Direct Learning: Linear Rule for Classification Problem
We consider two class problem: \( Y \in \{-1, 1\} \) and recall that the Bayes rule for the classification is

\[
f^*(x) = \text{sign} \left\{ P(Y = 1 | X = x) - 1/2 \right\}
\]

so it is natural to estimate \( P(Y = 1 | X = x) \) directly using training data.

Commonly used model is a logistic regression model by assuming

\[
P(Y = 1 | X) = \frac{\exp \{ \beta_0 + X^T \beta \}}{1 + \exp \{ \beta_0 + X^T \beta \}}
\]

then the resulting prediction rule is \( \text{sign} \left\{ \beta_0 + x^T \beta \right\} \).

Similar to the linear rule for regression problem, shrinkage can also be introduced to improve prediction and the regularization can be applied as before.
GLasso

- It is a Lasso-type regression with the likelihood function for the logistic model:

\[
\min_{\beta} \sum_{i=1}^{n} \left\{ -Y_i (\beta_0 + X_i^T \beta) + \log(1 + \exp\{\beta_0 + X_i^T \beta\}) \right\} + \lambda \| \beta \|_{L_1}.
\]

- It is a convex optimization problem so can be solved using Newton-method.

- Or, we can apply iteratively weighted least-square method (the algorithm for generalized linear model) with a Lasso penalty to solve the problem.

Donglin Zeng, Department of Biostatistics, University of North Carolina
Perceptron model for classification

- A more direct method (nothing to do with statistics) is to directly search a hyperplane separating two class data (perceptron model).
- This algorithm aims to find a separating hyperplane by minimizing the distance of misclassified points to the decision boundary:

\[- \sum_{i=1}^{n} Y_i (\beta_0 + X_i^T \beta).\]

- Stochastic gradient decent (popular in modern big data analysis) is used for optimization:

\[
\begin{pmatrix}
\beta \\
\beta_0
\end{pmatrix} \leftarrow \begin{pmatrix}
\beta \\
\beta_0
\end{pmatrix} + \rho \begin{pmatrix}
Y_i X_i \\
Y_i
\end{pmatrix},
\]

where \(\rho\) is the learning rate and \(i\) is a randomly selected subject. The algorithm may not have unique solutions and may cycle solutions when data are not separable.

Donglin Zeng, Department of Biostatistics, University of North Carolina
The idea behind discriminant analysis is to compare the distributions of feature variables for each class and identify the best rule to discriminate these distributions.

Let \( f_k(x) \) be the density of \( X \) in \( Y = k \) and \( \pi_k \) is the prevalence of \( Y = k, k \in \{-1, 1\} \). Clearly,

\[
P(Y = 1|X = x) = \frac{f_1(x)\pi_1}{f_1(x)\pi_1 + f_0(x)\pi_0}.
\]

Thus, the Bayes rule is

\[
I \left\{ \log f_1(x) - \log f_0(x) + \log \frac{\pi_1}{\pi_0} > 0 \right\}.
\]
Discriminant analysis under multivariate normality

- Assume $f_k(x) \sim N(\mu_k, \Sigma)$ for $k \in \{-1, 1\}$ (homogeneous variance). The prediction rule becomes the sign of

$$x^T \Sigma^{-1}(\mu_1 - \mu_0) - \frac{1}{2} \mu_1^T \Sigma^{-1} \mu_1 + \frac{1}{2} \mu_0^T \Sigma^{-1} \mu_0 + \log \frac{\pi_1}{\pi_0}.$$ 

- The decision boundary is a linear function of $X$ so this analysis is called linear discriminant analysis.

