Feature Selection using Misclassification Counts

Adil Bagirov[1] **Andrew Yatsko**[1] **Andrew Stranieri**[1] **Herbert Jelinek**[1,2]

[1] School of Science, Information Technology and Engineering,
University of Ballarat,
Ballarat, Victoria, 3353, Australia
E-mail: a.stranieri@ballarat.edu.au, a.bagirov@ballarat.edu.au,
andrewyatsko@students.ballarat.edu.au

[2] School of Community Health
Charles Sturt University,
Albury, New South Wales, 2640, Australia
Email: hjelinek@csu.edu.au

Abstract

Dimensionality reduction of the problem space through detection and removal of variables, contributing little or not at all to classification, is able to relieve the computational load and instance acquisition effort, considering all the data attributes accessed each time around. The approach to feature selection in this paper is based on the concept of coherent accumulation of data about class centers with respect to coordinates of informative features. Ranking is done on the degree to which different variables exhibit random characteristics. The results are being verified using the Nearest Neighbor classifier. This also helps to address the feature irrelevance and redundancy, what ranking does not immediately decide. Additionally, feature ranking methods from different independent sources are called in for the direct comparison.

Keywords: classification, feature ranking, feature selection, dimensionality reduction, optimization.

1 Introduction

Supervised Classification implies that unique association of instances with classes of data is known on the training stage for a data sample. This mapping is then used to develop an algorithm by which any new instance can be assigned to a correct class based on the data. A classification algorithm has to be able to deal with computational complexity commonly caused by the magnitude of instances often driven by the multitude of data attributes. This problem is huge in text categorization, every word expanding the attribute space to a whole new dimension. This area received much attention in the past, but continues to be in the focus despite the processing power of computers has increased dramatically. Some terminology has settled over the time. (Saeys et al. 2007) give a contemporary view of *feature selection* methods in bioinformatics.

Without knowing better, we can certainly assume that disengaging of variables, assumed all contributing, will cause reduction of the classification accuracy. We can stage experiments to ascertain influence of different variables, referred commonly to as *features*, indirectly, via responses we get from a classifier. Various models of feature-set are entered sequentially into the classifier, no matter what kind, and the best response is learned. This generic technique of feature selection is called *wrapping*. In this work we use accuracy of the k-NN classifier as an indirect measure of *fitness* of feature-set. Where a pre-selection of features is possible, it is termed *filtering*. Devices of different sorts are in employ, and if they can provide answers to feature *irrelevance* and *redundancy* - whether features align with no class or their input is equivalent to others - the better. Filtering, which can be rather elaborate, is independent from the method of classification, although it inevitably uses the class information. Information Gain and Relief are two filtering techniques considered widely a standard, each coming from a different perspective: probabilistic - the former, deterministic - the latter. Ultimately, there are methods of classification, selecting features to best suit the class distribution for the tune-up. This is referred to as *embedding*. SVM is an example of classifier where feature selection is embedded. We discuss these and other methods when comparing them to those introduced in this paper.

Only wrapping offers a universal approach for feature-set selection. A chosen set has to be *consistent* with the agenda of classification, that is, be sufficient for class discrimination. The enumeration of different subsets of features is computationally challenging. If *monotonicity* holds, so that any addition of a feature can only improve fitness of the current set, the exhaustive search can be escaped via *branch-and-bound* arrangement setting a qualifying level for fitness (Narendra and Fukunaga 1977). While same features may add differently to fitness of different sets, knowing fitness of individual features can be useful. Embedding may or may not produce a shortlist of features best describing data as a whole. In SVM it does. In Decision Trees, instead, one best feature is selected for spawning at different stages of tree growing (Quinlan 1993) with Information Gain often used as the criterion. The feature is different for different subsets of data included in subsequent branches of the tree. *Globally* or *locally*, it helps knowing how to rank features by *relevance*.

While *ranking* of features can be obtained as a byproduct of feature subset selection by a wrapper or embedded method, often ranking is an element of design of filter methods. Conversely, having features ranked is attractive for quick assembly of a desired feature-set. For example, (Huda et al. 2010) pair a Neural Network wrapper with a filter, akin to Information Gain, to facilitate selection of sufficient quality a feature-set. Feature ranking is also the

element of design of the proposed method, although we do not systematically explore the aspect of best feature-set, if only for verification. The ranking is done on the degree to which different variables exhibit random characteristics. Similar methods are known as filters. It may be interpreted as a method of classification adapted for feature ranking. So, it may be seen also as a wrapper or embedded technique. In fact, a number of methods, particularly those used for comparison in this paper, are exposed to such interpretations. These methods share the idea of misclassification counts. We pay Relief (Kira and Rendell 1992, Kononenko et al. 2008) a special attention for conceptual likeness to our method. The algorithm is given a remake to fit the new agenda.

2 Feature Ranking Algorithm

Introduction of a measure of similarity is a founding step in any approach to classification. If a class can be described as a cluster of data points in the problem space, then its center may be defined as a point most similar to them all, that is, containing the class information signature. One measure, commonly used, is distance in the problem space. Any new data can then be class assigned running the affinity check for different class centers. We imply sufficiency of the information signature for class identification.

2.1 Formulation

In this section we lay out an approach, whereby features are selected step-by-step, more informative / relevant first. In the formulation t stands for the iteration number and $I_t \subset \{1, \ \ldots, \ n\}$ are indices of the reduced set of features contending at time t.

Consider data A consisting of $m \geq 2$ classes (finite sets) $A_j \subset \mathbb{R}^n$, $j = 1 \ldots m$, so that: $A_j \neq \emptyset$; $A_{j1} \bigcap A_{j2} = \emptyset$, $\forall j^1, j^2, \ j^1 \neq j^2$; and $A = \bigcup_{j=1}^{m} A_j$. Let a^{ij} be elements of the sets, $i = 1 \ldots |A_j|$, where $|\cdot|$ is the notation for set cardinality.

Let $\|\cdot\|$ defines the metric for $\mathbb{R}^n$ space as follows:

$$\|a-x\| \ = \ (\sum_{l \in I_t} |a_l - x_l|^2)^{1/2} \ , \ \ \forall \, a, x \ \in \ \mathbb{R}^n \ , \ \ n > 1 \ .$$

$\mathbb{R}^n$ is a space of variable dimensionality from 1 to n.

Algorithm 1 (Forward Selection)

Step 1. (Initialization). Set $t = 0$, $I_t = \{1, \ \ldots, \ n\}$.

Step 2. (Class Centers). Determine class centers x^j assuming that sets A_j each form a unique cluster. Compute the centers by solving the following problem of convex programming:

$$\text{minimize} \ \ 1/|A_j| \cdot \sum_i \|a^{ij} - x^j\|^2 \ . \qquad (1)$$

(See Theoretical Aspects.)

Step 3. (Misclassified Points). Find points of sets A_j, which are closer to class centers of other sets coordinate-wise. Let x_*^j be solutions to the Problem 1.

Evaluate sets:

$$N^j = \left\{ a^{ij} \ : \ \min_{s \neq j} |a^{ij} - x_*^s|^2 \leq |a^{ij} - x_*^j|^2 \right\} \ ,$$

where $s = 1 \ldots m$ is the class index.

Get the resulting set:

$$N = \bigcup_{j=1}^{m} N^j \ .$$

The coordinate index $l \in \ I_t$ is implied in the above.

Step 4. (Relevant Attribute). To determine the coordinate of highest relevance find

$$l_* = \arg \min_{l \in I_t} (|N_l|/|A|) \ .$$

If ties exist choose arbitrary.

