Sequential Feature Analysis in a Floating Search Evaluation and Extraction of Weak Metaclassifiers

Análisis Secuencial de Parámetros en Evaluación de Búsqueda Flotante y Extracción de Metaclasificadores Débiles

EDWIN ALBERTO SILVA-CRUZ
Ingeniero Electrónico, Magister en Ingenierías, M2R Signaux, Images, Parole et Télécommunications, Candidato a Doctor
Universidad Industrial de Santander
edwin.silva.c@gmail.com
Bucaramanga, Colombia

CARLOS HUMBERTO ESPARZA-FRANCO
Ingeniero Electrónico, Candidato a Magister
Universidad Industrial de Santander
carlosesfra@gmail.com
Bucaramanga, Colombia

Fecha de recibido: 15/12/2013
Fecha de aceptado: 22/11/2014

Forma de citar: SILVA, Edwin, ESPARZA, Carlos. Sequential Feature Analysis in a Floating Search Evaluation and Extraction of Weak Metaclassifiers. Rev.UIS Ingenierías,2015,vol.14,n1,p.p 45 - 57.

RESUMEN

La extracción de parámetros es una de las tareas más exigentes en el diseño de un sistema de clasificación. En este artículo se presenta un nuevo algoritmo de evaluación y búsqueda flotante enfocado en parámetros débiles. En problemas de clasificación con un elevado número de parámetros débiles, un protocolo de búsqueda exhaustiva de parámetros es limitado por costos de cálculo. La propuesta reduce notablemente los costos de cálculo de la búsqueda de parámetros comparado con técnicas convencionales bottom-up, top-down y flotantes, así como otras técnicas recientes, sin reducir el desempeño del sistema de clasificación. La metodología propuesta fue probada en un problema de reconocimiento de 7 clases de expresión facial y los resultados muestran la viabilidad del método para problemas multiclase con parámetros débiles.

PALABRAS CLAVE: Búsqueda de Parámetros, Reconocimiento de Patrones, Minería de Datos, Reducción de Dimensionalidad, Sistema de Clasificación.

ABSTRACT

Feature extraction is one of the most challenging tasks in the design of a classification system. In this work we present a novel floating evaluation and search algorithm focused on weak features. In classification problem with a high number of weak features an exhaustive feature selection protocol is calculation cost prohibitive, so in our approach a floating method is proposed with restricted feature subset evaluation. Our proposal considerably decreases the calculation costs of feature search compared with conventional bottom-up, top-down and floating techniques, as well with other recent techniques, without reducing the classification performance. The proposed methodology was tested for 7-class facial expression recognition and the results show the viability of the approach for multiclass problems with weak features.

KEYWORDS: Feature Search, Pattern Recognition, Data Mining, Dimensionality Reduction, Classification System.

1. INTRODUCTION

Feature selection is defined as the activities performed to select relevant features from a full set of features such that the new subset of features is better or similar to perform classification. In general, there are two main reasons to exclude a feature from the original set. First, the feature does not provide information about the classes, which not only does not help in the classification problem, but it could decrease the classification accuracy due to the added statistical noise. Second, the feature is redundant. This is, the information conveyed in the feature is already present in other feature/features, via identical or very similar data or linear or nonlinear combination of data from other features. Feature selection is mainly used when the number of points in the main dataset is relatively small and the number of features is high.

The main objective of feature selection is to obtain a new dataset so that the complexities of the description and the classification system are reduced (Guyon, 2006) there is better generalization of the problem and the possibility of overfitting is prevented. There are several widely used techniques to perform feature selection. The simplest approach is to train and validate different classifiers using the whole probable subsets of features. However, this approach is rarely used but in cases with very small number of features, because the combination of features in different subsets increases very quickly when the number of features is higher. Another possibility more widely used when the number of features is high is to iteratively increase or decrease the size of the selected features dataset. Stepwise regression can be performed to either start with the full features dataset and iteratively discard features (backward elimination), to start without features and iteratively add features (forward selection) or a combination of the two techniques (bidirectional elimination)[1].

There are classification problems where the number of features is considerably high but the individual correlation between each feature and the classes is not high. On the other hand, the high number of available features makes it possible to build a strong classifier given adequate selection and weighting of the weak features. Unfortunately, conventional feature selection in these cases is generally cumbersome, because it involves either the individual ponderation of each feature, whose result is not accurate enough given the poor discrimination power of each feature, or wrapper methods involving the evaluation of feature subsets, which can be prohibitive given the iterative nature of the method plus the high number of features involved.

The exhaustive search of features by performing evaluation of combinations of subsets of features is a cumbersome problem due to the high number of calculations. In (Devijver,1982) it is shown how the exhaustive search of 10 features in an universe of 100 features requires more than 1013 feature subsets evaluations. As a consequence, more practical and faster feature search methodologies are needed. There have been recent works whose main goal is to optimize the feature search process, by reducing the calculation times without decreasing the classification performance. Some of these works are (Nakariyakul, 2009), (Peng, 20104), (Gheyas, 2010). With our work we provide a novel approach, especially suited to multiclass classification with high number of weak features.

Feature subsets are evaluated, but instead of doing a full evaluation of the possible feature subsets each added and deleted feature iteration, we perform evaluation of a limited number of feature subsets. The main drawback of a limited evaluation is that it can prevent the inclusion of a feature whose addition can considerably increase the classification power of the feature subset, but this inconvenience is solved by using individual performance metric per feature, so the likeliness of the addition of a relevant feature to the feature subset is increased. In section 2 a brief theoretical background of feature selection methodologies is presented. In section 3 our proposed methodology is presented. In section 4 we will discuss the results, including a case test. Finally, in section 5 the conclusion and final observations are shown, as well as future prospects of our proposed methodology.

