Dear Max

Please find attached six pages from Prof John Shawe-Taylor.

If you have any queries, please do not hesitate to contact me.

Many thanks
With kind regards

Eileen Simon

Email: ess@ecs.soton.ac.uk
Tel: 00 44 23 8059 4788
Ranking, clustering and data visualisation

distances for a fixed number of clusters automatically maximises the between-cluster distances.

There is another nice property of the solution of the optimisation criterion (8.6). If we simply expand the expression, we obtain

$$\text{opt} = \sum_{i,j:f_i=f_j} \| \phi(x_i) - \phi(x_j) \|^2$$

$$= \sum_{k=1}^{N} \sum_{i:f_i=k} \sum_{j:f_j=k} (\phi(x_i) - \phi(x_j), \phi(x_i) - \phi(x_j))$$

$$= \sum_{k=1}^{N} 2 \left( |f^{-1}(k)| \sum_{i:f_i=k} \kappa(x_i, x_i) - \sum_{i:f_i=k} \sum_{j:f_j=k} \kappa(x_i, x_j) \right)$$

$$= \sum_{k=1}^{N} 2 |f^{-1}(k)| \sum_{i:f_i=k} \| \phi(x_i) - \mu_k \|^2,$$

where the last line follows from (5.4) of Chapter 5 expressing the average-squared distance of a set of points from their centre of mass, and

$$\mu_k = \frac{1}{|f^{-1}(k)|} \sum_{i \in f^{-1}(k)} \phi(x_i) \quad (8.8)$$

is the centre of mass of those examples assigned to cluster $k$, a point often referred to as the centroid of the cluster. This implies that the optimisation criterion (8.6) is therefore also equivalent to the criterion

$$f = \arg \min_f \sum_{k=1}^{N} \left( \sum_{i:f_i=k} \| \phi(x_i) - \mu_k \|^2 \right) = \arg \min_f \sum_{i=1}^{N} \| \phi(x_i) - \mu_{f(x_i)} \|^2,$$

that seeks a clustering of points minimising the sum-squared distances to the centres of mass of the clusters. One might be tempted to assume that this implies the points are assigned to the cluster whose centroid is nearest. The following theorem shows that indeed this is the case.

Theorem 8.18 The solution of the clustering optimisation criterion

$$f = \arg \min_f \sum_{i,j:f_i=f_j} \| \phi(x_i) - \phi(x_j) \|^2$$

subject to the constraint of equal sized clusters

$$f(x_i) = \arg \min_{1 \leq k \leq N} \| \phi(x_i) - \mu_k \|,$$
8.2 Discovering cluster structure in a feature space

where $\mu_j$ is the centroid of the points assigned to cluster $j$.

Proof Let $\mu_k$ be as in equation (8.8). If we consider a clustering function $g$ defined on $S$ that assigns points to the nearest centroid

$$ g(x_i) = \arg\min_{1 \leq k \leq N} \| \phi(x_i) - \mu_k \|, $$

we have, by the definition of $g$

$$ \sum_{i=1}^l \| \phi(x_i) - \mu_{g(x_i)} \|^2 \leq \sum_{i=1}^l \| \phi(x_i) - \mu_{f(x_i)} \|^2. \quad (8.10) $$

Furthermore, if we let

$$ \hat{\mu}_k = \frac{1}{|g^{-1}(k)|} \sum_{i \in g^{-1}(k)} \phi(x_i) $$

it follows that

$$ \sum_{i=1}^l \| \phi(x_i) - \hat{\mu}_{g(x_i)} \|^2 \leq \sum_{i=1}^l \| \phi(x_i) - \mu_{g(x_i)} \|^2 \quad (8.11) $$

by Proposition 5.2. But the left-hand side is the value of the optimisation criterion (8.9) for the function $g$. Since $f$ was assumed to be optimal we must have

$$ \sum_{i=1}^l \| \phi(x_i) - \hat{\mu}_{g(x_i)} \|^2 \geq \sum_{i=1}^l \| \phi(x_i) - \mu_{f(x_i)} \|^2, $$

implying with (8.10) and (8.11) that the two are in fact equal. The result follows.$\Box$

