Mathematical Foundations of Data Sciences

Gabriel Peyré
CNRS & DMA, École Normale Supérieure

CHAPTER 13

Basics of Machine Learning

September 9, 2026

Machine learning seeks useful structure in data and predictors that remain accurate on new observations. We begin with dimensionality reduction and clustering, then develop regression and classification through empirical risk minimization, regularization, and kernel methods. Connections with inverse problems explain the role of model assumptions, while the final analysis of generalization balances approximation error against the statistical complexity of the chosen model class.

Imaging problems often concern the reconstruction or processing of an individual signal or image, whereas machine learning studies patterns across collections of observations. Their goals and performance measures differ, but both use tools such as linear models, regularization, and convex optimization.

13.1 Unsupervised Learning

In unsupervised learning, we observe n points $(x_i)_{i=1}^n$. The aim is to identify structure in these observations, for example to visualize them or group them into clusters. We assume for simplicity that the observations lie in Euclidean space, $x_i \in \mathbb{R}^p$, where p is the number of features. We store the observations as the rows of a matrix $X \in \mathbb{R}^{n\times p}$.

13.1.1 Dimensionality Reduction and PCA

Dimensionality reduction produces a compact representation of the data for visualization or subsequent analysis. By retaining informative features, it can also reduce computational cost and improve prediction. Principal component analysis (PCA) projects the data orthogonally onto the principal axes of the covariance matrix.

Presentation of the method. The empirical mean is defined as

$$\hat{m} \overset{\text{def.}}{=} \frac{1}{n}\sum_{i=1}^n x_i \in \mathbb{R}^p$$

and the empirical covariance as

$$\hat{C} \overset{\text{def.}}{=} \frac{1}{n}\sum_{i=1}^n (x_i - \hat{m})(x_i - \hat{m})^* \in \mathbb{R}^{p\times p}. \tag{13.1}$$

With the centered data matrix $\tilde{X} \overset{\text{def.}}{=} X - \mathbb{1}_n \hat{m}^*$, the covariance is $\hat{C} = \tilde{X}^*\tilde{X}/n$.

Suppose the points $(x_i)_i$ are i.i.d. with a finite second moment, and let $\mathbf{x}$ denote a random variable with their common distribution. The law of large numbers gives the following almost sure limits as $n \to +\infty$:

$$\hat{m} \to m \overset{\text{def.}}{=} \mathbb{E}(\mathbf{x}) \quad \text{and} \quad \hat{C} \to C \overset{\text{def.}}{=} \mathbb{E}((\mathbf{x}-m)(\mathbf{x}-m)^*). \tag{13.2}$$

Let μ be the distribution on $\mathbb{R}^p$ of $\mathbf{x}$. Then these limits can also be written as

$$m = \int_{\mathbb{R}^p} x \mathrm{d}\mu(x) \quad \text{and} \quad C = \int_{\mathbb{R}^p} (x-m)(x-m)^* \mathrm{d}\mu(x).$$

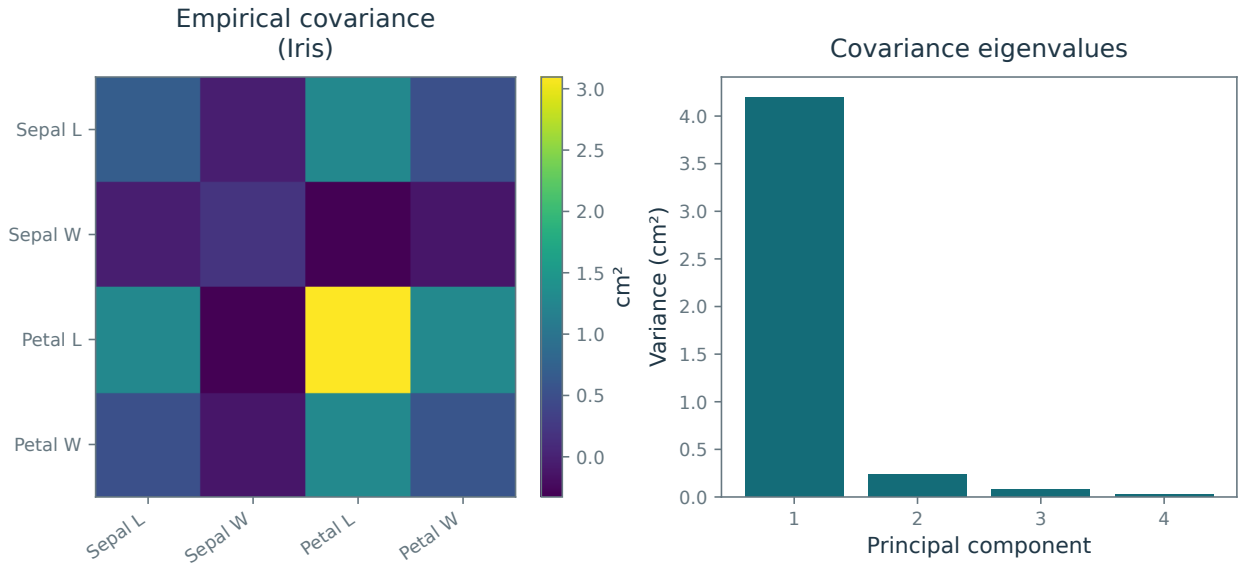


Figure 13.1. Empirical covariance of the Iris measurements and its eigenvalues.

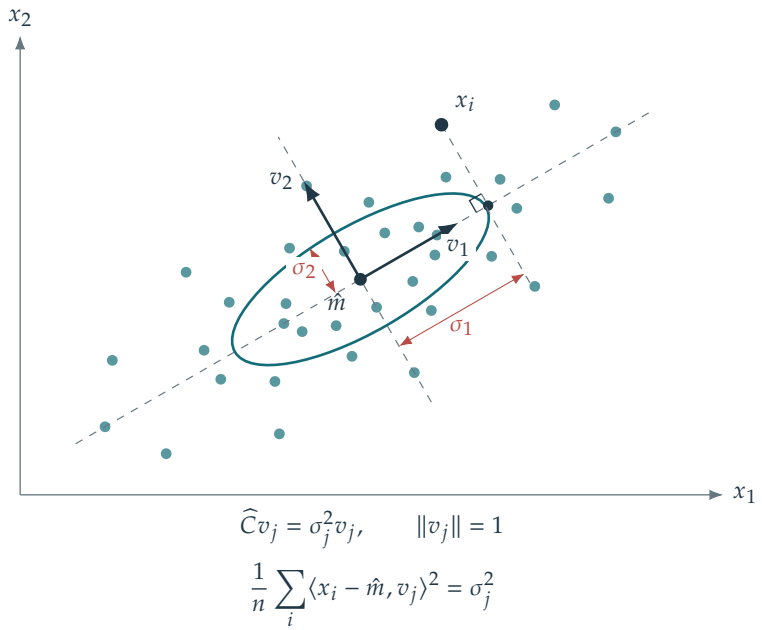


Figure 13.2. Principal directions of variation.

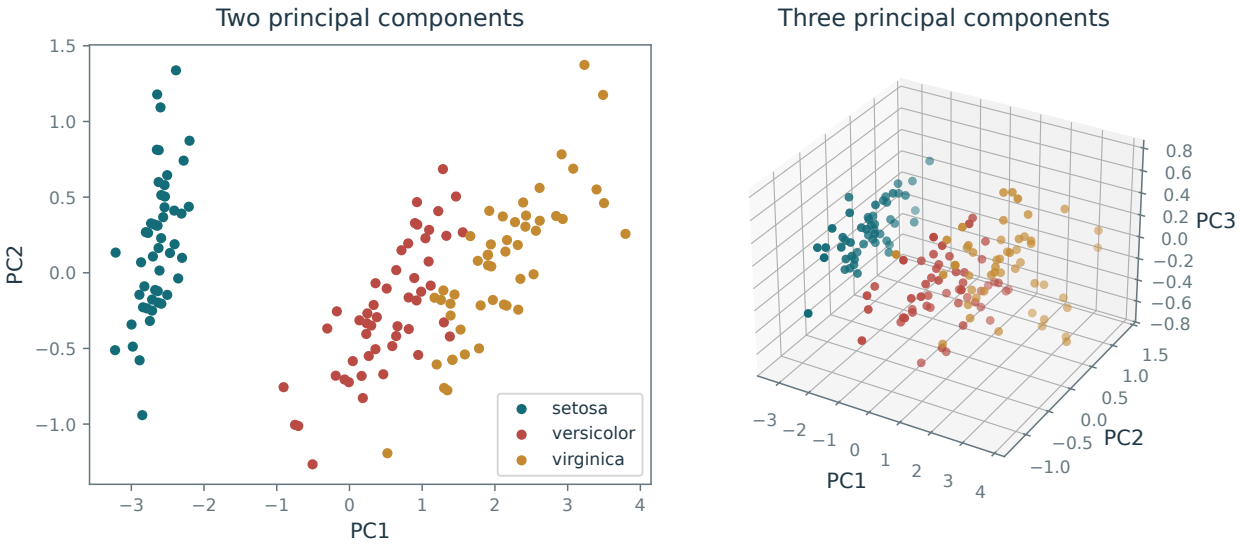


Figure 13.3. PCA projections of the data into two and three dimensions.

The PCA basis consists of the right singular vectors of the centered data matrix. Proposition 9.1 gives the reduced singular value decomposition

$$\tilde{X} = \sqrt{n} U \operatorname{diag}(\sigma) V^*$$

where $U \in \mathbb{R}^{n \times r}$ and $V \in \mathbb{R}^{p \times r}$ have orthonormal columns, and $r = \operatorname{rank}(\tilde{X}) \leqslant \min(n, p)$. Write $V = (v_k)_{k=1}^r$ for the orthonormal columns, which are eigenvectors of $\hat{C} = V \operatorname{diag}(\sigma^2) V^\top$, with $v_k \in \mathbb{R}^p$. These vectors describe the principal directions of variation in the point cloud $(x_i)_i$ in $\mathbb{R}^p$. We order the singular values as $\sigma_1 \geqslant \ldots \geqslant \sigma_r$, so the leading components account for the greatest variance.

Figure 13.1 displays an empirical covariance matrix and its spectrum $(\sigma_k^2)_k$ for the Iris dataset[1] introduced by Fisher. It contains 50 samples from each of three iris species: Iris setosa, Iris virginica, and Iris versicolor. The four features are sepal length, sepal width, petal length, and petal width.

To obtain a PCA embedding $x_i \in \mathbb{R}^p \mapsto z_i \in \mathbb{R}^d$ of dimension $d \leqslant p$, project onto the first d right singular vectors, extending the basis if $d > r$:

$$z_i \overset{\text{def.}}{=} (\langle x_i - \hat{m},\, v_k \rangle)_{k=1}^d \in \mathbb{R}^d. \tag{13.3}$$

The corresponding reconstruction is

$$\tilde{x}_i \overset{\text{def.}}{=} \hat{m} + \sum_{k=1}^{d} z_{i,k} v_k \in \mathbb{R}^p. \tag{13.4}$$

Thus $\tilde{x}_i = \operatorname{Proj}_{\tilde{T}}(x_i)$, where $\tilde{T} \overset{\text{def.}}{=} \hat{m} + \operatorname{Span}_{k=1}^d(v_k)$ is the affine reconstruction space.

Figure 13.3 shows an example of PCA for 2-D and 3-D visualization.

Optimality analysis. We prove that PCA minimizes the ℓ^2 reconstruction error among linear dimensionality-reduction methods. An optimal affine reconstruction space can be chosen to contain the empirical mean: for any fixed direction space, translating it to the mean cannot increase squared reconstruction error. After centering, we therefore assume $\hat{m} = 0$, hence $X = \tilde{X}$.

We recall that $X = \sqrt{n} U \operatorname{diag}(\sigma) V^\top$ with observations in rows, and $\hat{C} = X^\top X / n = V \operatorname{diag}(\sigma_i^2) V^\top$. We assume $1 \leqslant k \leqslant p$ and extend V to an orthogonal basis of $\mathbb{R}^p$ when necessary.

For the centered data, compare linear compression maps into dimension k followed by reconstruction:

$$\min_{R,S} \left\{ f(R,S) \overset{\text{def.}}{=} \sum_i \|x_i - R S^\top x_i\|_{\mathbb{R}^p}^2 \;;\; R, S \in \mathbb{R}^{p \times k} \right\} \tag{13.5}$$

Although $f(\cdot, S)$ and $f(R, \cdot)$ are convex separately, f is not jointly convex. Alternating minimization in R and S therefore need not reach a global minimizer. Nevertheless, a global minimizer has a simple explicit form.

[1] https://en.wikipedia.org/wiki/Iris_flower_data_set

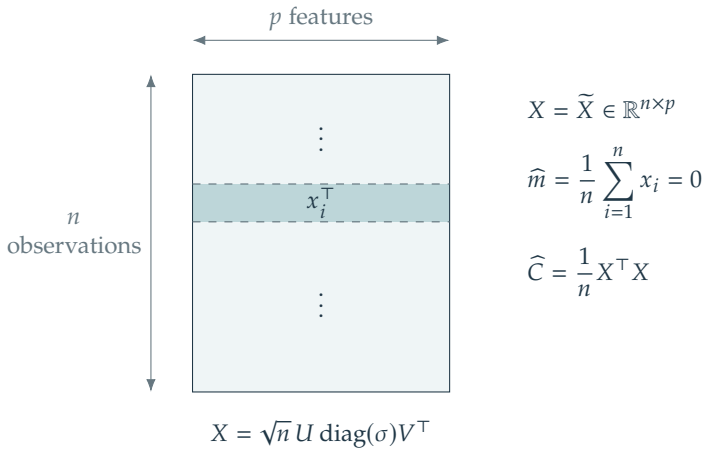


Figure 13.4. Rows of the centered data matrix are observations; columns are features.

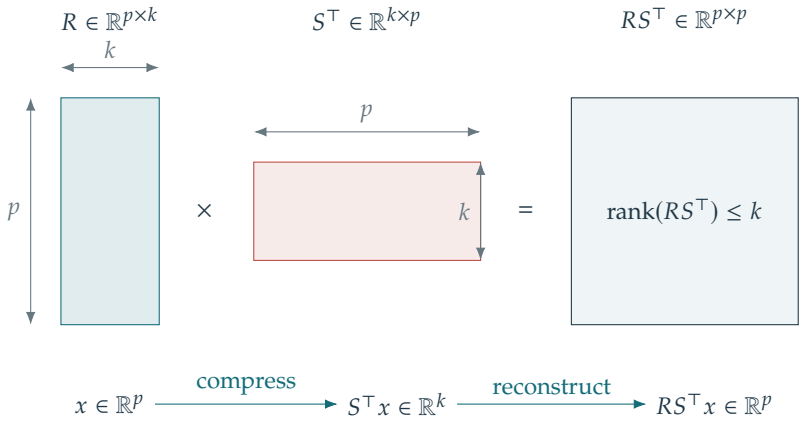


Figure 13.5. Linear compression followed by reconstruction.

Theorem 13.1. *A solution of* (13.5) *is* $S = R = V_{1:k} \overset{\text{def.}}{=} [v_1, \dots, v_k]$.

