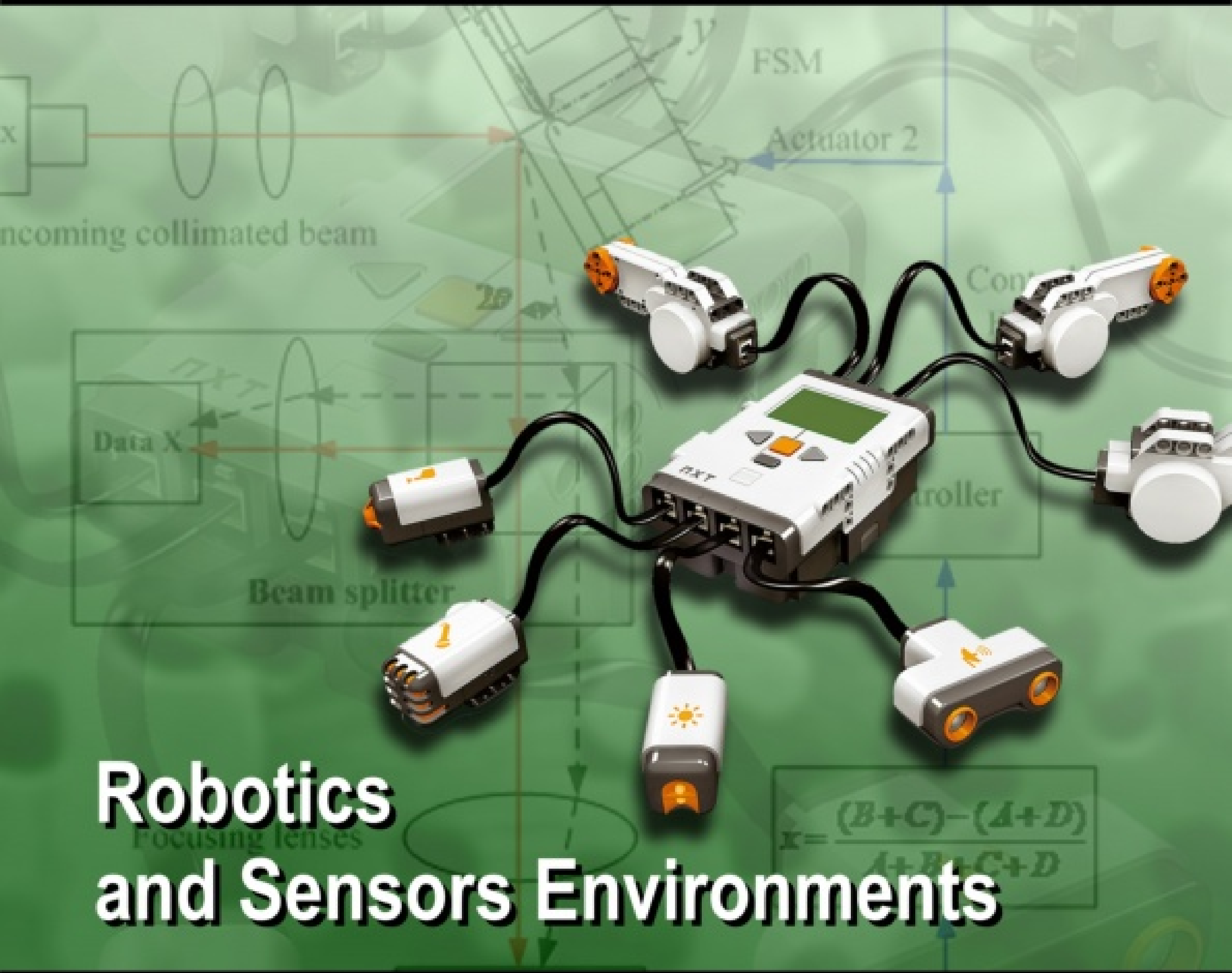
ISSN 1726-5749

SENSORS & TRANSDUCERS

3 /09
vol. 5
Special



Robotics and Sensors Environments

International Frequency Sensor Association Publishing
IFSA

Sensors & Transducers

Volume 4, Special Issue
March 2009

www.sensorsportal.com

ISSN 1726-5479

Sensors & Transducers Journal (ISSN 1726-5479) is a peer review international journal published monthly online by International Frequency Sensor Association (IFSA).
Available in electronic and CD-ROM.

Sensors & Transducers Journal

Contents

Volume 5
Special Issue
March 2009

www.sensorsportal.com

ISSN 1726-5479

Research Articles

Authors are encouraged to submit article in MS Word (doc) and Acrobat (pdf) formats by e-mail: editor@sensorsportal.com
Please visit journal's webpage with preparation instructions: http://www.sensorsportal.com/HTML/DIGEST/Submition.htm

International Frequency Sensor Association (IFSA).

Sensors & Transducers Journal, Vol. 5, Special Issue, March 2009, pp. 86-103

Sensors & Transducers

ISSN 1726-5479

http://www.sensorsportal.com

Feature Space Dimensionality Reduction for Real-Time Vision-Based Food Inspection

Mai Moussa CHETIMA and Pierre PAYEUR
School of Information Technology and Engineering, University of Ottawa
800 King Edward, Ottawa, ON, Canada, K1N 6N5
Tel.: 613-562-5800
E-mail: {m.chetima, ppayeur}@uottawa.ca

Received: 30 January 2009 /Accepted: 24 February 2009 /Published: 23 March 2009

Abstract: Machine vision solutions are becoming a standard for quality inspection in several manufacturing industries. In the processed-food industry where the appearance attributes of the product are essential to customer's satisfaction, visual inspection can be reliably achieved with machine vision. But such systems often involve the extraction of a larger number of features than those actually needed to ensure proper quality control, making the process less efficient and difficult to tune. This work experiments with several feature selection techniques in order to reduce the number of attributes analyzed by a real-time vision-based food inspection system. Identifying and removing as much irrelevant and redundant information as possible reduces the dimensionality of the data and allows classification algorithms to operate faster. In some cases, accuracy on classification can even be improved. Filter-based and wrapper-based feature selectors are experimentally evaluated on different bakery products to identify the best performing approaches. *Copyright © 2009 IFSA.*

Keywords: Machine vision, food inspection, quality control, feature selection, machine learning.

1. Introduction

As for several other manufacturing sectors, the food industry has been trying to automate the quality control processes in order to decrease production costs and increase the quality and uniformity of the production. Machine vision-based systems are of particular interest when it comes to measuring the superficial characteristics of a product for classification purposes. Most of the external quality attributes of a product can be inspected visually before the packaging line and items that do not satisfy the set standards are automatically rejected. Such machine vision systems have been used over a wide variety of inspection applications in the food manufacturing industry including meat, fruits and vegetables, bakery products, and prepared consumer foods [1, 2, 3].

Machine learning is at the core of several vision-based inspection systems. But in most applications, the exact set of features that are critical for the quality control is difficult to determine. It is therefore difficult and time-consuming to determine on which features to focus the system's attention when configuring the inspection system in order to sustain a high production rate with a maximum of reliability. One intuitive solution is to include all features that could possibly be relevant and let the learning algorithm decide which features are in fact worthwhile [4]. A more structured way is to identify the relevant features by means of rigorous feature selection techniques and make the inspection system concentrate only over a feature space of a reduced dimension. Such feature selection techniques are often categorized as filters or wrappers. In the filter approach, the feature selector is independent of any learning algorithm and serves as a filter to sieve the irrelevant and/or redundant attributes. The wrapper feature selectors are rather integrated with a learning algorithm to actively determine the relevant attributes for that particular learning algorithm.

The motivation for this research work is to evaluate the effectiveness of some state-of-the-art feature selection techniques for an application on real-time vision-based food inspection systems that operate on several types of bakery products such as hamburger buns, tortillas, and bread loaves. The development of an automated process for the determination of the most relevant features that can also automatically adapt its selection to the type of products being inspected represents a major evolution over the current technology where inspection parameters are mostly set through trial and error procedures. It makes the configuration and maintenance of inspection systems more straightforward, even for new products, while improving the uniformity of the production and reducing the costs of system's configuration.

