Modern Statistical Methods — Summary

Lucas Riedstra

October 18, 2020

Contents

Classical models rely on so-called "large n asymptotics" (where n is the sample size). This course focuses on the scenario where p, the number of variables, is larger or about as large as n. In this case, the classical theory breaks down, so we need new methods.

1 Kernel machines

We represent data are pairs $(Y_i, x_i) \in \mathbb{R} \times \mathbb{R}^p$ $(i = 1, \dots, n)$. The random variables Y_i are called the *responses*, and the (fixed) variables x_i are called *predictors*.

Recap 1.1. Let $X = (X_1, \dots, X_n)^\top$ be a multivariate random variable. Its distribution function is given by

$$F_X \colon \mathbb{R}^n \to [0,1] \colon \mathbf{x} \mapsto \mathbb{P}(X_1 \leq x_1, \dots, X_n \leq x_n).$$

Its expected value is given by

$$\mathbb{E}[X] \coloneqq (\mathbb{E}[X_1], \dots, \mathbb{E}[X_n])^\top \in \mathbb{R}^n.$$

Its covariance matrix is given by

$$\mathrm{Var}[X] = \mathbb{E}[(X - \mathbb{E}[X])(X - \mathbb{E}[X])^\top] = \mathbb{E}[XX^\top] - \mathbb{E}[X]\mathbb{E}[X]^\top.$$

The matrix $\mathrm{Var}[X]$ is symmetric positive semidefinite and satisfies $(\mathrm{Var}[X])_{ij} = \mathrm{Cov}(X_i, X_j)$.

Definition 1.2. In a *linear model*, we assume that

$$Y_i = x_i^\top \beta^0 + \varepsilon_i \quad (i = 1, \dots, n).$$

where $\beta \in \mathbb{R}^p$ is unknown and the multivariate random variable $\varepsilon = (\varepsilon_1, \dots, \varepsilon_n)$ satisfies $\mathbb{E}(\varepsilon) = 0$ and $\mathrm{Var}(\varepsilon) = \sigma^2 I$.

Definition 1.3. For an estimator $\tilde{\beta}$ of β^0, its *mean squared error* (MSE) is given by

$$\mathbb{E}_{\beta^0, \sigma^2}\left[(\tilde{\beta} - \beta^0)(\tilde{\beta} - \beta^0)^\top\right] = \mathrm{Var}(\tilde{\beta}) + \left[\mathbb{E}(\tilde{\beta} - \beta^0)\right]\left[\mathbb{E}(\tilde{\beta} - \beta^0)\right]^\top.$$

Note that if $\tilde{\beta}$ is unbiased, the second term will disappear and the MSE is simply the variance.

Recap 1.4. The maximum likelihood estimator (MLE) in this model is the ordinary least squares (OLS) estimator $\hat{\beta}^{\mathrm{OLS}} = (X^\top X)^{-1} X^\top Y$, where the *design matrix* $X \in \mathbb{R}^{n \times p}$ is the matrix whose rows are the vectors x_i. This estimator only exists if X has full column rank, so in particular, it is required that $p \leq n$.

The Cramér-Rao lower bown states that, out of all unbiased estimators, the MLE has the optimal variance *asymptotically* (i.e., for $n \to \infty$).

1.1 Ridge regression

Definition 1.5. Let $\lambda \geq 0$, and let $\mathbf{1} \in \mathbb{R}^n$ be the all-ones vector. Then we define the *Ridge regression* estimators

$$(\hat{\mu}_\lambda^{\mathrm{R}}, \hat{\beta}_\lambda^{\mathrm{R}}) \coloneqq \operatorname*{arg\,min}_{(\mu,\beta) \in \mathbb{R} \times \mathbb{R}^p} \left\{ \|Y - \mu\mathbf{1} - X\beta\|^2 + \lambda \|\beta\|^2 \right\},$$

where the used norm is the 2-norm. The parameter λ is called the *regularisation parameter*.

The parameter λ represents a penalty for large coefficients in the design matrix. The intercept is not penalised — this is because a shift in units should not affect the fitted values. However, $X\hat{\beta}$ is not invariant under scale transformations, so it is common practice to centre the columns of X to have mean 0, and then scale them to have ℓ_2-norm $\sqrt{n}$.