- Assume $f_k(x) \sim N(\mu_k, \Sigma_k)$. Then prediction rule becomes the sign of

$$-\frac{1}{2} x^T \Sigma_1^{-1} x + \frac{1}{2} x^T \Sigma_0^{-1} x + x^T (\Sigma_1^{-1} \mu_1 - \Sigma_0^{-1} \mu_0)$$

$$-\frac{1}{2} \mu_1^T \Sigma_1^{-1} \mu_1 + \frac{1}{2} \mu_0^T \Sigma_0^{-1} \mu_0 + \log \frac{\pi_1}{\pi_0}.$$ 

- The decision boundary is a quadratic function of $X$ so this analysis is called quadratic discriminant analysis.
Example of LDA

**FIGURE 4.5.** The left panel shows three Gaussian distributions, with the same covariance and different means. Included are the contours of constant density enclosing 95% of the probability in each case. The Bayes decision boundaries between each pair of classes are shown (broken straight lines), and the Bayes decision boundaries separating all three classes are the thicker solid lines (a subset of the former). On the right we see a sample of 30 drawn from each Gaussian distribution, and the fitted LDA decision boundaries.
Example of QDA

**FIGURE 4.6.** Two methods for fitting quadratic boundaries. The left plot shows the quadratic decision boundaries for the data in Figure 4.1 (obtained using LDA in the five-dimensional space $X_1, X_2, X_1 X_2, X_1^2, X_2^2$). The right plot shows the quadratic decision boundaries found by QDA. The differences are small, as is usually the case.
Extension to Nonlinear Rule
Nonlinear prediction rules can be obtained if we include high-order interactions, or replace $X$ with some basis functions in the feature space.

Commonly used and flexible basis functions includes splines and wavelets.

Splines are piecewise polynomials with varied degrees and differentiability from location to location.

There is a rich set of theories about functional approximation of B-splines in Sobolev and Besov spaces.

Splines for a multivariate function are obtained through tensor products: $B_{k_1}(x_1) \otimes B_{k_2}(x_2) \otimes ...$
We need to pre-define a sequence of knots, degrees of polynomials and smoothness at knots.

Commonly used splines consist of B-splines and natural cubic splines, and B-splines have a simple algorithm to recursively calculate.

Splines can also be represented by polynomials and truncated polynomials, where truncations occur at interior knots:

\[ x^k, (x - \xi_1)^k, (x - \xi_2)^k, \ldots \]
B-splines:
Cox-deBoor Recursion

- Cox-deBoor Algorithm: defines the blending functions for spline curves (not limited to deg 3)
  - curves are weighted avgs of lower degree curves
- Let $B_{k,i}(t)$ denote the $i$-th blending function for a B-spline of degree $d$, then:

$$B_{k,0}(t) = \begin{cases} 1, & \text{if } t_k \leq t < t_{k+1} \\ 0, & \text{otherwise} \end{cases}$$

$$B_{k,i}(t) = \frac{t - t_k}{t_{k+d} - t_k} B_{k,d-1}(t) + \frac{t_{k+d+1} - t}{t_{k+d+1} - t_{k+1}} B_{k+1,d-1}(t)$$

- Knot Vector
  \{0.0, 0.0, 0.0, 3.0, 4.0, 5.0, 6.0, 7.0\}
- Several consecutive knots get the same value
- Changes the basis functions!
R code of constructing B-splines

```r
>library(splines)
> x=rnorm(1, 0, 1)
> x
[1] -0.2355063
> bs(x, knots=c(-0.5, 0, 0.5), Boundary.knots=c(-4, 4), degree=3, intercept=T)
```

The above command will produce 7 basis functions. Note the set of all knot points are
`c(-4, -4, -4, -4, -0.5, 0, 0.5, 4, 4, 4, 4)` and the number of basis functions is
`m - n - 1`, where `m` is the total number of knot points which is 11 in this example, and
`n` is the degree, and for this example it is 3. Note that the knots points are ordered, and in R
the entire set of knots are obtained by adding `(n + 1)` lower boundary knot and `(n + 1)` upper
boundary knot with the inner knot points. Let the cubic spline basis functions be \( N_{7,3}(x) \),

heart disease data. Here we explore nonlinearities in the functions using natural splines. The functional form of the model is

\[
\logit[\Pr(\text{chd}|X)] = \theta_0 + h_1(X_1)^T \theta_1 + h_2(X_2)^T \theta_2 + \cdots + h_p(X_p)^T \theta_p, \quad (5.6)
\]

where each of the \( \theta_j \) are vectors of coefficients multiplying their associated vector of natural spline basis functions \( h_j \).