In the following sections, the mechanisms of four filter-based and wrapper-based feature selectors are introduced. Next the industrial food system used and the samples of food products are detailed. Finally, an experimental evaluation of the feature selection techniques applied on real datasets generated with the existing real-time vision-based food inspection system is conducted and their overall performance is analyzed.

2. Feature Selection Techniques

As mentioned previously, filter and wrapper approaches are examined for the application considered. Three different filter-based techniques are detailed: a correlation-based, a consistency-based, and the RELIEF feature selection methods respectively. A wrapper-based technique is also examined for three target learning schemes. Also, section 2.5 presents the strategy applied for searching the most relevant features among the initial unconstrained set.

2.1. Correlation-based Feature Selection

Correlation-based feature selection (CFS), introduced by Hall [5, 6], evaluates subsets of attributes rather than individual attributes. Hall's rationale for this technique is based on the hypothesis that *"a good feature subset is one that contains features highly correlated with (predictive of) the class, yet uncorrelated with (not predictive of) each other"* [5]. The first part of this hypothesis is inspired by Gennari *et al.* [7] who stated that features are relevant if their values vary systematically with category membership. This statement has been formalized by Kohavi and John [8] who formulated that a feature V_i is said to be relevant for a class C if and only if there exists some v_i and c for which $P(V_i = v_i) > 0$ such that:

$$P\left(C=c \middle| V_i = v_i\right) \neq P\left(C=c\right) \tag{1}$$

Theoretical and empirical evidence encourages removing redundant information along with irrelevant features [8, 9, 10]. A feature is considered redundant if it is highly correlated with one or more other features. CFS uses the following heuristic evaluation to rank feature subsets:

$$Merit_S = \frac{q\overline{r_{cf}}}{\sqrt{q + q\left(q-1\right)\overline{r_{ff}}}} \tag{2}$$

where $Merit_s$ is the heuristic "merit" of a feature subset S containing q features, $\overline{r_{cf}}$ is the average feature-class correlation, and $\overline{r_{ff}}$ is the average feature-feature correlation. The numerator can be interpreted as an indication of how predictive a group of features is. As for q fixed and greater than 1, the feature-class correlation average $\overline{r_{cf}}$ will be relatively large if the group of features is correlated with the class and small otherwise. Therefore the numerator allows discriminating irrelevant features. On the other hand, the denominator discriminates redundant features because in case of redundant attributes (respectively non redundant), the feature-feature correlation average $\overline{r_{ff}}$ will be large (small), which implies a larger (smaller) denominator, and therefore a smaller (larger) $Merit_s$. The correlation between features is computed using symmetrical uncertainty (SU):

$$SU = 2.0 \times \left[\frac{H\left(Y\right) + H\left(X\right) - H\left(X,Y\right)}{H\left(Y\right) + H\left(X\right)}\right] \tag{3}$$

where *H(Y)* is the entropy of a discrete feature *Y* and *H(X, Y)* is the entropy of a discrete feature *X* after observing *Y*. *H(Y)* and *H(X, Y)* are respectively given by:

$$H\left(Y\right) = -\sum_{y\in Y} p\left(y\right)\log_2\left(p\left(y\right)\right) \tag{4}$$

$$H\left(Y \mid X\right) = -\sum_{x\in X} p\left(x\right)\sum_{y\in Y} p\left(y \mid x\right)\log_2\left(p\left(y \mid x\right)\right) \tag{5}$$

In Hall's work, experiments conducted on 36 datasets [5, 6] demonstrated that CFS reduces the number of features by more than half for 70% of the discrete class datasets, while also enhancing accuracy on most of the datasets for various machine learning algorithms. The time complexity of CFS is $O\left(\frac{M\left(N^2 - N\right)}{2}\right)$ for *M* training instances of a dataset containing *N* attributes.

2.2. Consistency-based Feature Selection

In consistency-based feature selection, training instances are projected onto the subset of attributes and then the consistency of the subset is evaluated. It is therefore common practice to use a consistency-based subset evaluator in conjunction with a search algorithm that looks for the smallest subset with consistency. Liu and Setiono proposed an inconsistency evaluation criterion [11]. Two instances are considered inconsistent if they match except for their class labels. The inconsistency criterion is computed as follows: (1) suppose there are r possible class labels, $label_1, label_2, \ldots, label_r$, in a certain dataset which contains N instances; (2) suppose there are J distinct combinations of attribute values for a subset s of attributes (without considering the class labels of the instances); (3) suppose that D_i is the number of occurrences (or matching instances without considering the class labels of the instances) of the i^{th} combination of attribute values; (4) suppose that among the D_i instances, c_1 instances belong to class label $label_1$, c_2 instances belong to $label_2$, …, and c_r instances belong to class label $label_r$, such that $c_1 + c_2 + \ldots + c_r = D_i$, and let $M_i = \max\{ c_1 , c_2 , \ldots , c_r \}$. Then the inconsistency count is given by:

$$inconsistency\ \ count = \left(D_i - M_i\right) \tag{6}$$

In other words, for matching instances (without considering the class labels of the instances) in a subset s of attributes, the more the class labels match, the less inconsistent (or the more consistent) is the subset s with respect to the class. The inconsistency rate of an attribute subset s is given by the sum of all inconsistency counts divided by the total number of instances:

$$inconsistency_s = \frac{\sum_{i=1}^{J} D_i - M_i}{N} \tag{7}$$

Hall's experiments [6] showed consistency-based feature selection to be able to remove up to 92% of the attributes of some datasets while still improving the accuracy of classification. Moreover, the consistency criterion is relatively easy to implement in this framework. On the other hand, several search strategies can be used to look for the smallest most consistent subset of attributes.

2.3. RELIEF Feature Selection

The RELIEF attribute selection technique uses general characteristics of the data to evaluate attributes and operates independently of any learning algorithm. RELIEF was first introduced by Kira and Rendell [12] as a means of estimating the "quality" of attributes with and without dependencies among them. RELIEF is an instance-based attribute ranking scheme that works by randomly sampling an instance from the data and then locating its nearest neighbors from the same and opposite classes. The neighbor from the same class is named *nearest hit* and the one from the opposite class is called *nearest miss*. The values of the attributes of the nearest neighbors are compared to the sampled instance and used to update relevance scores for each attribute. As a matter of fact, the RELIEF's estimate *W[A]* of attribute *A* is an approximation of the following difference of probabilities:

$$\begin{aligned} W[A] = {} & P(\text{different value of A}|\text{nearest instance from different class}) \\ & - P(\text{different value of A}|\text{nearest instance from same class}) \end{aligned} \tag{8}$$

where $P(X|Y)$ is the conditional probability of some event X, given the occurrence of some other event Y. The rationale of the RELIEF algorithm is that useful attributes should differentiate between instances from different classes and have the same value for instances from the same class.