2. THEORETICAL BACKGROUND

In a classification problem there is a feature set that includes information that can or cannot be relevant as discriminator of the different classes (Z,Lu, 2010). As such, it is desirable to perform a methodology that reduces the size of the feature set without affecting the classification performance (Gheyas, 2010). There are

[1] In our case it was decided to use a different approach similar to forward selection but not starting from an empty features dataset. The logic was that it was clear that some spatial features would provide important information about facial expressions, such as features located around the eyes, eyebrows, mouth and frown regions, so it was not necessary to perform an algorithm to add these particular features to the classification system and instead, by manually including them, some computation cost was saved.

two main approaches to achieve this objective. The first approach is to try to eliminate features that are not relevant for the classification problem, either because they do not have discrimination capabilities or because even being relevant, their information is already present in another feature or set of features. The second approach is to perform some data transformation in order to project the original dataset into a lower dimensionality dataset where the class discrimination information is preserved or even enhanced.

In either case, for supervised feature selection or extraction there is a criterion function that has to be maximized. For feature selection, the criterion function provides a metric that typically depends on the classification accuracy, distance between probabilistic distributions, feature subset complexity and distances between classes. The general conventional criterion function for a full feature set of size *p, X_1,…,X_p*, evaluates all the feature subsets of size *d, χ_d*, as show in equation 1

$$J(\tilde{\chi}_d) = \max_{X \in \chi \kappa} J(X) \qquad (1)$$

Feature selection accomplishes several goals. Most important of them are:

- Only relevant features remain in the feature subset, so redundant or irrelevant information is excluded from the original set.
- Reduced size of the required feature subset.
- The classification algorithm may be less complex, which reduces the overfitting probabilities and increases the classification generalization.
- If the features are individually measured/obtained, the elimination of some of these features makes it possible to prevent some of these measurements, which improves the data acquisition times.

2.1 Feature relevance and redundancy

Given a full feature set $\mathcal{X}$, the relevance of a feature *X* is given by how important is the feature in the criterion function. Broadly, if the exclusion of the feature affects the classification performance, the feature is relevant; if there are cases in which the exclusion of the feature from a feature subset affects the classification performance, the feature is mildly relevant, and if the exclusion of the feature does not affect the classification performance in any case, the feature is completely irrelevant.

Redundancy is defined by the information provided by each feature. A feature may be apparently relevant if it conveys discrimination capabilities between two or more classes, but it is possible the information is already covered by one or more features in the feature subset, so there is redundancy and this feature can be discarded nonetheless given the classifier is strong enough to extract that redundant information from the rest of the subset.

The main difficulty of the evaluation of relevance and redundancy is that the features must be treated as a set of features in order to determine their importance. In figure 1 an example is shown.

In the example in figure 1, the features and would not provide discrimination capabilities if measured individually, because classes I and II are equally likely no matter the value of each feature treated individually. However, when the two features work together, a perfect classifier can be built, for instance using a very simple classifier with naïve Bayesian trees. This shows how it is important to work with feature subsets instead of treating each feature as an isolated entity when performing feature selection.

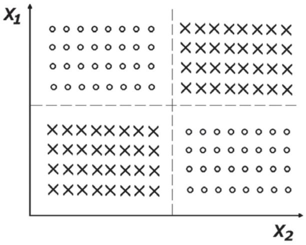

Figure 1. *Mutual Cooperation between two individual weak features*

Once the feature subset decision methodology is defined, it is necessary to use an algorithm to evaluate the performance of the new dataset. Whereas a simple metric using the classification accuracy with each subset sounds reasonable, there is a flaw in this approach. Given a powerful enough classifier, the best classification accuracy will possibly be obtained when more features than strictly needed are used. This phenomena was explained in (Trunk, 1979), which shows that the infinite addition of relevant features, even if redundant, decreases the classification accuracy when N→∞. However, the paper shows that this event is not so marked with smaller values of N, especially if the number of samples is considerable, such as in our case, so the classification can achieve good results with relevant but redundant features or with noisy features given the number of features is not higher than approximately 400 given the number of samples is higher than 100. This happens because a very complex classifier can be tolerant to some degree of noise and, especially, redundancy, without reducing the accuracy. Incidentally, that is the reasoning underneath the AdaBoost techniques, which use a high number of weak and severely redundant features to obtain an accurate classification. Consequently, when the classification system used to measure the performance of a feature subset is sophisticated to some degree it is more sensible to obtain a metric that optimizes a function, for instance an algorithm whose output is higher the most fitted the subset is, but whose output decreases the higher the number of needed features. Additionally, our classification algorithm was designed to be simple to specifically avoid this inconvenience.

2.2 Feature selection methodologies

There are three main approaches in order to perform feature selection. Filter technique is based on the features themselves, so the metrics involved are usually statistical measurements between features, typically distances and dependency. Wrapper technique is based on the classifier, so different subsets are evaluated and the conventional metrics are dependent on classification performance. Embedded methods include the feature search in the classifier design:

- Filter: Filter techniques are independent from the classifiers. Designed filters measure the individual features and depending on the obtained relevance, dependency between features and classes and other metrics, some features are eliminated and a feature set is obtained. The main drawback of this approach is that it is possible that a subset of features has great performance in opposition to the individual performance of each feature (such as in the example shown in figure 1), so the feature elimination may lead to worse classification.
- Wrapper: Wrapper techniques are computationally costly. Different feature subsets are evaluated and the performance is measured, generally via global classification accuracy and complexity of the dataset or the classifier. Whereas these techniques include the classification performance as a metric, their results are typically better than those obtained when using filter techniques. However, wrapper techniques are very limited in some real problems, for instance when the number of features is high, because in these cases it is difficult and lengthy to try an adequate number of feature sets that include a good range of feature likelihood.
- Embedded methods: Embedding methods are also depending on the classifiers. However, the feature set search is included in the classification design. As a consequence, the addition or elimination of features from the feature set is an integral part of the process instead of wrapping.