After that, we can compute $\hat{\mu}_\lambda^{\mathrm{R}}$ by taking the derivative:

$$\|Y - \mu\mathbf{1} - X\beta\|^2 = \sum_i (Y_i - \mu - \sum_j X_{ij}\beta_j)^2.$$

$$\frac{\partial}{\partial \mu}\|Y - \mu\mathbf{1} - X\beta\|^2 = -2\sum_i \left(Y_i - \mu - \sum_j X_{ij}\beta_j\right).$$

Setting this derivative equal to 0 yields

$$\begin{aligned} -2\sum_i \left(Y_i - \mu - \sum_j X_{ij}\beta_j\right) &= 0 \\ \sum_i Y_i - n\mu - \sum_j \beta_j(\sum_i X_{ij}) &= 0 \\ \sum_i Y_i - n\mu &= 0 \\ \mu = \frac{1}{n}\sum_i Y_i &= \bar{Y}. \end{aligned}$$

Therefore we conclude $\hat{\mu}_\lambda^{\mathrm{R}} = \bar{Y}$. After centering the responses (i.e. replacing Y_i by $Y_i - \bar{Y}$), the problem can be reduced to

$$\hat{\beta}_\lambda^{\mathrm{R}} = \operatorname*{arg\,min}_{\beta \in \mathbb{R}^p} Q(\beta) \coloneqq \operatorname*{arg\,min}_{\beta \in \mathbb{R}^p} \left[\|Y - X\beta\|^2 + \lambda\|\beta\|^2\right].$$

Since $Q(\beta)$ is convex quadratic, there is a unique root, and to find it we compute

$$\nabla_\beta Q(\beta) = 2X^\top(Y - X\beta) + 2\lambda\beta = 0 \iff \beta = (X^\top X + \lambda I)^{-1}X^\top Y.$$

We conclude that $\hat{\beta}_\lambda^{\mathrm{R}} = (X^\top X + \lambda I)^{-1}X^\top Y$. Note that, even if X does not have full column rank, this estimator exists for all $\lambda > 0$. In fact, for λ sufficiently small, the Ridge estimator outperforms the MLE in terms of mean squared error:

Theorem 1.6. *Fix β^0, σ^2, and assume that $\hat{\beta}^{\mathrm{OLS}}$ exists (i.e., X has full column rank). For some $\lambda > 0$ sufficiently small, it holds that the MSE of $\hat{\beta}^{\mathrm{OLS}}$ minus the MSE of $\hat{\beta}_\lambda^{\mathrm{R}}$ is positive definite.*

Proof. This is simply writing out the MSE's. In the end, we find that the result holds for $0 < \lambda < 2\sigma^2/\left\|\beta^0\right\|^2$. □

1.1.1 The SVD and PCA

Recap 1.7. Recall that any $X \in \mathbb{R}^{n\times p}$ can be factorised as $X = UDV^\top$, where U, V are $n \times n$ and $p \times p$ orthogonal matrices respectively, and $D \in \mathbb{R}^{n\times p}$ satisfies $D_{11} \geq \cdots \geq D_{mm} \geq 0$ where $m \coloneqq \min(n, p)$, and all other entries of D are 0. This is called the *singular value decomposition* or SVD of X.

If $n > p$, we can replace U by its first p columns and D by its first p rows to produce the so-called *thin SVD* of X. Then $U \in \mathbb{R}^{n\times p}$ has orthogo nal columns (so $U^\top U = I$) and $D \in \mathbb{R}^{p\times p}$

is square and diagonal.

Suppose $n \geq p$ and let $X = UDV^\top$ be the thin SVD of our design matrix X. Then we can write the fitted values from the Ridge regression as follows:

$$\begin{aligned} X\hat{\beta}_\lambda^{\mathrm{R}} &= X(X^\top X + \lambda I)^{-1} X^\top Y \\ &= UDV^\top (VD^2V^\top + \lambda I)^{-1} VDU^\top Y \\ &= UDV^\top \left(V(D^2 + \lambda I)V^\top\right)^{-1} VDU^\top Y \\ &= UD(D^2 + \lambda I)^{-1} DU^\top Y \\ &= UD^2(D^2 + \lambda I)^{-1} U^\top Y \\ &= \sum_{j=1}^{p} \frac{D_{jj}^2}{D_{jj}^2 + \lambda} U_j U_j^\top Y. \end{aligned}$$

Note that for OLS ($\lambda = 0$), this is simply the projection of Y onto the column space of X (if X has full column rank). If $\lambda > 0$, Y is still projected onto the column space of X, but the projection is shrunk in the directions of the left singular vectors, and the lower the corresponding singular value, the higher the shrinkage.