Step 5. (Contending Features). Make $t = t + 1$, and construct the new set of contributing factors:

$$I_t = I_{t-1} \setminus \ \{l_*\} \ .$$

If $|I_t| \ = 1$ then stop, else go to Step 2.

The $\|\cdot\|$, way we define it, is the radial, or Euclidian, distance. Square omission throughout the algorithm gives rise to formulation in so called Manhattan (the city block), or Hamming distances.

Plainly, the algorithm finds class centers and enumerates elements that belong to a class, but considering a particular feature, are closer to centers of other classes. The total of these counts, normalized by the number of elements in the whole set, establishes the feature rating. The higher the rating, the less relevant is the feature. The best performing feature out, search is repeated again to select a next one. The idea is stemming from the approach suggested in (Bagirov et al. 2003). However, the technique there engages subset selection directly, without having features ranked.

A variable in this method has the higher relevance, the more distant are values taken at class centers. If, instead, a variable has close readings for different classes, this results in the number of misclassified points growth. Obviously, a variable with the same value for all classes is irrelevant given data. Irrelevance correlates with rating close to unity obtained for a feature. However, it is theoretically impossible to judge irrelevance given only data. No matter how big is the set, it is merely a sample revealing the data concept, largely unknown, only partially.

At the same time, the algorithm is adaptable for other search tactics (Saeys et al. 2007). Generally, if the selection criterion is sensitive enough and consistency of the feature-set for classification is not violated, dismissal of insignificant has advantage over selection of significant - despite not the shortest, the complete reliable subset of features is immediately known after each elimination. Certain time saving is achievable on a big set of features if forward selection and backward elimination is done concurrently, that is, one best and one worst feature are taken out in a single swoop, steps 4 and 5 of Algorithm 1 adjusted accordingly for this mixed scheme.

Even with the full set of features, all being relevant, no classifier can guarantee the absolute precision. Harnessing minor features can not help overcoming this inherent classifier limitation. Instead, it may cause overfitting: a classifier gets perfectly trained, but performs poorly on a test data. Yet a classifier can have less overhead if fewer performance boosting features are used, explaining preference that we give to forward selection.

The one cluster per class representation holds by the slim assumption that classes may be described as "connected" and "convex" sets. This can be improved, if classes are subdivided into clusters, although the complexity of the algorithm increases significantly. The Incremental Global Search featuring k-Means by (Bagirov 2008) makes this possible, but other clustering methods are also available. (Kaufman and Rousseeuw 1987) partition data around medoids, so they call their algorithm PAM. They refer to the structural model assumed for data as k-Medoid. Confusion in the literature exists about origins of the k-Medoids algorithm. In fact, k-Medoids is a featured component of PAM. Both k-Means and k-Medoids find only local solutions for k clusters, for the Euclidian or Manhattan metric respectively, and corresponding to the formulation with or without squares. The algorithms do the same by redistributing data between clusters based on proximity of cluster centers. The difference is only in how cluster centers are obtained (Theoretical Aspects). The hard, unconstrained objective applies. That is, each element of data belongs to one and only one natural cluster. The algorithm by (Kaufman and Rousseeuw 1987) is reconfigurable for k-Means. Likewise, the algorithm by (Bagirov 2008) can be recast for k-Medoids. Both algorithms strive to find a near global solution for k clusters.

Algorithm 1 can be generalized as follows.

Algorithm 2 (Class Overlay Counts)

Step 1. (Initialization). Set $t = 0,\ I_t = \{1,\ \ldots,\ n\}$.

Step 2. (Class Centers). Compute centers $x^{jk} \in \mathbb{R}^n$ of clusters A_{jk} making class A_j by solving the following problem of convex programming:

$$\text{minimize}\ \ 1/|A_{jk}| \cdot \sum_i \|a^{ijk} - x^{jk}\|^2\ , \qquad (2)$$

where $a^{ijk} \in A_{jk}$ are the cluster elements, $i = 1 \ldots |A_{jk}|$, $j = 1 \ldots m$, $k = 1 \ldots p_j$. Subdivision of classes into clusters is assumed known.

Step 3. (Misclassified Points). Find points of sets A_{jk}, which are closer to cluster centers of other classes coordinate-wise.

Let x_*^{jk} be solutions to Problem 2. Evaluate sets:

$$N^{jk} = \left\{ a^{ijk} : \min_{s \neq j} \min_r |a^{ijk} - x_*^{sr}|^2 \leq |a^{ijk} - x_*^{jk}|^2 \right\},$$

where $s = 1 \ldots m$ and $r = 1 \ldots p_s$ are the class and the cluster within class indices respectively.

This results in the set:

$$N = \bigcup_{j=1}^{m} \bigcup_{k=1}^{p_j} N^{jk}\ .$$

The coordinate index $l \in\ I_t$ is implied.

Step 4. (Relevant Attribute). To determine the most relevant coordinate find

$$l_* = \arg\min_{l \in I_t} (|N_l|/|A|)\ .$$

If ties exist make an arbitrary choice.

Step 5. (Contending Features). Make $t = t + 1$, and construct the new set of contributing factors:

$$I_t = I_{t-1} \backslash \{l_*\}\ .$$

If $|I_t|\ = 1$ then stop, else go to Step 2.

If Algorithm 2 is reconfigured for backward elimination, it makes sense reclustering data after each cycle. Irrelevant features may cause misrepresentation of the instance space structure by significantly changing distances in concerned directions. This can make subsequent feature deselection less certain.

Even though the unconstrained clustering condition may be fulfilled by class, in the united set it is not guaranteed to hold. Conversely, partitioning of the superset leaves no tension between clusters. The tension between classes helps achieving the algorithm goal. However, the class interaction weakens with number of clusters increasing. Also, smaller clusters are rounder in shape, their eccentricity less expressed.

The circumstance of Algorithm 1 not taking parameters is attractive. In Algorithm 2 the number of clusters per class has to be selected. Generally, more clusters per class should be improving the model description. However, if this number is not small enough, the centers become close to each other and number of instances per cluster small, rendering less reliable counts. Clearly, having statistically sound data props precision of the algorithm.

This does not answer the question, though, of how to choose an appropriate number of clusters for each class - after all, classes may vary in size and have simple or complex mapping. In this regard we propose the following approach: the data is clustered first as a whole with a set number of clusters. A label is assigned to each cluster based on the leading class membership. The classes are then clustered independently using the information obtained. So, we search the data undivided by class for clusters to the best isolation as in (Bagirov 2008). After class labels are assigned to clusters we initiate the standard k-Means procedure (MacQueen 1967) to make clusters conform to the topology of individual classes.

Choice of parameters in Algorithm 2 involves preprocessing and this poses a significant setback. However, if the number of clusters is increased to the number of elements, each point becomes a cluster of its own. This seems to be solving the problem of parameter setting, neither clustering needs to be performed. At the same time, this removes tension between classes. Unless attribute values repeat, no instance can possibly cross to a different class because now the center of a cluster coincides precisely with its only element. All features important - makes the selection a futile exercise. This, however, inspires the idea of following approximation to Algorithm 2.

Algorithm 3 (Estimated Overlay)

Step 1. (Initialization). Set overlay encounter by feature to none: $N_l = \emptyset,\ l = 1 \ldots n$. Iterate by coordinate (index l is implied) with the following.