The compressor $S^\top$ and decompressor $R = S$ give precisely the PCA maps in (13.3) and (13.4).

We first prove that one can restrict attention to orthogonal projection matrices.

Lemma 13.2. *The minimization may be restricted to* $S = R$ *with* $S^\top S = \mathrm{Id}_{k\times k}$.

Proof. Fix an arbitrary pair (R, S). The matrix $RS^\top$ has rank $k' \leqslant k$. Choose $W \in \mathbb{R}^{p\times k'}$ whose columns form an orthonormal basis of $\mathrm{Im}(RS^\top)$, so that $W^\top W = \mathrm{Id}_{k'\times k'}$. Orthogonal projection gives

$$\operatorname*{argmin}_z \|x - Wz\|^2 = W^\top x$$

because the first-order optimality condition is $W^\top(Wz - x) = 0$. Hence, denoting $RS^\top x_i = Wz_i$ for some $z_i \in \mathbb{R}^{k'}$

$$f(R,S) = \sum_i \|x_i - RS^\top x_i\|^2 = \sum_i \|x_i - Wz_i\|^2 \geqslant \sum_i \|x_i - WW^\top x_i\|^2 \geqslant f(\tilde{W}, \tilde{W}).$$

where W has been extended to a matrix $\tilde{W} \in \mathbb{R}^{p\times k}$ with orthonormal columns and $\tilde{W}_{1:k'} = W$. □

Lemma 13.3. *Denoting* $C \overset{\text{def.}}{=} X^\top X = \sum_i x_i x_i^\top \in \mathbb{R}^{p\times p}$, *an optimal* S *is obtained by solving*

$$\max_{S\in\mathbb{R}^{p\times k}} \left\{ \mathrm{tr}(S^\top C S) \;;\; S^\top S = \mathrm{Id}_k \right\}.$$

Proof. By the previous lemma, restrict to $R = S$ with $S^\top S = \mathrm{Id}_k$. Expanding the residual gives

$$f(S,S) = \sum_i \|x_i - SS^\top x_i\|^2 = \sum_i \left(\|x_i\|^2 - 2x_i^\top SS^\top x_i + x_i^\top S(S^\top S)S^\top x_i \right).$$

Using that $S^\top S = \mathrm{Id}_k$, one has

$$f(S,S) = \sum_i \|x_i\|^2 - \sum_i \mathrm{tr}(S^\top x_i x_i^\top S) = \|X\|_{\mathrm{Fro}}^2 - \mathrm{tr}(S^\top C S).$$

The first term is independent of S, giving the stated maximization problem. □

The next lemma bounds the objective by a linear program. We then show that PCA attains this bound, proving optimality.

Lemma 13.4. *Denoting* $C = V\Lambda V^\top = n\hat{C}$ *with eigenvalues* $\lambda_1 \geqslant \cdots \geqslant \lambda_p \geqslant 0$ *(including zeros), one has*

$$\mathrm{tr}(S^\top C S) \leqslant \max_{\beta\in\mathbb{R}^p} \left\{ \sum_{i=1}^p \lambda_i \beta_i \;;\; 0 \leqslant \beta \leqslant 1, \sum_i \beta_i \leqslant k \right\} = \sum_{i=1}^k \lambda_i \tag{13.6}$$

with a maximizing vector $\beta = (1, \dots, 1, 0, \dots, 0)$.

Proof. Use the full orthogonal eigenbasis $V \in \mathbb{R}^{p\times p}$ introduced above, including eigenvectors corresponding to zero eigenvalues. Thus $V^\top V = VV^\top = \mathrm{Id}_p$. One has

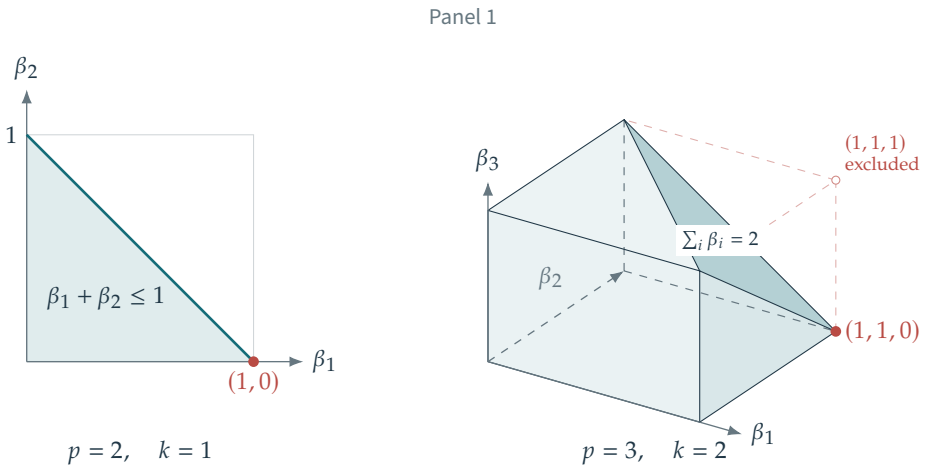
$$\mathrm{tr}(S^\top C S) = \mathrm{tr}(S^\top V\Lambda V^\top S) = \mathrm{tr}(B^\top \Lambda B) = \mathrm{tr}(\Lambda BB^\top) = \sum_{i=1}^p \lambda_i \|b_i\|^2 = \sum_i \lambda_i \beta_i$$

Here $B \overset{\text{def.}}{=} V^\top S \in \mathbb{R}^{p\times k}$. The vectors $(b_i)_{i=1}^p$, with $b_i \in \mathbb{R}^k$, are the rows of B, and $\beta_i \overset{\text{def.}}{=} \|b_i\|^2 \geqslant 0$. One has

$$B^\top B = S^\top VV^\top S = S^\top S = \mathrm{Id}_k,$$

so the columns of B are orthonormal. Consequently,

$$\sum_i \beta_i = \sum_i \|b_i\|^2 = \|B\|_{\mathrm{Fro}}^2 = \mathrm{tr}(B^\top B) = k.$$



Feasible region: $0 \le \beta_i \le 1$, $\sum_i \beta_i \le k$. Red dots: a maximizing vertex.

$$\lambda_1 \ge \cdots \ge \lambda_p \ge 0, \qquad \max \sum_i \lambda_i \beta_i = \sum_{i=1}^{k} \lambda_i$$

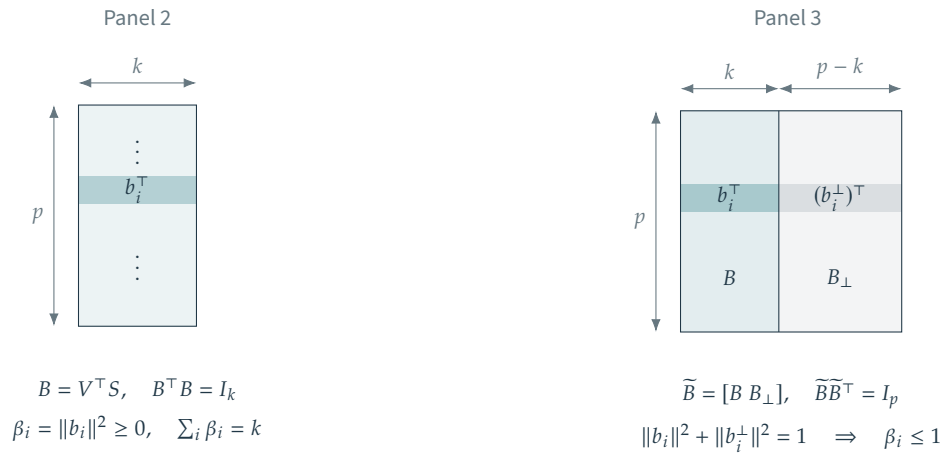
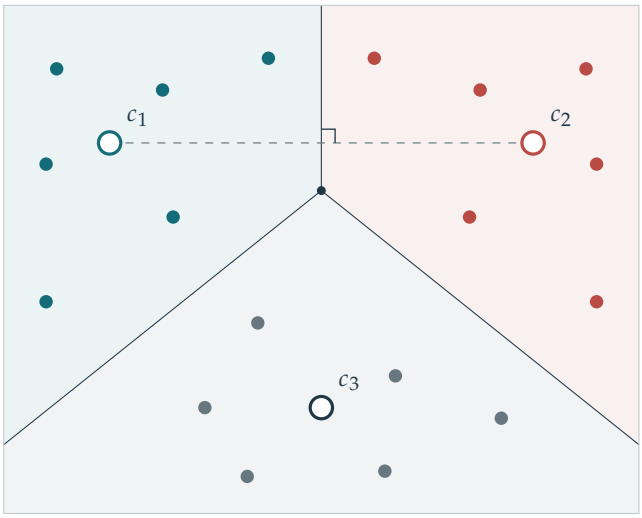


Figure 13.6. Panel 1: the linear-program bound in (13.6). Panel 2: matrix B. Panel 3: its orthogonal extension $\tilde{B}$.



$$V_j = \{x : \|x - c_j\| \le \|x - c_\ell\| \text{ for every } \ell\}$$

Dashed segment: two centroids. Its perpendicular bisector is their common cell edge.

Figure 13.7. k-means clusters according to Voronoi cells.

We extend the k columns of B into an orthogonal basis $\tilde{B} \in \mathbb{R}^{p\times p}$ such that $\tilde{B}\tilde{B}^\top = \tilde{B}^\top\tilde{B} = \mathrm{Id}_p$, so that

$$0 \leqslant \beta_i = \|b_i\|^2 \leqslant \|\tilde{b}_i\|^2 = 1$$

Thus $(\beta_i)_{i=1}^p$ is feasible for the linear program, so $\mathrm{tr}(S^\top CS)$ cannot exceed its maximum.

For any feasible β, use $\lambda_i \leqslant \lambda_k$ for $i > k$ and $\sum_i \beta_i \leqslant k$ to obtain

$$\sum_i \lambda_i\beta_i \leqslant \sum_{i\leqslant k} \lambda_i\beta_i + \lambda_k\left(k - \sum_{i\leqslant k}\beta_i\right) \leqslant \sum_{i\leqslant k}\lambda_i.$$

The choice $\beta_i = 1$ for $i \leqslant k$ and $\beta_i = 0$ otherwise attains this value. Figure 13.6 illustrates the argument. □

Proof of PCA optimality. For $S = V_{1:k} = [v_1, \dots, v_k]$, the eigenvector identities give $CS = V\Lambda V^\top V_{1:k} = V_{1:k}\,\mathrm{diag}(\lambda_i)_{i=1}^k$ and hence

$$\mathrm{tr}(S^\top CS) = \mathrm{tr}(S^\top S\,\mathrm{diag}(\lambda_i)_{i=1}^k) = \mathrm{tr}(\mathrm{Id}_k\,\mathrm{diag}(\lambda_i)_{i=1}^k) = \sum_{i=1}^k \lambda_i.$$

This attains the upper bound in Lemma 13.4, proving that S is optimal. □

13.1.2 Clustering and k-means

Clustering assigns a label $y_i \in \{1, \dots, k\}$ to each observation x_i. Observations with the same label form a cluster.

k-means One approach is to seek compact clusters by minimizing a measure of within-cluster dispersion. In Euclidean space, the k-means objective measures distances from observations to their centroids $c = (c_\ell)_{\ell=1}^k$, with $c_\ell \in \mathbb{R}^p$. Related formulations are possible in general metric spaces, although the centroid computations may be more difficult. The optimization problem is

$$\min_{(y,c)} \mathcal{E}(y,c) \overset{\text{def.}}{=} \sum_{\ell=1}^k \sum_{i:y_i=\ell} \|x_i - c_\ell\|^2.$$

The k-means algorithm alternates between updating the labels and updating the centroids, a form of block coordinate minimization. Initialize the centroids c, for example by choosing well-separated observations. We

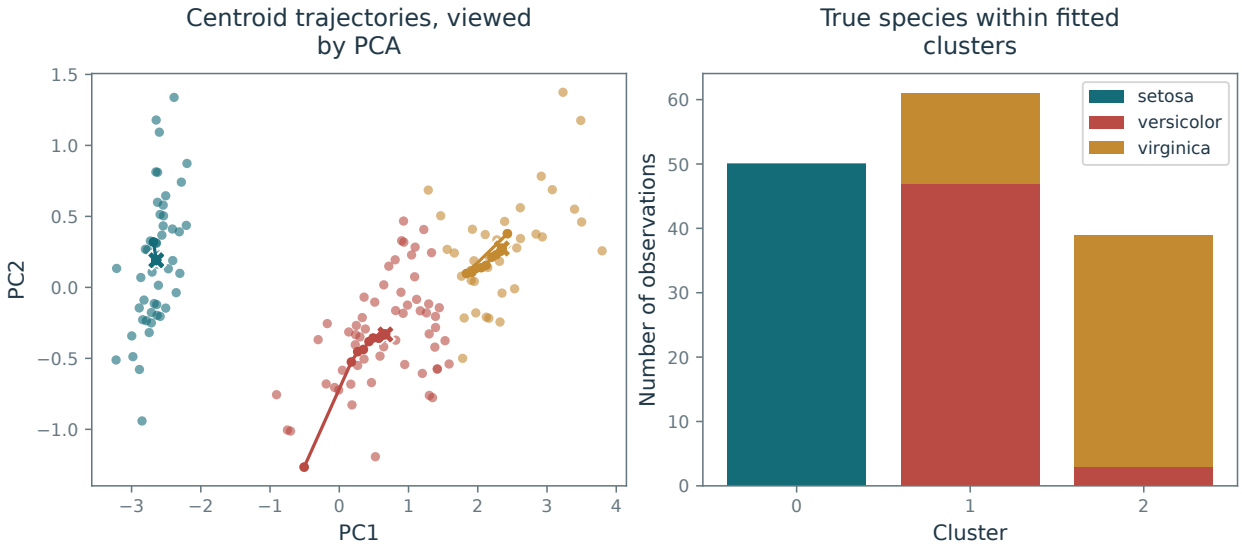


Figure 13.8. Left: iterations of the k-means algorithm. Right: class histograms after the k-means optimization.

discuss a more systematic initialization below. For fixed centroids c, minimize $y \mapsto \mathcal{E}(y, c)$ by assigning each observation to its nearest centroid:

$$\forall\, i \in \{1, \dots, n\}, \quad y_i \leftarrow \operatorname*{argmin}_{1 \leqslant \ell \leqslant k} \|x_i - c_\ell\|. \tag{13.7}$$

For fixed labels y, minimize $c \mapsto \mathcal{E}(y, c)$ by setting each centroid to the mean of its cluster:

$$\forall\, \ell \in \{1, \dots, k\}, \quad c_\ell \leftarrow \frac{\sum_{i : y_i = \ell} x_i}{|\{i\ ;\ y_i = \ell\}|} \tag{13.8}$$

If a cluster becomes empty, its centroid does not contribute to the objective for the current labels. One may leave it in place or reseed it at a data point before the next assignment step. The centroid formula applies only to nonempty clusters.

Each step decreases or preserves $\mathcal{E}$. With a fixed tie-breaking rule that retains existing assignments at ties, and without indefinite reseeding, a changed assignment strictly decreases the objective. There are finitely many label assignments, so the algorithm terminates after finitely many such changes.

