Feature subset selection using Thornton’s separability index and its applicability to a number of sparse proximity-based classifiers

John Greene

Department of Electrical Engineering
University of Cape Town, Private Bag, Rondebosch, 7001, South Africa.

jgreene@eng.uct.ac.za

Abstract

The nearest neighbour algorithm is a simple and intuitively appealing classifier with remarkably good performance on many data sets, but it is vulnerable to irrelevant inputs and may need to be preceded by an input feature subset-selection process. Wrapper methods are the most effective, since they are attuned to the inductive bias of the classifier, but they are computationally expensive. We propose the use of Thornton’s Separability Index as a simple measure of subset merit which is fast and easy to calculate, but gives results which are identical to the asymptotic result of multiple testing with random data splits. It would be attractive to apply this to more powerful proximity-based classifiers, such as the popular Support Vector Machine (which also exhibits vulnerability to irrelevant inputs, and is much slower to train than the nearest neighbour classifier). We show that this is indeed feasible. However the inductive bias induced by a kernel machine is more specific than that of a nearest neighbour classifier, because of the non-linear metric imposed by the kernel. We outline some ongoing work which investigates the use of Kernel Alignment as a way of evaluating potential subsets, and which exploits this specific bias. Some preliminary results are presented.

1. Introduction

An inductive learning algorithm predicts an appropriate response (or target) variable \( t \), given an input vector \( x \). If the task is one of regression, \( t \) is typically a real variable; if \( t \) is discrete, the task is one of classification. For simplicity we will deal here with the case of binary classification, and assume that \( t \) is either \(-1\) or \(+1\). This entails no loss of generality since multiclass classification can be implemented as a tree of sequential binary classifiers. The prediction is based on the similarity (in some defined sense) of \( x \) to the members of a corpus of \( x \)-vectors whose class is known, called the training set. The underlying assumption is that \( x \) is likely to share the class label of those vectors in the training set to which it is most similar. What exactly counts as ‘similar’ determines the ‘inductive bias’ of the algorithm.

A simple and obvious measure of similarity is Euclidean distance, where the vectors are considered as points in \( \mathbb{R}^d \), \( d \) being the number of elements in the vector. This leads to the Nearest Neighbour Classifier, an intuitively appealing approach in which a query point is assigned the same class label as the training point closest to it. Despite its simplicity, the nearest neighbour classifier performs rather well across a wide range of real-world problems, yielding results close to the state of the art on many of them. Inductive classification is impossible without some inductive bias, each of which has its own strengths and weaknesses. The ordering of points in a space by Euclidean distance is affected by the relative scaling of the axes of the space, and hence by the relative magnitude of the components of the input vectors. Frequently this is addressed by normalisation, scaling the range of each variable to unit standard deviation, for instance, though this can result in the loss of relevant information when the actual relative magnitudes of the components are significant.

2. Irrelevant inputs

A serious weakness of proximity-based classifiers is their sensitivity to irrelevant inputs – that is, input variables which (even when considered in combination with the other variables) are not significantly correlated with the target. The presence of such variables projects the data points into a higher dimensional space in an arbitrary way and dilutes the proximity measure on which the classification is based. The performance of a nearest neighbour classifier can be improved by first identifying these irrelevant features and eliminating them. This process, called ‘input feature subset selection’ has been the focus of many studies.

There are two main approaches, ‘filter’ methods and ‘wrapper methods’. In the former, feature selection proceeds independently of the learning algorithm. Characteristics such as correlation and mutual information between variables are used to select features which, \textit{a priori}, seem likely to provide a rich and diverse source of classificatory information. Since a filter-based feature selector can be followed by any classifier, it typically
takes no account of the latter’s specific inductive bias. In the wrapper method, on the other hand, the learning and testing phases of the classifier are integrated into the feature selection process. Each exploratory subset of features (whether selected by an exhaustive, greedy-eliminative or heuristic-stochastic search process) is evaluated by a train/test phase involving a specific classification algorithm. Thus the selection process is attuned to the specific inductive bias of the classifier used. As a result, wrapper methods usually outperform the more generalised filter methods, but often at the cost of a considerable computational burden.

Here we propose a method which does not closely fit either category. In speed and simplicity it resembles the filter approach. However, its inductive bias matches the nearest neighbour classifier and the figure of merit it returns for a particular feature subset is identical to that which would be obtained asymptotically through repeated testing.