Step 2. (Closest Points). Find two points for each point a^{ij}: indices of $a1$ are $i_1 \neq i,\ j_1 = j$, and indices of $a2$ satisfy $j_2 \neq j$; that is, points belonging to the same and a different class, but not a^{ij}, so that

$$|a1 - a^{ij}|^2 \ = \ \min_{a \ \in \ A_j \setminus \ \{a^{ij}\}} |a - a^{ij}|^2,$$

and

$$|a2 - a^{ij}|^2 \ = \ \min_{a \ \in \ A \ \setminus \ A_j} |a - a^{ij}|^2.$$

Step 3. (Misclassification). Add point a^{ij} to set N_l if

$$|a1 - a^{ij}|^2 \geq |a2 - a^{ij}|^2.$$

Keep iterating from Step 2 until point and coordinate cycles are exhausted.

Step 4. (Attribute Relevance). To determine relevance of coordinates find $|N_l|/|A|,\ l = 1 \ldots n$. Ordering on this ratio prioritizes the feature relevance.

In words, Algorithm 3 examines each attribute to establish whether it is good a class separator, as if for discretization purposes, that is, whether single class layers of data characterize the variable, or it is inundated by the class mix. This can be improved if k values of each class are drawn in the vicinity of current value and their averaged distances to the instance projection are compared to establish whether the instance is in the midst of its own class. It is clear that ranking obtained by this algorithm is independent from feature selection tactics. It is why usual steps articulating the tactics are not included. The ranking is also independent from the metric of problem space. We include squares for outward compatibility with other algorithms only.

Followed so far is the global feature weighting approach. It can be seen in the light of overlaying distributions for different classes along individual dimensions. Any overlapping of multivariate distributions, the class noise, adds uncertainty, but is not a problem nor the clue to feature weighting. It is the potential overlapping in respect of coordinates that matters. At the same time, data is borderless, represented by a finite set. Without fitting a structural model it is not possible to infer from the position of knowing the data concept. Nonetheless, it is possible to weight features locally by examining immediate neighborhoods of known instances. The results then can be generalized for the whole space hosting the data. This is the idea of Relief (Kira and Rendell 1992). In Relief feature-wise distance differences establish the rating. We use misclassification counts, and so let us call the version ReliefC.

Algorithm 4 (ReliefC)

Step 1. (Initialization). Set the encounter of class mix by feature to none: $N_l = \emptyset,\ l = 1 \ldots n$.

Step 2. (Closest Points). Find two points for each point a^{ij}: indices of $a1$ are $i_1 \neq i,\ j_1 = j$, and indices of $a2$ satisfy $j_2 \neq j$; that is, points belonging to the same and a different class, but not a^{ij}, so that

$$\|a1 - a^{ij}\|^2 \ = \ \min_{a \ \in \ A_j \setminus \ \{a^{ij}\}} \|a - a^{ij}\|^2,$$

and

$$\|a2 - a^{ij}\|^2 \ = \ \min_{a \ \in \ A \ \setminus \ A_j} \|a - a^{ij}\|^2.$$

Step 3. (Misclassification). Update coordinate sets N_l by including points a^{ij} if

$$|a1_l - a_l^{ij}|^2 \geq |a2_l - a_l^{ij}|^2.$$

Reiterate from Step 2 to cover all data.

Step 4. (Attribute Relevance). To determine relevance of coordinates find $|N_l|/|A|,\ l = 1 \ldots n$. Ordering on this ratio prioritizes the feature relevance.

Ranking obtained by this algorithm is space metric dependent. We can achieve refinement of the result if we adopt the tactics of backward elimination. Irrelevant dimensions may cause a significant distortion of the perceived data distribution. We can get a better understanding of other coordinate significance if we run Algorithm 4 again with the confusing attribute withheld, which, of course, can be repeated until each feature rank is adjusted. As in ReliefF (Kononenko et al. 2008) we can draw $k > 1$ nearest neighbors to the current instance for each present class to rely more on the distance statistics.

In Algorithm 4 each point is treated as a self-contained cluster, but having no other cluster elements required in Algorithm 2, we find a closest same class point to the instance of choice, which is to play the role of its cluster center. This makes the approximation. The data mapping used by Algorithm 2 has to be scaled down to fulfill the local feature weighting approach, which is impossible. We can notice semblance of Algorithms 4 and 3. Nevertheless, Algorithm 3 follows global, not the local generalized approach. It is value-wise, but not instance-wise.

2.2 Evaluation

We tested Algorithms 2, 3, 4 and ran a comparison with some other methods of feature ranking from this study and outside sources on data from (UCI Machine Learning Repository) and a proprietary data-set. The data space was assumed Euclidian in all metric dependent algorithms. Characteristics of data examples appear in Table 1. The numbers in columns reflect any transformations data required. "Clusters" applies to the data undivided by class. "k-NN" is the number of nearest neighbors in classification by the k-NN method to verify the results. Other columns are self-explanatory.

Data	Features	Instances	Classes
Housing Prices	13	336	7
Congressional Voting	16	435	2
Wall Following	24	5456	4
Diabetes Diagnostic	63	291	2
Data	Clusters	k-NN	
Housing Prices	21	1	
Congressional Voting	6	3	
Wall Following	20	5	
Diabetes Diagnostic	10	3	

Table 1: Data-sets used in experiments.

Housing Prices in suburbs and their defining factors is a snapshot of state of affairs in Boston, USA some time ago (Harrison and Rubinfeld 1978). The Housing Price is a continuous variable, and so the problem is of regression type. To represent it as a classification problem, we cluster the class variable by the same method we apply to data generally (Bagirov 2008). Whenever a conversion like this takes place, certain amount of noise is inevitably created, as data in the middle, between any two values defining adjacent classes, can not be successfully assigned to either of them. Therefore, the data was denoised after conversion using a technique we developed in (Stranieri and Yatsko 2009). Also, attributes of the data were re-scaled / standardized to zero mean and unit deviation (the absolute mean deviation).

Next two examples are exact classification tasks; neither the data requires a standardization. The Congressional Voting records on selected issues from a particular period in the past for each of the U.S. House of Representatives congressmen, either democrats or republicans, were interpreted by (Schlimmer 1987). The data can be treated as three-value numeric. In the Wall Following case a robot navigates around a room using ultrasonic sensors. This has to be seen as a time series; however, the moves are elementary and replicating: the robot either "follows and follows" directly or it "turns and turns". So, this is approached as a classification problem by (Freire et al. 2009), creators of the data. All the measurements are uniform, also having same upper limit defined by the sensor reachability.

The Diabetes Diagnostics data is a collection of medical records of various signs of presence or absence of this condition in patients and the expert opinion. This data array is available to the University of Ballarat Centre for Informatics and Applied Optimization Health Informatics Laboratory through collaboration with Charles Sturt University under provisions of DiScRi screening research initiative. Although this is a classification problem, it has specifics pertaining to diagnostic applications, with all focus given to a single small subset of data. A mix of attribute types required to be dealt with. So, where appropriate, ordinal attributes were converted to numeric. Otherwise, individual values of discrete attributes were turned into binary attributes. All the attributes except the class, numeric by the end throughout, were standardized to zero mean and unit deviation. Additional preprocessing relieved the data of several attributes inundated by missing values and involved generalization of the class for rare conditions.

Binary attributes qualify as numeric. At the same time, value combinations of binary attributes naturally subdivide the data, creating structures varying in detail, depending on how specific is the combination. This was used for setting missing values of numeric attributes, based on average. Unknown values of binary attributes themselves were entered using the same technique, but based on mode. Only selected binary attributes were used to narrow down the search, their number reduced step-by-step, until values left missing were set from all available data. The class attribute represents a special case. These missing values were set before any others from a predictor earlier identified as the best.

Note, the proposed algorithm of feature ranking can be adapted for missing values given cluster centers, and theoretically even the clustering algorithm can. However, this is not granted in respect of any other technique, and we do require a number of them for comparison. Generally, absence of certain values does not hurt predictability as this may seem - the data structure may make them redundant.