Because the objective is nonconvex, k-means need not find globally optimal clusters. A reseeding heuristic moves centroids c_ℓ from redundant or low-contribution clusters to regions with high distortion. This can help the iterations escape a poor local configuration.

Figure 13.8 shows an example of k-means iterations on the Iris dataset.

k-means++ The result of k-means depends strongly on initialization. Well-separated initial centers tend to cover the data more effectively than tightly grouped centers.

The randomized k-means++ initialization has an approximation guarantee even before Lloyd iterations are applied. In practice, Lloyd iterations usually improve the resulting centers. First choose c_1 uniformly among the sample points. After choosing $c_1, \dots, c_\ell$, select $c_{\ell+1}$ from the sample according to a probability $\pi^{(\ell)}$ on $\{1, \dots, n\}$ proportional to the squared distance from the nearest selected center

$$\forall\, i \in \{1, \dots, n\}, \quad \pi_i^{(\ell)} \overset{\text{def.}}{=} \frac{d_i^2}{\sum_{j=1}^{n} d_j^2} \quad \text{where} \quad d_i \overset{\text{def.}}{=} \min_{1 \leqslant r \leqslant \ell} \|x_i - c_r\|.$$

Points far from the previously seeded centers are more likely to be chosen. If every $d_i = 0$, the current centers already achieve zero distortion and the procedure can stop.

The following theorem, due to David Arthur and Sergei Vassilvitskii, bounds the expected seeding cost by a logarithmic factor times the optimal cost. Computing a global optimum is NP-hard in general.

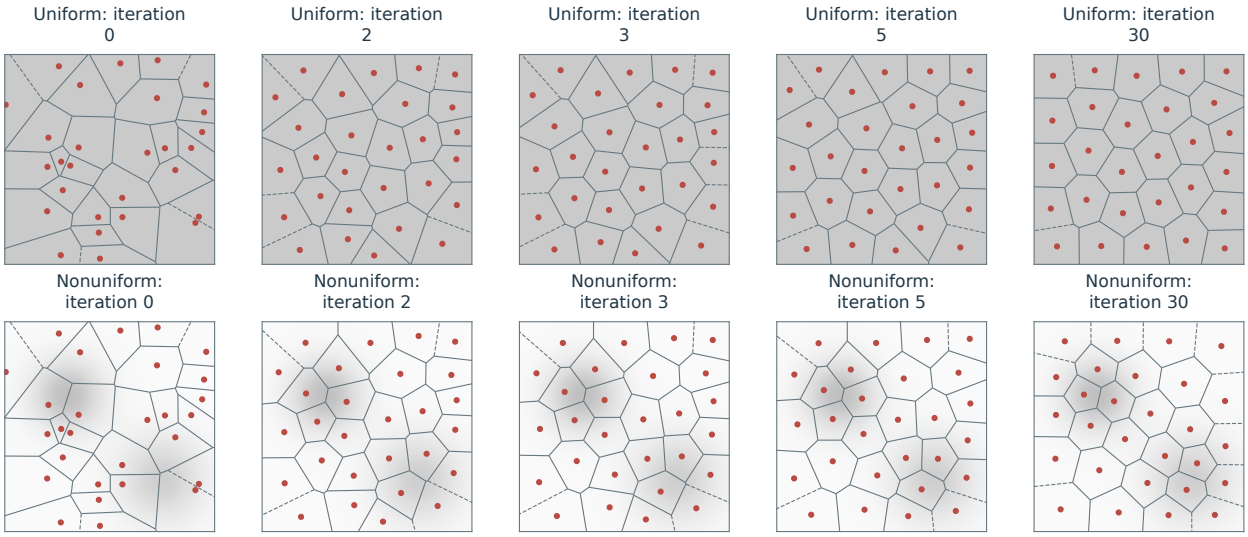


Figure 13.9. Continuous k-means (Lloyd) iterations $0, 2, 3, 5, 30$. The top row uses a uniform density; the bottom row uses the nonuniform density shown in grayscale.

Theorem 13.5. *Let $c^\star$ be the centroids selected by k-means++ and let $y^\star$ be their nearest-centroid labels, as in (13.7). Then*

$$\mathbb{E}(\mathcal{E}(y^\star, c^\star)) \leqslant 8(2 + \log(k)) \min_{(y,c)} \mathcal{E}(y, c),$$

where the expectation is over the random choices made during initialization.

Lloyd algorithm and continuous densities. The k-means iterations are also known as Lloyd's algorithm and are used in vector quantization for compression. The same alternating procedure applies to a probability measure μ on $\mathbb{R}^p$ with finite second moment. The energy to minimize becomes

$$\min_{(\mathcal{V},c)} \sum_{\ell=1}^{k} \int_{\mathcal{V}_\ell} \|x - c_\ell\|^2 \mathrm{d}\mu(x)$$

where $(\mathcal{V}_\ell)_\ell$ is a partition of the domain. Step (13.7) is replaced by the computation of a Voronoi cell

$$\forall\, \ell \in \{1, \ldots, k\}, \quad \mathcal{V}_\ell \stackrel{\text{def.}}{=} \{x \,;\, \forall\, \ell' \neq \ell, \|x - c_\ell\| \leqslant \|x - c_{\ell'}\|\}\,.$$

For distinct centroids, the Voronoi cells are polyhedra bounded by perpendicular bisectors. Resolve distance ties consistently to obtain a partition; this matters if μ assigns positive mass to a boundary. In low dimensions, computational geometry provides efficient algorithms for constructing the cells. Step (13.8) is then replaced by

$$\forall\, \ell \in \{1, \ldots, k\}, \quad c_\ell \leftarrow \frac{\int_{\mathcal{V}_\ell} x \mathrm{d}\mu(x)}{\int_{\mathcal{V}_\ell} \mathrm{d}\mu(x)}.$$

The centroid update applies to cells of positive μ-mass; zero-mass cells require a separate convention, as in the discrete algorithm. For a uniform planar density, hexagonal Voronoi patterns are characteristic of high-resolution quantization away from the boundary. They need not be exact finite-k minimizers on a bounded domain. Figure 13.9 displays two examples of Lloyd iterations on 2-D densities on a square domain.

13.2 Empirical Risk Minimization

We first describe empirical risk minimization, a framework shared by regression and classification. For classification, predictions may represent class probabilities rather than labels.

To make fitting $y_i \approx f(x_i)$ computationally tractable and obtain useful predictions, we restrict the admissible functions to a class of controlled complexity. The class may become richer as the sample size n grows, but its complexity must remain compatible with the available data.

13.2.1 Empirical Risk

Let $\mathcal{F}_n$ be a class of functions that may depend on the sample size. A common learning procedure is empirical risk minimization (ERM)

$$\hat{f} \in \operatorname*{argmin}_{f \in \mathcal{F}_n} \frac{1}{n} \sum_{i=1}^{n} L(f(x_i), y_i). \tag{13.9}$$

The loss function $L : \mathcal{Y}^2 \to \mathbb{R}^+$ measures prediction error for the chosen task. Classification and regression typically require different loss functions. We sometimes write $\hat{f}_n$ to emphasize the estimator's dependence on the sample size n.

13.2.2 Prediction and Consistency

For the analysis, assume the training pairs (x_i, y_i) are i.i.d. with distribution π on $\mathcal{X} \times \mathcal{Y}$. The corresponding population optimization problem is

$$\bar{f} \in \operatorname*{argmin}_{f \in \mathcal{F}_\infty} \int_{\mathcal{X}\times\mathcal{Y}} L(f(x), y) \mathrm{d}\pi(x, y) = \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim\pi}\big(L(f(\mathbf{x}), \mathbf{y})\big). \tag{13.10}$$

The desired consistency property is $\hat{f}_n \to \bar{f}$ as $n \to +\infty$. One way to express this is through expected prediction error over the training samples $(x_i, y_i)_i$:

$$E_n \overset{\text{def.}}{=} \mathbb{E}(\tilde{L}(\hat{f}_n(\mathbf{x}), \bar{f}(\mathbf{x}))) \longrightarrow 0.$$

Here the expectation includes both the test input $\mathbf{x}$, drawn from the marginal $\pi_{\mathcal{X}}$, and the n i.i.d. training pairs $(x_i, y_i) \sim \pi$ that determine $\hat{f}_n$. The comparison loss $\tilde{L}$ measures discrepancies between predictions in $\mathcal{Y}$; for example, we may take $\tilde{L} = L$. One can also study convergence in probability, i.e.

$$\forall\, \varepsilon > 0, \quad E_{\varepsilon,n} \overset{\text{def.}}{=} \mathbb{P}(\tilde{L}(\hat{f}_n(\mathbf{x}), \bar{f}(\mathbf{x})) > \varepsilon) \to 0.$$

These properties define consistency in expectation and in probability, respectively. A convergence rate gives an explicit upper bound on the decay of E_n or $E_{\varepsilon,n}$.

For $\tilde{L}(y, y') = |y - y'|^r$, convergence in expectation implies convergence in probability by Markov's inequality:

$$E_{\varepsilon,n} = \mathbb{P}(|\hat{f}_n(\mathbf{x}) - \bar{f}(\mathbf{x})|^r > \varepsilon) \leqslant \frac{1}{\varepsilon}\mathbb{E}(|\hat{f}_n(\mathbf{x}) - \bar{f}(\mathbf{x})|^r) = \frac{E_n}{\varepsilon}.$$

13.2.3 Parametric Approaches and Regularization

Instead of specifying $\mathcal{F}_n$ through a hard constraint, we can favor simple or regular functions through a penalty. A typical way to achieve this is by using a parametric model $y \approx f(x, \beta)$ where $\beta \in \mathcal{B}$ parametrizes the function $f(\cdot, \beta) : \mathcal{X} \to \mathcal{Y}$. The empirical risk minimization procedure (13.9) now becomes

$$\hat{\beta} \in \operatorname*{argmin}_{\beta \in \mathcal{B}} \frac{1}{n} \sum_{i=1}^{n} L(f(x_i, \beta), y_i) + \lambda_n J(\beta). \tag{13.11}$$

where J is a regularizer. For example, $J = \|\cdot\|_2^2$ controls parameter magnitude, while $J = \|\cdot\|_1$ promotes sparse coefficients and, in a linear feature model, feature selection. Here $\lambda_n > 0$ is a regularization parameter, and its asymptotic scaling must balance approximation and estimation; consistency often requires $\lambda_n \to 0$ at a controlled rate.

The population counterpart of (13.10) defines the parameter $\bar{\beta}$. A limiting estimator as $n \to +\infty$ then takes the form $\bar{f} = f(\cdot, \bar{\beta})$, with $\bar{\beta}$ satisfying

$$\bar{\beta} \in \operatorname*{argmin}_{\beta} \int_{\mathcal{X}\times\mathcal{Y}} L(f(x, \beta), y) \mathrm{d}\pi(x, y) = \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim\pi}\big(L(f(\mathbf{x}, \beta), \mathbf{y})\big). \tag{13.12}$$

Prediction vs. estimation risks. We may also ask how accurately $\hat{\beta}$ estimates $\bar{\beta}$, as measured by the parameter error $\|\hat{\beta} - \bar{\beta}\|$ for a chosen norm $\|\cdot\|$. Parameter estimation is generally more demanding than prediction: different parameters can produce similar predictions, especially when the model is poorly identified.

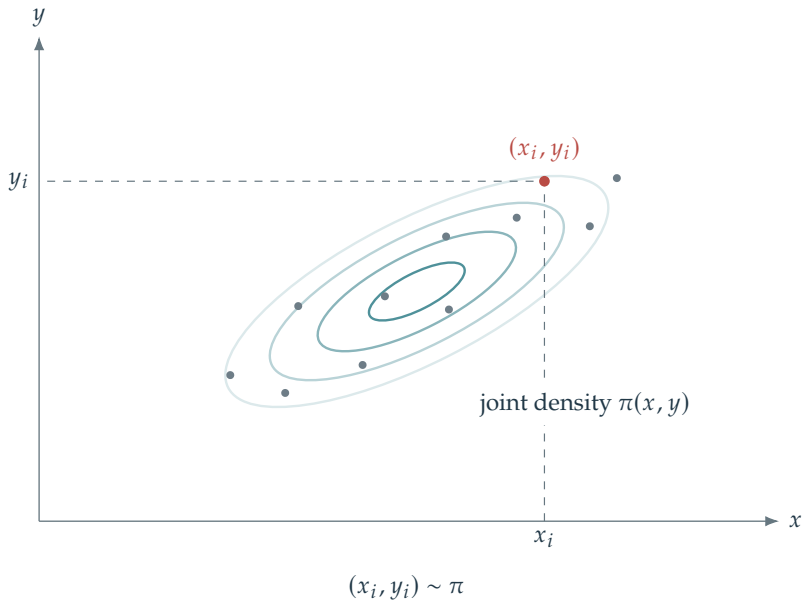


Figure 13.10. Probabilistic modeling.

13.2.4 Validation, Test Sets, and Cross-validation

The errors E_n and $E_{\varepsilon,n}$ cannot be evaluated directly because the population optimum $\bar{f}$ is unknown. To tune a parameter such as the regularization strength λ, we estimate the prediction risk $\mathbb{E}(L(\hat{f}(\mathbf{x}), \mathbf{y}))$ on data held out from training.

Use a second sample $(\bar{x}_j, \bar{y}_j)_{j=1}^{\bar{n}}$, called a validation set. In the statistical model, these observations are i.i.d. with distribution π and independent of the training sample. The validation risk is

$$R_{\bar{n}} = \frac{1}{\bar{n}} \sum_{j=1}^{\bar{n}} L(\hat{f}(\bar{x}_j), \bar{y}_j) \tag{13.13}$$

which converges to $\mathbb{E}(L(\hat{f}(\mathbf{x}), \mathbf{y}))$ for large $\bar{n}$. Minimizing $R_{\bar{n}}$ over hyperparameters (such as λ_n) is holdout validation. Cross-validation repeats training and validation across folds. A separate test set, untouched by hyperparameter selection, estimates the performance of the final selected procedure.

13.3 Supervised Learning: Regression

In supervised learning, the training data consist of pairs $(x_i, y_i) \in \mathcal{X} \times \mathcal{Y}$, with $\mathcal{X} = \mathbb{R}^p$ here for simplicity. We seek a function $f : \mathcal{X} \to \mathcal{Y}$ that captures the relationship $y_i \approx f(x_i)$ and predicts an output $f(x)$ for a new input x.

When $\mathcal{Y}$ is finite and discrete, the task is supervised classification, studied in Section 13.4. Binary classification uses $\mathcal{Y} = \{0, 1\}$; for example, in a medical application, $y_i = 0$ may indicate a healthy subject and $y_i = 1$ a subject with the condition of interest. When $\mathcal{Y}$ is continuous, typically $\mathcal{Y} = \mathbb{R}$, the task is regression.

13.3.1 Linear Regression

For regression, we specialize empirical risk minimization to $\mathcal{Y} = \mathbb{R}$ with the quadratic loss $L(y, y') = \frac{1}{2}|y - y'|^2$.