The characterisation given in Proposition 8.18 also indicates how new data should be assigned to the clusters. We simply use the natural generalisation of the assignment as

$$ f(x) = \arg\min_{1 \leq k \leq N} \| \phi(x) - \mu_k \|. $$

Once we have chosen the cost function of Computation 8.17 and observed that its test performance is bound solely in terms of the number of centres and the value of equation (8.6) on the training examples, it is clear that any clustering algorithm must attempt to minimise the cost function. Typically we might expect to do this for different numbers of centres, finally selecting the number for which the bound on $E_\Omega \min_{1 \leq k \leq N} \| \phi(x) - \mu_k \|^2$ is minimally acceptable.
Hence, the core task is given a fixed number of centres $N$ find the partition into clusters which minimises equation (8.6). In view of Proposition 8.18, we therefore arrive at the following clustering optimisation strategy.

**Computation 8.19 [Clustering optimisation strategy]** The clustering optimisation strategy is given by

\[
\text{input} \quad S = \{x_1, \ldots, x_d\}, \text{ integer } N \\
\text{process} \quad \mu = \arg\min_{\mu} \sum_{i=1}^{d} \min_{1 \leq k \leq N} \left\| \phi(x_i) - \mu_k \right\|^2 \\
\text{output} \quad f(\cdot) = \arg\min_{1 \leq k \leq N} \|\phi(\cdot) - \mu_k\|
\]

Figure 8.3 illustrates this strategy by showing the distances (dotted arrows) involved in computed the sum-squared criterion. The minimisation of this sum automatically maximises the indicated distance (dot-dashed arrow) between the cluster centres.

Fig. 8.3. The clustering criterion reduces to finding cluster centres to minimise sum-squared distances.

**Remark 8.20 [Stability analysis]** Furthermore we can see that this strategy suggests an appropriate pattern function for our stability analysis to
8.2 Discovering cluster structure in a feature space

The first term on the right-hand side is just the value of the Computation 8.19 that is minimised by the clustering function, while the value of the left-hand side is independent of the clustering. Hence, the clustering criterion automatically maximises the trace of the second term on the right-hand side. This corresponds to maximising

$$\sum_{k=1}^{N} |f^{-1}(k)| \|\mu_k - \phi_S\|^2;$$

in other words the sum of the squares of the distances from the overall mean of the cluster means weighted by their size. We again see that optimising the tightness of the clusters automatically forces their centres to be far apart.

8.2.2 Greedy solution: \(k\)-means

Proposition 8.18 confirms that we can solve Computation 8.17 by identifying centres of mass of the members of each cluster. The first algorithm we will describe attempts to do just this and is therefore referred to as the \(k\)-means algorithm. It keeps a set of cluster centroids \(C_1, C_2, \ldots, C_N\) that are initialised randomly and then seeks to minimise the expression

$$\sum_{i=1}^{\ell} \|\phi(x_i) - C_f(x_i)\|^2,$$

by adapting both \(f\) as well as the centres. It will converge to a solution in which \(C_k\) is the centre of mass of the points assigned to cluster \(k\) and hence will satisfy the criterion of Proposition 8.18.

The algorithm alternates between updating \(f\) to adjust the assignment of points to clusters and updating the \(C_k\) giving the positions of the centres in a two-stage iterative procedure. The first stage simply moves points to the cluster whose cluster centre is closest. Clearly this will reduce the value of the expression in (8.12). The second stage repositions the centre of each cluster at the centre of mass of the points assigned to that cluster. We have already analysed this second stage in Proposition 5.2 showing that moving the cluster centre to the centre of mass of the points does indeed reduce the criterion of (8.12).

