5.2 Computing projections: Gram–Schmidt, QR and Cholesky

The basic classification function of the previous section had the form of a thresholded linear function

$$h(\mathbf{x}) = \operatorname{sgn}\left(\langle \mathbf{w}, \boldsymbol{\phi}(\mathbf{x}) \rangle\right),$$

where the weight vector $\mathbf{w}$ had the form

$$\mathbf{w} = \frac{1}{\ell_+}\sum_{i=1}^{\ell_+} \boldsymbol{\phi}(\mathbf{x}_i) - \frac{1}{\ell_-}\sum_{i=\ell_++1}^{\ell} \boldsymbol{\phi}(\mathbf{x}_i).$$

Hence, the computation only requires knowledge of the inner product between two feature space vectors.

The projection $P_{\mathbf{w}}\left(\boldsymbol{\phi}(\mathbf{x})\right)$ of a vector $\boldsymbol{\phi}(\mathbf{x})$ onto the vector $\mathbf{w}$ is given as

$$P_{\mathbf{w}}\left(\boldsymbol{\phi}(\mathbf{x})\right) = \frac{\langle \mathbf{w}, \boldsymbol{\phi}(\mathbf{x}) \rangle}{\|\mathbf{w}\|^2}\mathbf{w}.$$

This example illustrates a general principle that also enables us to compute projections of vectors in the feature space. For example given a general vector

$$\mathbf{w} = \sum_{i=1}^{\ell} \alpha_i \boldsymbol{\phi}(\mathbf{x}_i),$$

we can compute the norm of the projection $P_{\mathbf{w}}\left(\boldsymbol{\phi}(\mathbf{x})\right)$ of the image of a point $\mathbf{x}$ onto the vector $\mathbf{w}$ as

$$\left\|P_{\mathbf{w}}\left(\boldsymbol{\phi}(\mathbf{x})\right)\right\| = \frac{\langle \mathbf{w}, \boldsymbol{\phi}(\mathbf{x}) \rangle}{\|\mathbf{w}\|} = \frac{\sum_{i=1}^{\ell} \alpha_i \kappa\left(\mathbf{x}_i, \mathbf{x}\right)}{\sqrt{\sum_{i,j=1}^{\ell} \alpha_i \alpha_j \kappa\left(\mathbf{x}_i, \mathbf{x}_j\right)}}.$$

Using Pythagoras's theorem allows us to compute the distance of the point from its projection as

$$\begin{aligned}
\left\|P_{\mathbf{w}}\left(\boldsymbol{\phi}(\mathbf{x})\right) - \boldsymbol{\phi}(\mathbf{x})\right\|^2 &= \|\boldsymbol{\phi}(\mathbf{x})\|^2 - \left\|P_{\mathbf{w}}\left(\boldsymbol{\phi}(\mathbf{x})\right)\right\|^2 \\
&= \kappa\left(\mathbf{x}, \mathbf{x}\right) - \frac{\left(\sum_{i=1}^{\ell} \alpha_i \kappa\left(\mathbf{x}_i, \mathbf{x}\right)\right)^2}{\sum_{i,j=1}^{\ell} \alpha_i \alpha_j \kappa\left(\mathbf{x}_i, \mathbf{x}_j\right)}.
\end{aligned}$$

If we have a set of orthonormal vectors $\mathbf{w}_1, \ldots, \mathbf{w}_k$ with corresponding dual representations given by $\boldsymbol{\alpha}^1, \ldots, \boldsymbol{\alpha}^k$, we can compute the orthogonal projection $P_V\left(\boldsymbol{\phi}(\mathbf{x})\right)$ of a point $\boldsymbol{\phi}(\mathbf{x})$ into the subspace V spanned by

$\mathbf{w}_1, \ldots, \mathbf{w}_k$ as

$$P_V(\boldsymbol{\phi}(\mathbf{x})) = \left(\sum_{i=1}^{\ell} \boldsymbol{\alpha}_i^j \kappa(\mathbf{x}_i, \mathbf{x})\right)_{j=1}^{k},$$

where we have used the vectors $\mathbf{w}_1, \ldots, \mathbf{w}_k$ as a basis for V.