These techniques require methods to measure the characteristics of either the features (typically for filter techniques) or the classifiers (typically for wrapper and embedding techniques) (Aha, 1996).

2.3 Feature and classifier metrics

In (Weed, 2011) there is a general description of these evaluation measures. We will present a brief review of these measures including personal observations about the utility, pros and cons of each measures, in order to have an introductory idea to the measures used in our work.

Feature metrics

Feature metrics involve the evaluation of the individual performance of each feature, mostly to eliminate statistical noise due to irrelevant features, and the feature redundancy. These metrics are independent from the classifiers and models, so they can perform poor discriminability compared to classifier metrics, but they are easily implemented and they are sometimes mandatory when classifier metrics are hard to perform due to the size of the feature dataset or the data complexities.

- Feature ranking: Each feature gets a score depending on its classification performance either individually of, if possible, within a feature subset. The principal issue of this metric is that the individual performance of a feature does not necessarily mean it is not useful for classification in the context of a feature subset.
- Mutual information and correlation: The subjacent idea is that if a feature has relevant information, the correlation and mutual information scores between the feature and the classes should represent that. This has a similar problem to feature ranking. For instance, it can exist two features and with lower mutual information and correlation to classes in a 2-class problem whereas there is a feature whose scores are higher, but the ensemble of and provides a global better classification.
- Intraclass distance: This metric involves the calculation of distances between samples from different classes. The idea is that if a feature serves as a class discriminator, the distances between

members from the same class should be relatively low (in comparison to interclass distances). If there is a feature whose presence increases the intraclass distances, it could be an indicator that this feature is not relevant in class discrimination and it is adding noise to the system.

- Probabilistic distance: Given the conditional probability density functions, this measure evaluates the probabilistic distances. The main inconvenience of probabilistic-based approaches is that they are not that useful for continuous variables, the number of samples has to be high enough to determine with certain accuracy the conditional probability density functions and sometimes the data does not fit conventional functions.

Classifier metrics

These metrics relies mainly on the classification accuracy depending on the feature subset used. As such, they are typically used with wrapper and embedded methods. Whereas literature commonly uses classification accuracy as the best classifier metrics, there are other possible measures that can evaluate the usefulness of different subsets of features.

- Classification accuracy: This is the most direct classifier metrics. After all, the goal of a classifier is to obtain the best classification accuracy, so one metric that evaluates this would most likely contribute to obtain a good features subset.
- Metrics based on learning: Instead of relying exclusively on the classification accuracy itself, this metric involves the evaluation of the learning processes of the classification system when different feature subsets are used. The evaluation of learning curves with horizontal axis representing number of features allows to determinate if there is likely an inadequate number of features (the learning error keeps getting lower the more features are used without entering overfitting zone) or if there are more features than needed (the learning error remains almost constant after a number of features are added). The main issue of using learning curves is that the calculation required for each evaluation is higher than accuracy based methods, which is sometimes very troublesome if the training and evaluation of each classifier depending on the different feature subsets is lengthy. This is aggravated because the training and evaluation have to be performed several times for a n-folded or a bootstrapping validation method.

Each of the related metrics have their own methodology approach, so in case the reader is interested in more detailed information, we suggest the bibliography on (Wedd, 2011).

3. METHODOLOGY

In the case of weak features, each feature is weak because the classification accuracy of each one is not high, so their utility is based on the combination of a high number of wear classifiers to build a strong classifier. Given that, our initial hypothesis was that the evaluation of features would not only be lengthy (considering weak features generally involve a high number of features to compensate the poor reliability of each one), but the evaluation of the individual performance of each feature would not likely lead to any accurate result. Another methodology based on the evaluation of feature subsets could be more accurate, but the time requirements made this option lengthy and cumbersome. Moreover, preliminary tests showed us the relevance of each feature is very small, which is sensible because of the weak nature of the features, so high relevance for one or more of its individual components could not be expected. Instead of trying a brute force approach whose results would probably be inaccurate nonetheless, we decided to solve this issue by performing a floating feature search with novel restriction contributions.

In figure 2 an example with p feature vectors is shown. Each feature is a vector of variable length (corresponding to our test case of features for facial expression recognition) and there are a high number of feature vectors, but the class discrimination capabilities of each one is relatively low.

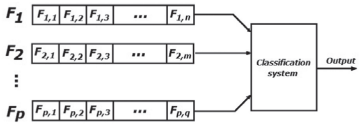

Figure 2. *Feature vectors in a multiclass classification problem*

There is a set of vectors corresponding to each feature per sample. In the preliminary classification tests each vector was given equal vote in the classification contest. However, it is expected that the contribution of each vector is different and even in some cases some of them are completely irrelevant or redundant for the classification, so they should be discarded. A 2-class problem is easier because an irrelevant feature is safely discarded, but our methodology was designed for multiclass problems (our case example is a 7-class problem), so some features may be irrelevant in the discrimination of one or more classes, but important in the discrimination of at least one of the classes, so the treatment of the feature selection process must be careful.

Our approach was a sequential search of the feature subset that optimized the classification accuracy given some restrictions. For this, the individual discrimination power of each feature was obtained. This part was tricky, because one feature whose accuracy in all the 1 vs. 1 contests except one is 50% and the other contest accuracy is 100% would have low global accuracy (52,38% in a 7-classes problem), which can be regarded as very low because a random classifier in 1 vs. 1 contests would have approximately 50% accuracy. However, this particular feature is very good in one of the contests, so it is a very reliable discriminator of two classes. As a consequence, the score of each feature was double, given by its overall classification accuracy and the highest individual 1 vs. 1 accuracy. With these regards, this process is incomplete. In a full multiclass classification problem every 1 vs. 1 contest should be taken in account. However, a 7-class problem has 21 1 vs. 1 contests and if we consider each feature selection iteration has a n-folded validation stage, it would be impossible to perform such kind of complete feature selection algorithm in a reasonable period of time. The sequential feature selection was neither top-down nor bottom-up, because we had intuitive information about which features could be adequate for the starting feature set[2].