Results of application of feature ranking Algorithm 2 to different data-sets are shown in Tables 2, 3, 4, 5 and 6. In these tables: "Order" is the feature informativeness from highest to lowest - the rank; and "Rating" is the actual value corresponding to the rank as obtained by the algorithm after the first cycle.

2.2.1 Housing Prices

The factors affecting Housing Prices are listed in Table 2, their actual meaning can be found at the source. Representation of factors and specific circumstances have bearing on the ranking. Standing of several aspects of housing generally may be different.

Feature	Order	Rating
Rooms	1	0.6905
Income	2	0.7381
Employment	3	0.7768
Crime	4	0.7857
Pollution	5	0.8006
Industrial Area	6	0.8095
Education	7	0.8274
Building Age	8	0.8452
Black Culture	9	0.8631
Transport Access	10	0.8720
Tax	11	0.9315
Residential Area	12	0.9583
Natural Reserves	13	1.0000

Table 2: Housing Prices: factor significance.

The listing order corresponds to the result of forward selection. However, no shift of position of residual factors occurs through the factor set reduction. This is a characteristic of the formulation used and applies to all data-sets.

First impression of the ranking is that it does not betray the common sense, especially the two factors at the top. Indeed, housing price is higher for more room and with less population on low income. (Harrison and Rubinfeld 1978) also note the clean environment as a factor gaining in significance.

For the Housing Prices data-set results by other authors are also available (Bi et al. 2003), where feature weighting is a byproduct of classification using Support Vectors. Results of numerical experiments are presented graphically as star plots. We estimated feature ranks for comparison out of this representation. The authors specifically mention the number of rooms as the leading factor, influencing positively the housing price. Interestingly, ranks are positively or negatively charged. The next important factor appears to be the income, and it is charged negatively.

2.2.2 Congressional Voting

The Congressional Voting example is good that it refracts feature significance as heat of the debate, whether due to issue controversy or its actuality, and this is what Table 3 is meant to reflect. The topical context has to be examined carefully to fully understand significance of different issues and also be seen in the historical frame, whether they were routine or new matters at that time.

Feature	Order	Rating
Physicians	1	0.0552
Budget	2	0.1356
Education	3	0.1931
Crime	4	0.2299
Nicaragua	5	0.2391
$El - Salvador$	6	0.2506
Missiles	7	0.2897
Superfunds	8	0.3448
Synfuels	9	0.3586
Exports	10	0.3862
Satellites	11	0.3931
Handicapped	12	0.4069
Religious	13	0.4115
Immigration	14	0.6529
$South - Africa$	15	0.7678
Water	16	0.7954

Table 3: Congressional Voting: issue controversy.

Although given identifiers do not reveal the full story, it is clear that some up-to-date or pressing issues do occupy leading positions on the list and, instead, some issues of consensus appear down the list. However, there is no clear divide between parties only on two issues of physicians and budget at the top.

2.2.3 Wall Following

This data-set mirrors the Wall Following Robot moves. Table 4 shows significance of one sensor readings above others as obtained by the proposed feature ranking algorithm. The robot has 24 ultrasonic sensors around its "waist", but it is clear that the robot can get away with only two sensors: one tracking the wall, and another the obstacle ahead - the orthogonal wall, what the robot is actually programmed for. At the same time, it is obvious that in a small room or narrow space all or some readings interpret the same information. Represented appropriately, velocity of the robot and / or radius inverse of the turn could capture it all.

Feature	Order	Rating	Feature	Order	Rating
$US15$	1	0.6171	$US24$	13	0.8048
$US19$	2	0.7392	$US05$	14	0.8070
$US06$	3	0.7546	$US13$	15	0.8116
$US18$	4	0.7680	$US14$	16	0.8141
$US08$	5	0.7835	$US02$	17	0.8286
$US20$	6	0.7887	$US11$	18	0.8380
$US17$	7	0.7896	$US16$	19	0.8455
$US22$	8	0.7927	$US21$	20	0.8475
$US01$	9	0.7953	$US10$	21	0.8510
$US23$	10	0.7997	$US03$	22	0.8563
$US07$	11	0.8013	$US04$	23	0.8563
$US12$	12	0.8024	$US09$	24	0.8671

Table 4: Wall Following: sensor informativeness.

According to how the robot circumnavigates the room (clockwise), half of its sensors is on its side nearer to the wall, and other half is sounding the outer space. The sensor numbers (not the rank) closer to the wall are between 13 and 24 with US13 pointing exactly in the opposite direction of the robot and US19 exactly towards the wall. Indeed, we find US19 the second leading feature. Also, among the eight leading features we encounter six sensors next to the wall and, vise versa, among the eight trailing features there are six sensors further from the wall. It is reasonable to assume the sensor range is insufficient to cover the space of the room, which makes sensors next to the wall more valuable predictors. However, US01 pointing directly ahead is not in the leading third. Actually, only one of many comparison methods in the next section places US01 at the top, and none the adjacent sensors. In this regard we have to clarify that shape of the room in (Freire et al. 2009) is not simply rectangular, but has a rectangular concession in one corner, which coerces the robot to make turns not only to the same side (right) but also to the other (left).

Instead of the sensor pointing directly ahead, we have US15 as the leading factor, pointing almost backwards, which is sensible to rely on when making a turn without arriving at the obstacle. US15 actually sounds parallel to the wall in places, because for whatever reason trajectory of the robot is turned by about the same angle as the misdirection of US15 against the robot opposite, as appears on images in (Freire et al. 2009), which also makes the sensor sounding distance shorter. In the limited space of the room many sensors provide reasonable whereabouts, which explains appearance of versions of the data-set with four or even two features, although they are not the readings from sensors pointing in "compass" directions. Indeed, sensors rate close, due likely to their mutual redundancy. Yet the situation comprehensiveness can be improved via sensor combination.

2.2.4 Diabetes Diagnostics

The Diabetes Diagnostics is a medical data-set and without specialist knowledge it is difficult to comment on significance of different symptoms and results of tests. At the same time, because the publicity acknowledged burden of the spread condition on health funds and its link to the obesity, some general awareness exists.

Feature	Order	Rating
DM Diagnostic	1	0.0584
Screening Glucose	2	0.1478
Glucose	3	0.2062
LDL	4	0.2887
HT Diagnostic	5	0.3058
TC	6	0.3127
HbA1c	7	0.4467
HT Status	8	0.4708
LSBP	9	0.4777
DM Family History	10	0.4880
$Ewing - Early$	11	0.4880
Ewing Score	12	0.5292
DBHR	13	0.5395
BMI	14	0.5533
VAHR	15	0.5704
TC/HDL ratio	16	0.5704
Age	17	0.5876
$Ewing - Normal$	18	0.6186
DBHR result	19	0.6426
Grade 10 *sec*	20	0.6598
LSHR	21	0.6667
Lying DBP	22	0.6976
HDL	23	0.7320
Triglyceride	24	0.7388
Lying SBP	25	0.7801
PQ 10 *sec*	26	0.7938
Waist Circumference	27	0.8007
QRS 10 *sec*	28	0.8419
HGBP	29	0.8522
QRS Axis 10 *sec*	30	0.8832
QTc 10 *sec*	31	0.9003
$Ewing - Atypical$	32	0.9141

Table 5: Diabetes Diagnostics: symptom significance.