Nonlinear regression can be formulated using a dictionary of features, such as polynomials. This lifts the data into a higher-dimensional space and leads to the kernel methods of Section 13.5.

Least squares and conditional expectation. If f ranges over measurable functions and $\mathbb{E}(\mathbf{y}^2) < \infty$, a population minimizer $\bar{f}$ in (13.10) is the conditional mean of the response given the input. It is defined up to equality almost everywhere under the input distribution. Suppose for simplicity that π has density $\frac{\mathrm{d}\pi}{\mathrm{d}x\mathrm{d}y}$ with

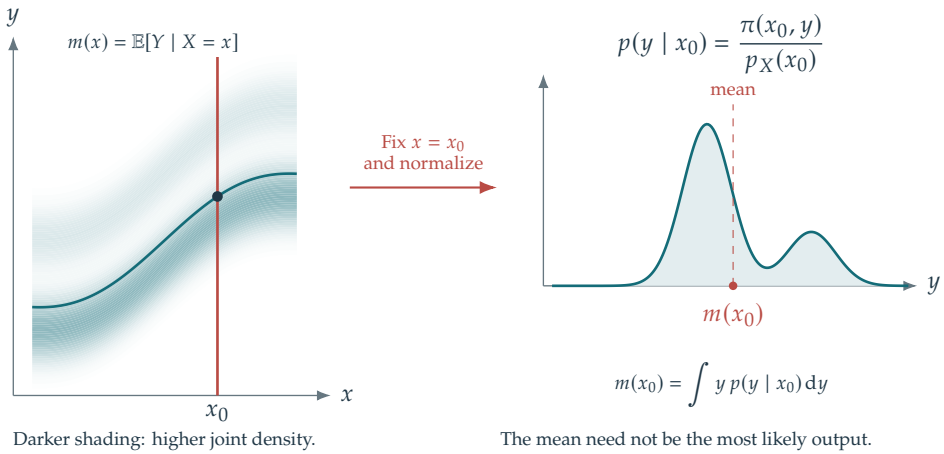


Figure 13.11. Conditional expectation.

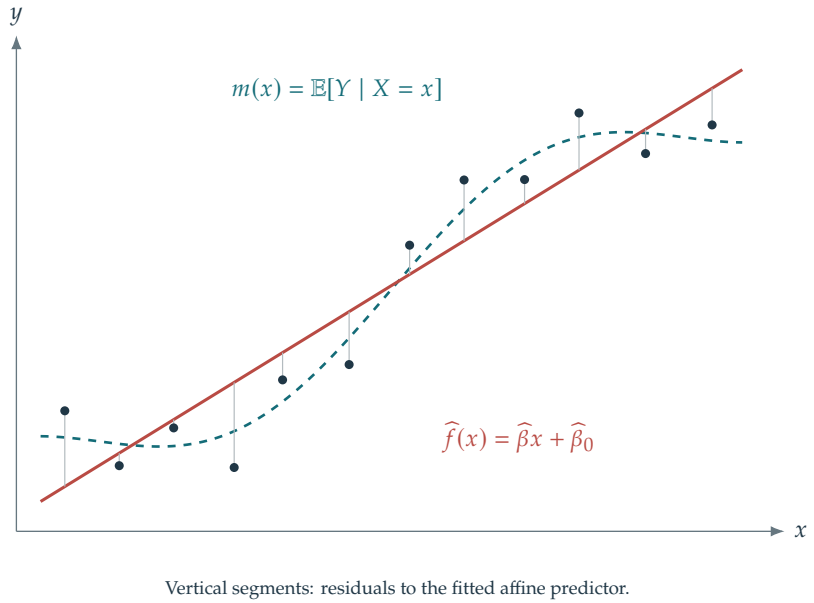


Figure 13.12. Linear regression.

respect to a product measure $\mathrm{d}x\mathrm{d}y$, such as Lebesgue measure. Then

$$\forall\, x \in \mathcal{X}, \quad \bar{f}(x) = \mathbb{E}(\mathbf{y}|\mathbf{x} = x) = \frac{\int_{\mathcal{Y}} y \frac{\mathrm{d}\pi}{\mathrm{d}x\mathrm{d}y}(x, y)\mathrm{d}y}{\int_{\mathcal{Y}} \frac{\mathrm{d}\pi}{\mathrm{d}x\mathrm{d}y}(x, y)\mathrm{d}y}$$

where $(\mathbf{x}, \mathbf{y})$ has law π and the formula applies where the denominator is positive.

If $\mathcal{X}$ and $\mathcal{Y}$ are discrete, let $\pi_{x,y}$ denote the probability of $(\mathbf{x} = x, \mathbf{y} = y)$. Then

$$\forall\, x \in \mathcal{X}, \quad \bar{f}(x) = \frac{\sum_y y\pi_{x,y}}{\sum_y \pi_{x,y}}$$

with no prescribed value when the marginal of π on $\mathcal{X}$ vanishes at x.

Replacing π by the empirical distribution averages responses at each observed input but leaves predictions unspecified elsewhere. Generalization therefore requires assumptions on regularity or model complexity.

Penalized linear models. A linear model constrains f to have the form $f(x, \beta) = \langle x, \beta \rangle$ with parameters $\beta \in \mathcal{B} = \mathbb{R}^p$. Affine predictors are included through the identity $\langle x, \beta \rangle + \beta_0 = \langle (x, 1), (\beta, \beta_0) \rangle$, by appending a constant coordinate to x to form $(x, 1)$. We therefore use the linear notation below.

Under the square loss, the regularized ERM (13.11) is conveniently rewritten as

$$\hat{\beta} \in \operatorname*{argmin}_{\beta \in \mathcal{B}} \frac{1}{2}\langle \hat{C}\beta, \beta\rangle - \langle \hat{u}, \beta\rangle + \lambda_n J(\beta) \tag{13.14}$$

where we introduced the empirical second-moment matrix (equal to the covariance (13.1) for centered data) and cross-moment vector

$$\hat{C} \overset{\text{def.}}{=} \frac{1}{n}X^*X = \frac{1}{n}\sum_{i=1}^{n} x_i x_i^* \quad \text{and} \quad \hat{u} \overset{\text{def.}}{=} \frac{1}{n}\sum_{i=1}^{n} y_i x_i = \frac{1}{n}X^*y \in \mathbb{R}^p.$$

As $n \to +\infty$, suitable moment assumptions on π give the following almost sure limits by the law of large numbers:

$$\hat{C} \to C \overset{\text{def.}}{=} \mathbb{E}(\mathbf{x}\mathbf{x}^*) \quad \text{and} \quad \hat{u} \to u \overset{\text{def.}}{=} \mathbb{E}(\mathbf{y}\mathbf{x}). \tag{13.15}$$

For appropriately chosen $\lambda_n \to 0$, the estimator can converge as $n \to +\infty$ to the population parameter

$$\bar{\beta} \in \operatorname*{argmin}_{\beta} \{J(\beta) \,;\, C\beta = u\}.$$

Problem (13.14) is equivalent to the regularized resolution of inverse problems (9.9), with $X/\sqrt{n}$ in place of Φ and $y/\sqrt{n}$ as the observation vector. Random design introduces sampling error into the operator as well: $\hat{C}$ estimates C. The typical scale is $1/\sqrt{n}$, in the sense that

$$\mathbb{E}(\|\hat{C} - C\|) = O(n^{-1/2}) \quad \text{and} \quad \mathbb{E}(\|\hat{u} - u\|) = O(n^{-1/2}),$$

assuming $\mathbb{E}(\mathbf{y}^4) < +\infty$ and $\mathbb{E}(\|\mathbf{x}\|^4) < +\infty$, which give finite second moments for $\mathbf{x}\mathbf{x}^*$ and $\mathbf{x}\mathbf{y}$. Using a linear predictor does not require a correctly specified model $\mathbf{y} = \langle \mathbf{x}, \beta\rangle + w$ with noise w independent of $\mathbf{x}$. The aim is to estimate the best linear predictor $\bar{\beta}$.

Extensions of Theorems 9.4, 11.9, and 11.11 can account for covariance-estimation error and yield prediction bounds of the form

$$\mathbb{E}(|\langle \hat{\beta}, \mathbf{x}\rangle - \langle \bar{\beta}, \mathbf{x}\rangle|^2) = O(n^{-\kappa})$$

and estimation rates of the form

$$\mathbb{E}(\|\hat{\beta} - \bar{\beta}\|^2) = O(n^{-\kappa'}),$$

under suitable source conditions involving C and u. Since the noise level is roughly $n^{-\frac{1}{2}}$, the ideal cases are when $\kappa = \kappa' = 1$, which is the so-called linear rate regime. Support-recovery guarantees can also be derived by extending Theorem 11.15. We now focus on quadratic regularization, a standard regression method. Sparse penalties such as ℓ^1 are useful when feature selection or a sparse model is appropriate; their statistical benefit depends on the data and assumptions. In imaging inverse problems, sparse regularization expresses prior structure of the object being reconstructed.

Ridge regression (quadratic penalization). For $J = \|\cdot\|^2/2$, the estimator (13.14) is obtained in closed form as

$$\hat{\beta} = (X^*X + n\lambda_n \mathrm{Id}_p)^{-1}X^*y = (\hat{C} + \lambda_n \mathrm{Id}_p)^{-1}\hat{u}. \tag{13.16}$$

This is often called ridge regression in the literature. The Woodbury identity gives the equivalent expression

$$\hat{\beta} = X^*(XX^* + n\lambda_n \mathrm{Id}_n)^{-1}y. \tag{13.17}$$

When $n \gg p$, formula (13.16) requires solving the smaller, $p \times p$ system. When $p \gg n$, formula (13.17) uses an $n \times n$ system and is preferable; it also extends to infinite-dimensional feature spaces.

Convergence to the minimum-norm population solution $\bar{\beta} = C^+u$ requires control of both the regularization bias and sampling error. The condition $\lambda_n \to 0$ alone is insufficient when small eigenvalues amplify noise. Here is one elementary bound that also applies in a Hilbert feature space.

Theorem 13.6. *Assume $\|\mathbf{x}\| \leqslant \kappa$ almost surely, $\mathbb{E}(\mathbf{y}^2) < \infty$, and $u = C\bar{\beta}$. Suppose*

$$\bar{\beta} = C^\gamma z, \qquad \|z\| \leqslant \rho, \qquad 0 < \gamma \leqslant 1. \tag{13.18}$$

Set $B^2 = \kappa^2\big(\mathbb{E}(\mathbf{y}^2) + \kappa^2\|\bar\beta\|^2\big)$. *For every* $\lambda > 0$,

$$\mathbb{E}\|\hat\beta - \bar\beta\|^2 \leqslant 2\rho^2\lambda^{2\gamma} + \frac{4B^2}{n\lambda^2}.$$

If $B, \rho > 0$, *choosing* $\lambda_n = (B/(\rho\sqrt{n}))^{1/(\gamma+1)}$ *gives*

$$\mathbb{E}\|\hat\beta - \bar\beta\|^2 \leqslant 6\rho^{2/(\gamma+1)}B^{2\gamma/(\gamma+1)}n^{-\gamma/(\gamma+1)}. \tag{13.19}$$

Proof. Let $\beta_\lambda = (C + \lambda\mathrm{Id})^{-1}u$. Spectral calculus gives $\|\beta_\lambda - \bar\beta\| \leqslant \rho\lambda^\gamma$ and $\|\beta_\lambda\| \leqslant \|\bar\beta\|$. Moreover,

$$\hat\beta - \beta_\lambda = (\hat C + \lambda\mathrm{Id})^{-1}\big((\hat u - u) - (\hat C - C)\beta_\lambda\big).$$

The inverse has norm at most $1/\lambda$. Independence and the second-moment bound yield $\mathbb{E}\|\hat\beta - \beta_\lambda\|^2 \leqslant 2B^2/(n\lambda^2)$. Combining the two errors with $\|a+b\|^2 \leqslant 2\|a\|^2 + 2\|b\|^2$ proves the first claim; the stated choice of λ_n gives the second. □

In finite dimensions, a source representation exists for every $\bar\beta \in \mathrm{Im}(C)$, but its radius ρ depends on the spectrum and on the target. In infinite dimensions, it is an additional regularity assumption. The bound displays no explicit feature dimension, although κ, B, and ρ may depend on it. Standard ridge regularization has bias saturation beyond source exponent $\gamma = 1$; stronger source assumptions alone do not justify extending this estimate to $\gamma = 2$.

13.4 Supervised Learning: Classification

For classification, the labels are discrete: $y_i \in \mathcal{Y} = \{1, \dots, k\}$. We study two standard methods: nearest neighbors and logistic classification. Both methods provide useful baselines for assessing more elaborate models. Nearest-neighbor methods also apply to regression.

13.4.1 Nearest-Neighbor Classification

The R-nearest-neighbor method (R-NN) classifies an input using the labels of its R nearest training observations. Increasing R reduces sensitivity to individual noisy labels, but excessive smoothing can obscure class boundaries. For consistency, a standard asymptotic choice has $R = R_n \to \infty$ and $R_n/n \to 0$. Nearest-neighbor classification is a useful baseline, especially when the feature dimension p is small.

The prediction $\hat f(x) \in \mathcal{Y}$ is the most frequent label among those R observations, with a fixed rule for breaking ties. The method is nonparametric: its representation depends on the training sample rather than on a fixed-dimensional parameter vector.

First compute the Euclidean distance from the query x to each training observation x_i. Sorting the distances generates an indexing σ (a permutation of $\{1, \dots, n\}$) such that

$$\|x - x_{\sigma(1)}\| \leqslant \|x - x_{\sigma(2)}\| \leqslant \ldots \leqslant \|x - x_{\sigma(n)}\|.$$

For a given R, one can compute the "local" histogram of classes around x

$$h_\ell(x) \overset{\text{def.}}{=} \frac{1}{R}\#\{i \in \{1, \dots, R\} : y_{\sigma(i)} = \ell\}.$$

The predicted class at x is an index attaining the largest histogram value:

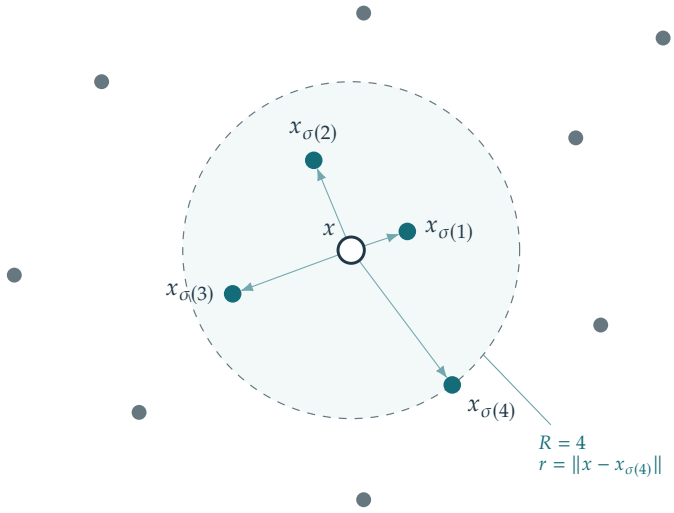
$$\hat f(x) \in \operatorname*{argmax}_\ell h_\ell(x).$$

Choose R by validation or cross-validation. For classification, the validation risk (13.13) typically uses the 0–1 loss, giving the fraction of misclassified observations

$$R_{\bar n} \overset{\text{def.}}{=} \frac{1}{\bar n}\sum_{j=1}^{\bar n}\delta(\bar y_j - \hat f(\bar x_j))$$

where $\delta(0) = 0$ and $\delta(s) = 1$ if $s \neq 0$. The method also applies to features in a general metric space in place of $\mathbb{R}^p$. Fast nearest-neighbor search can avoid explicitly sorting every distance.