Definition 5.8 A *projection* is a mapping P satisfying

$$P(\boldsymbol{\phi}(\mathbf{x})) = P^2(\boldsymbol{\phi}(\mathbf{x})) \text{ and } \langle P(\boldsymbol{\phi}(\mathbf{x})), \boldsymbol{\phi}(\mathbf{x}) - P(\boldsymbol{\phi}(\mathbf{x}))\rangle = 0,$$

with its dimension $\dim(P)$ given by the dimension of the image of P. The orthogonal projection to P is given by

$$P^{\perp}(\boldsymbol{\phi}(\mathbf{x})) = \boldsymbol{\phi}(\mathbf{x}) - P(\boldsymbol{\phi}(\mathbf{x}))$$

and projects the data onto the orthogonal complement of the image of P, so that $\dim(P) + \dim(P^{\perp}) = N$, the dimension of the feature space. ■

Remark 5.9 [Orthogonal projections] It is not hard to see that the orthogonal projection is indeed a projection, since

$$P^{\perp}\left(P^{\perp}(\boldsymbol{\phi}(\mathbf{x}))\right) = P^{\perp}(\boldsymbol{\phi}(\mathbf{x})) - P\left(P^{\perp}(\boldsymbol{\phi}(\mathbf{x}))\right) = P^{\perp}(\boldsymbol{\phi}(\mathbf{x})),$$

while

$$\begin{aligned} &\left\langle P^{\perp}(\boldsymbol{\phi}(\mathbf{x})), \boldsymbol{\phi}(\mathbf{x}) - P^{\perp}(\boldsymbol{\phi}(\mathbf{x}))\right\rangle \\ &\quad = \left\langle P^{\perp}(\boldsymbol{\phi}(\mathbf{x})), \boldsymbol{\phi}(\mathbf{x}) - (\boldsymbol{\phi}(\mathbf{x}) - P(\boldsymbol{\phi}(\mathbf{x})))\right\rangle \\ &\quad = \langle(\boldsymbol{\phi}(\mathbf{x}) - P(\boldsymbol{\phi}(\mathbf{x}))), P(\boldsymbol{\phi}(\mathbf{x}))\rangle = 0. \end{aligned}$$

■

Projections and deflations The projection $P_{\mathbf{w}}(\boldsymbol{\phi}(\mathbf{x}))$ of $\boldsymbol{\phi}(\mathbf{x})$ onto $\mathbf{w}$ introduced above are onto a 1-dimensional subspace defined by the vector $\mathbf{w}$. If we assume that $\mathbf{w}$ is normalised, $P_{\mathbf{w}}(\boldsymbol{\phi}(\mathbf{x}))$ can also be expressed as

$$P_{\mathbf{w}}(\boldsymbol{\phi}(\mathbf{x})) = \mathbf{w}\mathbf{w}'\boldsymbol{\phi}(\mathbf{x}).$$

Hence, its orthogonal projection $P_{\mathbf{w}}^{\perp}(\boldsymbol{\phi}(\mathbf{x}))$ can be expressed as

$$P_{\mathbf{w}}^{\perp}(\boldsymbol{\phi}(\mathbf{x})) = \left(\mathbf{I} - \mathbf{w}\mathbf{w}'\right)\boldsymbol{\phi}(\mathbf{x}).$$