From Tables 5 and 6 we notice that some top ranking factors do imply the high content of sugars in specimens and, consulting the dictionary, the leading factor, Diabetes Mellitus (DM) diagnostic, appears to be a very specific carbohydrate metabolism disorder, besides reoccurring. While the DM diagnostic may be lacking analytic qualities as a forgone conclusion,

Feature	Order	Rating
QTd 10 sec	33	0.9313
Atrial Fibrillation	34	0.9588
Ewing − Definite	35	0.9588
Hearth Attack	36	0.9656
Pain in Left Arm	37	0.9794
CVD Diagnostic	38	0.9863
Palpitations	39	0.9863
Smoking	40	0.9897
Stroke	41	0.9931
Nausea	42	0.9931
Vomiting	43	0.9931
LSHR result	44	0.9931
VAHR result	45	0.9931
QTc 10 sec $> 1/2$	46	0.9931
Gender	47	0.9966
CVD Status	48	0.9966
Angina	49	0.9966
Hearth Failure	50	0.9966
Chest Pain	51	0.9966
CA Neuropathy	52	0.9966
Bloating	53	0.9966
Abdominal Pain	54	0.9966
Alcohol	55	0.9966
CVD Family History	56	0.9966
HGBP result	57	0.9966
Ewing − Severe	58	0.9966
QTc 5 min $> 1/2$	59	0.9966
Dizziness	60	1.0000
Pacemaker	61	1.0000
LSBP negative	62	1.0000
LSBP result	63	1.0000

Table 6: Diabetes Diagnostics: symptom significance (continued).

a number of factors immediately after it show a very strong predictive ability of the condition according to the rating. General awareness factors have an advanced position on the list, but can not be a match for specialist testing. The DM family history and the age appear in the first third of the list. Perhaps the waist circumference in the first half of the list by itself does not offer a measure sensitive enough without linking to other sizes, like height. It is likely, though, that BMI (the body mass index) in the first third of the list does take this into account. The host of factors towards the end of the list is either general complicities or those having a circumstantial effect. However, misplacement of two last features may have had occurred because the data scarcity and substitution of missing values. A number of features in this data-set are present in both numerical and categorical forms, one implying the other.

At the end of this section let us restate that no additional passes through Algorithm 2 are required to fine-tune the ranking, unless Algorithm 2 is run in the mode of backward elimination and the data-set is restructured. The observation that the order of significance of features does not change in the reduced set is a characteristic of the formulation used. Following result holds (applicable also to Algorithm 2):

Proposition 1 *Rating of features as obtained after application of Algorithm 1 does not change after a feature is removed from the set.*

This follows directly from the lemma proved in Theoretical Aspects. △

3 Comparison

Classification opens a way for indirect comparison, as previously explained, and there are different methods of feature ranking that can be compared directly with.

3.1 Classification

The results were undergone a verification using a classification method. The purpose was to ascertain that performance of the classifier is predictable. This is exactly the wrapping technique, except the ranking is known beforehand and only designated combinations of features need testing. Specifically, we are interested to find how the accuracy changes when features are subtracted from the end or beginning of the ranked list. The accuracy is found via the leave-one-out procedure, always fetching unwavering results, a special case of multi-fold cross-validation whereby credibility of each instance class is tested in turn against the whole instance base excepting that instance. The result will fluctuate if folds of cross-validation contain more than one instance.

The Nearest Neighbor classifier (k-NN) is easy implementable and suits our approach that it deals in distances. A good survey of instance-based methods, those with k-NN in their core, is contained in (Wilson and Martinez 2000). Precisely, the crisp version of fuzzy k-NN algorithm described in (Keller et al. 1987) is used. The only difference we introduce is that all neighbors are included in the radius given by the farthest of k initially selected nearest neighbors of the instance to be classed from the reference base. This allows to capture all repeating instances. Where data was treated for noise, a single nearest neighbor is often the best. Where it was not, a bigger k may be more optimal, as appears in Table 1, although in the Wall Following example $k = 1$ is still the best.

Computation results appearing next as accuracy percentile charts represent classification series driven by leading (or trailing) feature-sets, that is, with features below (or above) any given line on the ranked list removed to obtain each series element.

3.1.1 Housing Prices

The k-NN classification accuracy on the Housing Prices data-set in the forward run, first series on Figure 1, exhibits a slow decreasing trend at the beginning, getting more intense as factors are discarded one-by-one from the end of ranked list, that is, less informative first. The slide of accuracy at the run end reflects its accumulated loss as weighed against significance of the topmost factors. Conversely, the backward run, second series on Figure 1, where more informative factors are discarded first, is notable for abrupt, going less dramatic fall of accuracy; respective of the order but with some grouping; although unable to contain the accumulated loss at the end, having no factors of significance left. These two observations thus support the result of ranking.

(Bi et al. 2003), while applying Support Vector Machines, compare responses from the classifier with and without some unrelated variables mixed in, so that features performing no better than the artificial ones could be discarded as irrelevant. They could not find any not contributing, however, in the Housing Prices example. This corresponds to our finding, despite the other authors scheme is for regression, not for classification, which required us to discretize the class variable. We should expect a temporary increase of accuracy when discarding irrelevant features and this does not happen.

However, irrelevance can be full or it can be partial. We may find a number of features being a burden on a classifier overall, while contributing for

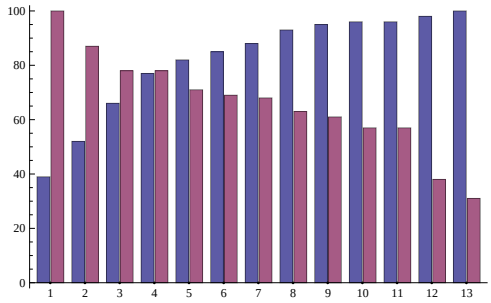

Figure 1: Housing Prices classification accuracy change with leading features removed first (2nd series, left-to-right) or last (1st series, right-to-left).

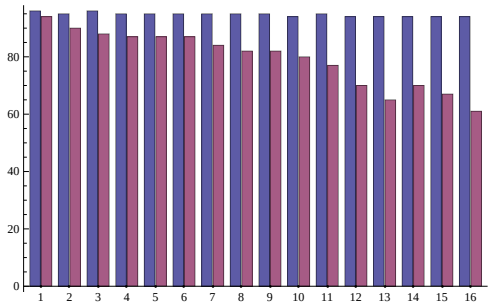

Figure 2: Congressional Voting classification accuracy change with leading features removed first (2nd series, left-to-right) or last (1st series, right-to-left).

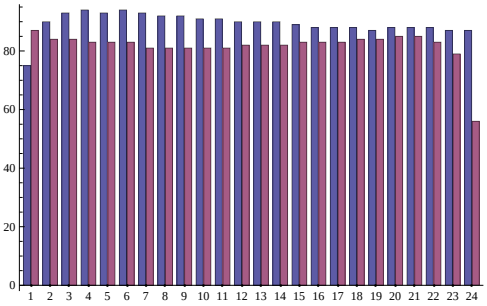

Figure 3: Wall Following classification accuracy change with leading features removed first (2nd series, left-to-right) or last (1st series, right-to-left).

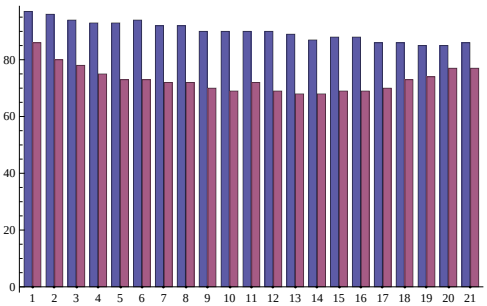

Figure 4: Diabetes Diagnostics classification accuracy change with leading features removed first (2nd series, left-to-right) or last (1st series, right-to-left).

a small subset of data. If no such subset exists then, of course, these features are fully irrelevant, as long as the data truly represents the underlying concept. The subtle differences may be lost in data conversion.