Figure 13.14 shows the predicted class regions $\big\{x\,;\,\hat f(x) = \ell\big\}$, for $\ell = 1, \dots, k$, in a two-dimensional projection of the Iris dataset. Increasing R leads to smoother class boundaries.



$$\|x - x_{\sigma(1)}\| \leq \cdots \leq \|x - x_{\sigma(4)}\| < \|x - x_{\sigma(5)}\|$$

Figure 13.13. Nearest neighbors.

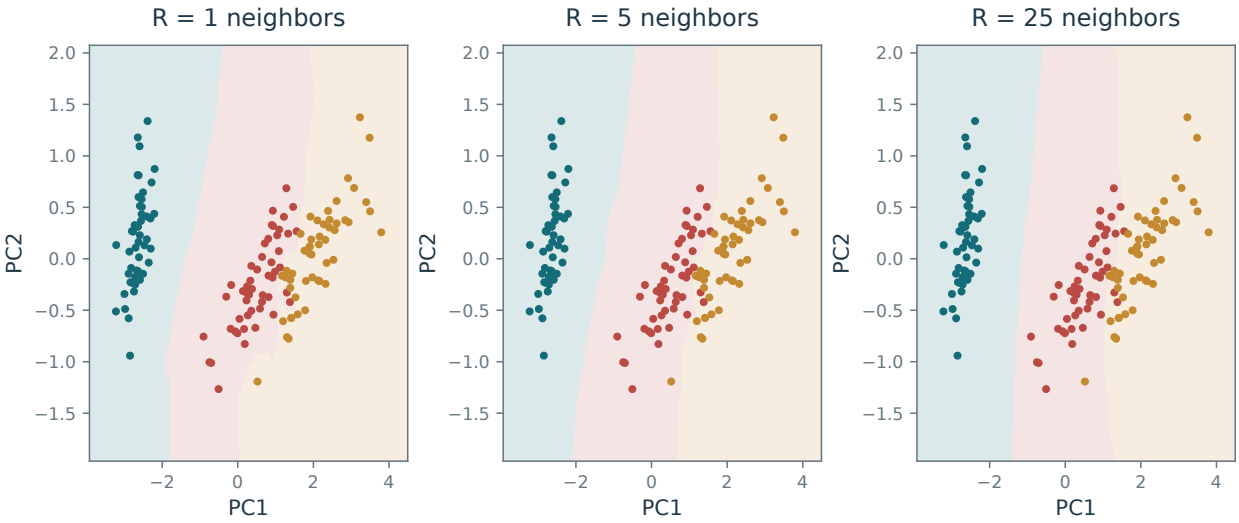


Figure 13.14. Classification boundaries produced by R-nearest neighbors.

13.4.2 Binary Logistic Classification

Logistic classification is a widely used baseline for binary and multiclass prediction. It outputs a probability for each class, providing a richer prediction than a class label alone. The accuracy of these probabilities must still be assessed on held-out data. The standard name for this model is logistic regression, even when the prediction task is classification.

Support vector machines provide another common approach. Their hinge loss is nonsmooth, and their scores are not probabilities without an additional calibration step. The choice between these methods depends on the task and evaluation criterion.

For the binary formulas below, use labels $y_i \in \mathcal{Y} = \{-1, 1\}$. In logistic classification, the prediction $f(\cdot, \beta) \in [0, 1]$ is a probability for the positive class rather than a class label. We retain the notation f for this probability predictor.

Approximate risk minimization. A linear score $\langle x, \beta \rangle$ defines the classifier $\operatorname{sign}(\langle x, \beta \rangle)$, with either class assigned when the score is zero. The empirical 0–1 objective counts classification errors, leading to the minimization problem

$$\min_{\beta} \sum_{i=1}^{n} \ell_0(-y_i \langle x_i, \beta \rangle) \tag{13.20}$$

where $\ell_0 = 1_{[0,+\infty)}$ counts zero margins as errors. Away from ties, misclassification corresponds to $\langle x_i, \beta \rangle$ and y_i having different signs, so that in this case $\ell_0(-y_i \langle x_i, \beta \rangle) = 1$ (and 0 otherwise for correct classification).

The loss ℓ_0 is nonconvex, and solving (13.20) globally is NP-hard in general. Convex upper bounds on ℓ_0 provide tractable surrogate objectives.

Two common surrogates are

$$\ell(u) = (1 + u)_+ \quad \text{and} \quad \ell(u) = \log(1 + \exp(u))/\log(2)$$

These are, respectively, the nonsmooth hinge loss used in support vector machines and the smooth logistic loss. The $1/\log(2)$ is just a constant which makes $\ell_0 \leqslant \ell$. AdaBoost uses the exponential loss $\ell(u) = e^u$. Least squares uses $\ell(u) = (1 + u)^2$. Its growth for large negative margins makes it a poor pointwise approximation of ℓ_0, although it can still give useful classifiers. Unregularized hinge-loss minimization is equivalent to a linear program with nonnegative slack variables:

$$\min_{u \geqslant 0, \beta} \left\{ \sum_i u_i \; ; \; y_i \langle x_i, \beta \rangle \geqslant 1 - u_i \text{ for all } i \right\}.$$

Logistic loss probabilistic interpretation. Logistic regression applies a sigmoid to the linear score from Section 13.3.1. The resulting probability predictor is nonlinear. The predicted probability of class +1 is

$$f(x, \beta) \overset{\text{def.}}{=} \theta(\langle x, \beta \rangle) \quad \text{where} \quad \theta(s) \overset{\text{def.}}{=} \frac{e^s}{1 + e^s} = (1 + e^{-s})^{-1}, \tag{13.21}$$

which is often called the "logit" model. Despite its simplicity, a linear classifier can be effective when the feature dimension p is large. With $s = \langle x, \beta \rangle$, the predicted probability of class -1 is $1 - f(x, \beta) = \theta(-s)$. The identity $\theta(-s) = 1 - \theta(s)$ makes the two probabilities sum to one. It does not require balanced class frequencies; an intercept can account for unequal prior frequencies.

For a nonzero parameter, $\beta/\|\beta\|$ determines the normal direction of the separating hyperplane, while $1/\|\beta\|$ controls the width of the probability transition. As $\|\beta\| \to +\infty$ along a fixed direction, the transition approaches a sharp decision boundary.

The predictor $f(x, \beta)$ is also a single-layer perceptron with a logistic (sigmoid) activation; see Chapter 17.

For conditionally independent labels $y_i \in \{-1, +1\}$, write $\bar{y}_i = (y_i + 1)/2 \in \{0, 1\}$, $s_i = \langle x_i, \beta \rangle$, and $p_i = \theta(s_i)$. The likelihood is

$$\mathbb{P}(\mathbf{y} = y_i \mid \mathbf{x} = x_i) = p_i^{\bar{y}_i} (1 - p_i)^{1 - \bar{y}_i}.$$

Thus the negative log-likelihood is

$$\sum_i \{-\bar{y}_i \log p_i - (1 - \bar{y}_i) \log(1 - p_i)\} = \sum_i \log(1 + \exp(-y_i s_i)).$$

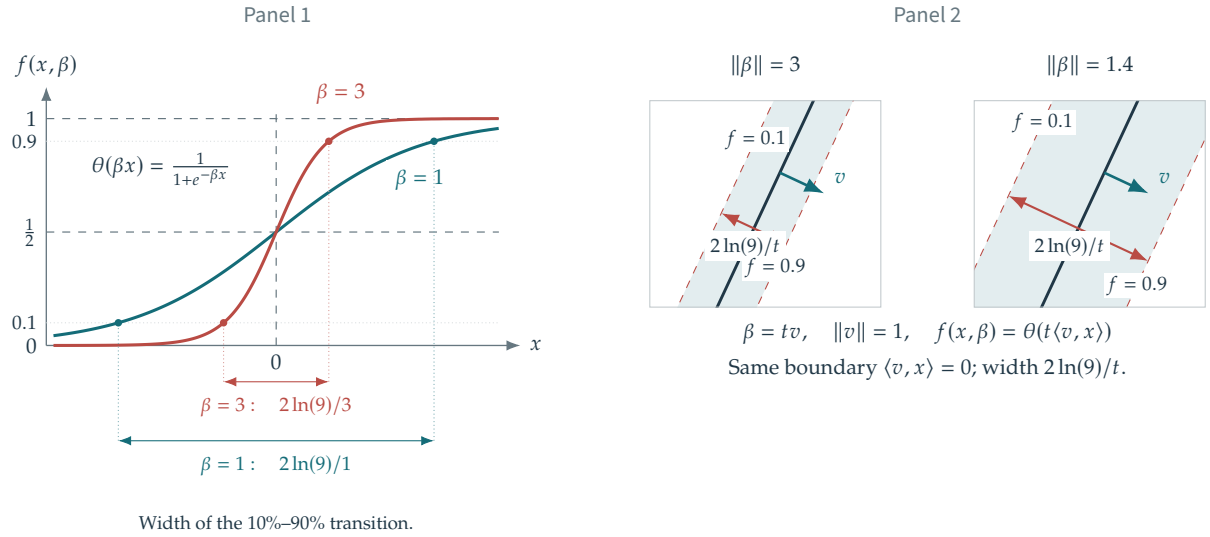


Figure 13.15. Logistic classification in one and two dimensions, showing how $\|\beta\|$ controls the width of the probability transition.

After division by the sample size, this gives the empirical-risk objective (13.11) with logistic loss:

$$\hat{\beta} \in \operatorname*{argmin}_{\beta \in \mathbb{R}^p} E(\beta) = \frac{1}{n}\sum_{i=1}^{n} L(\langle x_i, \beta\rangle, y_i) \tag{13.22}$$

where the logistic loss reads

$$L(s,y) \overset{\text{def.}}{=} \log(1+\exp(-sy)). \tag{13.23}$$

Problem (13.22) is a smooth convex minimization. If X has full column rank, E is strictly convex, so it has at most one minimizer. A finite minimizer need not exist for separable data. A positive quadratic penalty guarantees existence and uniqueness.

Figure 13.16 compares the 0–1, logistic, and hinge losses.

Gradient descent method. Write the objective as

$$E(\beta) = \mathcal{L}(X\beta, y) \quad \text{where} \quad \mathcal{L}(s,y) = \frac{1}{n}\sum_i L(s_i, y_i),$$

Its gradient is

$$\nabla E(\beta) = X^* \nabla \mathcal{L}(X\beta, y) \quad \text{where} \quad \nabla \mathcal{L}(s,y) = -\frac{y}{n} \odot \theta(-y \odot s),$$

where $\odot$ is the pointwise multiplication operator, i.e. `.*` in Matlab. Once $\beta^{(\ell=0)} \in \mathbb{R}^p$ is initialized (for instance at 0_p), one step of gradient descent (14.13) reads

$$\beta^{(\ell+1)} = \beta^{(\ell)} - \tau_\ell \nabla E(\beta^{(\ell)}).$$

Figure 13.17 illustrates the method on samples from a Gaussian mixture, whose overlap is controlled by an offset ω. Overlaying the data on the probability map highlights the separating hyperplane $\{x \,;\, \langle \beta, x\rangle = 0\}$.

13.4.3 Multiclass Logistic Classification

For multiclass logistic classification, introduce one weight vector for each of the k classes. Collect the vectors $\beta = (\beta_\ell)_{\ell=1}^k$ as columns of $\beta \in \mathbb{R}^{p\times k}$.

For an input $x \in \mathbb{R}^p$, convert the class scores into probabilities using the softmax model:

$$f(x,\beta) = \left(\frac{e^{\langle x, \beta_\ell\rangle}}{\sum_m e^{\langle x, \beta_m\rangle}}\right)_\ell \tag{13.24}$$

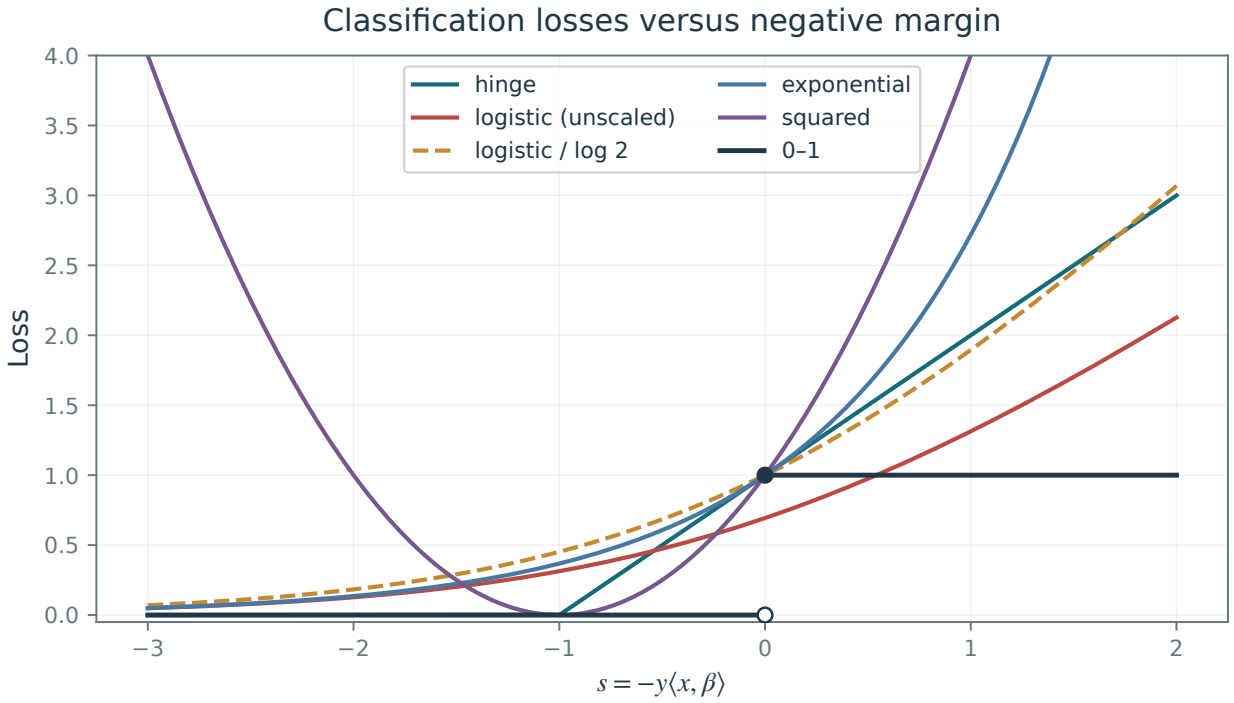


Figure 13.16. Classification losses as functions of the negative margin $s = -y\langle x, \beta\rangle$, with zero margin counted as an error. The unscaled logistic loss need not upper-bound the 0–1 loss; division by log 2 gives the upper-bound normalization used above.