The original version of RELIEF is limited to only two-class problems, which led Kononenko [13] to extend the original RELIEF to deal with noisy, incomplete, and multi-class datasets. The first enhancement that Kononenko addressed was to increase the reliability of probability approximation by searching the k-nearest hits/misses instead of only one near hit/miss, where k is a positive integer such that $k > 1$. The enhanced version, called RELIEF-F, finds nearest neighbors from each class different than the current sampled instance and averages their contribution for updating estimates *W[A]*, and finally weights the average with prior probability of each class as follows [14]:

set all weights *W[A]* := 0.0;
for i := 1 to *m* do
begin
randomly select an instance *R*;
find *k* nearest hits H_j;
for each class $C \neq$ class(*R*) do
find *k* nearest misses $M_j(C)$
for *A* := 1 to NumberOfAttributes do

$$W\left[A\right] := W\left[A\right] - \sum_{j=1}^{k} diff\left(A, R, H_j\right) / \left(m \times k\right) +$$

$$\sum_{C \neq class(R)} \left[\frac{P\left(C\right)}{1 - P\left(class\left(R\right)\right)} \sum_{j=1}^{k} diff\left(A, R, M_j\left(C\right)\right) \right] / \left(m \times k\right)$$

End

where *NumberOfAttributes* is the total number of attributes in the original dataset, and *diff(Attribute, Instance1, Instance2)* computes the difference between the values of *Attribute* for two instances. For discrete attributes, the difference is either 1 (the values are different) or 0 (the values are the same). For continuous attributes the difference is the actual difference normalized to the interval [0, 1]. The same function *diff(.)* is used for calculating the distance between instances to find the nearest neighbors. The expression "continuous attributes" refers here to features that can be measured on a continuum or scale, and can have almost any numeric value, as opposed to discrete data like good or bad, on or off, etc, where the attribute can take a value within a limited set of values. In the algorithm of RELIEF-F above, *m* is a user specified number of instances. Kononenko [13] notes that the higher the value of *m*, the more reliable the estimates of RELIEF-F are, but increasing *m* also increases running time. For the purpose of the experiments presented here, the number of instances sampled *m* is set to 250 and the number of nearest neighbors *k* is set to 10 as suggested by the experiments of Kononenko in [13]. Kononenko's experiments on both artificial and real world data showed RELIEF to be a good candidate for selecting relevant features out of imperfect data.

2.4. Wrapper Feature Selection

A wrapper feature selection approach uses a machine learning algorithm as a black box, therefore needing only the interface of the induction algorithm. In fact, knowledge of the learning algorithm itself is not necessary [8]. The wrapper feature selector repeatedly searches for a "good" feature subset by using the induction algorithm as part of the evaluation function. In [8], Kohavi and John suggest using *five-fold cross validation* accuracy evaluation function for feature selection. This accuracy evaluation function is repeated multiple times. In *t*-fold cross-validation, where *t* is a positive integer, the data is randomly split into *t* mutually exclusive subsets (the folds) of approximately equal size as illustrated in Fig. 1 for *t* = 3.

The learner is trained and tested *t* times, each time with (*t*-1) training folds and one different test fold. For instance for *t=3* as illustrated in Fig. 1, the training set {1, 2, 3} is divided into three training set/test set groups {1, 2}/{3}, {1, 3}/{2}, and {2, 3}/{1}. The induction algorithm is then trained three times with each of the three training sets {1, 2}, {1, 3} and {2, 3}, but only using a subset of the features. The feature subset is proposed by the feature search engine. Each one of the three training sets produces a certain classifier represented by "c" in Fig. 1. Each classifier "c" is tested using the remaining test set: the classifier from training set {1, 2} is tested using test set {3}, the classifier from {1, 3} tested using {2}, and the classifier from {2, 3} tested using {1}. The overall accuracy is then averaged over the *t* folds and if the standard deviation of the accuracy estimate is above 1% and *t* iterations of the cross validations have not been executed; then another cross-validation is run. The accuracy estimation time is the product of the induction algorithm running time and the cross-validation time. A complexity penalty was added to the evaluation function in such a way that the smaller of two feature subsets that have the same estimated accuracy is always picked.

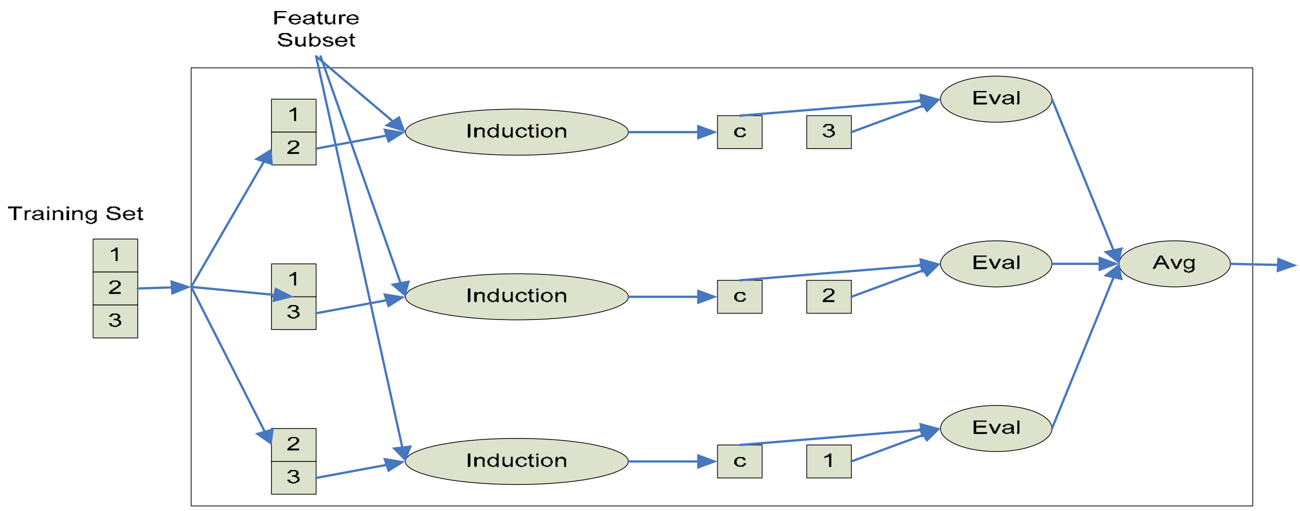


Fig. 1. Example of 3-fold cross-validation.

2.5. Feature Search Engine

Feature selection algorithms employ several search strategies which can be grouped into two main categories: exhaustive search and heuristic search. Each category of search can generally be further broken down into forward selection and backward elimination. Forward selection starts with no variables and adds them one by one, at each step adding the one that decreases the error the most, until any further addition does not significantly decrease the error (or increase it only slightly). On the contrary, backward elimination starts with all the variables and removes them one by one, at each step removing the one that decreases the error the most, until any further removal increases the error significantly.

Exhaustive search explores all possible subsets of M features chosen from N attributes (where typically M ≤ N) in order to theoretically find the optimal solution. However, exhaustive search is generally impractical when the number of attributes in the original dataset is relatively large. As a matter of fact, there exist 2^a possible subsets of *a* attributes. Therefore, in most implementations of exhaustive search algorithms, at some operator defined stopping point, the subset of features with the highest score discovered up to that point is selected as the satisfactory feature subset. Instead of exploring all possible subsets of features, heuristic search algorithms on the other hand ignore whether the solution to the problem can be proven to be correct, but usually produce a good solution or solve a simpler problem that contains, or intersects with, the solution of the more complex problem. Heuristics are typically used when there is no known way of finding an optimal solution, or when it is desirable to give up finding the optimal solution for an improvement in run time.

A heuristic search algorithm, known as *best first,* was used with the CFS, the Consistency-based subset evaluation, and the wrapper techniques presented here. Best-first search is a search algorithm which explores a graph by expanding the most promising node chosen according to some heuristic evaluation rule. Kohavi and John [8] conducted an evaluative comparison between hill climbing and best first search algorithms and concluded that the latter was generally a more thorough technique. The principle of best-first search heuristic optimization is described as follows:

1. Begin with the OPEN list containing the start state, the CLOSED list empty, and BEST $\leftarrow$ start state.
2. Let subset $s = \arg\max e(x)$ (get the state from OPEN with the highest evaluation).
3. Remove s from OPEN and add to CLOSED.
4. If $e(s) \geq e(\text{BEST})$; then BEST $\leftarrow s$.
5. For each child t of s that is not in the OPEN or CLOSED list, evaluate and add to OPEN.
6. If BEST changed in the last p expansions, goto 2.
7. Return BEST.