We use four natural spline bases for each term in the model. For example, with \( X_1 \) representing \( \text{sbp} \), \( h_1(X_1) \) is a basis consisting of four basis functions. This actually implies three rather than two interior knots (chosen at uniform quantiles of \( \text{sbp} \)), plus two boundary knots at the extremes of the data, since we exclude the constant term from each of the \( h_j \).

**Figure 5.4.** Fitted natural-spline functions for each of the terms in the final model selected by the stepwise procedure. Included are pointwise standard-error bands. The rug plot at the base of each figure indicates the location of each of the sample values for that variable (jittered to break ties).
Smooth splines

- They are also piecewise polynomials as B-splines.
- However, instead of specifying knots in advance, smoothing spline approximation is obtained by minimizing the following penalized least squares:

\[
\sum_{i=1}^{n} (Y_i - f(X_i))^2 + \lambda \int \{f''(t)\}^2 dt.
\]

- Note that \(\lambda\) regularizes the smoothness of \(f(t)\).
- The solution is a linear combination of natural cubic splines whose knots are placed on \(X_1, ..., X_n\).
- Tuning is performed via Cp, CV or GCV:

\[
GCV = \frac{1}{n} \sum_{i=1}^{n} \left( \frac{Y_i - \hat{f}_\lambda(X_i)}{1 - S_\lambda(i, i)} \right)^2,
\]

where \(S_\lambda\) is some projection matrix on the space spanned by the cubic splines.
Wavelet approximation

- Wavelets can be used to approximate possibly irregular function/surfaces; to localize and identify such accumulation of "wavelets" (small waves) through both frequency and spatial resolution.

- They are extensively used in data compression, turbulence analysis, image & signal processing and statistical estimation.

- Comparatively, Fourier spectrum analysis is a global frequency decomposition approach. Alternative approximation such as blocked Fourier analysis, Splines, local polynomial approximation also have local adaptivity; however, wavelet analysis is more "elegant and mathematically consistent".
Wavelet construction

- We start with some father wavelet $\phi(x) = I(x \in (0, 1])$.

**Example: Haar system:**

- Consider $V_0 = \{ f \in L_2(R) : f = \sum_k c_k I_{(k,k+1]}(x) \}$ then 
  $\{ \phi_{0k} = \phi(x - k) \}$ is orthonormal bases (ONB) of $V_0$.
- Let $V_1 = \{ f(x) : f(x) = h(2x) \text{ for some } h \in V_0 \}$ then 
  $\{ \phi_{1k} = \sqrt{2} \phi_{0k}(2x) = \sqrt{2} \phi(2x - k) \}$ is ONB of $V_1$. 
- Generally, let $V_j = \{ f(x) : f(x) = h(2^j x) \text{ for some } h \in V_0 \}$ then 
  $\{ \phi_{jk} = 2^{j/2} \phi(2^j x - k) \}$ is ONB of $V_j$ for any $j$.
- $V_{-1} \subset V_0 \subset V_1 \subset ...; \bigcup_{j=0}^{\infty} V_j$ is dense in $L_2(R)$
- $V_{j+1} = V_j \oplus W_j$, where $\{ \psi_{jk} = 2^{j/2} \psi(2^j x - k) \}$ is ONB of $W_j$
- $\psi(x) = -I_{[0,1/2]}(x) + I_{[1/2,1]}(x)$ (mother wavelet).
First,

\[ L_2(R) = V_j \oplus W_j \oplus W_{j+1} \oplus W_{j+2} \cdots. \]

Second,

\[ V_0 = W_{-1} \oplus W_{-2} \cdots \oplus W_{-j} \oplus V_{-j}. \]

In other words, any \( f(x) \in L_2(R) \) can be recovered using lower resolution projection \( (V_j) \) and additional details from higher resolutions \( W_j, W_{j+1}, \cdots \). This is called the multiresolution analysis of \( f \).