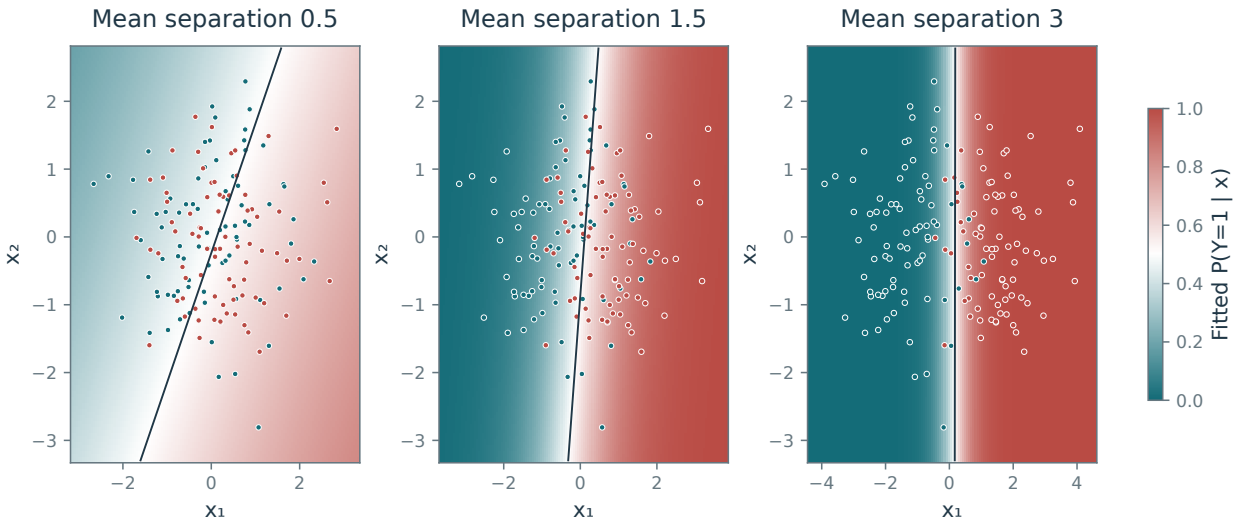


Figure 13.17. Effect of class separation on predicted classification probabilities.

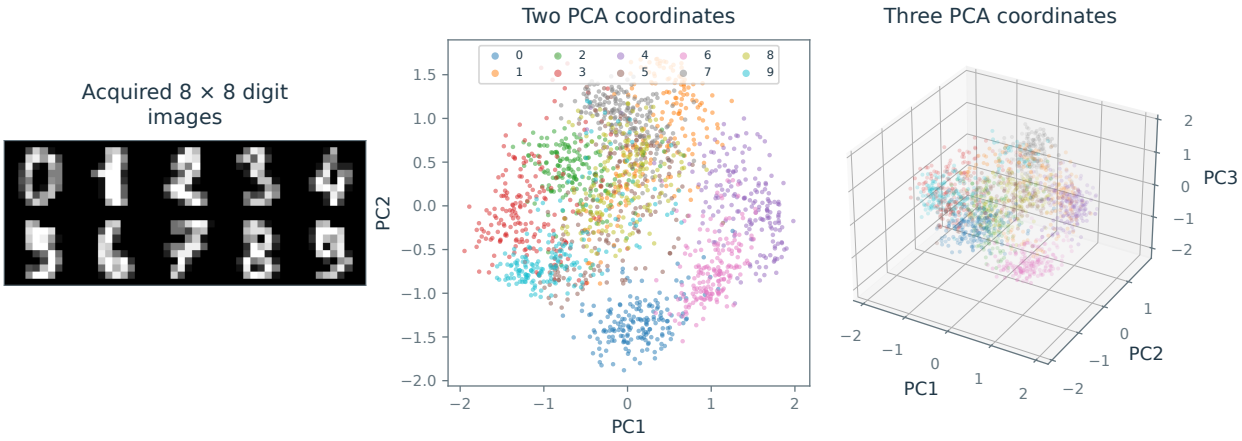


Figure 13.18. Digit images and their two- and three-dimensional PCA projections.

The vector $f(x,\beta) \in [0,1]^k$ gives the probabilities that x belongs to each class, and satisfies $\sum_\ell f(x,\beta)_\ell = 1$.

Estimate β by maximizing the log-likelihood over the training sample:

$$\max_{\beta\in\mathbb{R}^{p\times k}} \frac{1}{n}\sum_{i=1}^{n} \log(f(x_i,\beta)_{y_i})$$

where $y_i \in \mathcal{Y} = \{1,\dots,k\}$ is the label of observation x_i.

This is conveniently rewritten as

$$\min_{\beta\in\mathbb{R}^{p\times k}} \mathcal{E}(\beta) \stackrel{\text{def.}}{=} \frac{1}{n}\left(\sum_i \mathrm{LSE}(X\beta)_i - \langle X\beta,\, D\rangle\right)$$

where $D \in \{0,1\}^{n\times k}$ is the one-hot class matrix

$$D_{i,\ell} = \begin{cases} 1 & \text{if} \quad y_i = \ell, \\ 0 & \text{otherwise.} \end{cases}$$

and LSE is the log-sum-exp operator

$$\mathrm{LSE}(S)_i = \log\left(\sum_\ell \exp(S_{i,\ell})\right), \qquad \mathrm{LSE}(S) \in \mathbb{R}^n.$$

For $k = 2$, identify the first class with label $+1$ and the second with label -1. The model then agrees with binary logistic classification in Section 13.4.2, with parameter $\beta_1 - \beta_2$. A single weight vector therefore suffices in that case.

Direct evaluation of log-sum-exp can overflow when $S_{i,\ell}$ is large. A stable evaluation subtracts the largest entry in each row before exponentiation, using the identity

$$\mathrm{LSE}(S + a) = \mathrm{LSE}(S) + a$$

for a rowwise constant a. This shift is especially useful for well-separated classes, whose parameter vectors β_ℓ may grow large.

The gradient of the sum of the rowwise log-sum-exp values is the softmax operator

$$\nabla_S \sum_i \mathrm{LSE}(S)_i = \mathrm{SM}(S) \stackrel{\text{def.}}{=} \left(\frac{e^{S_{i,\ell}}}{\sum_m e^{S_{i,m}}}\right)$$

As with log-sum-exp, subtract the largest entry in each row before evaluating the exponentials.

Once the matrix D is computed, the gradient of $\mathcal{E}$ is computed as

$$\nabla\mathcal{E}(\beta) = \frac{1}{n}X^*(\mathrm{SM}(X\beta) - D).$$

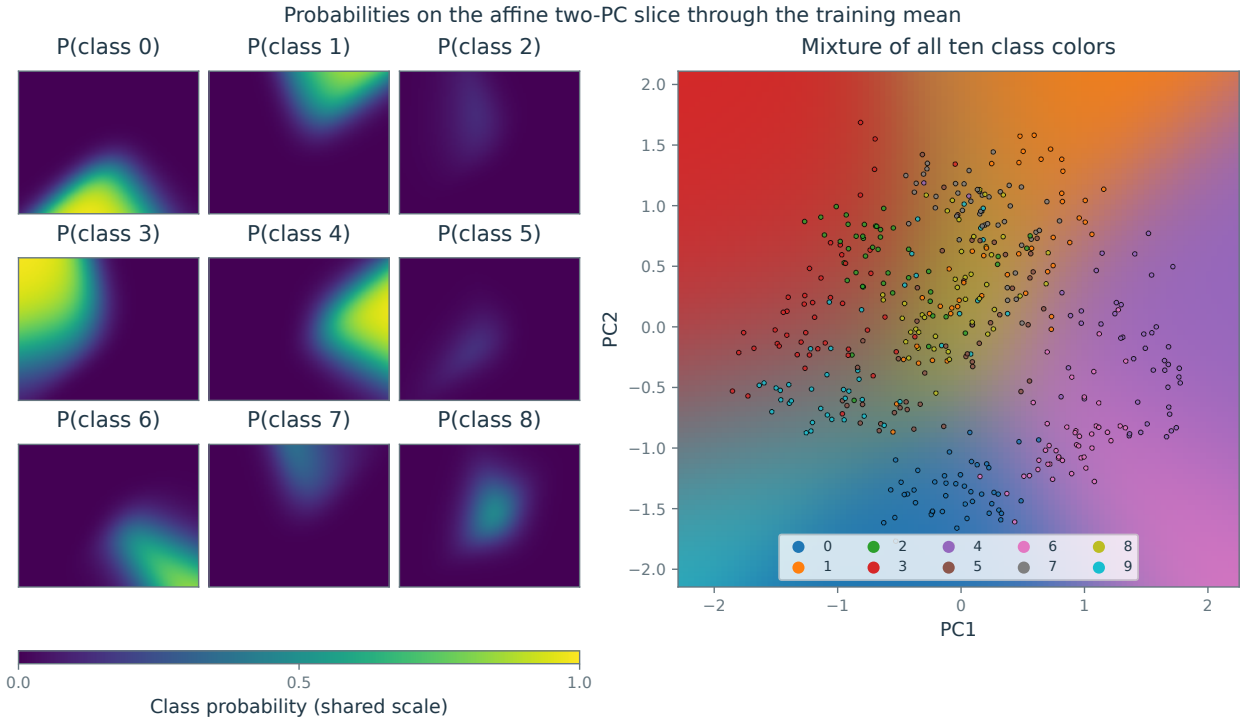


Figure 13.19. Digit classification. Left: the first nine class probabilities on an affine two-dimensional PCA slice through the training mean. Right: the mixture of all ten class colors, with projected observations overlaid.

The objective $\mathcal{E}$ can then be minimized by gradient descent.

To illustrate the method, we use a dataset of n images of size $p = 8 \times 8$, representing digits from 0 to 9 (so there are $k = 10$ classes). Figure 13.18 displays a few representative examples as well as 2-D and 3-D PCA projections. Figure 13.19 visualizes the fitted class probabilities $f(x, \hat{\beta})$ by assigning colors on a regular image grid.

13.5 Kernel Methods

Linear models may fail to capture the geometry needed for a regression or classification task. Moreover, inputs such as strings and graphs need not belong to a vector space.

Kernel methods introduce a feature map into a richer space and apply linear prediction there. This produces nonlinear regression and classification rules while retaining linear-system solvers and convex optimization. The kernel trick expresses the computations through similarities between the n observations, without constructing the feature vectors explicitly. Such methods are nonparametric when the number of fitted coefficients grows with n, allowing the model to become more expressive as data accumulate.

Algorithms that use the observations x_i only through inner products can often be kernelized. Examples include linear regression, nearest-neighbor methods, SVMs, logistic classification, and PCA. We introduce the construction through ridge regression and then extend it to other losses and distance-based methods.

13.5.1 Feature Map and Kernels

Let $\varphi : \mathcal{X} \to \mathcal{H}$ be a feature map into a real Hilbert space $\mathcal{H}$, which may be infinite dimensional. The input space $\mathcal{X}$ need not be a vector space. When $\mathcal{X} = \mathbb{R}^p$, the choice $\varphi(x) = x$ recovers ordinary linear methods. For regression, we consider predictions which are linear functions over this lifted space, i.e. of the form $x \mapsto \langle \varphi(x), \beta \rangle_{\mathcal{H}}$ for some weight vector $\beta \in \mathcal{H}$. For binary classification, one uses $x \mapsto \operatorname{sign}(\langle \varphi(x), \beta \rangle_{\mathcal{H}})$.

For scalar inputs ($p = 1$), the feature map $\varphi(x) = (1, x, x^2, \dots, x^k) \in \mathbb{R}^{k+1}$ gives polynomial regression. Multivariate monomials extend this construction to higher-dimensional inputs. Let $\Phi = (\varphi(x_i)^*)_{i=1}^n$ be the feature matrix, whose rows are $\varphi(x_i)$. For $\mathcal{H} = \mathbb{R}^{\bar{p}}$, it is an ordinary matrix $\Phi \in \mathbb{R}^{n \times \bar{p}}$. In an infinite-dimensional feature space, the same notation represents a linear operator.

For $\lambda > 0$, consider the regularized empirical-risk problem

$$\min_{\beta \in \mathcal{H}} \sum_{i=1}^{n} \ell(y_i, \langle \varphi(x_i), \beta \rangle) + \frac{\lambda}{2}\|\beta\|_{\mathcal{H}}^2 = \mathcal{L}(\Phi\beta, y) + \frac{\lambda}{2}\|\beta\|_{\mathcal{H}}^2 \tag{13.25}$$

Start with regression under the squared loss $\ell(y, z) = \frac{1}{2}(y - z)^2$:

$$\beta^\star = \operatorname*{argmin}_{\beta \in \mathcal{H}} \|\Phi\beta - y\|_{\mathbb{R}^n}^2 + \lambda\|\beta\|_{\mathcal{H}}^2.$$

which is the ridge regression problem studied in Section 13.3.1. Setting the gradient to zero gives

$$\beta^\star = (\Phi^*\Phi + \lambda \mathrm{Id}_{\mathcal{H}})^{-1}\Phi^* y.$$

Direct computation may be impractical or impossible when $\mathcal{H}$ is infinite dimensional. The following Woodbury identity converts the problem into a finite system. This is the feature-space counterpart of (13.17), with Φ replacing X and λ replacing $n\lambda_n$.

Proposition 13.7. *One has*

$$(\Phi^*\Phi + \lambda \mathrm{Id}_{\mathcal{H}})^{-1}\Phi^* = \Phi^*(\Phi\Phi^* + \lambda \mathrm{Id}_{\mathbb{R}^n})^{-1}.$$

Proof. Denoting $U = (\Phi^*\Phi + \lambda \mathrm{Id}_{\mathcal{H}})^{-1}\Phi^*$ and $V = \Phi^*(\Phi\Phi^* + \lambda \mathrm{Id}_{\mathbb{R}^n})^{-1}$, one has

$$(\Phi^*\Phi + \lambda \mathrm{Id}_{\mathcal{H}})U = \Phi^*$$

and

$$(\Phi^*\Phi + \lambda \mathrm{Id}_{\mathcal{H}})V = (\Phi^*\Phi\Phi^* + \lambda\Phi^*)(\Phi\Phi^* + \lambda \mathrm{Id}_{\mathbb{R}^n})^{-1} = \Phi^*(\Phi\Phi^* + \lambda \mathrm{Id}_{\mathbb{R}^n})(\Phi\Phi^* + \lambda \mathrm{Id}_{\mathbb{R}^n})^{-1} = \Phi^*.$$

Since $(\Phi^*\Phi + \lambda \mathrm{Id}_{\mathcal{H}})$ is invertible, one obtains the result. □

The identity gives the alternative representation

$$\beta^\star = \Phi^* c^\star \quad \text{where} \quad c^\star \stackrel{\text{def.}}{=} (K + \lambda \mathrm{Id}_{\mathbb{R}^n})^{-1} y$$

where the empirical kernel matrix is

$$K = \Phi\Phi^* = (\kappa(x_i, x_j))_{i,j=1}^n \in \mathbb{R}^{n \times n} \quad \text{where} \quad \kappa(x, x') \stackrel{\text{def.}}{=} \langle \varphi(x), \varphi(x') \rangle.$$

The coefficients $c^\star \in \mathbb{R}^n$ depend on the inputs only through κ, without explicit computation of $\varphi(x)$. If κ is inexpensive to evaluate, forming K takes $O(n^2)$ kernel evaluations. A dense direct solve costs $O(n^3)$ operations; an iterative solve costs $O(n^2)$ per matrix-vector product, with the iteration count determined by conditioning and accuracy.