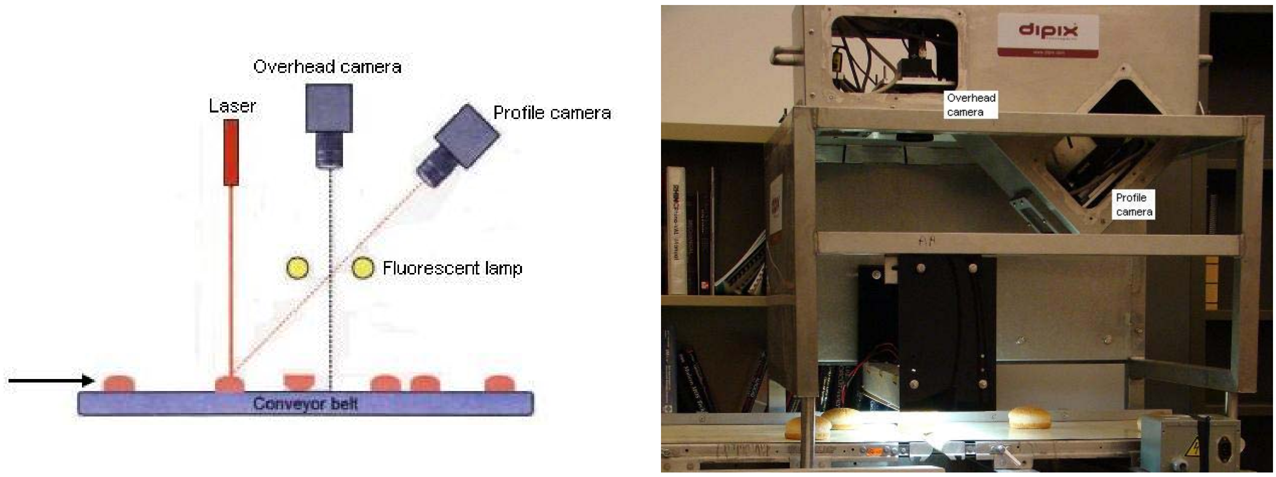
In the algorithm above, the function e(.) is the heuristic evaluation of the feature subset. The heuristic evaluation used depends on the goal of the application. Nonetheless, the idea is always to select the subset which maximizes the heuristic evaluation function. For instance, in the case of CFS, this heuristic evaluation function is computed using Equation (2). In the case of consistency-based feature selection, the heuristic evaluation is $(1 - inconsistency_S)$ where $inconsistency_S$ is given by Equation (7). In the case of feature wrappers, the heuristic evaluation used is the actual estimated accuracy of the induced classifier. In all our implementations, the number of expansions p for the best first search algorithm was set to five.

3. Food Inspection System and Product Datasets

3.1. Industrial Vision-Based Food Inspection Setup

The vision-based food inspection system used for our experimentation is a technology developed and manufactured by Dipix Technologies Inc. that automates visual identification and classification of bakery products such as buns, cookies, tortillas, and pizzas. Fig. 2a presents a conceptual view of the system. The inspection station is equipped with a conveyor belt which moves the bakery products from the output of the oven to the rejection and packaging systems in an industrial setting. One camera is mounted above the conveyor belt and produces real-time line scans of the top view of products while they move with the conveyor belt. Two fluorescent lamps continuously illuminate the field of view of the overhead camera to ensure uniform lighting across the entire width of the conveyor. A laser light strip is also projected vertically on the conveyor belt and a second camera equipped with an optical filter fitted to the laser wavelength is mounted in diagonal to collect an image that reveals the profile information on every single product. Fig. 2b shows the actual system used for our experiments.

The profile data captured by the diagonal camera is marked with a time stamp which allows real-time image processing algorithms to combine the height information with the overhead data collected by the line scan camera for every single product that passes under the inspection head, assuming constant speed of the conveyor belt. From the resulting tridimensional and color model of every product under inspection, a large number of parametric features characterizing the items under inspection are estimated. Depending on the product, up to 200 features can be extracted. Examples of such features include color information, topping coverage, heights, diameters, slopes, lengths, surface area, circularity, and volume. The system analyzes the features of every product and then orders rejection of the product if it is classified as defective; or orders acceptation if the product is judged acceptable, that is within the preset standards.



a) b)

Fig. 2. Food inspection station: a) conceptual structure, and b) actual experimental system.

For the purpose of our experimentation, new modules were developed and embedded into the existing visual food inspection system to record on disk all extracted features on every item and for each category of products tested. When available the original system's decision ("accept" or "reject") was also recorded for every item to provide a comparison basis against which the classification with a reduced set of features can be monitored to evaluate the performance of the proposed feature reduction techniques [15].

3.2. Product Datasets

The bakery products over which experimentation was conducted belong to two categories. The first one corresponds to products (called "known" products) for which the system already had a classification rule defined and implemented. In that case, both the raw feature measurements and the decision of the system were recorded. The second category contains products (called "unknown" products) for which the system was not set to permit the correct classification of the products under inspection. These mainly correspond to a new type of products to inspect for which the system was never used. In this situation, an operator was asked to present the samples in two successive rounds. During the first round, the operator passes a set of product samples considered as "acceptable" to a typical customer on the conveyor belt. During the second round, sample products considered as "unacceptable" are passed on the conveyor belt. In both rounds, the inspection system extracts all the features from all the presented samples and tags each item as "acceptable" or "unacceptable" respectively.

For the category of "known" products, experiments were conducted on two bakery products that have been inspected for several years by the vision-based food inspection system available: burger buns and tortillas as illustrated in Fig. 3. Seeded buns were selected as they contain more features and more complexity than regular unseeded ones. Buns and tortillas both have "irregular" shapes as none of them consistently exhibits a perfectly defined geometrical shape. For both the buns and the tortillas datasets, 82 continuous features are extracted per product item, plus one Boolean feature representing the decision to reject or accept the item. These datasets were collected on two different industrial production lines and each contains 3287 product items.

a) b)

Fig. 3. Sample of products with known inspection presets: a) seeded burger buns, b) tortillas.

For the category of "unknown" products, a substantially different type of bun was selected to ensure a complete decoupling from the pre-tuned inspection machine. Ciabatta breads with a triangular shape and a non-uniform height, seen in Fig. 4a, were selected as they were never processed by the inspection system used. This extra dataset was generated in the laboratory using the machine shown in Fig. 2b. Because there was no predefined classifier available in this case, the samples were subjectively divided into an "acceptable" and an "unacceptable" batches. If not already apparent, defects were manually created on the ciabatta buns belonging to the "unacceptable" batch. Defects can take the form of holes, colors corresponding to overbaked buns, packing plastic pieces that made their way onto the buns, broken buns or weirdly shaped buns (not triangular), as shown in Fig. 4b. In this case, 160 samples containing approximately 50% "acceptable" and 50% "unacceptable" ciabatta buns were inspected. The same 82 features were extracted per sample as with the seeded buns and tortillas, along with the predefined Boolean class attribute ("accept" or "reject").

a) b)

Fig. 4. Sample of products without known inspection presets: a) acceptable products, b) unacceptable products.

Prior to feature selection, the data extracted from those samples are randomized. Randomizing data, that is changing their order of presentation in the datasets, is particularly important in the case where the inspected products were inspected in two rounds ("acceptable" then "unacceptable" products), because in a real production line the defects usually come in an unordered manner. Therefore, this factor should not impact on the analysis of feature selection and dimensionality reduction. The recorded continuous valued features are also discretized using a supervised discretization technique introduced by Fayyad and Irani [16] which combines an entropy-based splitting criterion with a minimum description length stopping criterion [14].

4. Experimental Evaluation

4.1. Dimensionality Reduction on Samples of Different Products

The features extracted by the inspection system along with the related Boolean decision are successively used as input to the different feature selection approaches described in section 2. Table 1 and Table 2 show the number of features selected for tortillas and seeded buns respectively, sorted in ascending order of the number of features selected by the different dimensionality reduction techniques. The feature wrappers are trained with three target learning algorithms: Naïve Bayes, C4.5 decision tree and Multi-Layer Perceptron (MLP). Those three learning schemes are summarized in Section 4.2.