Once the initial feature subset was obtained, we defined the feature criterion function as the simple classification accuracy of each one of the expressions, taken individually. There are several ways of measuring the performance of a feature subset, for instance the methodologies used in (Sun, 2010), (Gu, Q., et al. 2010), (Hancazar, B., et al. 2010). In our work we decided to use classification performance because it is a direct way to measure how the classification system is accurately performing its job, while keeping low calculation costs. By doing this individually, we tried to guarantee that a feature whose classification power contribution was low for some classes but high for one or two of them was still included. In equation 2 the score per feature is shown. c corresponds to the classes in the multiclass problem.

$$J\,(Xi) = \max(Sc(Xi)), \quad c = 1, \ldots, C \qquad (2)$$

Full training and classification validation were performed with the initial feature subset and the performance scores were obtained. The next step in most reviewed sequential algorithms is to iteratively include or discard one or more features such that the score function was optimized (maximum increase in bottom-up methods and minimal decrease in top-down methods). However, in our proposal we do not proceed this way due to two main considerations. First, the number of features is considerably high, so each iteration would imply the training and validation of a high number of different feature set possibilities[3].

Second, we need to guarantee the classification validity for each one of the classes, so the problem size is multiplied. Instead, we previously obtained an individual feature score for each of the features in the complete set. For the first added feature we performed 4 tests, including each one of the four highest scorers from the features outside of the selected set. We reckon this novel approach has an important drawback, because the fact that one feature has relatively high individual score does not mean it will have an important classification contribution. This happens because the feature can have high correlation with one or several of the features already included in the set, so its inclusion would not contribute to the classification accuracy no matter its individual relevance. The opposite is possible too, in a feature discarding iteration: a low individual score does not necessarily mean the feature is irrelevant, due to the fact it can work together as a team with other features to produce a stronger classification. However, performing full training and validation with the whole different possible arrangements of new feature subsets is lengthy and costly, whereas 4 training and validation stages per iteration are more reasonable and we consider 4 was an adequate number to provide the algorithm with different possibilities so to choose the best one, and that way preventing the inclusion of a high individual score feature whose global contribution was very small or null.

[2] In our case example, they correspond to the spatial facial areas in a facial expression video sequence, including the eyes, eyebrows, mouth and frown, but discarding areas around the chin, cheeks and peripheral facial regions whose contribution was not evident for facial expression.

[3] For example, in a problem with 256 features performed bottom-up for sequential forward selection (SFS) with 1 feature included each step, the first iteration has 256 training and validation stages, the second iteration has 255 training and validation stages and so on. The stages per iteration are increased if the number of features included each selection is higher, because different feature subset combinations have to be tested.

The feature subset score per iteration is given by the simple error estimation of the classification in each stage, as in equation 3.

$$J\,(X) = \frac{1}{n}\sum_{nf=1}^{n} \;(1 - CE\,(X_{nf}\,)) \tag{3}$$

n is the number of folds in the leave-subjects-out process per step, *nf* is the n-folded step, is the classification error in the nf-th folded step. As such, the feature subset score is the average classification accuracy for the *n* folds in the training and validation per iteration.

When the first sequential feature selection is performed, one sequential feature discard follows, using a similar approach: the 4 worse individual scorers are selected and training and validation are performed with the current set minus each one of these features. The new feature set whose performance is the best, probably the one that guarantees a minimal accuracy decrease, was selected and the remaining feature is discarded. However, if the new accuracy is worse than the original one, the feature is not discarded, because that would mean the original subset with the same number of features had better performance than the new one, so the eligible feature to be eliminated has non negligible relevance. So far, the process is very similar to a floating search method (Pudil, P., et al. 1994), (Somol, P., et al. 1999). However, in floating search methods features continue to be discarded until the feature set function score is lower than in the previous iteration. In our case we did not perform it that way because our feature function score is not as simple as in a 2-classes problem, so the recursive elimination of features could lead to discard features important for some of the class discriminations. Evidently, this could have been helped by performing feature selection for each one of the classes, but this would mean this whole process had to be repeated several times due to the number of classes.

The process is iteratively repeated until some condition is met. Each time a feature is added, it gets a flag, so it is not removed in the same iteration. However, if features that were added in previous iteration start to get discarded and the function score does not increase between the previous and current iteration, it probably means there are not features remaining in the eligible feature subset (features that do not belong to the current feature subset) whose inclusion improves the classification accuracy. If this is the case, the process stops and the final feature subset is the subset from the last iteration.

The process is better explained in the pseudoalgorithm 1. In the pseudoalgorithm, represents the K features for the i-th sample in the complete dataset.