### 3. The proposed method

Consider wrapper-based feature selection in which trial subsets are evaluated by running a nearest neighbour classifier. Since generalisation accuracy is at issue, each cycle of evaluation entails a training phase and a test phase using independent data. For this purpose the available data is split randomly into train- and test-data (multiple cross-validation or leave-one-out testing being far too computationally expensive in the context of an iterative search). The detail of the split raises a dilemma – too large a test set will leave us with depleted training data; too small a one results in test sampling error and consequent noisy evaluation, which impairs the convergence of the search. The problem could be solved in principle by multiple train/test cycles for each subset using different random splits, and averaging the generalisation error to reduce its variance. However once again this imposes an intolerable computational burden. Thornton’s separability index [1] yields a result identical to the asymptotic result of a large number of train/test cycles with random splits, but it requires only a simple non-iterative calculation. As a bonus, it can be efficiently computed using Matlab’s matrix-oriented data sorting facility.

Thornton’s separability index \( s \) is defined as the fraction of a set of data points whose classification labels are the same as those of their nearest neighbours. Thus it is a measure of the degree to which inputs associated with the same output tend to cluster together. It may be written

\[
s = \sum_{i=1}^{n} \left( f(x_i) + f(x'_i) + 1 \right) \mod 2
\]

where \( x' \) is the nearest neighbour of \( x \) and \( n \) is the number of points.

It is intuitively obvious that \( s \) will be close to 1 for a set of points in which those with opposite labels exist in tight, well-separated clusters. As the clusters move closer together and points from opposing classes begin to overlap, the index will begin to fall. If the centroids of

\begin{table}[h]
\centering
\begin{tabular}{|l|}
\hline
% Matlab code for Thornton’s separability index  \\
\hline
\% Accepts a p x d matrix X in which each row is a vector  \\
\% of d numeric features (usually norm. into the range 0-1)  \\
\% t is a vector of labels (usually +1 or -1)  \\
\% Returns s, a number between 0 and 1, a measure of  \\
\% degree to which classes are geometrically separable.  \\
\% s is the fraction of instances whose classification  \\
\% label is shared by its nearest neighbour (determined  \\
\% on the basis of simple Euclidean distance)  \\
\% X = X + 1e-3*randn(size(X)); % tie-breaker for toy data  \\
\% p = length(t); % number of points  \\
\% d2 = dist2(X,X); % squared euclidean distances  \\
\% [S,I] = sort(d2); % sort each col by distance; I = index  \\
\% t1 = t(I(1,:)); % labels of points (row 1 of d2)  \\
\% t2 = t(I(2,:)); % labels of n-neighbours (row 2 of d2)  \\
\% s = sum(t1==t2)/p; % frac. of points with same class n.nr  \\
\hline
\end{tabular}
\end{table}

the clusters coincide, or the points are uniformly distributed in the space without clustering, the nearest neighbour of a point will have no more than a 50% probability of having the same class label as its neighbour, and the separability index will be close to 0.5. A regular intermeshed grid of alternately-labelled points (as would be generated by the exclusive-OR or parity problems) would have \( s = 0 \).

In a loose sense \( s \) can be seen as a generalisation of the idea of linear separability (though this is represented by a boolean variable or predicate rather than a real-valued index).

### 4. Nearest neighbour to kernel machines

A second weakness of the simple nearest neighbour classifier is its sensitivity to outliers in the training data, and its tendency to ‘overfit’ the data (generalise poorly) when classes are not cleanly separated. This is due to the fact that the label associated with the region of space in the neighbourhood of a data point is totally dominated by that single point, and overlapping class distributions lead to highly convoluted and irregular decision boundaries. This can be mitigated by considering a number of points in an extended neighbourhood of the query point and ‘voting’ as in the k-nearest-neighbour algorithm, or through local averaging by diffusing the region of influence of each point through the use of radial-basis functions or ‘kernels’ associated with them. The effect of
overlapping kernels is to compute a weighted mean of the influence of a number of training points in a neighbourhood. This has a regularising effect on the decision boundary as well as minimising the effect of outliers or mis-classified points. Adjusting the sphere of influence or ‘width’ of the radial kernel gives us control over the degree of regularisation. The shape of the kernel too is not without significance, since it induces a non-linear metric in terms of which proximity is assessed. This is even more striking if the kernel used is local but non-radial (generalised correlation kernels, for instance).

From this we can see that, while kernel-based methods are proximity-based, and therefore share aspects of the inductive bias of the nearest neighbour classifier, there may also be significant differences in this inductive bias. It is therefore of interest to ask whether feature subset selection can also be used when the classification step to follow is kernel-based. A simple test on a number of standard data sets showed that it can – in fact the correlation between the separability index for various subsets and the actual generalisation accuracy is remarkably good. To determine the latter, we ran an iterative support vector machine (the kernel Adatron algorithm) on these data sets, and determined generalisation accuracy using tenfold cross-validation. We plotted generalisation accuracy against separability index for randomly-selected feature subsets, using a number of well-known data sets (liver, sonar, Wisconsin breast cancer, Cleveland heart disease). The results for the liver data are typical, and indicated below. Clearly separability index is effective in this role.

