1999.
- [8] U. M. Fayyad and K. B. Irani. Multi-interval discretisation of continuous-valued attributes. In *Proc. of the Thirteenth International Joint Conference on Artificial Intelligence*, 1993.
- [9] Y. Freund and R. Shapire. A decision-theoretic generalization of online learning and an application to boosting. *Journal of Computer and System Science*, 55(1), 1997.
- [10] K. Fukunaga. *Introduction to Statistical Pattern Recognition*. San Deigo, California: Academic Press, Inc., 1990.
- [11] T. Golub, D. Slonim, and P. T. *et al.* Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. *Science*, 286(15):531 – 537, 1999.
- [12] M. A. Hall. Correlation-based feature selection for discrete and numerical class machine learning. In *Proc. of Intl. Conf. on Machine Learning*, 2000.
- [13] M. A. Hall and L. A. Smith. Practical feature subset selection for machine learning. In *Proc. of the 21st Australian Computer Science Conference*, 1998.
- [14] T. M. Mitchell. *Machine Learning*. McGraw Hill, 1997.
- [15] W. Press, S. A. Teukolsky, W. T. Vetterlin, and B. P. Flannery. *Numerical Recipes in C*. Cambridge, 1995 edition, 1988.
- [16] K. Tieu and P. Viola. Boosting image retrieval. In *IEEE Int’l Conf. Computer Vision and Pattern Recognition (CVPR’00)*, June 2000.
- [17] N. Vasconcelos and A. Lippman. A probabilistic architecture for content-based image retrieval. In *IEEE Conference on CVPR*, 2000.
- [18] Y. Wu, Q. Tian, and T. S. Huang. Discriminant-em algorithm with application to image retrieval. In *IEEE Int’l Conf. on CVPR*, 2000.
- [19] Y. Wu and A. Zhang. Adaptive pattern discovery for interactive multimedia retrieval. In *IEEE Int’l Conf. Computer Vision and Pattern Recognition (CVPR’03)*, June 2003.
- [20] Y. Wu and A. Zhang. Interactive pattern analysis for relevance feedback in multimedia information retrieval. *ACM Multimedia Systems Journal (to appear)*, 2004.