Algorithm 1 Pseudoalgorithm for feature selection

```
Procedure Input
    $X_k = X_0$                    // $X_0$ is manually initialized
    $Y_k = U - X_k$
    $X_{k,i} = \{x_1, x_2, ..., x_N\}$
While End conditions are not met do
Procedure Leave-subjects-out data generation
    $X_{k,i}$ is separated in disjoint leave-subjects-out training
    and validation sets
Procedure Training
    Train the classification algorithm using $X_{k,i}^{tr}$
Procedure Validation
    The classification accuracy is obtained using the
    validation subsets
    $W_k = Y_k$ ;   $b = \Phi$ ;   $w = \Phi$ ;   $V_k = X_k^{val}$
    for $i = 1$ to 4 do        the value 4 can be changed according
                               to the problem
        $t$ = argmax $c \in W_k S(c)$ // $S\,(\cdot)$ is the individual score
                                             per feature
        $b = b + \{t\}$ ; $W_k = W_k - \{t\}$
        $t$ = argmin $c \in V_k S(c)$
        $w = w + \{t\}$;  $V_{k =} V_{k -}$ $\{t\}$
        $b$ and $w$ contain the 4 most and least individually
        relevant features from the eligible subset respectively
    end

Procedure Addition and Elimination of features
    y = argmax$_b$ J $(X_k^{val} \cup \{b\})$ // $J\,(\cdot)$  is the classification
                                                performance according
                                                to the feature subset.
    $X_k^{val} = X_k^{val} \cup \{y\}$
    $x$ = argmax$_w$ $J\,(X_k^{val} - \{w\})$
    $X_k^{val} = X_k^{val} - \{x\}$
Procedure End conditions
    The global accuracy does not increase after several
    iterations.
    The same features are recurrently chosen for the
    algorithm.
End
```

4. RESULTS

Our feature search algorithm was tested with real data from a set of facial expression recognition features obtained from the Cohn-Kanade databases CK and CK+ (Tian, Y., et al. 2001), (Lucey, P., et al. 2010) using Volumetric Patterns of Oriented Edge Magnitudes (VPOEM) and Temporal Patterns of Oriented Edge Magnitudes (TPOEM)[4], for a total of 256 facial

[4] The features VPOEM and TPOEM are described in the unpublished paper currently in review process that is attached as an annex

expression features, 4 features per each one of 8 x 8 spatial cells within the facial region. The VPOEM and POEM are based in the algorithms proposed in (Vu, Caplier. 2010), that was probed successfully on a facial expression case in (Silva, E., et al. 2010). The VPOEM and TPOEM codifications produce vectors of average length 33, so unlike many applications where each feature in a n-dimensional problem is a scalar or a discrete value, in our case each feature is a high dimensional vector itself. This makes for an interesting problem to test our methodology.

The initial feature subset was manually set, including areas that we considered important in the facial expression recognition. Not surprisingly, in the first iterations of the feature inclusion and elimination no feature was eliminated, because these first iterations sets were relevant for expression classification. However, after a number of iterations the feature set started to both include and eliminate features, which was the main point of the combined search methodology. At this point the relative simplicity of the classification system proved to be a strength. Given a complex enough classification system, the introduction of new features would almost always improve the accuracy of the validation tests. This happens because even if the recently added features do not provide important discrimination capabilities, the classifier would learn the new inputs, maybe incurring in some overfitting, but the classification results would not decrease.

For that reason some common function scores are created in a way that a higher number of features decreases the subset score. As a consequence, there is a point when even if a new feature slightly increases the classification accuracy, the subset score is lower than in the previous iteration, so the recursive algorithm stops. In our case this was not necessary, because for the classification we used a simple Mahalanobis distance metaclassification, where the average of each feature per expression was obtained and the Mahalanobis distance from each feature to the corresponding expression feature average was obtained and this metric was used to ponderate the expression score per feature. The final classification was the unweighted sum of the expression scores per feature. Given this simplicity, when a new subset of candidate features was evaluated, if neither of them had relevant classification information neither for global accuracy or at least one expression accuracy, the feature search stops. On the other hand, this approach has one important inconvenience. In a 2-class classification problem the proposed architecture was strong enough to indirectly measure the mutual information between some features. This is, if in the feature subset there were already one or more features that provided more information than one candidate feature, the tested subset with the new candidate would not have increased accuracy.

On the other hand, in a multiclass problem with a fusion classifier this is not necessarily the case. We will explain this issue with an example. Consider a feature subset with 10 features in a 3-classes classification problem. 5 of these features (features 1 to 5) provide important information in the classification of class III, whereas the other 5 features (features 6 to 10) are more fitted to discriminate classes I and II. Moreover, consider the information of the latter 5 features as statistical noise for the discrimination of class III. Now if there is a new candidate with relevant information for classification of class III, the tested classification accuracy of class III will probably increase no matter if the information was already included in the features 1 to 5, because the new feature subset has 6 relevant features and 5 statistical noise features for class III discrimination opposed to 5 and 5 in the former subset respectively. Consequently, our proposal was not necessarily strong enough to prevent the inclusion of redundant features from time to time, especially when these new features were very specialized. This issue could be theoretically solved by performing a similar feature search algorithm but with different classification architecture. However, this was not practically convenient for this work, because of the already related time consumption of the feature search algorithm that includes n-folded training and testing of several candidates per iteration, which would take several weeks of calculation if using more sophisticated classification architectures. Moreover, with a high number of features the design of classifiers that take in consideration the mutual information between features is complex, because of the exponentially increasing number of possible interfeatures dependencies when the number of features is higher. On the other hand, later tests showed that the final feature subset obtained by our methodology was good enough to reduce the feature extraction calculation while maintaining high classification accuracy even when the classification architecture was more powerful.

In figure 3 the normalized objective score against the iteration is shown. Whereas we mentioned the stop mechanism was when there were not any feature candidates that could manage to improve the score, we decided to extend the test for a few more iterations in order to see if this stopping point was effectively adequate. The plot is not very smooth at some points, but that was a limitation due to the election of 5-folded for the validation protocol. Had we chosen a higher number of folds, the plot would be smoother, but there was still the calculation costs problem, which would duplicate with a 10-folded methodology. However, a simple inspection of the plot shows that the variations in each iteration are relatively small, so the use of 5-folds in the leave-subjects-out validation is not harmful for the evaluation of the algorithm. Additionally, the plot is adequate enough to show the progression of the feature search: added features start increasing the value of the objective score, then it tends to stabilize with smaller increases and finally a point where the increases stop, which is the stopping command for the regular algorithm. However, as previously stated, a few more iterations were manually performed to continue evaluating the progress, and effectively, it seemed more added features would not lead to further progress.