If we have a data matrix $\mathbf{X}$ with rows $\boldsymbol{\phi}(\mathbf{x}_i)$, $i = 1, \ldots, \ell$, then deflating the matrix $\mathbf{X}'\mathbf{X}$ with respect to one of its eigenvectors $\mathbf{w}$ is equivalent to projecting the data using $P_{\mathbf{w}}^{\perp}$. This follows from the observation that projecting the data creates the new data matrix

$$\tilde{\mathbf{X}} = \mathbf{X}\left(\mathbf{I} - \mathbf{w}\mathbf{w}'\right)' = \mathbf{X}\left(\mathbf{I} - \mathbf{w}\mathbf{w}'\right), \tag{5.8}$$

so that

$$\begin{aligned}
\tilde{\mathbf{X}}'\tilde{\mathbf{X}} &= \left(\mathbf{I} - \mathbf{w}\mathbf{w}'\right)\mathbf{X}'\mathbf{X}\left(\mathbf{I} - \mathbf{w}\mathbf{w}'\right) \\
&= \mathbf{X}'\mathbf{X} - \mathbf{w}\mathbf{w}'\mathbf{X}'\mathbf{X} - \mathbf{X}'\mathbf{X}\mathbf{w}\mathbf{w}' + \mathbf{w}\mathbf{w}'\mathbf{X}'\mathbf{X}\mathbf{w}\mathbf{w}' \\
&= \mathbf{X}'\mathbf{X} - \lambda\mathbf{w}\mathbf{w}' - \boldsymbol{\lambda}\mathbf{w}\mathbf{w}' + \lambda\mathbf{w}\mathbf{w}'\mathbf{w}\mathbf{w}' \\
&= \mathbf{X}'\mathbf{X} - \lambda\mathbf{w}\mathbf{w}',
\end{aligned}$$

where λ is the eigenvalue corresponding to $\mathbf{w}$.

The actual spread of the data may not be spherical as is implicitly assumed in the novelty detector derived in the previous section. We may indeed observe that the data lies in a subspace of the feature space of lower dimensionality.

We now consider how to find an orthonormal basis for such a subspace. More generally we seek a subspace that fits the data in the sense that the distances between data items and their projections into the subspace are small. Again we would like to compute the projections of points into subspaces of the feature space implicitly using only information provided by the kernel.

Gram–Schmidt orthonormalisation We begin by considering a well-known method of deriving an orthonormal basis known as the *Gram–Schmidt* procedure. Given a sequence of linearly independent vectors the method creates the basis by orthogonalising each vector to all of the earlier vectors. Hence, if we are given the vectors

$$\boldsymbol{\phi}(\mathbf{x}_1), \boldsymbol{\phi}(\mathbf{x}_2), \ldots, \boldsymbol{\phi}(\mathbf{x}_\ell),$$

the first basis vector is chosen to be

$$\mathbf{q}_1 = \frac{\boldsymbol{\phi}(\mathbf{x}_1)}{\|\boldsymbol{\phi}(\mathbf{x}_1)\|}.$$

The ith vector is then obtained by subtracting from $\boldsymbol{\phi}(\mathbf{x}_i)$ multiples of $\mathbf{q}_1, \ldots, \mathbf{q}_{i-1}$ in order to ensure it becomes orthogonal to each of them

$$\boldsymbol{\phi}(\mathbf{x}_i) \longrightarrow \boldsymbol{\phi}(\mathbf{x}_i) - \sum_{j=1}^{i-1} \langle \mathbf{q}_j, \boldsymbol{\phi}(\mathbf{x}_i) \rangle \mathbf{q}_j = \left(\mathbf{I} - \mathbf{Q}_{i-1}\mathbf{Q}_{i-1}'\right)\boldsymbol{\phi}(\mathbf{x}_i),$$

where $\mathbf{Q}_i$ is the matrix whose i columns are the first i vectors $\mathbf{q}_1, \ldots, \mathbf{q}_i$. The matrix $(\mathbf{I} - \mathbf{Q}_i\mathbf{Q}_i')$ is a projection matrix onto the orthogonal complement of the space spanned by the first i vectors $\mathbf{q}_1, \ldots, \mathbf{q}_i$. Finally, if we let

$$\nu_i = \left\| \left(\mathbf{I} - \mathbf{Q}_{i-1}\mathbf{Q}'_{i-1} \right) \boldsymbol{\phi}\left(\mathbf{x}_i\right) \right\|,$$

the next basis vector is obtained by normalising the projection

$$\mathbf{q}_i = \nu_i^{-1} \left(\mathbf{I} - \mathbf{Q}_{i-1}\mathbf{Q}'_{i-1} \right) \boldsymbol{\phi}\left(\mathbf{x}_i\right).$$