For the tortillas dataset, the C4.5 wrapper algorithm offers maximum reduction of the feature space dimension by selecting only 4 features out of 82, followed equally by the CFS feature selector and the MLP wrapper that both select 5 features. The RELIEF algorithm is last on the list as it keeps up to 33 attributes out of 82. For the buns dataset, the C4.5 wrapper also provides maximum reduction with 2 features selected, followed by the consistency-based subset evaluation (5 out of 82). The Naïve Bayes wrapper and the MLP wrapper both found 6 attributes to be relevant to their respective learning algorithms. RELIEF is again the one rejecting the least attributes.

Table 1. Feature space reduction performance on the tortillas dataset.

	Number of features selected	Number of features rejected	Feature rejection ratio (%)
C4.5 wrapper	4	78	95.12
CFS	5	77	93.90
MLP wrapper	5	77	93.90
NB wrapper	7	75	91.46
Consistency	10	72	87.80
RELIEF	33	49	59.76

Table 2. Feature space reduction performance on the seeded buns dataset.

	Number of features selected	Number of features rejected	Feature rejection ratio (%)
C4.5 wrapper	2	80	97.56
Consistency	5	77	93.90
MLP wrapper	6	76	92.68
NB wrapper	6	76	92.68
CFS	7	75	91.46
RELIEF	59	23	28.04

For the buns and the tortillas datasets, the C4.5 wrapper, the MLP wrapper, the Naïve Bayesian wrapper, CFS and the consistency-based subset evaluation tend to consider 10 or less features out of 82 as relevant to qualify the product, whereas RELIEF seems to be cautious by keeping many more features. The fact that only positive differences of probability were kept in the RELIEF implementation implies that weakly relevant features are very likely to be preserved. Kohavi and John also mentioned that the RELIEF algorithm tends to keep most of the relevant features of a dataset even if they are redundant and if only a fraction of them is necessary for the concept description [8]. Moreover, wrapper feature selection techniques globally tend to select fewer features than filter-based feature selectors. This could be explained by the fact that wrappers are meant to optimize the feature selection for a particular given algorithm with which they interact during the attribute selection process. One interesting point is the fact that all feature subsets selected by any of the feature selectors, except RELIEF, are all also selected by RELIEF for the seeded buns dataset.

When considering the “unknown” category of products for which the classification of the samples is done subjectively by a human operator before extracting the features using the vision-based food inspection system, the same information is presented to the different feature selection approaches. Table 3 shows the number of features selected for the triangular ciabatta buns, sorted in ascending order of the number of features selected.

Table 3. Feature space reduction performance on the ciabatta buns dataset.

	Number of features selected	Number of features rejected	Feature rejection ratio (%)
C4.5 wrapper	4	78	95.12
NB wrapper	4	78	95.12
MLP wrapper	9	73	89.02
Consistency	9	73	89.02
CFS	14	68	82.93
RELIEF	36	46	56.10

As with the other types of products, the feature wrappers keep the least attributes: C4.5 wrapper and Naïve Bayes wrapper share the first rank by selecting only 4.88% of the features (4 out of 82); followed by the MLP wrapper which selects 9 attributes (10.98%). The consistency-based subset selector also selects 9 features out of 82. RELIEF ends up again the one removing the least attributes by keeping 36 features. RELIEF is preceded by CFS which judges 14 features (17.07%) to be relevant and non redundant. All the features selected by the different techniques are either also selected by RELIEF, or have a maximum of two features which are not selected by RELIEF. Except RELIEF, all the attribute subset selectors kept less than 15 features (18.29% of all features) for all the experimented datasets.

These experiments with three datasets of products with different characteristics point out that feature wrappers generally achieve better results than filters due to the fact that they are tuned to the specific interaction between an induction algorithm and its training data. However, they tend to be much slower than feature filters because they must interact with an induction algorithm, as will be detailed in the following section.

4.2. Classification Accuracy with Reduced Feature Space

In spite of the observations made in the previous section, the number of attributes selected as relevant by the different feature selectors should be interpreted with caution. For artificial datasets, it is pretty straightforward to evaluate the performance of a feature selection algorithm. However, for real world datasets such as the ones reported in this work, it is not necessarily clear what the relevant features are. Therefore, whether the selected features are relevant or not can only be determined indirectly, by observing the effects of feature space dimensionality reduction over the performance of a learning algorithm. In fact, although a dataset with fewer features could be preferred for production rate enhancement, the accuracy of prediction with the reduced datasets remains of capital importance for proper classification of the products as acceptable or not to the customer.

In order to better guide the compromise that must be found between prediction accuracy and dimension reduction of the dataset for industrial quality inspection applications, the study was extended to involve three different machine learning techniques: Naïve Bayes (a probabilistic learner), C4.5 (a decision tree learner) and Multi-Layer Perceptron (MLP, a neural networks learner). The Naïve Bayes algorithm assumes that features are conditionally independent given their label and computes the posterior probability of each class given the feature values present in the class, and assigns the instance to the class with the highest probability [17][18]. C4.5 is an algorithm which builds a decision tree top-down by recursively finding the best single feature test to conduct at the root node of the tree [19]. The MLP is a hierarchical structure of several perceptrons with weighted interconnections able to capture complex input/output relationships from training data [20]. For wrapper-based feature selectors an interaction with the learning algorithm is already required during feature selection.

The holdout and cross-validation methods [21] are considered to evaluate the accuracy of the prediction (as “accept” or “reject”) on the class of the sample products. The accuracy represents the percentage of products that were classified from the reduced set of features in the exact same way as the original classification of the system, without feature reduction. The holdout method, also called test sample estimation, separates the data into two mutually exclusive subsets: the training set and the test set, or holdout set. As the name of the sets indicates, the training set is used for the training phase, and the test set is used later to evaluate performance. In *t*-fold cross-validation, the data is randomly split into *t* mutually exclusive subsets (the folds) of approximately equal size as explained in section 2.4. For the present evaluation, 10-fold cross-validation was used for model selection as suggested by Kohavi in [21], and the cross-validation was repeated 10 times with different random seeds. Holdout was repeated 100 times on the datasets and the overall accuracy was averaged.

Note that Kohavi and John [8] suggest only five fold cross validation for feature reduction purposes where the interest is to identify relevant and non redundant features. Their experiments showed that five-fold cross validation gives satisfactory results. The other reason is the fact that during feature selection, cross-validation is applied to every feature subset proposed by the feature search algorithm until a satisfactory subset is found, with respect to the defined stopping criterion; therefore having a relatively small number of folds accelerates the process without sacrificing a lot of quality. On the other hand, during model selection where the focus is more to find a reliable classifier rather than to find a subset of relevant and non redundant features, the cross-validation is done only on one feature-reduced subset (or the full set of features if no dimensionality reduction has been accomplished). In this later case, increasing the number of folds to 10 in the *t*-fold cross validation only moderately impacts running time.

It is worth mentioning that the continuous features generated by the inspection system are discretized only for feature selection purposes. After feature selection, the reduced datasets are extracted from the original continuous datasets and then passed to the learning algorithms for accuracy estimation. The same train/test sets and the same folds were used for all learning schemes in order to establish a common base for comparison.

Fig. 5 and Fig. 6 show the accuracy estimation on the class prediction with the different feature selection techniques for the tortillas dataset, evaluated using the three learning schemes. On both figures “Tortilla” represents the original full dataset which did not undergo any feature selection and the vertical bars at the edge of the columns represent the standard deviation. Fig. 5 shows the result of 10 repetitions of the 10-fold cross-validation, and Fig. 6 presents the result of 100 repetitions of the holdout accuracy test.