The fitted predictor at a new input $x \in \mathcal{X}$ also uses only kernel evaluations:

$$\langle \beta^\star, \varphi(x) \rangle_{\mathcal{H}} = \langle \sum_i c_i^\star \varphi(x_i), \varphi(x) \rangle_{\mathcal{H}} = \sum_i c_i^\star \kappa(x, x_i). \tag{13.26}$$

13.5.2 Kernel Design

One can also start directly from a kernel $\kappa(x, x')$ rather than an explicit feature map φ. This is often convenient because kernels can be designed through their geometric properties and combined, for example by addition. The required condition is that every Gram matrix $(\kappa(x_i, x_j))_{i,j}$ be symmetric positive semidefinite. Such kernels are commonly called positive definite kernels; the condition is equivalent to the existence of a Hilbert-space feature map generating the kernel. The underlying input space need not be Euclidean.

When using the linear kernel $\kappa(x, y) = \langle x, y \rangle$, one retrieves the linear models studied in the previous section, and the lifting is trivial $\varphi(x) = x$. A family of popular kernels are polynomial ones, $\kappa(x, x') = (\langle x, x' \rangle + c)^a$ for $a \in \mathbb{N}^*$ and $c > 0$, which corresponds to a lifting in finite dimension. For instance, for $a = 2$ and $p = 2$, one has a lifting in dimension 6

$$\kappa(x, x') = \big(x_1 x_1' + x_2 x_2' + c\big)^2 = \langle \varphi(x), \varphi(x') \rangle \quad \text{where} \quad \varphi(x) = (x_1^2, x_2^2, \sqrt{2}x_1x_2, \sqrt{2c}x_1, \sqrt{2c}x_2, c)^* \in \mathbb{R}^6.$$

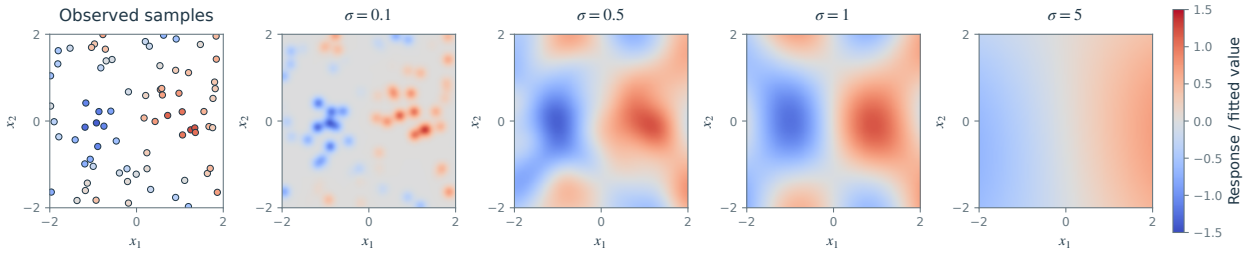


Figure 13.20. Regression using a Gaussian kernel.

A widely used choice on Euclidean spaces is the Gaussian kernel

$$\kappa(x, y) \overset{\text{def.}}{=} e^{-\frac{\|x-y\|^2}{2\sigma^2}}. \tag{13.27}$$

The bandwidth $\sigma > 0$ controls the locality of the model and is typically tuned by cross-validation. The Gaussian kernel has the infinite-dimensional feature map $\varphi(x)(t) = (2/(\pi\sigma^2))^{p/4} \exp(-\|x - t\|^2/\sigma^2) \in L^2(\mathbb{R}^p)$. Another related popular kernel is the Laplacian kernel $\exp(-\|x - y\|/\sigma)$. More generally, Bochner's theorem states that a continuous translation-invariant kernel $\kappa(x, x') = k(x - x')$ on $\mathbb{R}^p$ is positive definite if and only if k is the Fourier transform of a finite nonnegative measure. If this measure has an integrable density, a feature map can be built from the square root of that density. When k is integrable and its Fourier transform is an integrable function, the condition is $\hat{k} \geqslant 0$; strict positivity is not necessary. Figure 13.20 shows an example of regression using a Gaussian kernel.

Kernel on non-Euclidean spaces. On a general input space $\mathcal{X}$, a real kernel κ must produce a symmetric positive semidefinite Gram matrix $K = (\kappa(x_i, x_j))_{i,j}$ for every finite collection of inputs. Equivalently, there is a Hilbert space $\mathcal{H}$ and a feature map $\varphi : \mathcal{X} \to \mathcal{H}$ such that $\kappa(x, x') = \langle \varphi(x), \varphi(x') \rangle_{\mathcal{H}}$.

For instance, if $\mathcal{X} = \mathcal{P}(\mathcal{M})$ is the set of measurable sets of finite measure in a measure space $(\mathcal{M}, \mu)$, one can define $\kappa(A, B) = \mu(A \cap B)$ which corresponds to the lifting $\varphi(A) = 1_A$ so that $\langle \varphi(A), \varphi(B) \rangle = \int 1_A(x) 1_B(x) \mathrm{d}\mu(x) = \mu(A \cap B)$. Kernels can also be defined on strings and graphs, although their construction is more involved.

When $\kappa(x, x) > 0$ for every input under consideration, the kernel can be normalized to have a unit diagonal. For a translation-invariant kernel on $\mathbb{R}^p$, the diagonal equals the constant $k(0)$, so normalization only requires division by $k(0) > 0$. This corresponds to replacing $\varphi(x)$ by $\varphi(x)/\|\varphi(x)\|_{\mathcal{H}}$ and thus replacing $\kappa(x, x')$ by $\frac{\kappa(x,x')}{\sqrt{\kappa(x,x)}\sqrt{\kappa(x',x')}}$.

13.5.3 General Case

The least-squares result extends to other losses under the assumptions below. With a squared Hilbert-space norm penalty, the minimizer belongs to a subspace generated by the data, of dimension at most n. This representer theorem reduces the problem to finitely many coefficients even when the feature space $\mathcal{H}$ is infinite dimensional.

Proposition 13.8. *Assume $\lambda > 0$ and the losses $\ell(y_i, \cdot)$ are finite, nonnegative, and convex. The solution $\beta^\star \in \mathcal{H}$ of*

$$\min_{\beta \in \mathcal{H}} \sum_{i=1}^{n} \ell(y_i, \langle \varphi(x_i), \beta \rangle) + \frac{\lambda}{2} \|\beta\|_{\mathcal{H}}^2 = \mathcal{L}(\Phi\beta, y) + \frac{\lambda}{2} \|\beta\|_{\mathcal{H}}^2 \tag{13.28}$$

is unique and can be written as

$$\beta = \Phi^* c^\star = \sum_i c_i^\star \varphi(x_i) \in \mathcal{H} \tag{13.29}$$

where $c^\star \in \mathbb{R}^n$ is a solution of

$$\min_{c \in \mathbb{R}^n} \mathcal{L}(Kc, y) + \frac{\lambda}{2} \langle Kc, c \rangle_{\mathbb{R}^n} \tag{13.30}$$

where we defined

$$K \overset{\text{def.}}{=} \Phi\Phi^* = (\langle \varphi(x_i), \varphi(x_j) \rangle_{\mathcal{H}})_{i,j=1}^n \in \mathbb{R}^{n \times n}.$$

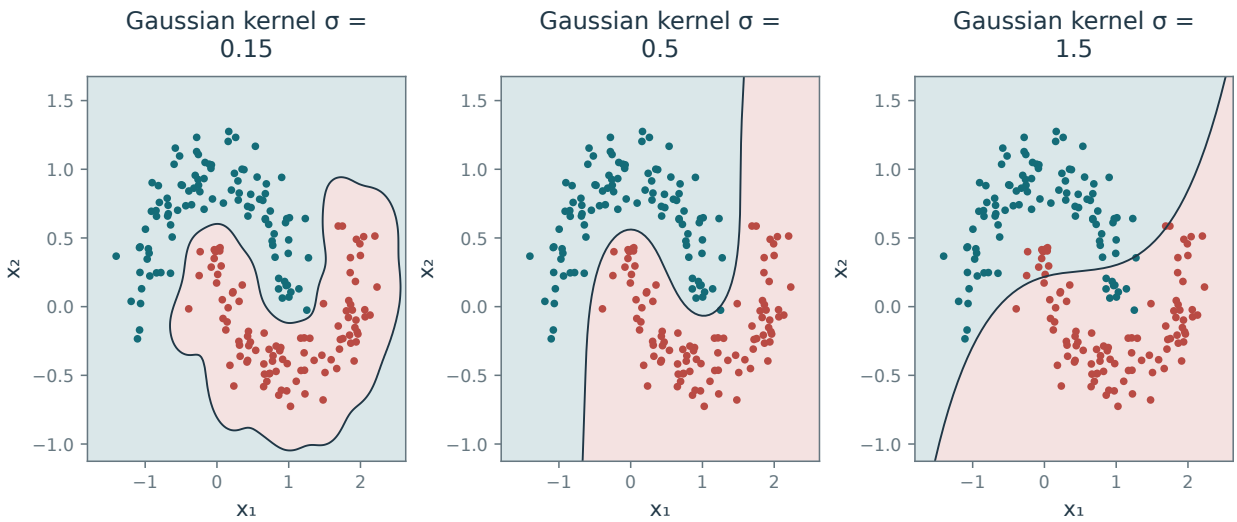


Figure 13.21. Nonlinear classification using a Gaussian kernel.

Proof. Let $V = \mathrm{Span}\{\varphi(x_i) : 1 \leqslant i \leqslant n\}$ and decompose $\beta = \beta_V + \beta_\perp$ orthogonally. Predictions on the data depend only on β_V, while $\|\beta\|^2 = \|\beta_V\|^2 + \|\beta_\perp\|^2$. Since $\lambda > 0$, a minimizer has $\beta_\perp = 0$. The restriction to the finite-dimensional space V is continuous, coercive, and strictly convex, hence has a unique minimizer. Substituting $\beta = \Phi^* c$ gives (13.30); the coefficient vector c need not be unique if K is singular. □

Equation (13.29) places the minimizer in a subspace of dimension at most n, spanned by the observed feature vectors $\varphi(x_i)$. The potentially infinite-dimensional problem (13.28) becomes the finite-dimensional problem (13.30). A dense first-order iteration costs $O(n^2)$ once the kernel matrix is available. A low-rank approximation $K \approx \tilde{\Phi}^* \tilde{\Phi}$ can reduce this cost further by introducing a smaller approximate feature space.

For classification applications, one can use for ℓ a hinge or a logistic loss function, and then the decision boundary is computed following (13.26) using kernel evaluations only as

$$\mathrm{sign}(\langle \beta^\star, \varphi(x) \rangle_{\mathcal{H}}) = \mathrm{sign}(\sum_i c_i^\star \kappa(x, x_i)).$$

Figure 13.21 illustrates a nonlinear decision boundary in two dimensions. The decision rule is linear in the feature space $\mathcal{H}$; evaluating it through the nonlinear feature map produces a curved boundary in the input space.

Nearest-neighbor classification (Section 13.4.1) and regression can likewise use distances in feature space, computed entirely from the kernel:

$$\|\varphi(x_i) - \varphi(x_j)\|_{\mathcal{H}}^2 = \kappa(x_i, x_i) + \kappa(x_j, x_j) - 2\kappa(x_i, x_j).$$

13.6 Probably Approximately Correct Learning

This section introduces Probably Approximately Correct (PAC) learning, which studies generalization guarantees under explicit complexity and sampling assumptions. Distribution-free bounds avoid further restrictions on the data distribution. The main reference (and in particular the proofs of the mentioned results) for this section is the book "Foundations of Machine Learning" by Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar https://cs.nyu.edu/~mohri/mlbook/.

Assume the observations $\mathcal{D} \overset{\text{def.}}{=} (x_i, y_i)_{i=1}^n$ are independent and identically distributed copies of a random pair (X, Y) taking values in $\mathcal{X} \times \mathcal{Y}$. From this sample, we seek a predictor $f : \mathcal{X} \to \mathcal{Y}$ with risk close to the smallest achievable value:

$$L(f) \overset{\text{def.}}{=} \mathbb{E}(\ell(Y, f(X))) = \int_{\mathcal{X}\times\mathcal{Y}} \ell(y, f(x)) \mathrm{d}\mathbb{P}_{X,Y}(x, y)$$

where $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ is some loss function. To estimate this minimizer, we select a class of functions $\mathcal{F}$ and minimize the empirical risk

$$\hat{f} \in \operatorname*{argmin}_{f \in \mathcal{F}} \hat{L}(f) \overset{\text{def.}}{=} \frac{1}{n} \sum_{i=1}^{n} \ell(y_i, f(x_i)).$$

We assume a measurable minimizer exists. It is random because it depends on the sample $\mathcal{D}$.

Example 13.9 (Regression and classification). In regression, $\mathcal{Y} = \mathbb{R}$ and the most common choice of loss is $\ell(y, z) = (y - z)^2$. For binary classification, $\mathcal{Y} = \{-1, +1\}$ and the ideal choice is the 0-1 loss $\ell(y, z) = 1_{y \neq z}$. For many useful classes $\mathcal{F}$, minimizing the empirical 0–1 risk is computationally difficult. A common alternative uses a surrogate loss of the form $\ell(y, z) = \Gamma(-yz)$ where Γ is a convex function upper-bounding $1_{\mathbb{R}^+}$, which makes empirical risk minimization convex when the real-valued score class is convex.

PAC learning seeks a bound on the excess risk $L(\hat{f}) - \inf(L) \geqslant 0$ that holds with probability at least $1 - \delta$. The bound depends on the sample size n and confidence parameter δ. For the excess risk to vanish, the approximation class must become sufficiently rich while its statistical complexity remains controlled. Quantitative approximation rates require distributional assumptions, although some consistency results hold without them.

13.6.1 Nonparametric Models and Calibration

If $\mathcal{F}$ contains all measurable functions, a minimizer $f^\star$ of L is called a Bayes estimator: it achieves the smallest possible population risk.

Risk decomposition Denoting

$$\alpha(z|x) \overset{\text{def.}}{=} \mathbb{E}_Y(\ell(Y, z)|X = x) = \int_{\mathcal{Y}} \ell(y, z) \mathrm{d}\mathbb{P}_{Y|X}(y|x)$$

the average loss of predicting $z \in \mathcal{Y}$ at input $x \in \mathcal{X}$, the risk decomposes as

$$L(f) = \int_{\mathcal{X}} \left[\int_{\mathcal{Y}} \ell(y, f(x)) \mathrm{d}\mathbb{P}_{Y|X}(y|x) \right] \mathrm{d}\mathbb{P}_X(x) = \int_{\mathcal{X}} \alpha(f(x)|x) \mathrm{d}\mathbb{P}_X(x)$$

Consequently, $f^\star$ can be chosen separately at each input x by solving

$$f^\star(x) = \operatorname*{argmin}_z \alpha(z|x).$$

Example 13.10 (Regression). For squared-loss regression with $\mathbb{E}(Y^2) < \infty$, one has $f^\star(x) = \mathbb{E}(Y|X = x) = \int_{\mathcal{Y}} y \mathrm{d}\mathbb{P}_{Y|X}(y|x)$, which is the conditional expectation.