The selected feature subset has a considerably smaller size than the original feature set, with 112 features compared to the original 256 features. Additionally, the spatial localization of the new subset greatly corresponds with the intuitively important features for facial expression recognition. Whereas there was not a priori information about the evident irrelevance of some features beyond intuitive estimation, the process supported the initial hypothesis. Moreover, the designed feature search is not limited to weak metaclassification in a multiclass problem where the relevance of each feature can be relatively estimated a priori by intuition, so our proposed feature search can be used in similar applications where the relevance per feature is not necessarily well known.

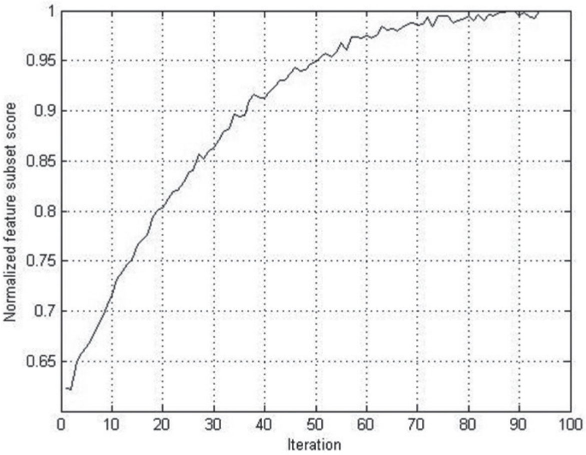

Figure 3. *Normalized criterion score vs. iterations*

In order to measure the stability and convergence of the feature search, a Jaccard coefficient (Jaccard, 1901), (Cha, 2007) was obtained. The Jaccard coefficient is shown in equation 4.

$$S(f_i, f_j) = \frac{|f_i \cap f_j|}{|f_i \cup f_j|} \tag{4}$$

The idea of this metric is to evaluate the ratio of common features and total features between two different feature subsets. We performed the feature search algorithm twice and we obtained the Jaccard coefficient at several iterations. The results are shown in table 1.

Table 1. *Jaccard coefficient vs. number of iterations. Identical initial subsets*

Iteration	0	20	40	60	80	94
$S(f_i, f_j)$	1	0,897	0,946	0,962	0,981	0,991

Naturally, before the first iteration the Jaccard coefficient is 1, because both subsets are equal. After the first iterations the score decreases because there are different discarded and added features in some iterations, so in iteration 20 there are 70 common features from a total of 78 features, in iteration 40 there are 88 common features from a total of 93 features, in iteration 60 there are 100 common features from a total of 104 features and the trend continues.

Consequently, we performed tests with different random initial feature subsets, without overlapping between the two subsets. These tests were limited to 25 iterations due to the time requirements of the full feature search. The goal of the tests was to establish if the Jaccard coefficient would increase from the initial value of zero, due to the non-overlapping subsets, to higher values due to common features being included. The average results for 3 tests are shown in table 2. More tests would have provided better results, because the variability of results between the 3 tests was considerable, mostly due to the different initial feature subsets. However, the tests were enough to show the convergence trend of the subsets when more iterations are performed.

Table 2. *Jaccard coefficient vs. number of iterations. Non-overlapping initial subsets*

Iteration	0	5	10	15	20	25
$S(f_i, f_j)$	0	0,056	0,101	0,158	0,209	0,248

The table 2 shows how the Jaccard coefficient steadily increases when 2 non-overlapping subsets of 54 features each are used to start the feature search. After 25 iterations the Jaccard coefficient is 0,248. This may seem low compared to the values of the previous table, but there is an important issue. Considering both feature subsets had 54 different features, best case scenario is if after 25 iterations both searches have included the same 25 features and eliminated 25 of the initial non-overlapping features. If that was the case, the number of common features after iteration 25 would be 25 and the total number of features would be 83, so the Jaccard coefficient would be 0,301. However, this best case scenario is only achieved if each iteration 1 feature is eliminated from both subsets, which is an unrealistic expectation because that would imply there are at least 25 useless features in both feature subsets and the feature search algorithm adds exactly the same features for both subsets.

Had we used a conventional floating search approach starting with the same 54 features initial subset, and assuming the average size of the floating search subset in the full process was around 90, which is a very generous estimation, this would have meant that the average number of training and validation n-folded steps in the same 94 iterations are: ((256 - 90) * 2) * 94 = 31208 , compared to (4 * 2) * 94) = 752 steps with our proposal, which proves the noticeable calculation cost reduction using our approach.

Our next tests are performed to do a benchmark between our work and some related works in the literature. The first comparison was with the work in (Yu, 2003). In this methodology a concept called predominant correlation is introduced with a filter method used to try to identify relevant and redundant features. The main idea is to use the conditional entropy of the variables to obtain the information gain.

The metrics based on entropy require discrete values, so in order to implement them we discretized our TPOEM and VPOEM vectors. This proved to be the main challenge of implementing this approach, because it is not possible to properly discretize the information of vectors of average length 34 without probably losing considerable information[5]. Naturally, the discretization cannot be manually performed and any unsupervised technique leads to the embedding of the data in lower dimensionality spaces where the interclass discrimination is probably lost if the high dimensional folds are complex and nonconvex, as in our case. Accordingly, the discretization was performed as a 1 class-based supervised dimensionality reduction problem using the metric learning by collapsing classes.