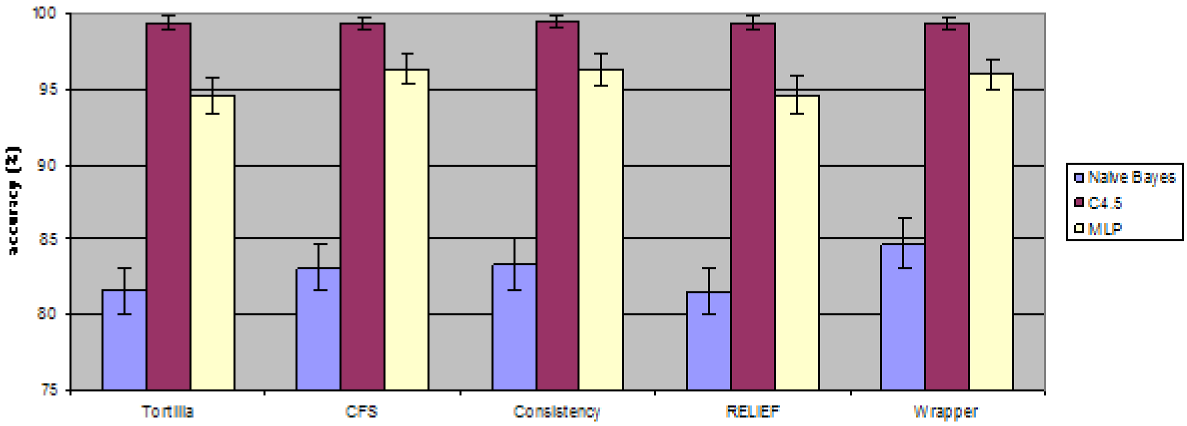


Fig. 5. Ten repetitions of the 10-fold cross-validation accuracy estimation for the tortillas dataset.

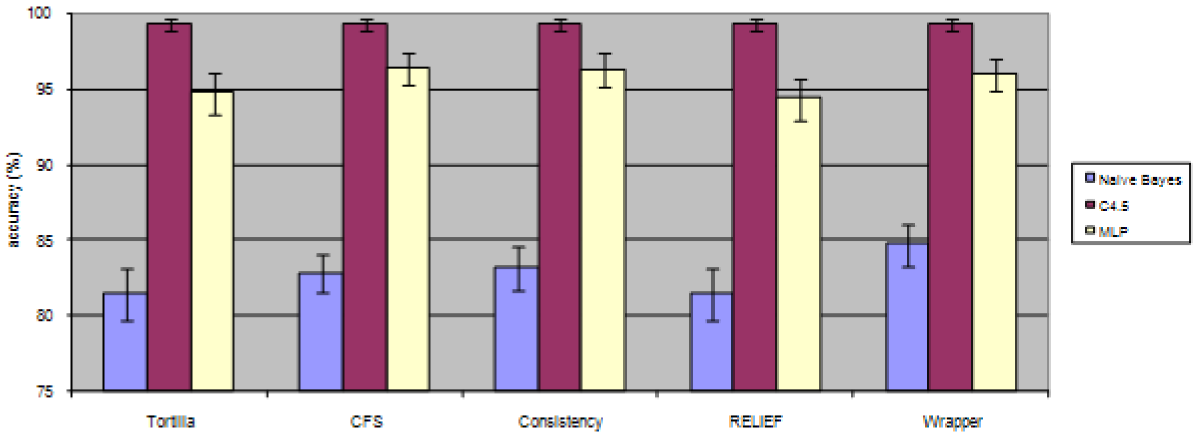


Fig. 6. One hundred repetitions of the holdout accuracy estimation for the tortillas dataset.

For all feature selectors, the C4.5 learning scheme globally gives the highest accuracy and the lowest standard deviation at the same time, followed closely by the MLP learner and far beyond by the Naïve Bayes learner. Holdout and cross-validation gave comparable results in terms of accuracy and standard deviation. It is important to note that in the experiments presented, both cross-validation and holdout use 90% of the data for training and 10% for test. For 10-fold cross-validation, the full original tortillas dataset had an accuracy of 99.39% with a standard deviation of 0.47% using the C4.5 learning scheme. Consistency-based subset evaluation gave a slightly better accuracy (99.44%) than the full dataset. The RELIEF feature selector is the second best by having the same accuracy estimation as the original dataset, followed by the C4.5 wrapper and the correlation-based feature selector with an accuracy of respectively 0.03% and 0.04% below that of the full dataset. The holdout tests gave approximately the same order by ranking consistency and RELIEF equally accurate to the original full dataset (accuracy of 99.37%), followed by CFS and C4.5 wrapper achieving an equal accuracy of 0.01% inferior to that of the original dataset. The standard deviation for all feature-reduced datasets are all between 0.39% and 0.47% and are considered not high enough to impact the interpretation of the results.

Fig. 7 shows the results of 10 repetitions of 10-fold cross-validation for the seeded buns dataset. The holdout test results are not presented here as they are similar to the cross-validation results. C4.5 is once again the learning scheme giving globally the best accuracy estimation, followed by the MLP and the Naïve Bayes respectively. The original full buns dataset has an accuracy of approximately 99.81%

with a standard deviation of 0.25% and none of the reduced dataset is able to achieve a better accuracy. However, the consistency-based subset evaluation and the C4.5 wrapper closely follow the original full buns dataset by both achieving an estimated accuracy only 0.58% inferior to that of the full dataset. RELIEF occupies the third place and CFS the fourth with accuracy less than 1% lower than that of the full buns dataset.

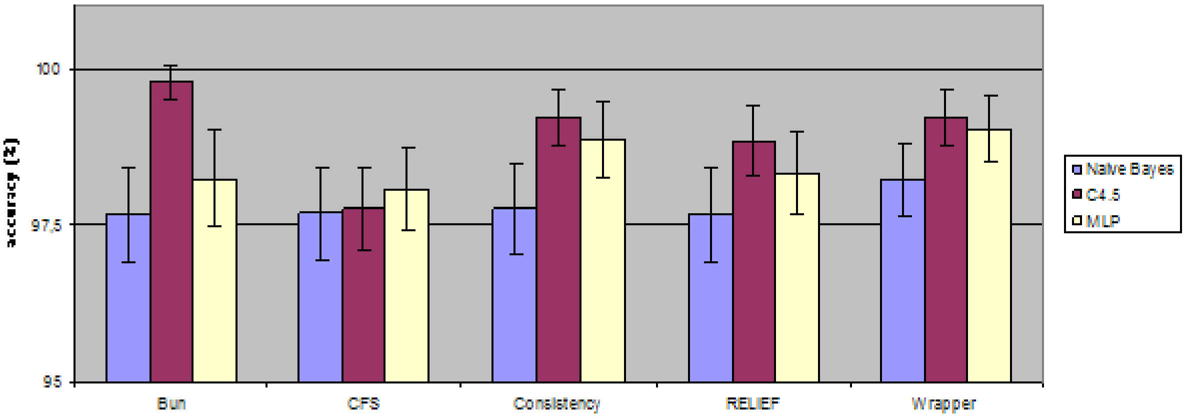


Fig. 7. Ten repetitions of the10-fold cross-validation accuracy estimation for the buns dataset.

The fact that the Naïve Bayes classifier gives lower accuracy estimations compared to the C4.5 and the Multi-Layer Perceptron can be attributed to the assumption that the algorithm makes about features being conditionally independent. In fact, several of the features in the dataset are correlated, for example features such as the mean diameter of an approximately circular product and its surface area are clearly correlated. MLP was able to capture a certain rule for the classification of the products considered because of the structure inherent to the MLP which allows capturing complex input/output relationships. C4.5 giving very high classification accuracy can be explained by the fact that the vision-based food inspection system inherently uses a decision structure very close to a decision tree.