3.1.2 Congressional Voting

It is remarkable that the Congressional Voting data shows no visible loss of classification accuracy throughout in the forward run, represented by the first series on Figure 2. At the same time, the backward run exhibits a profile suggesting significance of leading factors when removed - the second series on Figure 2. The increase of accuracy at the end of backward run belies the insignificance of left features as k-NN switches to the implicit mode of predicting the biggest class anyway with all irrelevant features, which is an issue with imbalanced data-sets. Closer, behind the chart, result examination reveals that parity of class prediction deteriorates abruptly, reducing to zero by the end. Note, irrelevance does not mean the question on agenda is unimportant, simply parties both agree or disagree. One could be interested to see the feature list from this perspective.

At the same time, the accuracy in the forward run does not grow noticeably. This can be simply because the accuracy is already high at the beginning, and there is a limit of achievable with k-NN. Although, there are may be a background connection between debated topics. For instance, despite all of them may seem independent, there is a monetary component to any of them, which the budget attribute, emerging at the top of significance list, fully embraces.

3.1.3 Wall Following

In predicting Wall Following Robot moves the features are likely much related. Nonetheless, the forward run, first series on Figure 3, is characterized by a notable growth of accuracy, continuing up to the moment when only four top features is left. This can not be explained by irrelevance. The second series on Figure 3 certainly does not confirm this. These features are all redundant, and the improvement is purely due to reduction of overhead - distance calculations for k-NN classification become simpler. Although the backward run has a weak profile, it supports correctness of the feature ranking. Despite the accuracy elevates slightly again by the end, it is still below the level the forward run even takes off.

This leads to quite a different idea of how a feature list may be shortened, not only from the position of little relevance. If groupings of similar features are known, then keeping top features of each lot is sufficient. Yet, if it is not for the knowledge of domain, then how to tell that features are redundant?

3.1.4 Diabetes Diagnostics

The Diabetes Diagnostics is certainly an example of feature little relevance or even irrelevance. At the same time, features in the first half of the list are much different in relevance, spanning rating from very small to very high, see Table 5. We observe from the forward run, first series on Figure 4, that the classification accuracy is only increasing, allowed some fluctuation. This chart shows features in groups of three counted off the end of ranked relevance list.

The backward run, depicted as the second series on Figure 4, arranged similarly, portrays the

increasing insignificance of left features. Again, because this set is imbalanced, which is usual for medical diagnostics, it has the problem previously explained, causing the accuracy to increase at the end of run, while the relevance of features left is vanishing.

3.2 Ranking

The end result of feature rating is the order of informativeness, to know what features to keep and which to discard. It is only by luck that ratings calculated by different algorithms are compatible. There must be a way to compare methods based only on ranking they produce, and this is what we deal with in this section. A number of independent methods of feature ranking is brought in for comparison, (Weka Data Mining Tools) being the main source, find guidance where required from the book by (Witten and Frank 2005). We identify all these methods in Table 7.

ID	Name	Design	Origin
NNS	Single Feature k-NN	Wrapper	Proposed
NNX	Feature Excepted k-NN	Wrapper	Proposed
EO	Estimated Overlay	Filter	Proposed
RC	ReliefC	Filter	Proposed
RF	ReliefF	Filter	Weka
$H2$	Chi Square	Filter	Weka
IG	Information Gain	Filter	Weka
$1R$	One Attribute Rule / One R	Embed.	Weka
SVM	Support Vector Machine	Embed.	Weka

Table 7: Feature ranking methods for comparison.

3.2.1 Alternative Methods of Ranking

Let us recount methods sourced from (Weka Data Mining Tools) first, although this does not explain their exact implementation.

Chi Square statistic and Information Gain are probabilistic filters, which expect nominal data, but this can be arranged through discretization.

Chi Square statistic is the mean quadratic deviation of observed against expected frequencies, approaching one of the classic types of probability distribution introduced by Pearson when the number of data points increase. Each attribute produces a different result depending on how closely it follows the expected frequency for each class. No difference means that the feature is contributing nothing special to classification. So, smaller values of the statistic correspond to higher independence of the class from a given attribute, and feature ranks are assigned accordingly. (Liu and Setiono 1997) adapt a Chi Square discretization method by (Kerber 1992) to rank features on the number of intervals, different for different attributes, but obtained with the same significance level for the statistic.

Information Gain is often a choice among methods using probabilities. It is based on calculation of Entropy, a measure of uncertainty of a particular outcome. Entropy is calculated for each class and the total is found. It is then reduced by entropy calculated on class posterior probabilities for each value of a variable. The difference is the Information Gain. The less uncertain outcome from using a feature, the more is the information gain, and this makes the basis for ranking. The technique is widely used in Decision Trees (Quinlan 1993). (Fayyad and Irani 1993) extend the splitting mechanism of decision trees to discretization of features. Because they utilize the Minimum Description Length principle, put into theory by (Rissanen 1978), in the stopping criteria, potentially, this also can be used for ranking of features by the number of intervals.

Methods not using probabilities but having a statistical interpretation are as follows.

One Attribute Rule, used for classification, compares different attributes and relies solely on the attribute giving the least error (Holte 1993). The error is how features are rated. This is an embedded technique that could be identified as a filter, the wrapping clad kind, if not the design hierarchy.

The idea of Relief by (Kira and Rendell 1992) is that a feature should have distinct readings for different classes about same locality. Therefore, we can find two closest instances of data to the instance acting as probe, of a different and the same class, and subtract coordinate distances to these points. Positive differences characterize inner points of a class. The feature-wise differences are then averaged for a random selection of instances or all data to obtain weights for ranking. Larger weights identify features of better class separation. The technique is a filter: its design has a connection to, but does not include classification by k-NN. ReliefF is a multi-class implementation of Relief, taking care of noisy and incomplete data by (Kononenko et al. 2008).

The simplification One Attribute Rule implies makes it more of a filter than embedded type, and so is ReliefF. Both have certain design similarities with our method. The algorithm of Class Overlay Counts is of filter type, although a substantial preparatory phase is involved if it is not run in the simple mode.

Support Vector Machine (SVM), as a method of classification, finds separation hyperplanes maximizing the margin between classes, for which purpose it locates base points called support vectors. This results in weighting of variables, establishing their ranks. Clearly, this is an example of embedded method. Other authors result of using SVM on the Housing data (Bi et al. 2003) is also available.

Also included are: two k-NN wrapper estimators and two alternative ranking schemes of own making. The wrapping technique for obtaining ranking from accuracy of classification by nearest neighbors is using either single features or sets found by exception of single feature. No forward selection or backward elimination was pursued on this occasion to enhance the selection. Estimated Overlay and ReliefC are the two alternative schemes suggested in this study to circumvent necessity to cluster the data. ReliefC is an interpretation of Relief, counting occurrences of class overlap instead of summing up the standard feature-wise distance differences.

3.2.2 Method of Ranking Comparison

One approach to ranking goodness evaluation is extraction of longest sequence of preserved order of features of a scheme taken for a standard. The discrepancy with total number of features is then relative error, or variation. This method of comparison, while clear for understanding, on implementation side is not trivial. Also, it is not taking into account local changes of the position, tending to overestimation

of error, especially on long feature-sets. Sequences of different length may be compared instead. By this method features of the alternative ranking, the principal sequence includes up to a given position, are counted. This turns to be similar to a method of Kendall discussed in (Bhamidipati and Pal 2006) if rankings are compared instead of ratings.