It follows that

$$\begin{aligned} \boldsymbol{\phi}\left(\mathbf{x}_i\right) &= \mathbf{Q}_{i-1}\mathbf{Q}'_{i-1}\boldsymbol{\phi}\left(\mathbf{x}_i\right) + \nu_i \mathbf{q}_i = \mathbf{Q}_i \begin{pmatrix} \mathbf{Q}'_{i-1}\boldsymbol{\phi}\left(\mathbf{x}_i\right) \\ \nu_i \end{pmatrix} \\ &= \mathbf{Q} \begin{pmatrix} \mathbf{Q}'_{i-1}\boldsymbol{\phi}\left(\mathbf{x}_i\right) \\ \nu_i \\ \mathbf{0}_{\ell-i} \end{pmatrix} = \mathbf{Q}\mathbf{r}_i, \end{aligned}$$

where $\mathbf{Q} = \mathbf{Q}_\ell$ is the matrix containing all the vectors $\mathbf{q}_i$ as columns. This implies that the matrix $\mathbf{X}$ containing the data vectors as rows can be decomposed as

$$\mathbf{X}' = \mathbf{Q}\mathbf{R},$$

where $\mathbf{R}$ is an upper triangular matrix with ith column

$$\mathbf{r}_i = \begin{pmatrix} \mathbf{Q}'_{i-1}\boldsymbol{\phi}\left(\mathbf{x}_i\right) \\ \nu_i \\ \mathbf{0}_{\ell-i} \end{pmatrix}.$$

We can also view $\mathbf{r}_i$ as the respresentation of $\mathbf{x}_i$ in the basis

$$\left\{\mathbf{q}_1, \ldots, \mathbf{q}_\ell\right\}.$$

QR-decomposition This is the well-known *QR-decomposition* of the matrix $\mathbf{X}'$ into the product of an orthonormal matrix $\mathbf{Q}$ and upper triangular matrix $\mathbf{R}$ with positive diagonal entries.

We now consider the application of this technique in a kernel-defined feature space. Consider the matrix $\mathbf{X}$ whose rows are the projections of a dataset

$$S = \left\{\mathbf{x}_1, \ldots, \mathbf{x}_\ell\right\}$$

into a feature space defined by a kernel κ with corresponding feature mapping $\boldsymbol{\phi}$. Applying the Gram–Schmidt method in the feature space would lead to the decomposition

$$\mathbf{X}' = \mathbf{Q}\mathbf{R},$$

defined above. This gives the following decomposition of the kernel matrix

$$\mathbf{K} = \mathbf{X}\mathbf{X}' = \mathbf{R}'\mathbf{Q}'\mathbf{Q}\mathbf{R} = \mathbf{R}'\mathbf{R}.$$

Definition 5.10 This is the *Cholesky decomposition* of a positive semi-definite matrix into the product of a lower triangular and upper triangular matrix that are transposes of each other.