For the “unknown” products, Fig. 8 and Fig. 9 respectively show the results of 10 repetitions of 10-fold cross-validation accuracy estimation and the results of 100 repetitions of holdout accuracy estimation for the ciabatta buns. Unlike the “known” products where C4.5 seems to generally perform better than the MLP and the Naïve Bayes learning schemes, results on “unknown” products do not unanimously exhibit the best learning scheme independently of the used feature selection. Nevertheless, both cross-validation and holdout tests results show that the accuracy of almost all the reduced datasets is greater or equal to the accuracy obtained with the full dataset.

For all three learning algorithms, the ciabatta buns dataset that was reduced with a wrapper feature selection gave the greatest cross-validation accuracy estimation: 86% for the Naïve Bayes wrapper, 86.63% for the C4.5 wrapper and 83.38% for the MLP wrapper, which correspond respectively to 9.33%, 12.57% and 4.19% estimated accuracy enhancements compared to that of the full dataset. This conclusion also holds for the holdout accuracy estimation with 8.93% enhancement using the Naïve Bayes wrapper, 5.71% for the C4.5 wrapper and 2.82% for the MLP wrapper. For both the C4.5 and the MLP learning schemes, the consistency-based attribute subset filter occupies the second rank with respect to the estimated accuracy from a reduced feature space, followed by CFS. For the Naïve Bayes Learner, CFS is the second best choice and the consistency-based filter the third choice. Recalling that Naïve Bayes assumes conditional independence between attributes, one could argue that the removal of correlated attributes by CFS improves the performance of Naïve Bayes. RELIEF is the last choice for all three candidate machine learning schemes. However, it is noticeable in Fig. 8 and Fig. 9 that even though RELIEF performs the least in terms of estimated classification accuracy, the feature subset selected by RELIEF still provides better accuracy than when all the features are considered, for a product for which the inspection system is not finely tuned.

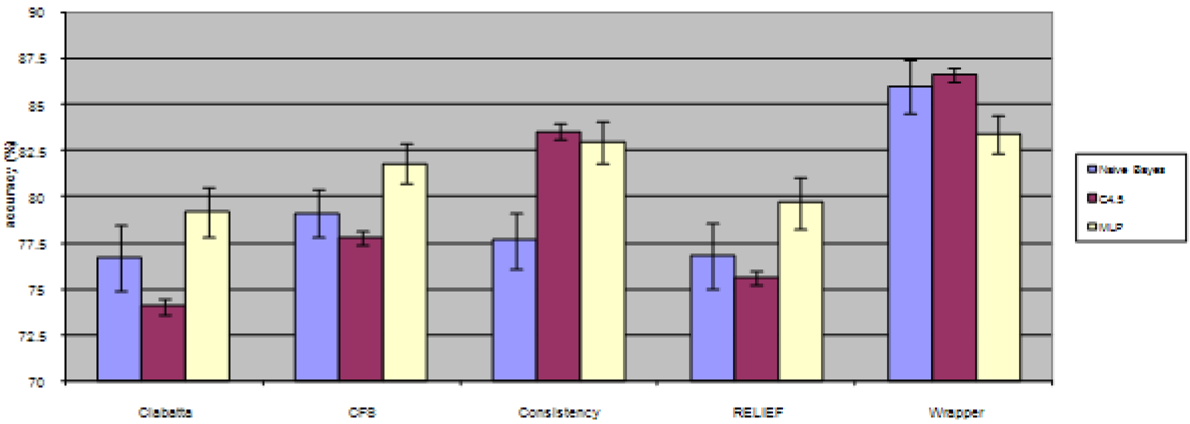


Fig. 8. Ten repetitions of the10-fold cross-validation accuracy estimation for the ciabatta buns dataset.

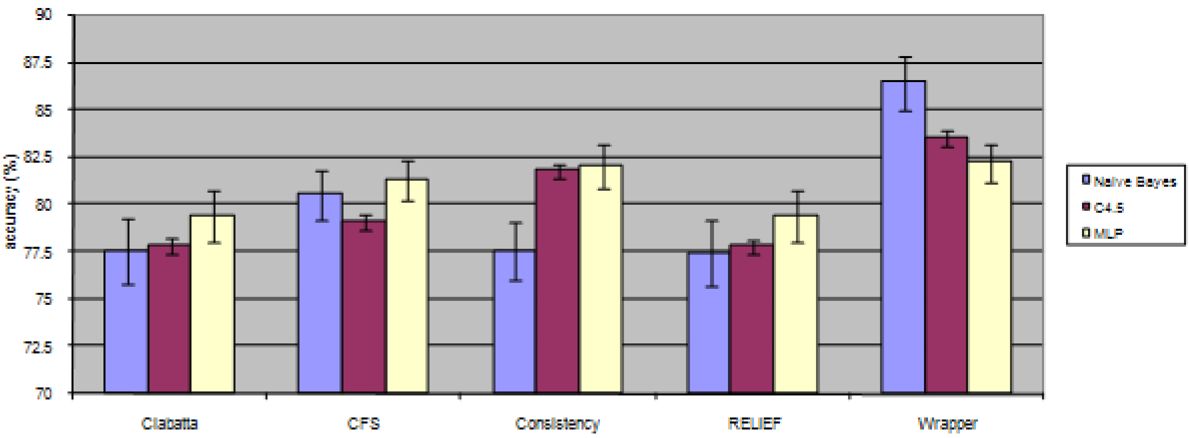


Fig. 9. One hundred repetitions of the holdout accuracy estimation for the ciabatta buns dataset.

Globally, the advantage of a consistency-based subset evaluation is undisputed accuracy-wise for both the buns and the tortillas datasets. Considering the number of attributes retained by the different feature selectors, consistency-based subset evaluation acquires another advantage over RELIEF by proposing only 10 attributes versus 33 for the tortillas dataset and 5 versus 59 for the seeded buns dataset. One might explain the success of consistency-based subset evaluation and RELIEF techniques by their ability to capture attribute interactions. CFS also gives reasonably good results, especially when the relatively small number of features selected by this filter is considered. In the case considered here, several features in the dataset are correlated and CFS appears to be able to identify these correlations. The C4.5 wrapper has a net advantage with respect to the number of features selected. Wrappers also tend to give better classification results than filters in general. But this benefit is compromised by the time it takes to train wrappers, which can reach several minutes rather than only a few seconds with filter-based selectors, due to the repetitive interaction with the learning schemes.

For the ciabatta buns, the Naïve Bayes wrapper, C4.5 wrapper and MLP wrapper outperformed all the other feature selectors. The fact that the reference training samples have been subjectively classified by a human operator reduces the chances of ciabatta buns dataset's classifier to be confidently represented by a single structure independently of the applied feature selector. Consequently, the performance of the feature selectors depended more on the target learning scheme, which explains the exceptional performance of feature wrappers for this category of products. Consistency-based subset evaluation would be the second best candidate, followed by CFS and RELIEF. The order of performance of all the evaluated feature selectors for the ciabatta buns with respect to estimated classification accuracy actually coincides with the order of feature space reduction presented in Table 3. That is the feature selector retaining the lower number of attributes is also the one achieving the best classification accuracy estimates. This provides a good direction for actual implementation of optimized classification systems that can adapt automatically to new products.

According to these experiments, feature wrappers and consistency-based subset evaluation appear to be the best suited feature selector for application on a real-time vision-based food inspection system, provided that the products under classification are similar to the ones analyzed here. Feature wrappers unfortunately come at the expense of longer training time. RELIEF is the next best candidate on two third of the experimented datasets in terms of classification accuracy and training time, but tends to keep a lot more features than needed. In fact, not only does consistency-based subset evaluation on two out of three datasets outperform all the other feature selectors' accuracy for the C4.5 algorithm, but it also generally gives a better accuracy for all the learning schemes we experimented with and for almost all the datasets. This work demonstrated that instead of analyzing and interpreting 82 features for every product under classification in-real time, the vision-based food inspection system can now focus on less than 15 features and produce classification accuracy similar or even better than with the full 82 features. Reducing the number of features extracted and analyzed in real-time reduces the time taken for processing a single product and therefore allows the food inspection system to support higher production rates. It is however necessary to emphasize that having training data that cover as many cases as possible is a definite key in helping the algorithms generalize. Therefore particular caution in choosing the data samples shall be applied to prevent poor generalization of the learner.