Other choices of father wavelets: Daubechies wavelets, Coiflet wavelets, Symmlets wavelets (smoother than Haart wavelets).
Wavelet 1D demo

Analyzed signal.

Discrete Transform, absolute coefficients.

Absolute Values of C_{a,b} Coefficients for a = 1 2 3 4 5 ...
Wavelet 2D demo

Reconstructed image $a_0$.

- $X$
- $a_1$
- $h_1$
- $v_1$
- $d_1$
- $a_2$
- $h_2$
- $v_2$
- $d_2$

Donglin Zeng, Department of Biostatistics, University of North Carolina

Instruct $X$ from the wavelet
representation structure $\{c, s\}$.

= waverec2(c, s, 'db1');
Instead of using splines or wavelets as bases, the method based on RKHS looks for a kernel-based approximation.

(Kernel function) a function \( k : X \times X \to \mathbb{R} \) is called a positive and symmetric kernel on the feature space if

\[
\sum_{i=1}^{n} \sum_{j=1}^{n} a_i a_j k(x_i, x_j) \geq 0
\]

for any \( a_1, \ldots, a_n \in \mathbb{R} \) and \( x_1, \ldots, x_n \in X \), and \( k(x, y) = k(y, x) \).

Recall the definition of Hilbert space: it is a linear space equipped with an inner product, \((\mathcal{H}, \langle, \rangle)\), where \( \langle f, g \rangle = \langle g, f \rangle, \langle f, f \rangle \geq 0 \) and \( \langle f, g \rangle \) is linear in \( f \). Furthermore, \( \mathcal{H} \) is complete under the metric defined as \( \|f - g\| = \sqrt{\langle f - g, f - g \rangle} \).
Construction of RKHS

We consider all functions of form
\[ \sum_{i=1}^{n} \alpha_i k(x, x_i), \quad \alpha_1, \ldots, \alpha_n \in \mathbb{R}, x_1, \ldots, x_n \in \mathcal{X}. \]

For any functions \( f = \sum_{i=1}^{n} \alpha_i k(x, x_i) \) and \( g = \sum_{j=1}^{m} \beta_j k(x, y_j) \)

in this space, we define an inner product

\[ \langle f, g \rangle = \sum_{i=1}^{n} \sum_{j=1}^{m} \alpha_i \beta_j k(x_i, y_j) \]

thus, the resulting norm is \( \|f\|_{\mathcal{H}} = \sqrt{\langle f, f \rangle} \).

We complete this space to obtain a Hilbert space called the RKHS associated with \( k(x, y) \).
Basic properties of RKHS

- for any \( f(x) \in \mathcal{H}, f(x) = \langle f(\cdot), k(\cdot, x) \rangle \) (reproducing).
- In particular, \( k(x, y) = \langle k(\cdot, x), k(\cdot, y) \rangle \) so \( k(\cdot, x) \) is called a canonical feature map, which can be considered as some transformation mapping every \( x \) to a function \( k(\cdot, x) \).
- (Mercer’s theorem) Let \( \mathcal{X} \) be compact. Then there exists a series of orthogonal bases in \( \mathcal{H} \), denoted by \( e_1(x), e_2(x), \ldots \) such that
  \[
  k(x, y) = \sum_{i=1}^{\infty} \lambda_i e_i(x)e_i(y)
  \]
  and \( \lambda_i = \|e_i(x)\|_{\mathcal{H}}^{-1} \).
- This is similar to eigenvalue decomposition (spectral decomposition).
Examples of RKHS

- Linear kernel:  
  \[ k(x, y) = x^T y \] 
  then \( \mathcal{H} \) consists of all linear functions of \( x \).

- Polynomial kernel: 
  \[ k(x, y) = (1 + x^T y)^m \] 
  for \( m \geq 0 \) then \( \mathcal{H} \) contains the function of polynomial order \( m \).