A common methodology mistake when using supervised dimensionality reduction is that in many works the data in the dimensionality reduction is also used in the validation. However, that is an error, because the supervised dimensionality reduction is a sort of a classifier, so the data should not be also used in the validation. Consequently, we separated the full dataset in 10 leave-subject-outs folds; 4 of them to be used in the discretization, 5 in the training and the remaining set in the validation[6].

Once the discretized subsets were obtained, the protocol to obtain the predominant features was followed. Notice that the algorithm is quite demanding because it requires the calculation of the symmetrical uncertainty (Press, W., et al. 1996) for all the features and classes. Moreover, the fact that a feature is not predominant does not mean it is not relevant for the classification problem, as in figure 1. Finally, the calculation costs of this methodology are very high, because instead of the evaluation of the performance of some feature subsets per iteration, it requires the calculation of the symmetrical uncertainty between the candidate feature and the current feature subset, which is highly demanding. We used the same initial subset as one of our previous tests, again with 94 iterations. This it self is not a fair comparison because the calculation of 94 iterations with this approach was approximately 7 times slower than with our approach, plus the dimensionality reduction and discretization problem, which was even more demanding.

Finally, we implemented the work proposed in (Deng, S., et al. 2013). In order to use a very similar approach, we built 21 SVM classifiers ($(\left(\frac{k(k-1)}{2}\right)$, where is the number of classes) using the SVM-MMFS and SVM-BFFS protocols. This, again, is a delicate methodology due to several issues. First of all, since the features are obtained based on their classification performance, the protocol was performed using leave-subjects-out. Secondly, as we found out in preliminary tests using 1 vs. 1 SVM classification per feature, this does not mean the selected features actually provide better class representation, because some features get easy high scores for the classification of easy classes (in our facial

[5] This discretization problem can be viewed as a dimensionality reduction problem from a very high dimensional space to a 1-dimensional discretized space, which, evidently, is prone to many errors.

[6] The methodology is similar to random n-folded but since there are samples from the same subjects (persons) in the facial expression databases, it is better not to use samples from the same individual, even if belonging to different classes, in both the training and validation sets.

expression problem, joy and surprise), so features that have to deal with hard to classify expressions get unfair low scores.

In this case we did not run a fixed number of iterations unlike the previous test, because this algorithm has its own end condition rules. The computation costs per iteration were higher due to the requirement of calculation of 21 SVM classifiers per iteration besides the calculation of weights and stabilization per iteration, which is highly demanding because it has to be done with each candidate feature, and the size of the candidate features subset is elevated, plus the classification performance per iteration. The use of the default end conditions meant another issue, because in most of our tests that meant the algorithm ended after a very high number of iterations (this is, it achieved maximum performance when a high number of features were added), so the final feature set was not drastically reduced in size from the original one. We reckon this technique can be very useful when there are some very powerful features and some weak features in the dataset, because the weak features are rarely added. This is clear in the cited paper, because the maximum number of iterations shown there is 10 iterations, and the example shows 100% of classification performance after just 2 iterations (so only 2 features in the subset). However, when using a diverse set where most of the features are weak plus some irrelevant, the algorithm is not bold enough to eliminate features that provide a slight but almost negligible help in the class discrimination.

The calculation costs per iteration are shown in table 3 (notice the FC-BF calculation costs do not include the calculation costs of the MCML discretization we used to estimate the conditional entropy, which in our case took more than 2 hours calculation).

Table 3. *Calculation costs per iteration using several techniques*

Technique	Calculation cost per iteration
SFA-WC	11.397s
FC-BF	3.104s
SVM-MFFS	87.822s

While it is possible that with the optimization of the code some calculation costs may be reduced, the results are similar to the results in the bibliography, with iteration time close to 450ms per iteration for FC-BF (in our case each iteration is actually 10 iterations, due to the 10-leave-n-subjects-out validations by iteration) and 41.92s by iteration for SVM-MFFS, but with a set of parameters drastically lower than the used in our work.

The results show that the reduction of the dimensions by FC-BF has a considerably lower computational cost by iteration compared to our technique and the reduction by SVM-MFFS is the most computationally expensive. Nevertheless, in the FC-BF the shown results are the computational cost by iteration, without including the reduction cost of the data to 1 dimension by MCML. The MCML stage cost was higher than the total computational cost of the parameter extraction algorithm.

Naturally, the performance of the parameter search algorithms should not be measured based on computing time, but also on the ability of parameter extraction discrimination without affecting the classification rate.

Table 4. *Classification and numbers of parameters with the complete data and reduced by SFA-MC, FC-BF and SVM-MFFS*

Tec.	Full	SFA-WC	FC-BF	SVM-MFFS
Ang	93.52	94.02	92.94	94.87
Dis	91.58	91.95	90.40	90.23
Fear	89.33	89.06	86.88	85.94
Hap	100.00	100.00	99.50	100.00
Sad	87.00	88.06	88.41	84.06
Sur	94.75	95.47	95.06	96.30
Neu	90.64	91.30	90.28	90.64
N. Feat.	256	112	150	163

In table 4 we show the classification by class with the original data and reduced by SFA-WM, FC-BF and SVM-MFFS, and the number of parameters by each technique, using the TPOEM data from the CK+ facial expression database. The number of samples per class are: anger, 117 samples; disgust, 174 samples; fear, 63 samples; joy, 198 samples; sadness, 69 samples; surprise, 243 samples and neutral, 288 samples, with 3 samples per individual per expression, without repetition of individuals in the training and validation sets (not even samples from the same individual but different class).

5. CONCLUSIONS

In this work we have shown how the sequential floating search of features from a dataset when each feature is weak and the number of features is high can be performed with reduced costs compared to full search techniques. Our algorithm is suitable for both binary and multiclass classification problems, because the individual score of each feature depends on its discrimination capabilities between 2 or more different classes, so a very specialized feature is included in the feature subset no matter if its global discrimination accuracy is not high.