Principal component analysis Consider $v \in \mathbb{R}^p$ with norm 1, then since the columns of X have been centered, the sample mean of Xv is 0, and the sample variance is therefore

$$\frac{1}{n}\sum_i (Xv)_i^2 = \frac{1}{n}(Xv)^\top Xv = \frac{1}{n} v^\top X^\top Xv = \frac{1}{n} v^\top VD^2V^\top v.$$

Writing $a = V^\top v$ (with $\|a\| = 1$), we find

$$\frac{1}{n} v^\top VD^2V^\top v = \frac{1}{n} a^\top D^2 a = \frac{1}{n}\sum_j a_j^2 D_{jj}^2$$

Therefore, we see that the above is maximised if $a = \pm e_1$, or equivalently $v = \pm V_1$. Therefore, V_1 determines which combination of columns of X has the largest variance (subject to having norm 1), and $XV_1 = D_{11}U_1$ is known as the *first principal component* of X. Analogously, it can be shown that $D_{22}U_2, \ldots, D_{pp}U_p$ have maximum variance D_{jj}^2/n, subject to being orthonormal to all earlier principal components.

We see that Ridge regression shrinks Y most in the smaller principal components of X. Therefore it will work well if most of the information is in the larger principal components of X.

A comment on computation By analogous calculations as before, one cam compute $\hat{\beta}_\lambda^{\mathrm{R}} = V(D^2 + \lambda I)^{-1} DU^\top Y$. Since calculating the inverse of a diagonal matrix is trivial, we see that the complexity of computing $\hat{\beta}_\lambda^{\mathrm{R}}$ for any λ lies in $O(np)$. Of course, this is after computation of the SVD of X, which lies in $O(np\min(n,p))$.

1.2 v-fold cross validation

Of course, we are still left with the problem of choosing λ in ridge regression. We consider one possible way of doing so, namely v-fold cross validation, which is a general way of selection a good regression method from several competing methods. Here, we assume that our predictors are random, so that we have i.i.d. data pairs (x_i, Y_i) $(i = 1, \ldots, n)$. Suppose (x^*, Y^*) is a new data pair, independent of

(X, Y) and identically distributed. Ideally, we want to pick λ which minimises the prediction error (averaged over Y^* and x^*)

$$\mathbb{E}\left[\left(Y^* - (x^*)^\top \hat{\beta}^{\mathrm{R}}_\lambda(X, Y)\right)^2 \mid X, Y\right],$$

where the dependence of $\hat{\beta}^{\mathrm{R}}_\lambda$ on the training data (X, Y) is made explicit by denoting it $\hat{\beta}^{\mathrm{R}}_\lambda(X, Y)$.

This is impossible to minimise, but it may be possible to minimise the expected prediction error (averaged over the training data)

$$\mathbb{E}\left\{\mathbb{E}\left[\left(Y^* - (x^*)^\top \hat{\beta}^{\mathrm{R}}_\lambda(X, Y)\right)^2 \mid X, Y\right]\right\}. \tag{1}$$

This is still not possible to compute directly, but we estimate it using v-fold cross validation. Split the data into v groups or *folds* of roughly equal size $(X^{(1)}, Y^{(1)}), \ldots, (X^{(v)}, Y^{(v)})$ and let $(X^{(-k)}, Y^{(-k)})$ denote all data except that in the k-th fold. Then we define

$$\mathrm{CV}(\lambda) \coloneqq \frac{1}{n}\sum_{i=1}^{n}\left[Y_i - x_i^\top \hat{\beta}^{\mathrm{R}}_\lambda(X^{(-\kappa(i))}, Y^{(-\kappa(i))})\right]^2,$$

and choose the value of λ that minimises $\mathrm{CV}(\lambda)$.

The function $\mathrm{CV}(\lambda)$ is called the *out-of-sample error*, since the training data does not include x_i.

Recap 1.8. The *tower rule* states that for random variables X, Y we have $\mathbb{E}[X] = \mathbb{E}[\mathbb{E}[X \mid Y]]$.

Note that for each i, we have

$$\mathbb{E}\left[\left\{Y_i - x_i^\top \hat{\beta}^{\mathrm{R}}_\lambda(X^{(-\kappa(i))}, Y^{(-\kappa(i))})\right\}^2\right] = \mathbb{E}\left[\mathbb{E}\left[\left\{Y_i - x_i^\top \hat{\beta}^{\mathrm{R}}_\lambda(X^{(-\kappa(i))}, Y^{(-\kappa(i))})\right\}^2 \mid X^{-\kappa(i)}, Y^{-\kappa(i)}\right]\right].$$

This equals the expected prediction error in eq. (1), except that the training data X, Y are replaced with a smaller data set.

