Feature Selection in Data Mining

classical and new techniques

Guido Sciavicco

Università degli Studi di Ferrara

11 Novembre 2015

in collaboration with dr. Enrico Marzano, CIO Gap srl
Active Contact System Project

Contents

Features - 1

- Both *supervised* and *unsupervised* knowledge discovery processes share the same basis.

A_1	A_2	...	L_1?
a_{11}	a_{12}	...	L_1?
a_{21}	a_{22}	...	L_2?
a_{31}	a_{32}	...	L_3?

Features - 2

- The information we are searching for is hidden in a table (view) in such a way that we can ask ourselves:
  1. Given all *instances* with a certain label, what are the rules that govern the label value in terms of the attributes (or features) values? - *supervised classification*
  2. Given my *unlabeled* instances, is it possible to identify *clusters* to which all instances belong? - *unsupervised classification*
  3. Given my *unlabeled* instances, and given the result of a clustering process, is it possible to link the clusters to the the attributes via a set of rules? - *unsupervised* to *supervised* classification
- All such questions, as they are posed, implicitly involve *all attributes*, whether they are meaningful or not.

Features - 3

- A typical classification problems includes hundreds to millions instances, each described by tens to hundreds of attributes.
- Attributes are generically collected *brute force*: as I do not know not only the rules I am searching for, but even if such a rule exists, I choose to consider as many characteristics of each instance I can possibly think of.
- Having non meaningful attributes contributing to a classification process produces a *slow*, *difficult to interpret*, and sometimes even *wrong* result.
- And, given N attributes, the search space in which I have to select my subset of meaningful attributes is 2^N, which makes it very difficult, if not impossible, to perform a manual selection.

Features - 4

- It is important to stress that it does not matter what algorithm we want to use to perform a classification or a clustering.
- Every algorithm will perform better is if it run with:
  1. *less* features;
  2. *more correlated* features;
  3. features that *more clearly* influence the label.
- There are at least three categories of Feature Selection algorithms:
  1. *Filter*-based;
  2. *Wrapper*-based;
  3. *Embedded*.

Filters

- A *filter* is an algorithm that can be applied to a data set in order to select only a subset of the features, with the following characteristics:
  1. A filter is *independent* from the successive clustering/classification step;
  2. A filter can be *supervised* or *unsupervised* - it follows the same nomenclature that we used before: supervised filters can be applied in presence of the label, and unsupervised filters must be used in the other case.
- Filters not only improve the performance and the result of the successive classification/clustering, but they can and must be used to perform a more fundamental *pre-processing* of the original data set.
- Pre-processing our data set is necessary, and, at some stage, this includes a feature selection step.

Unsupervised Filters: An Example

- One of the most fundamental unsupervised filter consists of eliminating features with zero or near to zero *variance*. For each feature that is considered, it is eliminated if:
  1. It presents a single possible value (zero variance), or
  2. It has very few unique values relative to the number of samples, and the ratio of the frequency of the most common value to the frequency of the second most common value is large (near to zero variance).
- So, for example, if we have 1000 instances with an attribute name *A*, which presents the value *a* 999 times, and *b* just once, *A* is eliminated by the filter; the result is that my original data set is then *projected* over the remaining attributes.

Supervised Filters: An Example - 1

- Another class of filter technique is called *feature weighting*.
- A feature weighting algorithm assigns weights to features (individually) and rank them based on their relevance.
- One way to do this is to estimate the relevance of a feature according to how well its values distinguish among the instances of the same and different classes (labels): suppose that A is a feature, and that its values among the instances changes *pretty much* as the label does, then its relevance is high.

Supervised Filters: An Example - 2

- Another approach used in filters is based on the selection of a subset of features and the evaluation of 'goodness' of the entire subset.
- For example, we can use a selection filter to eliminate *redundant* features: we can establish the *consistency* of a subset w.r.t. the original set.
- To this end, we can establish that A is redundant if eliminating it does not change the behaviour of the label over the instances.

Wrapper-Based Methods - 1

- A *wrapper-based* method considers the selection of a set of features as a *search problem*, where different combinations are prepared, evaluated and compared to other combinations.
- A predictive model is used to evaluate a combination of features and assign a score based on model accuracy, thus:
  1. A wrapper-based method *depends* on the model;
  2. A wrapper-based method can be *supervised* or *unsupervised* - but this time, only supervised wrapper methods are well-known, and unsupervised ones are under investigation; we'll be more precise later!

Wrapper-Based Methods - 2

- A wrapper model is based on the *predictive accuracy* of a *predetermined* learning algorithm to determine the quality of selected features, so that for each selection the learning algorithm is asked to check the accuracy of the classifier built over it.
- Four basic steps:
  1. *generation*: depends on the search strategy;
  2. *evaluation*: depends on the learning algorithm;
  3. *stopping criterion*: depends on the wrapper strategy;
  4. *validation*: usually final-user dependant.

Supervised Wrapper Methods - 1

- In supervised wrapper-based feature selection we can use *any* supervised learning algorithm as *black box*.
- For a given selection we just need to ask the learning algorithm to return a learning model *and* its performance.
- If you recall, there are many performance indicators:
  1. *accuracy*;
  2. *True Positive, True Negative, False Positive, False Negative. . .* ;
  3. *ROC curve*;
  4. . . .
- One parameter that can be used in stepping from a generic wrapper-based method to a specific one is precisely the performance indicator(s) used in the evaluation phase.

Supervised Wrapper Methods - 2

- So, for example, we can have a wrapper-based feature selection by using:
  1. A search strategy to produce a subset of the features (we'll be more precise in a moment!);
  2. The decision tree C4.5 as evaluation method;
  3. The accuracy of the prediction (correctly classified instances VS total instances) as evaluation meter.
- Also, we can decide to run the decision tree as full training (to speed up the process) or in cross-validation mode (to avoid false accuracy measures due to over-fitting).