Since the Cholesky decomposition is unique, performing a Cholesky decomposition of the kernel matrix is equivalent to performing Gram–Schmidt orthonormalisation in the feature space and hence we can view Cholesky decomposition as the dual implementation of the Gram–Schmidt orthonormalisation. ■

Cholesky implementation The computation of the (j, i)th entry in the matrix $\mathbf{R}$ corresponds to evaluating the inner product between the ith vector $\boldsymbol{\phi}(\mathbf{x}_i)$ with the jth basis vector $\mathbf{q}_j$, for $i > j$. Since we can decompose $\boldsymbol{\phi}(\mathbf{x}_i)$ into a component lying in the subspace spanned by the basis vectors up to the jth for which we have already computed the inner products and the perpendicular complement, this inner product is given by

$$\nu_j \langle \mathbf{q}_j, \boldsymbol{\phi}(\mathbf{x}_i) \rangle = \langle \boldsymbol{\phi}(\mathbf{x}_j), \boldsymbol{\phi}(\mathbf{x}_i) \rangle - \sum_{t=1}^{j-1} \langle \mathbf{q}_t, \boldsymbol{\phi}(\mathbf{x}_j) \rangle \langle \mathbf{q}_t, \boldsymbol{\phi}(\mathbf{x}_i) \rangle,$$

which corresponds to the Cholesky computation performed for $j = 1, \ldots, \ell$

$$\mathbf{R}_{ji} = \nu_j^{-1} \left(\mathbf{K}_{ji} - \sum_{t=1}^{j-1} \mathbf{R}_{tj} \mathbf{R}_{ti} \right), \; i = j+1, \ldots, \ell,$$

where ν_j is obtained by keeping track of the residual norm squared d_i of the vectors in the orthogonal complement. This is done by initialising with the diagonal of the kernel matrix

$$d_i = \mathbf{K}_{ii}$$

and updating with

$$d_i \leftarrow d_i - \mathbf{R}_{ji}^2$$

as the ith entry is computed. The value of ν_j is then the residual norm of the next vector; that is

$$\nu_j = \sqrt{d_j}.$$

Note that the new representation of the data as the columns of the matrix $\mathbf{R}$ gives rise to exactly the same kernel matrix. Hence, we have found a new projection function

$$\hat{\boldsymbol{\phi}} : \mathbf{x}_i \longmapsto \mathbf{r}_i$$

which gives rise to the same kernel matrix on the set S; that is

$$\kappa\left(\mathbf{x}_i, \mathbf{x}_j\right) = \hat{\kappa}\left(\mathbf{x}_i, \mathbf{x}_j\right) = \left\langle \hat{\boldsymbol{\phi}}\left(\mathbf{x}_i\right), \hat{\boldsymbol{\phi}}\left(\mathbf{x}_j\right) \right\rangle, \text{ for all } i, j = 1, \ldots, \ell.$$

This new projection maps data into the coordinate system determined by the orthonormal basis $\mathbf{q}_1, \ldots, \mathbf{q}_\ell$. Hence, to compute $\hat{\boldsymbol{\phi}}$ and thus $\hat{\kappa}$ for new examples, we must evaluate the projections onto these basis vectors in the feature space. This can be done by effectively computing an additional column denoted by $\mathbf{r}$ of an extension of the matrix $\mathbf{R}$ from an additional column of $\mathbf{K}$ denoted by $\mathbf{k}$

$$\mathbf{r}_j = \nu_j^{-1}\left(\mathbf{k}_j - \sum_{t=1}^{j-1} \mathbf{R}_{tj}\mathbf{r}_t\right), \; j = 1, \ldots, \ell.$$

We started this section by asking how we might find a basis for the data when it lies in a subspace, or close to a subspace, of the feature space. If the data are not linearly independent the corresponding residual norm d_j will be equal to 0 when we come to process an example that lies in the subspace spanned by the earlier examples. This will occur if and only if the data lies in a subspace of dimension $j-1$, which is equivalent to saying that the rank of the matrix $\mathbf{X}$ is $j-1$. But this is equivalent to deriving

$$\mathbf{K} = \mathbf{R}'\mathbf{R}$$

with $\mathbf{R}$ a $(j-1) \times \ell$ matrix, or in other words to $\mathbf{K}$ having rank $j-1$. We have shown the following result.