In the results section it has been determined that the proposed algorithm has good performance and stability. The Jaccard coefficients are high and increasing after a few stabilization iterations, which is an indicator of the convergence and stability of our proposed methodology.

The case test we used in this work for facial expression recognition proved how the original feature set can be reduced from 256 vectors to 112 vectors. Whereas these vectors are obtained by the VPOEM and TPOEM codification of the detected face, this means the calculation and memory costs are reduced by more than 50% using the proposed technique, without affecting the classification accuracy, or even more drastically reduced with just slight performance decrease.

We performed tests to compare the results of classification with our dataset using raw data (no feature extraction), SFA-MC (our proposal), FC-BF and SVM-MMFS. The calculation costs using our proposal were dramatically lower than SVM-MFFS and higher than FC-BF. However, SFA-MC obtained a higher degree of reduction of features, whereas the classification accuracy did not seem to be harmed (actually, it seems to be slightly increased, but the data size is not enough to have statistical certainty of this improvement). Naturally, the use of other floating feature searches that requires the evaluation of a high number of subsets per iteration would be even costlier than SVM-MMFS.

Some possible improvements to this work are related to enhanced improvements on multiclass classification problems. It is possible to obtain different feature subsets, specialized in the classification of each class, instead of relying on the global classification. This inclusion would likely improve the final feature subset, more optimized to not only increase the global performance, but minimizing the classification errors of the most difficult to classify classes. This improvement was not included in this work because even if the feature search calculation costs were dramatically reduced, the multiclass individual contests inclusion would have required full feature search for each 1 vs. 1 contest, which would have been lengthy, with 21 different feature searches corresponding to each one of the 21 1 vs. 1 contests in our 7-class problem.

6. REFERENCES

GUYON, I *Feature extraction: foundations and applications*.Springer, 2006.

DEVIJVER, P; KITTLER, J. *Pattern recognition: A statistical approach*. Prentice/Hall International Englewood Cliffs, NJ, 1982.

NAKARIYAKUL, S; CASASENT, D “An improvement on floating search algorithms for feature subset selection,” *Pattern Recognit.*, vol. 42, no. 9, pp. 1932–1940, 2009.

PENG, Y; WU, Z; JIANG, J.“A novel feature selection approach for biomedical data classification,” *J. Biomed. Inform.*, vol. 43, no. 1, pp. 15–23, 2010.

GHEYAS, I; SMITH, L. “Feature subset selection in large dimensionality domains,” *Pattern Recognit.*, vol. 43, no. 1, pp. 5–13, 2010.

Z. Q; LU, J.“The elements of statistical learning: data mining, inference, and prediction,” *J. R. Stat. Soc. Ser. A (Statistics Soc.*, vol. 173, no. 3, pp. 693–694, 2010.

TRUNK, G. “A problem of dimensionality: A simple example,” *Pattern Anal. Mach. Intell. IEEE Trans.*, no. 3, pp. 306–307, 1979.

AHA, D; BANKERT, R. “A comparative evaluation of sequential feature selection algorithms,” in *Learning from Data. Springer New York*, 1996, pp. 199–206.

WEDD, A; COPSEY, K. *Statistical Pattern Recognition, third Edition*, Third Edit. Wiley, 2011, p. 642.

SUN, D; ZHANG, D.“Bagging constraint score for feature selection with pairwise constraints,” *Pattern Recognit.*, vol. 43, no. 6, pp. 2106–2118, 2010.

GU, Q. et al. “Generalized fisher score for feature selection,” *arXiv Prepr. arXiv1202.3725*, 2012.

HANCAZAR, B. et al. “Small-sample precision of ROC-related estimates,” *Bioinformatics*, vol. 26, no. 6, pp. 822–830, 2010.

PUDIL, P. et al. “Floating search methods in feature selection,” *Pattern Recognit. Lett.*, vol. 15, no. 11, pp. 1119–1125, 1994.

SOMOL, P. et al.“Adaptive floating search methods in feature selection,” *Pattern Recognigion Lett.*, vol. 20, no. 11, pp. 1157–1163, 1999.

TIAN, Y. et al. “Recognizing Facial Actions by Combining Geometric Features and Regional Appearance Patterns,” *Robot. Institute, Carnegie Mellon Univ. Pittsburgh, PA 15213*, p. 31, 2001.

LUCEY, P. et al. “The Extended Cohn-Kanade Dataset (CK+): A complete dataset for action unit and emotion-specified expression,” in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on*, 2010, no. July, pp. 94–101.

VU, N-S; CAPLIER, A.“Face recognition with patterns of oriented edge magnitudes,” in *Computer Vision--ECCV 2010*, Springer, 2010, pp. 313–326.

SILVA, E. et al. “POEM-based facial expression recognition, a new approach,” in *Image, Signal Processing, and Artificial Vision (STSIVA), 2012 XVII Symposium of*, 2012, pp. 162–167.

JACCARD, P. Coefficient: Jaccard, *Étude comparative de la distribution florale dans une portion des Alpes et des Jura*, vol. 37. Impr. Corbaz, 1901.

CHA, S-H. “Comprehensive survey on distance/similarity measures between probability density functions,” *City*, vol. 1, no. 2, 2007.

YU, L; LIU, H.“Feature selection for high-dimensional data: A fast correlation-based filter solution,” in *ICML*, 2003, vol. 3, pp. 856–863.

PRESS, W. et al. Flannery, *Numerical recipes in C*, vol. 2. Citeseer, 1996.

DENG, S. et al. “A feature-selection algorithm based on Support Vector Machine-Multiclass for hyperspectral visible spectral analysis,” *J. Food Eng.*, vol. 119, no. 1, pp. 159–166, 2013.