![Liver Data: generalisation vs separability index for randomly selected subsets](image)

The specificity of the kernel-induced inductive bias raises the question of whether other easily-calculated measures of feature-subset figures-of-merit exist which also could short-circuit the rather slow training/testing phase, and which, in addition could effectively exploit the more specialised inductive bias of the kernel machines. Recent work suggests that the concept of kernel alignment may furnish the basis for such a measure. After explaining the concept of kernel alignment we will outline the proposal and report briefly on ongoing work in this direction.

5. Sparse Kernel machines

First we must place the Support Vector Machine (SVM) and more recent variants [3] in a common context with the work described above. In the support vector machine with radial kernels, the input data is mapped via the kernels (one of which is centred at each training point) into a high- or infinite-dimensional feature space, where a (linear) hyperplane separator can almost certainly be found. In this respect it is the same as many well-established kernel-based classifiers such as the Parzen-window classifier or Probabilistic Neural Network (PNN). However, in the case of the SVM, it is not just any hyperplane separator which is found, but a maximal margin hyperplane – i.e. one which maximises the distance between the hyperplane and the data points. This has the effect of establishing bounds on generalisation error. In the case of the kernel Adatron algorithm this is achieved in a very simple way: an ‘importance weight’ $\alpha_i$ (a lagrangian multiplier in another context) is associated with each training data point. These weights are iteratively adjusted in such a way that the weights of those points near the decision boundary (the “support vectors”) increase in value, while many of those further from the boundary approach zero. This not only forces the decision boundary into the maximal-margin position, it also enforces sparsity through a compression of the training data, with consequent run-time speedup (those points whose $\alpha$-values are zero can be ignored).

The SVM neatly circumvents the dilemma of radial basis function selection, which it performs automatically. Simple and effective methods also exist for the a priori choice of the kernel spread parameter. Thus the design of an SVM classifier requires little in the way of arbitrary user choices or parameter-setting through validation testing (though there is a relatively non-critical parameter $C$ which limits local deviations of the decision boundary by erroneous or outlier training points, and is often set through cross-validation). Very recently, variant forms of sparse radial basis machines have appeared. However, as will be clear from the above, these algorithms, along with the SVM, share the basic nature of all proximity-based classifiers, and we may assume that they will be vulnerable to irrelevant input features (a fact which has recently been acknowledged in the literature in the case of the SVM). The seriousness of this is exacerbated by relatively long training times (compared with the nearest neighbour classifier) which makes the use of wrapper-based feature selection non-viable.


In any kernel classifier the kernel- or gram-matrix $K$ is a square symmetrical semipositive-definite matrix which summarises the pairwise geometric relationship between
the training points in terms of the metric induced by the
kernel function (each element $k_{ij}$ is the result of applying
the kernel function to each pair of data points $x_i, x_j$). Let
us assume that the training point labels ($\pm 1$) are contained
in a column vector $t$. We can define the outer-product
matrix $T = t x t'$. $T$ is also a square symmetrical matrix
the same size as $K$, with $t_{ij} = +1$ if $t_i = t_j$ and $t_{ij} = -1$ if
$t_i \neq t_j$. Thus $T$ summarises the available information about
the training data in respect of its class labelling.

Inductive learning exploits some relationship between
the geometrical distribution of the input data and the class
labelling. Since $K$ summarises geometrical proximity and
$T$ the class-related similarity of corresponding pairs of
points, we expect these matrices to exhibit some
similarity. This similarity can be measured by matrix
alignment, which is a generalisation of the normalised dot
product of two vectors.

If we consider a matrix as a vector-of-vectors, it is
natural to extend the idea of the dot- or inner-product of
vectors to the sum of the products of corresponding
elements of two matrices. For a general pair of
identically-shaped matrices $A$ and $B$ we will write this
generalised inner product

$$\sum_i \sum_j a_{ij} b_{ij} \quad \text{as} \quad \langle A, B \rangle_F.$$

The $F$ is a reference to the Frobenius norm of matrix $A$
which, in the above notation, could be written $\langle A, B \rangle_F^{1/2}$.
A natural measure of the similarity of two vectors is the
cosine of the generalised angle between them, calculated
as the normalised dot product:

$$\cos(\phi) = \frac{x^T y}{\sqrt{x^T x * y^T y}}$$

The matrix alignment factor $A$ is an obvious generali-
sation of this idea:

$$A(A, B) = \frac{\langle A, B \rangle_F}{\sqrt{\langle A, A \rangle_F * \langle B, B \rangle_F}}$$

$$\quad = \frac{\langle A, B \rangle_F}{( \langle A \rangle_F * \langle B \rangle_F)^{1/2}}$$

where $\langle X \rangle_F$ is the Frobenius norm of $X$.