Hence, each stage can only reduce the expression (8.12). Since the number of possible clusterings is finite, it follows that after a finite number of iterations the algorithm will converge to a stable clustering assignment provided ties are broken in a deterministic way. If we are to implement in a dual form we must represent the clusters by an indicator matrix \(A\) of dimension \(\ell \times N\).
8.2 Discovering cluster structure in a feature space

```
% original kernel matrix stored in variable K
% clustering given by a ell x N binary matrix A
% and cluster allocation function f
% d gives the distances to cluster centroids
A = zeros(ell,N);
f = ceil(rand(ell,1)* N);
for i=1,ell
    A(i,f(i)) = 1;
end
change = 1;
while change = 1
    change = 0;
    E = A * diag(1./sum(A));
    Z = ones(ell,1)* diag(E'*K*E)' - 2*K*E;
    [d, ff] = min(Z, [1, 2]);
    for i=1,ell
        if f(i) == ff(i)
            A(i,ff(i)) = 1;
            A(i, f(i)) = 0;
            change = 1;
        end
    end
end
f = ff;
```

Code Fragment 8.3. Matlab code to perform k-means clustering.


to obtain a closed form approximation. We can then study the approximation and statistical properties of its solutions.

Clustering into two classes We first consider the simpler case when there are just two clusters. In this relaxation we represent the cluster assignment by a vector \( y \in \{-1,+1\}^\ell \), that associates to each point a \{\(-1,+1\)\} label. For the two classes case the clustering quality criterion described above is minimised by maximising

\[
\sum_{y_i \neq y_j} \| \phi(x_i) - \phi(x_j) \|^2.
\]

Assuming that the data is normalised and the sizes of the clusters are equal this will correspond to minimising the so-called cut cost

\[
2 \sum_{y_i \neq y_j} \kappa(x_i, x_j) = \sum_{i,j=1}^\ell \kappa(x_i, x_j) - \sum_{i,j=1}^\ell y_i y_j \kappa(x_i, x_j),
\]

subject to \( y \in \{-1,+1\}^\ell \), \( \sum_{i=1}^{\ell} y_i = 0 \).
Ranking, clustering and data visualisation

since it measures the kernel 'weight' between vertices in different clusters. Hence, we must solve

\[
\max \quad y'Ky \\
\text{subject to} \quad y \in \{-1,+1\}^\ell \\
\sum_{i=1}^{\ell} y_i = 0
\]

We can relax this optimisation by removing the restriction that \( y \) be a binary vector while controlling its norm. This is achieved by maximising the Raleigh quotient (3.2)

\[
\max \frac{y'Ky}{y'y}.
\]

As observed in Chapter 3 this is solved by the eigenvector of the matrix \( K \) corresponding to the largest eigenvalue with the value of the quotient equal to the eigenvalue \( \lambda_1 \). Hence, we obtain a lower bound on the cut cost of

\[
0.5 \left( \sum_{i,j=1}^{\ell} \kappa(x_i, x_j) - \lambda_1 \right),
\]

giving a corresponding lower bound on the value of the sum-squared criterion. Though such a lower bound is useful, the question remains as to whether the approach can suggest useful clusterings of the data. A very natural way to do so in this two-cluster case is simply to threshold the vector \( y \) hence converting it to a binary clustering vector. This naive approach can deliver surprisingly good results though there is no a priori guarantee attached to the quality of the solution.

Remark 8.23 [Alternative criterion] It is also possible to consider minimising a ratio between the cut size and a measure of the size of the clusters. This leads through similar relaxations to different eigenvalue problems. For example if we let \( D \) be the diagonal matrix with entries

\[
D_{ii} = \sum_{j=1}^{\ell} K_{ij},
\]

then useful partitions can be derived from the eigenvectors of

\[
D^{-1}K, \quad D^{-1/2}KD^{-1/2} \quad \text{and} \quad K-D
\]

with varying justifications. In all cases thresholding the resulting vectors delivers the corresponding partitions. Generally the approach is motivated using the Gaussian kernel with its useful properties discussed above.