Example 13.11 (Classification). For classification applications, where $\mathcal{Y} = \{-1, 1\}$, it is convenient to introduce $\eta(x) \overset{\text{def.}}{=} \mathbb{P}_{Y|X}(y = 1|x) \in [0, 1]$. If the two classes are separable, then $\eta(x) \in \{0, 1\}$ for $\mathbb{P}_X$-almost every x. For the 0-1 loss $\ell(y, z) = 1_{y \neq z} = 1_{(0,+\infty)}(-yz)$, one may choose $f^\star = \operatorname{sign}(2\eta - 1)$, with either class selected at a tie and $L(f^\star) = \mathbb{E}_X(\min(\eta(X), 1 - \eta(X)))$. The conditional probability η is unknown and must be estimated from $\mathcal{D}$. Direct 0–1 optimization is computationally difficult for many useful classes $\mathcal{F}$. For a margin loss $\ell(y, z) = \Gamma(-yz)$, an unconstrained Bayes score satisfies

$$f^\star(x) = \operatorname*{argmin}_z \Big(\alpha(z|x) = \eta(x)\Gamma(-z) + (1 - \eta(x))\Gamma(z) \Big),$$

Thus the optimal score is a function of $\eta(x)$. This argmin notation presumes attainment; for logistic loss and $\eta \in \{0, 1\}$, the infimum is approached only as the score tends to the appropriate infinity.

Calibration in the classification setup We would like the classifier $\operatorname{sign}(f^\star)$ to agree with a Bayes classifier for the 0–1 loss, namely $\operatorname{sign}(2\eta - 1)$ away from ties. This property is called classification calibration. It does not require the scores $f^\star$ and $2\eta - 1$ themselves to agree. The following result characterizes calibration for finite convex margin losses.

Proposition 13.12. *A loss ℓ associated to a finite, nonnegative convex Γ is calibrated if and only if Γ is differentiable at 0 and $\Gamma'(0) > 0$.*

Both hinge and logistic loss satisfy this calibration criterion. Let L_Γ be the risk for $\ell(y,z) = \Gamma(-yz)$ and let $L_{0\text{–}1}$ be classification risk, with a fixed tie-breaking convention. Quantitative calibration bounds have the form

$$0 \leqslant L_{0\text{–}1}(\operatorname{sign} f) - \inf L_{0\text{–}1} \leqslant \Psi(L_\Gamma(f) - \inf L_\Gamma) \tag{13.31}$$

for some increasing function $\Psi : \mathbb{R}^+ \to \mathbb{R}^+$ satisfying $\Psi(0) = 0$. Such a control ensures in particular that if $f^\star$ minimizes L_Γ, its induced classifier has Bayes-optimal 0–1 risk. At $\eta = 1/2$, either class is optimal; away from ties, its sign agrees almost surely with that of $2\eta - 1$. For hinge loss, the calibration bound holds with $\Psi(r) = r$. For logistic loss, it holds with $\Psi(s) = \sqrt{2s}$.

13.6.2 PAC bounds

Approximation–estimation decomposition. For a class $\mathcal{F}$ of functions, the excess risk of the empirical estimator

$$\hat{f} \in \operatorname*{argmin}_{f \in \mathcal{F}} \hat{L}(f)$$

decomposes into a random estimation error and a deterministic approximation error:

$$L(\hat{f}) - \inf L = \left[L(\hat{f}) - \inf_{\mathcal{F}} L \right] + \mathcal{A}(\mathcal{F}) \quad \text{where} \quad \mathcal{A}(\mathcal{F}) \overset{\text{def.}}{=} \left[\inf_{\mathcal{F}} L - \inf L \right]. \tag{13.32}$$

This separates the effects of sampling and model approximation.

Enlarging a nested class $\mathcal{F}$ reduces approximation error but typically increases the estimation-error bound. Choose the class size as a function of n to balance these effects and avoid overfitting.

Approximation error. A quantitative approximation bound requires assumptions on $f^\star$. No-free-lunch results rule out a uniform distribution-free approximation rate over all measurable targets. The following example illustrates such a bound.

Example 13.13 (Linearly parameterized functionals). A popular class of functions consists of linearly parameterized maps of the form

$$f(x) = f_w(x) = \langle \varphi(x),\, w \rangle_{\mathcal{H}}$$

where $\varphi : \mathcal{X} \to \mathcal{H}$ maps inputs to a Hilbert feature space. For $\mathcal{X} = \mathbb{R}^p$, the identity map $\varphi(x) = x$ recovers ordinary linear models. One can also consider for instance polynomial features, $\varphi(x) = (1, x_1, \ldots, x_p, x_1^2, x_1 x_2, \ldots)$, giving rise to polynomial regression and polynomial classification boundaries. Restrict the parameters to the ball $\mathcal{F} = \{f_w \,;\, \|w\|_{\mathcal{H}} \leqslant R\}$. Suppose $f^\star = f_{w^\star}$ for some $w^\star \in \mathcal{H}$, the loss $\ell(y,\cdot)$ is uniformly Q-Lipschitz, and $\mathbb{E}\|\varphi(X)\|_{\mathcal{H}} < \infty$. Projecting $w^\star$ onto the radius-R ball gives

$$\mathcal{A}(\mathcal{F}) \leqslant Q\mathbb{E}(\|\varphi(X)\|_{\mathcal{H}}) \max(\|w^\star\|_{\mathcal{H}} - R, 0).$$

Remark 13.14 (Connections with RKHS). Note that this lifting actually corresponds to using functions f in a reproducing kernel Hilbert space, denoting

$$\|f\|_k \overset{\text{def.}}{=} \inf_{w \in \mathcal{H}} \{\|w\|_{\mathcal{H}} \,;\, f = f_w\}$$

with associated kernel $k(x,x') \overset{\text{def.}}{=} \langle \varphi(x),\, \varphi(x') \rangle_{\mathcal{H}}$. We use the feature-map representation directly below.

Estimation error. Concentration inequalities control sampling fluctuations, while a bound on the complexity of $\mathcal{F}$ makes this control uniform over the model class.

The estimation error is controlled by uniform deviations between L and $\hat{L}$. Let $g \in \mathcal{F}$ attain $\inf_{\mathcal{F}} L$, assuming for simplicity that such a minimizer exists. Then

$$L(\hat{f}) - \inf_{\mathcal{F}} L = \left[L(\hat{f}) - \hat{L}(\hat{f}) \right] + \left[\hat{L}(\hat{f}) - \hat{L}(g) \right] + \left[\hat{L}(g) - L(g) \right] \leqslant 2 \sup_{f \in \mathcal{F}} |\hat{L}(f) - L(f)| \tag{13.33}$$

since $\hat{L}(\hat{f}) - \hat{L}(g) \leqslant 0$. Define the two one-sided deviations

$$\Delta_+(\mathcal{D}) = \sup_{f \in \mathcal{F}} (L(f) - \hat{L}(f)), \qquad \Delta_-(\mathcal{D}) = \sup_{f \in \mathcal{F}} (\hat{L}(f) - L(f)).$$

The preceding decomposition gives the sharper bound $L(\hat{f}) - \inf_{\mathcal{F}} L \leqslant \Delta_+ + \Delta_-$. Assume all suprema below are measurable (or use outer expectations).

Proposition 13.15 (McDiarmid concentration)**.** *Assume* $0 \leqslant \ell(Y, f(X)) \leqslant \ell_\infty$ *almost surely, uniformly over* $f \in \mathcal{F}$*. With probability at least* $1-\delta$*, both signs satisfy*

$$\Delta_\pm(\mathcal{D}) - \mathbb{E}_{\mathcal{D}}\Delta_\pm(\mathcal{D}) \leqslant \ell_\infty \sqrt{\frac{\log(2/\delta)}{2n}}. \tag{13.34}$$

To bound $\mathbb{E}_{\mathcal{D}}\Delta_\pm(\mathcal{D})$, we need to control the complexity of $\mathcal{F}$. VC dimension gives one approach, but for norm-constrained linear models it can yield loose bounds that depend on the ambient dimension. Rademacher complexity gives a finer measure for a class $\mathcal{G}$ of functions from $\mathcal{X} \times \mathcal{Y}$ to $\mathbb{R}$:

$$\mathcal{R}_n(\mathcal{G}) \overset{\text{def.}}{=} \mathbb{E}_{\varepsilon,\mathcal{D}}\Big[\sup_{g\in\mathcal{G}} \frac{1}{n}\sum_{i=1}^n \varepsilon_i g(x_i, y_i)\Big]$$

where ε_i, independent of the data, are independent Rademacher random variables (i.e. $\mathbb{P}(\varepsilon_i = \pm 1) = 1/2$). The quantity $\mathcal{R}_n(\mathcal{G})$ depends on the distribution of (X, Y).

Apply this complexity measure to the loss class $\mathcal{G} = \ell[\mathcal{F}]$, defined by

$$\ell[\mathcal{F}] \overset{\text{def.}}{=} \{(x, y) \in \mathcal{X} \times \mathcal{Y} \mapsto \ell(y, f(x)) \; ; \; f \in \mathcal{F}\},$$

Symmetrization gives the following bounds.

Proposition 13.16. *One has*

$$\mathbb{E}\Delta_\pm(\mathcal{D}) \leqslant 2\mathcal{R}_n(\ell[\mathcal{F}]).$$

If ℓ *is* Q*-Lipschitz with respect to its second variable, one furthermore has* $\mathcal{R}_n(\ell[\mathcal{F}]) \leqslant Q\mathcal{R}_n(\mathcal{F})$ *(here the class of functions only depends on* x*).*

Combining (13.32), (13.33), and Propositions 13.15 and 13.16 gives the following result.

Theorem 13.17. *Assume* $\ell(y, \cdot)$ *is uniformly* Q*-Lipschitz and* $0 \leqslant \ell \leqslant \ell_\infty$ *uniformly over the class. Then, with probability at least* $1-\delta$

$$0 \leqslant L(\hat{f}) - \inf L \leqslant \ell_\infty \sqrt{\frac{2\log(2/\delta)}{n}} + 4Q\mathcal{R}_n(\mathcal{F}) + \mathcal{A}(\mathcal{F}). \tag{13.35}$$

Example 13.18 (Linear models)*.* In the case where $\mathcal{F} = \{\langle \varphi(\cdot), w\rangle_{\mathcal{H}} \; ; \; \|w\| \leqslant R\}$ where $\|\cdot\|$ is some norm on $\mathcal{H}$, one has

$$\mathcal{R}_n(\mathcal{F}) = \frac{R}{n}\mathbb{E}_{\varepsilon,\mathcal{D}}\Big\|\sum_i \varepsilon_i \varphi(x_i)\Big\|_*$$

where $\|\cdot\|_*$ is the dual norm, defined by

$$\|u\|_* \overset{\text{def.}}{=} \sup_{\|w\|\leqslant 1} \langle u, w\rangle_{\mathcal{H}}.$$

When $\|\cdot\| = \|\cdot\|_{\mathcal{H}}$ is the Hilbert norm, the expression simplifies further: $\|u\|_* = \|u\|_{\mathcal{H}}$ and

$$\mathcal{R}_n(\mathcal{F}) \leqslant \frac{R\sqrt{\mathbb{E}(\|\varphi(X)\|_{\mathcal{H}}^2)}}{\sqrt{n}}.$$

The bound has no explicit dependence on feature dimension and also applies to infinite-dimensional RKHS feature maps with finite second moment. For fixed R and fixed loss constants, the estimation bound in (13.35) decays as $1/\sqrt{n}$; the approximation error remains a separate term. The radius R controls the balance between approximation and estimation. In practice, it can be selected by cross-validation.

Example 13.19 (Application to SVM classification)*.* The 0–1 loss is not Lipschitz in a real-valued score. For a margin $\rho > 0$, use the bounded ramp loss

$$\Gamma_\rho(s) = \min\{1, \max\{0, 1 + s/\rho\}\}.$$

It is $1/\rho$-Lipschitz and satisfies

$$1_{[0,+\infty)}(s) \leqslant \Gamma_\rho(s) \leqslant \max\{1 + s/\rho, 0\}.$$

The left inequality counts score ties as errors, so it also bounds the risk of any fixed tie-breaking classifier. For $\mathcal{F}_R = \{f_w : \|w\|_{\mathcal{H}} \leqslant R\}$, the one-sided concentration and symmetrization bounds imply, simultaneously for every $f \in \mathcal{F}_R$, with probability at least $1 - \delta$,

$$L_{0\text{–}1}(\operatorname{sign} f) \leqslant \frac{1}{n}\sum_i \max\{1 - y_i f(x_i)/\rho, 0\} + \frac{2R}{\rho}\sqrt{\frac{\mathbb{E}\|\varphi(X)\|_{\mathcal{H}}^2}{n}} + \sqrt{\frac{\log(1/\delta)}{2n}}.$$

This bound applies to a data-dependent predictor, including an empirical hinge-loss minimizer in the ball. It does not require minimizing the nonconvex ramp loss. In practice, the norm constraint is often replaced by a penalty:

$$\min_{w} \frac{1}{n}\sum_i \max\{1 - y_i f_w(x_i), 0\} + \lambda\|w\|_{\mathcal{H}}^2, \qquad \lambda > 0. \tag{13.36}$$

This is the kernel support vector machine.

Remark 13.20 (Resolution using the kernel trick). The kernel trick reduces problems such as (13.36), of the form

$$\min_{w \in \mathcal{H}} \frac{1}{n}\sum_{i=1}^{n} \ell(y_i, \langle w,\, \varphi(x_i)\rangle) + \lambda\|w\|_{\mathcal{H}}^2 \tag{13.37}$$

to optimization over n coefficients, even when the feature space is infinite dimensional. The representer theorem gives $w^\star = \sum_{i=1}^n c_i^\star \varphi(x_i)$ for some $c^\star \in \mathbb{R}^n$: an orthogonal component would leave the predictions unchanged and strictly increase the positive norm penalty. Substituting this representation into (13.37) reduces the data dependence to the empirical kernel matrix $K = (k(x_i, x_j) \overset{\text{def.}}{=} \langle \varphi(x_i),\, \varphi(x_j)\rangle)_{i,j=1}^n \in \mathbb{R}^{n\times n}$ since it reads

$$\min_{c \in \mathbb{R}^n} \frac{1}{n}\sum_{i=1}^{n} \ell(y_i, (Kc)_i) + \lambda\langle Kc,\, c\rangle$$

From any optimal $c^\star$ of this convex program (which need not be strictly convex when K is singular), the estimator is evaluated as $f_{w^\star}(x) = \langle w^\star,\, \varphi(x)\rangle = \sum_i c_i^\star k(x_i, x)$.