However, our way of implementing it gave no different result than simply summing up absolute displacements of the rank for each feature, which is attributed to Spearman (Bhamidipati and Pal 2006). Therefore, we use this simple and recognized method to represent results of comparison. Thus found totals are rated by a like result, obtained for opposite ordering of the principal sequence, to measure how "wrong" the contending ranking is. Approaches of forward selection and backward elimination explain interest to validity of leading or trailing features ranked by significance. Therefore, comparison of ranking as produced by Algorithm 2 with other methods is conducted not only for all features but also for leading and trailing thirds of the list.

3.2.3 Ranking Comparison Results

Table 8 summarizes results of feature ranking using different methods, identified in Table 7, for each of examined sets. The comparison elements are: the absolute rank displacement for all features (1st-), for top (-2nd) and bottom (-3rd) portions of the ranked list. Columns denote alternative ranking schemes stated in Table 7. Because ranking differences on short data-sets can be rather imprecise, and to get a qualitative rather than quantitative evaluation of different methods, we use tenths rather than hundredths (percents) parts of the whole.

Dataset	NNS	NNX	EO
Housing Prices	6-4-4	5-1-3	2-2-1
Congressional Voting	2-1-1	6-5-3	0-0-0
Wall Following	4-3-3	5-2-3	7-6-5
Diabetes Diagnostic	5-4-4	4-3-3	2-3-0
Dataset	H2	IG	1R
Housing Prices	3-3-1	4-4-2	3-3-3
Congressional Voting	2-1-1	1-1-1	2-1-1
Wall Following	4-2-3	4-3-3	4-3-3
Diabetes Diagnostic	1-1-0	1-1-0	5-3-4
Dataset	RC	RF	SVM
Housing Prices	2-0-1	6-6-4	6-6-3 4-1-4
Congressional Voting	5-3-4	5-3-3	6-5-5
Wall Following	5-4-4	7-6-5	5-3-5
Diabetes Diagnostic	3-2-2	4-2-5	6-5-4

Table 8: Ranking method comparison summary on variation scale of 0 to 10.

Of all represented methods Chi Square and Information Gain give the best support for the proposed method of Class Overlay Counts. This is not a surprise because Chi Square and Information Gain are based on the same idea in the guise of probabilities. Estimated Overlay is of the same type, but is much dependent on data. On one occasion we see a significant departure from the principal method, and on a different occasion we obtain fully indifferent ranking, thus making the comparison trivial. Estimated Overlay offers, otherwise, a very undemanding alternative to the main method.

One Attribute Rule and Single Feature k-NN give the proposed method a more cautious support than Chi Square or Information Gain. Single Feature k-NN is a wrapper and One Attribute Rule can be interpreted as a wrapper, because it calculates the prediction error. Single Feature k-NN has specifics that can make it insensitive to irrelevant features, as k-NN switches into the mode that simply predicts the biggest class, and on imbalanced data-sets this accuracy can be high. This, however, affects only the end of ranked list. Wrapper methods are thus potentially exposed to a loss of detective ability on features of little relevance, and so the backward elimination makes a wrong design of feature selection algorithms relying on wrapping.

While ReliefF is not a wrapper and Feature Excepted k-NN is, these two methods use the same principle of k-NN and do produce similar results but are less supportive of the proposed ranking, even to the point of disagreement. Interestingly, ReliefC, despite being akin to ReliefF, gives much closer results overall. Feature-wise distance differences in ReliefF do appear more ambiguous than class overlap in ReliefC. As to Feature Excepted k-NN, it has the limitation that the impact on classification accuracy of a single feature missing from the set can be very small, resulting in features rating close assigned same rank. Also, the method can mistake redundant features for those with little expression.

A Support Vector Machine (SVM) has a very different design than the rest of methods, although it is not a fact that SVM has much a different idea about what the correct ranking should be like, because the independent results by other authors for the Housing data, appearing as the second line, are encouraging. We found though that SVM can be computationally very demanding and unpredictable even on small feature-sets. The necessity to output a unique ranking possibly makes the algorithm loop.

Overall, probabilistic schemes used for comparison are in a good alliance with the feature ranking method we propose, better than techniques not using probabilities, while the proposed two alternatives to the main method are competitive.

4 Conclusion

In this paper an approach to dimensionality reduction of the problem space through feature selection is proposed. It is based on the concept of coherent accumulation of data about class centers for informative features. Those in accord with this property can be short-listed to represent the data or, alternatively, discordant features can be removed, allowing for faster classification and data acquisition. The conclusive rating of features becomes known after the first cycle of the algorithm, making it possible to do without selection refinement of residual feature-sets.

Comparison with other methods of feature ranking shows a good correlation in many cases. However, assumptions the proposed algorithm relies on must be upheld. Firstly, the model should allow interpretation of classes as unique, rotund in shape sets, or classes can be subdivided into such clusters. Secondly, better results can be expected on statistically abundant data. The former poses a dilemma between getting quick results and getting the data model right first. Quicker results may be desirable in some circumstances, so alternatives in the spirit of main algorithm are considered, although the clustering does not bear hugely on performance. The latter is rather broad. In this regard the method shares assumptions of many other algorithms using

misclassification counts, whether in the guise of probability or accuracy of classification.

The algorithm outputs a ranked list, where it is only possible to say that a feature up the list is more relevant than a feature down the list. It is impossible to brand a feature irrelevant, although wrapping, supplementing results of ranking, can help to make the deselection. It appears that removal of less informative features from the end of the list may result in a temporary increase of classification accuracy before its starts to fall. This indicates that features removed may be irrelevant. The classification on results of ranking may also show that some, listed one after another features are similar by their action, that the top feature in a lot carries essentially the same information as the rest. The accuracy plateaus when these features get removed. However, ranking by itself cannot answer the question of redundancy.

5 Theoretical Aspects

Assume, without loss of generality, that there is just one set A. The objective function in Problem 1 then can be expressed as follows:

$$1/|A| \cdot \sum_i \|x - a^i\|^2 \; = \;$$

$$\sum_l (1/|A| \cdot \sum_i |x_l - a_l^i|^2). \qquad (3)$$

Proposition 1 is consequent from following.

Lemma 1 *Minimizer of Problem 1 obtained on Step 2 of Algorithm 1 is the by-coordinate minimizer.*

Proof. The objective function is representable in a form of sum of non-negative continuous functions of their arguments, according to Expression 3. Because a global minimum exists for each of the components it exists for the compound function. The exact location of the minimum is governed by interaction between components. In this case components are independent of each other. Therefore, the compound minimum is sum of minimums of the components. Besides, each component is represented by a single variable and all variables are included in the total. Thus, minimizers of Problems 1 in respect of coordinates together make the minimizer of Problem 1. △

The above applies to the Euclidian metric. However, it is easy to see that Lemma 1 also holds for the Manhattan metric, all what is required is omission of squares in Expression 3.

Lemma 1 is essential for Algorithm 1. Nevertheless, even a stronger result holds.

Proposition 2 *Solution to Problem 1 is the centroid of elements making the class in the Euclidian metric.*

Proof. Taking partial derivatives from Expression 3 for the objective function by each of coordinates l and equating them to zero we obtain:

$$\sum_i (2 \cdot x_l - 2 \cdot a_l^i) = 0.$$

It immediately follows that

$$x_l = 1/|A| \cdot \sum_i a_l^i,$$

which is exactly the by-coordinate expression for the centroid vector. The solution delivers a minimum, because second derivatives are all greater than zero, and so the Hessian matrix of the objective function at the point is positive definite. It is also the only minimum as no constraints apply. △

To do the same in the Manhattan metric we do require Lemma 1 though.