Proposition 5.11 *The rank of the dataset S is equal to that of the kernel matrix $\mathbf{K}$ and by symmetry that of the matrix $\mathbf{X}'\mathbf{X}$.*

We can therefore compute the rank of the data in the feature space by computing the rank of the kernel matrix that only involves the inner products between the training points. Of course in high-dimensional feature spaces we may expect the rank to be equal to the number of data points. If we use the Gaussian kernel this will always be the case if the points are distinct.

Clearly the size of d_j indicates how independent the next example is from the examples processed so far. If we wish to capture the most important dimensions of the data points it is therefore natural to vary the order that the examples are processed in the Cholesky decomposition by always choosing the point with largest residual norm, while those with small residuals are eventually ignored altogether. This leads to a reordering of the order in which the examples are processed. The reordering is computed by the statement

```
[a, I(j + 1)] = max(d);
```

in the Matlab code below with the array `I` storing the permutation.

This approach corresponds to pivoting in Cholesky decomposition, while failing to include all the examples is referred to as an *incomplete Cholesky decomposition*. The corresponding approach in the feature space is known as *partial Gram–Schmidt orthonormalisation*.

Algorithm 5.12 [Cholesky decomposition or dual Gram–Schmidt] Matlab code for the incomplete Cholesky decomposition, equivalent to the dual partial Gram–Schmidt orthonormalisation is given in Code Fragment 5.4. ■

Notice that the index array I stores the indices of the vectors in the order in which they are chosen, while the parameter η allows for the possibility that the data is only approximately contained in a subspace. The residual norms will all be smaller than this value, while the dimension of the feature space obtained is given by T. If η is set small enough then T will be equal to the rank of the data in the feature space. Hence, we can determine the rank of the data in the feature space using Code Fragment 5.4.

The partial Gram–Schmidt procedure can be viewed as a method of reducing the size of the residuals by a greedy strategy of picking the largest at each iteration. This naturally raises the question of whether smaller residuals could result if the subspace was chosen globally to minimise the residuals. The solution to this problem will be given by choosing the eigensubspace that will be shown to minimise the sum-squared residuals. The next section begins to examine this approach to assessing the spread of the data in the feature space, though final answers to these questions will be given in Chapter 6.

5.3 Measuring the spread of the data

The mean estimates where the data is centred, while the variance measures the extent to which the data is spread. We can compare two zero-mean uni-

```
% original kernel matrix stored in variable K
% of size ell x ell.
% new features stored in matrix R of size T x ell
% eta gives threshold residual cutoff
j = 0;
R = zeros(ell,ell);
d = diag(K);
[a,I(j+1)] = max(d);
while a > eta
  j = j + 1;
  nu(j) = sqrt(a);
  for i = 1:ell
    R(j,i) = (K(I(j),i) - R(:,i)’*R(:,I(j)))/nu(j);
    d(i) = d(i) - R(j,i)^2;
  end
  [a,I(j+1)] = max(d);
end
T = j;
R = R(1:T,:);
% for new example with vector of inner products
% k of size ell x 1 to compute new features r
r = zeros(T, 1);
for j=1:T
  r(j) = (k(I(j)) - r’*R(:,I(j)))/nu(j);
end
```

Code Fragment 5.4. Matlab code for performing incomplete Cholesky decomposition or dual partial Gram–Schmidt orthogonalisation.

variate random variables using a measure known as the *covariance* defined to be the expectation of their product

$$\operatorname{cov}(x, y) = \mathbb{E}_{xy}[xy].$$

Frequently, raw feature components from different sensors are difficult to compare because the units of measurement are different. It is possible to compensate for this by standardising the features into unitless quantities. The standardisation $\hat{x}$ of a feature x is

$$\hat{x} = \frac{x - \mu_x}{\sigma_x},$$

where μ_x and σ_x are the mean and standard deviation of the random variable x. The measure $\hat{x}$ is known as the standard score. The covariance

$$\mathbb{E}_{\hat{x}\hat{y}}[\hat{x}\hat{y}]$$