Supervised Wrapper Methods - 3

- Now we know that the evaluation phase is not a problem: we have literally hundreds of possible choices as supervised learning algorithms.
- But what about the search strategy?
- Classical choices include:
  1. deterministic: *sequential* and *exponential* search methods, or
  2. random:
     - 2.1 *neural networks*;
     - 2.2 *particle swarm optimization*;
     - 2.3 *evolutionary algorithms*,
     - 2.4 ...

EA-based Supervised Wrapper Methods - 1

- How can an Evolutionary Algorithm serve as search method in a wrapper-based feature selection mechanism?
- To understand this, we need to understand how an EA works in the first place.
- The basic elements of a EA are:
  1. Individuals' representation and initialization;
  2. Fitness/Ojective function(s);
  3. Evolution strategy.

EA-based Supervised Wrapper Methods - 2

- In the particular case of feature selection, individual may be represented as:

$$\langle b_{A_1}, b_{A_2}, \ldots, b_{A_n}, c_1, c_2, \ldots \rangle$$

where:

1. each b_{A_i} is a bit (0=do not select this feature, 1=select this feature);
2. each c_j is a particular characteristic of the individual that depends on the search method - might be present or not.

EA-based Supervised Wrapper Methods - 3

- Typically, we randomly initialize P individuals (P possible selections of features);
- Following the generic EA methodology, we *evaluate* each one of them, by running a classifier on the *projection* of the original data set over the selection specified by the individual, and by taking a predetermined quality measure of the run as evaluation (e.g.: the accuracy).
- If the EA is *multi-objective*, we can add another quality measure: typically it should hold that the two measures are *inversely proportional*. For example, we can add the *number of selected features*.
- In this way, judging the relative quality of two individuals is more complex; the set of ”best” individuals is called *Pareto*-set.

EA-based Supervised Wrapper Methods - 4

- Given a set of individuals, we can only *exclude* those that do not belong to the Pareto set (or *front*); *I* does not belong to the front if and only if there exists *J* such that
  1. *J improves I* under at least one objective (e.g.: the accuracy), and
  2. *J is not worse than I* under every other objective (e.g.: the number of features).
- Under such conditions, *I* is *dominated* by *J* and thus does not belong to the front.

EA-based Supervised Wrapper Methods - 5

- From the population P we build de sub-set Q of Pareto-optimal individuals.
- Those in Q, plus some individuals in $P \setminus Q$ (which are randomly chosen in order to avoid local optima) are *combined* to form a new population P' of individuals *generically better* of those in P under the objective(s).
- The process is then iterated a predetermined number of times.
- The last population, and, in particular, the non-dominated individuals of the last generation, are those we consider for the selection. But which one?

EA-based Supervised Wrapper Methods - 6

- We design an *a posteriori* decision making process that takes into account domain-specific elements, and we choose one individual.
- The experiments tell us that this choice behaves much better than the original data set.
- A huge number of experiments are possible:
  1. by changing the classification model;
  2. by changing the objective function(s);
  3. by changing the selection mechanism hidden in the EA.

Unsupervised Classification - 1

- We already know what *unsupervised* classification is.
- Our data set is not labeled: the purpose is to separate the instances into disjoint subsets in the cleanest possible way.
- Our purpose now is to eliminate those features that make it difficult for a clustering algorithm to give us a proper result.

Unsupervised Classification - 2

- There are several way to measure the goodness of a clustering.
- A very simple way to do this is to measure the ratio between the biggest and the smallest obtained clusters: the heuristics here is to stay below 3.
- Another way, possible only with certain clustering algorithms, is to measure the *likelihood* (for example, in the EM algorithm).
- Now, suppose that we choose any clustering method (C) and any meaningful measure of its performance (M).

Wrapper-based Feature Selection for Unsupervised Classification - 1

- To eliminate noisy features we can proceed precisely the same way as in supervised classification.
- We can choose evolutionary-based wrapper feature selection, and set the following objectives:
  1. Accuracy of the classification model obtained by a selection *after the clustering*;
  2. Measure of the goodness of the clustering.

Wrapper-based Feature Selection for Unsupervised Classification - 2

- Therefore, for each selection:
  1. We run the clustering algorithm C and we obtain its performance M;
  2. We run a classification learning algorithm, where the cluster becomes the class;
  3. We set M and the accuracy (or any other measure) to be maximized.
- Since there is no direct relation between the number of the features and M, we do not set the cardinality of the set of features to be minimized.

Experiments - 1

- As for the GAP data set:
  1. We have a data set composed by 49 features (search space: 2^{49}: operational, service, and central data;
  2. Each line was classified by the *outcome*;
  3. We ran a wrapper-based feature selection mechanism based on evolutionary computation, to minimize the number of features and maximize the accuracy.
- The accuracy increased from 95.09 to 95.45, and the number of features dropped to 8. Therefore, 41 features previously collected do to influence the outcome of a session.

Experiments - 2

- Psychology data set:
  1. We started by 152 features for 159 unlabeled instances: each instance corresponds to a child who undergone a psychological test (search space 2^{152}).
  2. We ran a wrapper-based feature selection mechanism based on evolutionary computation, to maximize both the likelihood of the clusterization (via EM) and the accuracy (via J48).
  3. We obtained a set of 10 features, and each child has been classified into one of two categories: the interpretation shows that this choice is perfectly coherent with the semantics of the questions in the test.
- The accuracy increased from 83.98 to 96.04, and the number of features dropped to 10. In this case we can conclude that *children examined by the test can be classified into two categories* using their answer in 10 of the 152 questions.