To measure the alignment of a kernel matrix $K$ with a data
set labelled $t = [t_1, t_2, \ldots, t_n]^T$ we calculate the outer-
product matrix $T = t x t'$ and then compute the kernel-data
alignment $A_{id} = A(K, T)$ as in the above expression.

Various uses [2] have been proposed for the kernel
alignment measure in both supervised and unsupervised
learning. It has been shown that kernel parameters which
optimise the alignment index also result in optimal
generalisation of a classifier based on density estimation
using Parzen windowing (e.g. the Probabilistic Neural
Network). Here we propose its use to tune the width
parameter of a gaussian kernel (a use which has been
implied in the literature but not reported on in detail) and
a novel possible use as a figure of merit for feature subset
selection.

7. Parameter Tuning

The first is easily achieved; we simply calculate the
alignment of the kernel with the data for a range of values
of the width parameter $\sigma$. We tried this for a number of
toy data sets and several publicly-available benchmark
data sets. Although extensive empirical testing would be
needed to make a precise statement about its effectiveness
(work which is in progress), it is striking that typically the
alignment shows a clear peak at values of $\sigma$ close to those
identified by cross-validation as minimising the
generalisation error. For illustrative purposes we will
briefly discuss results obtained with one particular data
set – the Wisconsin breast cancer data (subsequently
referred to as WBC). In general form the results were
similar for many other data sets and comprehensive
results will be presented in due course.

![Alignment Index vs $\sigma$](image)

The multiple curves in the above graph are plots for
different feature subsets. For our present purposes, only
the general shape of any one of them is significant, but the
similarity of the curves and the near coincidence of their
peaks points to a near-independence of optimal kernel
width on feature subset. This simplifies the proposed
method and contributes to its robustness.
8. Alignment-based Feature Selection

As a first approach to investigating the sensitivity of kernel/data alignment to feature subsets, we tried computing both the separability index and the alignment index on (512) fully enumerated subsets of the WBC data (which has 9 features). The relationship between them, for all possible subsets is shown by a scattergram.

A portion of this information is shown on an expanded scale below, allowing more detail to be seen:

Both separability index and alignment index appear to be satisfactory as a way of selecting good feature subsets – both produce low generalisation error in cross-validation testing. Any differences in performance are subtle, and quantifying them would require careful empirical testing. It appears that the use of the separability index induces more sparse feature subsets, but with a much greater variability than is the case with the alignment index. The latter seems much more consistent in its evaluations of which features are important. Surprisingly, the top twenty subsets as selected by the two methods had only one subset in common (and that not near the top of the list).

The lack of consistency in the elements predominantly selected in favoured subsets is a puzzling feature of many methods of subset selection, and points to a fundamental ill-posed-ness of the problem. This is currently a focus of our attention.

9. Conclusions

Thornton’s separability index is an efficient alternative to multiple validation testing for simple proximity classifiers like the nearest neighbour classifier, where its asymptotic optimality has been demonstrated. Limited empirical testing suggests that this behaviour carries over to classifiers which are less directly describable as proximity-based. In particular, separability index appears to be a good criterion of selection for feature subsets when classification is being done by support vector machines.

Kernel/data alignment is another simple predictor of generalisation performance which can be applied to kernel selection and design. It can also be used for subset selection, although here no clear competitive advantage has been shown in this preliminary work. However its specific relation to the inductive bias implied by the kernel suggests that it may well offer significant advantages which are not evident when simple radial kernels are used. This deserves further investigation.

An ongoing quest (always tempered by the phantom of the ‘no free lunch’ theorems) is for learning machines and algorithms free from a panoply of more-or-less arbitrary user-selected parameters and empirical choices. Very significant steps in that direction have been taken in the last few years. Using separability index for feature selection, followed by kernel alignment to select the width of a gaussian radial basis function, followed by a modern sparse kernel classifier (such as Boudat and Anouar’s remarkable algorithm [4]) yield performance which is close to the state of the art for many datasets without a single arbitrary choice having to be made by the user, and without any trial and error experimentation.

10. References