- Gaussian RBF kernel: 
  \[ k(x, y) = \exp\left\{ -\|x - y\|^2 / \sigma^2 \right\} \].

- Both polynomial kernel and Gaussian RBF kernel generate RKHS with nonlinear smooth functions.
More on RKHS associated with Gaussian RBF

- This RKHS is the most commonly used when fitting a nonlinear prediction rule (note that the RKHS does not contain constant).
- Its ONS consists of $e_{n_1}(x_1) \otimes ... \otimes e_{n_p}(x_p)$, where

$$e_n(x) = \sqrt{2^n \sigma^2} x^n e^{-x^2 / \sigma^2}.$$ 

- For any function in $g \in L_q(\mu) (q > 1)$, when $\sigma$ is small enough, there is a function in $\mathcal{H}$ close to $g$: $\int k(x, y)g(y)d\mu(y)$ (Gaussian mollifier).
Why RKHS is useful

- When $k(x, y)$ is nonlinear, the corresponding $\mathcal{H}$ contains nonlinear functions.
- For some choice of kernel (in particular, Gaussian RBF), the space $\mathcal{H}$ can approximate any arbitrary function when certain tuning parameter is small enough. In other words, some RKHS has universal approximation property.
- More important, the following **representer theorem** holds:
  - Let $\mathcal{H}$ be a RKHS associated with a kernel function $k(x, y)$. We consider $f$ (assumed it exists) minimizing the following penalized objective function
    $$G(f(x_1), ..., f(x_n)) + Q(\|f\|_{\mathcal{H}}),$$
    where $Q$ is an increasing function. Then $f$ takes form
    $$f(x) = \sum_{i=1}^{n} \alpha_i k(x, x_i).$$
Proof of the representer theorem

Any $f$ in $\mathcal{H}$ can be written as the summation of one component in the linear space spanned by $\{k(x, x_i), i = 1, \ldots, n\}$ and a projection on its orthogonal complement, denoted by $g$. I.e., $f(x) = \sum_{i=1}^{n} \alpha_i k(x, x_i) + g(x)$. Thus, $<g, k(\cdot, x_i)> = 0$ for $i = 1, \ldots, n$.

By the reproducing property,

$$f(x_j) = <f, k(\cdot, x_j)> = \sum_{i=1}^{n} \alpha_k <k(\cdot, x_i), k(\cdot, x_j)> + <g, k(\cdot, x_j)> = \sum_{i=1}^{n} \alpha_i k(x_i, x_j),$$

which only depends on $\alpha$’s, but not $g$.

On the other hand,

$$Q(\|f\|_\mathcal{H}) = Q(\sqrt{\sum_{i=1}^{n} \alpha_i k(x, x_i)\|_\mathcal{H}^2 + \|g\|_\mathcal{H}^2})$$

$$\geq Q(\sqrt{\sum_{i=1}^{n} \alpha_i k(x, x_i)\|_\mathcal{H}^2}).$$

Combined these results, we conclude $\sum_{i=1}^{n} \alpha_i k(x, x_i)$ always gives the smaller or equal value of the objective function, as compared to $f$. The proof is completed.

Donglin Zeng, Department of Biostatistics, University of North Carolina
The proof of the representer theorem implies that the minimization problem can be solved by solving

\[
\min_{\alpha_1, \ldots, \alpha_n} \left[ G \left( \sum_{i=1}^{n} \alpha_i k(x_i, x_1), \ldots, \sum_{i=1}^{n} \alpha_i k(x_i, x_n) \right) \right]
\]

\[
Q \left( \sqrt{\sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j k(x_i, x_j)} \right) .
\]

In other words, we only need the input of \( k(x_i, x_j), i, j = 1, \ldots, n \) to solve the problem. There are two advantages:

- it is dimension-free, i.e., between-subject distance matters instead of the dimensionality of the feature variable. Moreover, we only need to store \( n \times n \) matrix instead of \( n \times p \) feature matrix (e.g., \( x \) consists of millions of SNPs).
- there is a great flexibility to choose kernel function (e.g., multiple kernels, tensor kernels).