Proposition 3 *Solution to Problem 1 is the medoid of elements making the class in the Manhattan metric.*

Proof. Expression 3 for the objective function by coordinate, unsquared, with index l omitted for clarity, and scaling factor $1/|A|$, a positive constant, dropped for convenience, can be rewritten as:

$$E \; = \; \sum_i |x - a^i| \; = \; E_1 \; + \; \Delta E \; + \; E_2 \; ,$$

where

$$E_1 \; = \; \sum_i (a^p - a^i) \;\; , \;\; i \; \leq \; p \; ,$$

$$E_2 \; = \; \sum_i (a^i - a^p) \;\; , \;\; i \; > \; p \; ,$$

$$\forall \; p \; \in \; \{1 \; \ldots \; |A| - 1\}$$

and

$$\Delta E \; = \; (p - (|A| - p)) \cdot (x - a^p) \;\; , \;\; a^p \; \leq \; x \; \leq \; a^{p+1} \;\; .$$

This describes change of the objective function, linear on a segment between any two element values, all arranged in increasing sequence, which can be done without loss of generality. The first derivative, or slope of function E on this segment is

$$s \; = \; 2 \; \cdot \; p \; - \; |A| \;\; , \;\; a^p \; \leq \; x \; \leq \; a^{p+1} \;\; .$$

For small p it is negative as $|A| \; > \; 2 \; \cdot \; p$ and is increasing with p. Conversely, $s > 0$ for large p. Thus, E being continuous decreases with p, but reaches a minimum when the slope is minimal. This depends on whether $|A|$ odd or even.

If $|A|$ is odd, and so the number of intervals is even, the minimizer is

$$x^* \; = a^p \;\; , \;\; p \; = \; \lfloor |A| \; / \; 2 \rfloor \; + \; 1 \;\; .$$

If $|A|$ is even, and so the number of intervals is odd, the whole middle interval is the minimizer:

$$a^p \; \leq \; x^* \; \leq \; a^{p+1} \;\; , \;\; p \; = \; [|A| \; / \; 2] \;\; .$$

Often a single reference point is taken to represent a minimizer that is not unique. In this case by convention it is calculated as mean of interval ends:

$$x^* \; = \; (a^p \; + \; a^{p+1}) \; / \; 2 \;\; .$$

Thus found point for any $|A|$ is known as the median of increasing sequence of values and for all coordinates as the medoid. △

The term of medoid was introduced by (Kaufman and Rousseeuw 1987), so as not to confuse a new notion with that of median. In their clustering algorithm medoid is the instance chosen to be central

in a cluster. Elsewhere, the notion of medoid is being used in the sense of definition above. The two do not contradict. Indeed, the reference point can be shifted to any of vertices of the minimum.

Acknowledgement

The authors are thankful to anonymous reviewers of this submission for supplying valuable and stimulating comments on how this paper can be improved.

References

Bagirov A. M. (2008), Modified global k-means algorithm for minimum sum-of-squares clustering problems. *in* 'Pattern Recognition', 10(41): 3192-3199.

Bagirov A. M., Rubinov A. M., Soukhoroukova N. V., Yearwood J. (2003), Unsupervised and supervised data classification via nonsmooth and global optimization. *in* 'TOP: Spanish Operations Research Journal', 11(1): 1-93.

Bhamidipati N.L., Pal S.K. (2006), Comparing rank-inducing scoring systems. *in* 'Proceedings of ICPR 2006 - 18th International Conference on Pattern Recognition', 3: 300-303. IEEE.

Bi J., Bennett K. P., Embrechts M., Breneman C. M., Song M. (2003), Dimensionality reduction via sparse support vector machines. *in* 'Journal of Machine Learning Research', 3: 1229-1243.

Fayyad U.M., Irani K.B. (1993), Multi-interval discretization of continuous-valued attributes for classification learning. *in* 'Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence', 1022-1027. Morgan Kaufmann.

Freire A.L., Barreto G.A., Veloso M., Varela A.T. (2009), Short-term memory mechanisms in neural network learning of robot navigation tasks: a case study. *in* 'Proceedings of LARS 2009 : the 6th Latin American Robotics Symposium', 1-6. Valparaiso, Chile.

Harrison D., Rubinfeld D.L. (1978), Hedonic prices and the demand for clean air. *in* 'Journal of Environmental Economics and Management', 5: 81-102.

Holte R.C. (1993), Very simple classification rules perform well on most commonly used databases. *in* 'Machine Learning', 11: 63-91.

Huda S., Jelinek H., Ray B., Stranieri A., Yearwood J. (2010), Exploring novel features and decision rules to identify cardiovascular autonomic neuropathy using a hybrid of wrapper-filter based feature selection. *in* 'Proceedings of the Sixth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP)', 297-302. IEEE.

Kaufman L., Rousseeuw P.J. (1987), Clustering by means of medoids. *in* 'Y. Dodge (editor) Statistical Data Analysis based on L1 Norm', 405-416. Elsevier / North-Holland.

Keller J.M., Gray M.R., Givens J.A. (1985), A fuzzy k-nearest neighbor algorithm. *in* 'IEEE Transactions on Systems, Man and Cybernetics', 15(4): 580-585.

Kerber R. (1992), Chimerge: discretization of numeric attributes. *in* 'Proceedings of the Tenth National Conference on Artificial Intelligence', 123-128. AAAI / MIT Press.

Kira K., Rendell L. (1992), The feature selection problem: traditional methods and a new algorithm. *in* 'Proceedings of the Ninth National Conference on Artificial Intelligence', 129-134, AAAI Press.

Kononenko I., Robnik-Šikonja M. (2008), Non-myopic feature quality evaluation with (R)ReliefF. *in* 'Liu H., Motoda H. (editors): Computational Methods of Feature Selection', 169-191. Chapman & Hall / CRC.

Liu H., Setiono R. (1997), Feature selection via discretization. *in* 'IEEE Transactions on Knowledge and Data Engineering', 9(4): 642-645.

MacQueen J. (1967), Some methods for classification and analysis of multivariate observations. *in* 'Proceedings of the Fifth Berkley Symposium on Mathematical Statistics and Probability', 281-297.

Narendra P.M., Fukunaga K. (1977), A branch and bound algorithm for feature subset selection. *in* 'IEEE Transactions on Computers', 26(9): 917-922.

Quinlan J. (1993), C4.5: Programs for machine learning. Morgan Kaufmann.

Rissanen J. (1978), Modeling by shortest data description. *in* 'Automatica', 14: 465-471.

Saeys Y., Inza I., Larrañaga P. (2007), A review of feature selection techniques in bioinformatics. *in* 'Bioinformatics', 23(19): 2507-2517.

Schlimmer J.C. (1987), Concept acquisition through representational adjustment. *in* 'Doctoral dissertation', University of California at Irvine, USA.

Stranieri A., Yatsko A. (2009), Capped k-NN editing in definition lacking problems of classification. *in* 'University of Ballarat Research Repository', 1-16. Internet: "http://arrow.edu.au".

UCI Machine Learning Repository. Internet: "`http://mlearn.ics.uci.edu/`"

Weka Data Mining Tools. Internet: "`http://www.cs.waikato.ac.nz/ml/weka/`"

Wilson D. R., Martinez T. R. (2000), Reduction techniques for instance-based learning algorithms. *in* 'Machine Learning', 38: 275-286.

Witten I.H., Frank E. (2005), Data mining: practical machine learning tools and techniques. 2-nd edition. Morgan Kaufmann.