5. Conclusions

This paper investigated and evaluated the application of four feature selection techniques for parametric space reduction in a real-time vision-based food inspection system. Three machine learning algorithms were used to evaluate and compare the accuracy of the classification from the extracted features with and without feature selection. Experimental results on seeded buns and tortillas demonstrated that consistency-based subset evaluation outperforms all other feature selectors in terms of classification accuracy, and is also very competitive in terms of the number of selected attributes. Experiments on another dataset composed of products manually classified as acceptable or defective also placed consistency-based subset evaluation as the second candidate after feature wrappers. Wrapper approaches tend to give excellent classification accuracy results, especially with the C4.5 decision tree inducer, but take a longer time to train. The RELIEF technique also revealed good performance, but has the disadvantage of keeping more features than the other selectors, which might impede production rates in an industrial setting. CFS performed better than RELIEF on one third of the experimented datasets.

Most of the feature-reduced datasets provided by the attribute selectors gave an estimated classification accuracy very close to the accuracy achieved with the full datasets, and even higher when extracted with either the consistency-based subset evaluation technique or the feature wrappers with a target learning algorithm. Apart from the RELIEF algorithm, all feature selectors reduced the feature space dimensionality by more than 80%. This evaluation with realistic datasets extracted from bakery products demonstrates the relevance of integrating feature selectors into the vision-based food inspection system. With this solution, the system can readily focus on fewer features that get automatically identified according to the characteristics of the product being inspected. Moreover, it still provides inspection decisions and classification of a comparable reliability as with the full set of measured features while allowing for a higher rate of production.

Acknowledgements

The authors acknowledge the partial financial support of Precarn Inc., and the collaboration of Dipix Technologies Inc. to the first phase of this research.

References

[1]. M. R. Chandraratne, D. Kulasiri, S. Samarasinghe, Classification of Lamb Carcass Using Machine Vision: Comparison of Statistical and Neural Network Analyses, *Journal of Food Engineering*, Vol. 82, No. 1, 2007, pp. 26-34.

[2]. J. Blasco, N. Aleixos, E. Molto, Computer Vision Detection of Peel Defects in Citrus by Means of a Region Oriented Segmentation Algorithm, *Journal of Food Engineering*, Vol. 81, No. 3, 2007, pp. 535-543.

[3]. T. Brosnan, D.-W. Sun, Improving Quality Inspection of Food Products by Computer Vision – a Review, *Journal of Food Engineering*, Vol. 61, 2004, pp. 3-16.

[4]. H. Almuallim, T. G. Dietterich, Learning with Many Irrelevant Features, in *Proc. of the 9th National Conf. on Artificial Intelligence, 1991,* pp. 547-552.

[5]. M. A. Hall, Correlation-Based Feature Selection for Machine Learning, *PhD Thesis,* Department of Computer Science, University of Waikato, Hamilton, New Zealand, 1998.

[6]. M. A. Hall, Correlation-Based Feature Selection for Discrete and Numeric Class Machine Learning, *Proc. of the 17th Intl Conf. on Machine Learning,* 2000, pp. 359-366.

[7]. J. H. Gennari, P. Langley, D. Fisher, Models of Incremental Concept Formation, *Artificial Intelligence,* Vol. 40, No. 4, 1999, pp. 11-61.

[8]. R. Kohavi, G. H. John, Wrappers for Feature Subset Selection, *Artificial Intelligence,* Vol. 97, 1997, pp. 273-324.

[9]. P. Langley, S. Sage, Induction of Selective Bayesian Classifiers, *Proc. of the 10th Conf. on Uncertainty in Artificial Intelligence,* Seattle, W.A, Morgan Kaufmann Publishers, 1994, pp. 399-406.

[10]. R. Kohavi, D. Sommerfield, Feature Subset Selection Using the Wrapper Method: Overfitting and Dynamic Search Space Topology, *Proc. of the 1st Intl Conf. on Knowledge Discovery and Data Mining,* 1995, pp. 192-197.

[11]. H. Liu, R. Setiono, A Probabilistic Approach to Feature Selection: a Filter Solution, in *Proc. of the 13th Intl Conf. on Machine Learning, 1996,* pp. 319-327.

[12]. K. Kira, L. Rendell, A Practical Approach to Feature Selection, *Proc. of the 9th Intl Conf. on Machine Learning,* 1992, pp. 249-256.

[13]. I. Kononenko, Estimating Attributes: Analysis and Extensions of RELIEF, in *Proc. of the 7th European Conference on Machine Learning,* 1994, pp. 171-182.

[14]. M. A. Hall, G. Holmes, Benchmarking Attribute Selection Techniques for Discrete Class Data Mining, *IEEE Transactions on Knowledge and Data Engineering*, Vol. 15, No. 6, 2003, pp. 1437-1447.

[15]. I. H. Witten, E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques*, 2nd Edition, Morgan Kaufmann, San Francisco, 2005.

[16]. U. M. Fayyad, K. B. Irani, Multi—Interval Discretization of Continuous-Valued Attributes for Classification Learning, in *Proc. of the 13th Intl Joint Conf. on Artificial Intelligence*, 1993, pp. 1022-1027.

[17]. J. R. Anderson, M. Matessa, Explorations of an Incremental, Bayesian Algorithm for Categorization, *Machine Learning*, Vol. 9, 1992, pp. 275-308.

[18]. P. Domingos, M. Pazzani, Beyond Independence: Conditions for the Optimality of the Simple Bayesian Classifier, in *Proc. of the 13th Intl Conf. on Machine Learning*, Bari, Italy, 1996, pp. 105-112.

[19]. J. R. Quinlan, *C4.5: Programs for Machine Learning*, San Mateo, California: Morgan Kaufmann Publishers, 1993.

[20]. S. Haykins, Neural Networks: a Comprehensive Foundation, Upper Saddle River, N. J., *Prentice Hall*, 1999.
[21]. R. Kohavi, A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection, in *Proc. of the Intl Joint Conf. on Artificial Intelligence*, Montreal, QC, 1995, pp. 1137-1143.

Guide for Contributors

Aims and Scope

Sensors & Transducers Journal (ISSN 1726-5479) provides an advanced forum for the science and technology of physical, chemical sensors and biosensors. It publishes state-of-the-art reviews, regular research and application specific papers, short notes, letters to Editor and sensors related books reviews as well as academic, practical and commercial information of interest to its readership. Because it is an open access, peer review international journal, papers rapidly published in *Sensors & Transducers Journal* will receive a very high publicity. The journal is published monthly as twelve issues per annual by International Frequency Association (IFSA). In additional, some special sponsored and conference issues published annually.

Topics Covered

Contributions are invited on all aspects of research, development and application of the science and technology of sensors, transducers and sensor instrumentations. Topics include, but are not restricted to:

- Physical, chemical and biosensors;
- Digital, frequency, period, duty-cycle, time interval, PWM, pulse number output sensors and transducers;
- Theory, principles, effects, design, standardization and modeling;
- Smart sensors and systems;
- Sensor instrumentation;
- Virtual instruments;
- Sensors interfaces, buses and networks;
- Signal processing;
- Frequency (period, duty-cycle)-to-digital converters, ADC;
- Technologies and materials;
- Nanosensors;
- Microsystems;
- Applications.

Submission of papers

Articles should be written in English. Authors are invited to submit by e-mail editor@sensorsportal.com 6-14 pages article (including abstract, illustrations (color or grayscale), photos and references) in both: MS Word (doc) and Acrobat (pdf) formats. Detailed preparation instructions, paper example and template of manuscript are available from the journal's webpage: http://www.sensorsportal.com/HTML/DIGEST/Submition.htm Authors must follow the instructions strictly when submitting their manuscripts.

Advertising Information