We now have a bias-variance tradeoff in the size of the folds: if $v = n$ (known as "leave-one-out" cross-validation), the estimation will be almost unbiased, but the averaged quantities in $\mathrm{CV}(\lambda)$ will be highly correlated which leads to high variance. Typical choices of v are 5 or 10.

Instead of finding the single best λ, we can also aim to find the best weighted combination of λ's. For example, suppose λ is restricted to a grid $\lambda_1 > \cdots > \lambda_L$. Then we can use any nonnegative least-squares optimization algorithm to minimise

$$\frac{1}{n}\sum_{i=1}^{n}\left[Y_i - \sum_{\ell=1}^{L} w_\ell x_i^\top \hat{\beta}^{\mathrm{R}}_{\lambda_\ell}(X^{(-\kappa(i))}, Y^{(-\kappa(i))})\right]^2,$$

over all $w \in \mathbb{R}^L_{\geq 0}$. This procedure is known as *stacking* and often outperforms cross-validation.

1.3 The kernel trick

We note that

$$X^\top(XX^\top + \lambda I) = (X^\top X + \lambda I)X^\top,$$

and multiplying from the left with $(X^\top X + \lambda I)^{-1}$ and from the right with $(XX^\top + \lambda I)^{-1}$ gives

$$(X^\top X + \lambda I)^{-1}X^\top = X^\top(XX^\top + \lambda I)^{-1}.$$

Using this, we see that we can rewrite the fitted values from ridge regression as follows:

$$X\hat{\beta}^R_\lambda = X(X^\top X + \lambda I)^{-1}X^\top Y = XX^\top(XX^\top + \lambda I)^{-1}Y.$$

Two important remarks:

1. Computing the LHS of this equation takes roughly $O(np^2 + p^3)$ operations, while computing the RHS takes $O(n^2p + n^3)$ operations (this is because in the LHS we invert an $p \times p$ matrix, while in the RHS we invert a $n \times n$ matrix). Therefore, if $p \gg n$, the RHS can be much cheaper to compute.

2. The LHS depends only on the matrix $K = XX^\top$ (this matrix is called the *kernel matrix*). Intuitively, since $K_{ij} = \langle x_i, x_j \rangle$, the entries of the kernel matrix show how 'similar' the corresponding predictors are.

Example 1.9. Suppose we have data $(Y_i, z_i)_{i=1,\dots,n}$ with $z_i = (z_{i1}, \dots, z_{id})^\top$, and we believe the following quadratic relation holds:

$$Y_i = \sum_k \sqrt{2}\gamma_k z_{ik} + \sum_{k,\ell} \vartheta_{k\ell} z_{ik} z_{i\ell} + \varepsilon_i.$$

To compute fitted values using ridge regression, we can rewrite this as a linear model $Y = X\beta + \varepsilon$ where

$$\beta = \begin{pmatrix} \gamma_1 \\ \vdots \\ \gamma_d \\ \vartheta_{11} \\ \vartheta_{12} \\ \vdots \\ \vartheta_{dd} \end{pmatrix}, \quad x_i = \begin{pmatrix} \sqrt{2}z_{i1} \\ \vdots \\ \sqrt{2}z_{id} \\ z_{i1}z_{i1} \\ z_{i1}z_{i2} \\ \vdots \\ z_{id}z_{id} \end{pmatrix}.$$

In this case, we have $p = d^2 + d$ variables, which means computing $(X^\top X + \lambda I)^{-1}$ takes $O(d^6)$ operations. In this case, computing $(XX^\top + \lambda I)^{-1}$ is probably easier.

We are still left with the problem of computing $K \coloneqq XX^\top$, which can take $O(n^2 p) = O(n^2 d^2)$ operations if done naively. However, observe that

$$K_{ij} = x_i^\top x_j = 2\sum_k z_{ik} z_{jk} + \sum_{k,\ell} z_{ik} z_{i\ell} z_{jk} z_{j\ell} = \left(1 + \sum_k z_{ik} z_{jk}\right)^2 - 1 = (1 + z_i^\top z_j) - 1.$$

This quantity can be computed in $O(d)$, and therefore K can be computed in $O(n^2 d)$ operations: we have a factor d improvement.

The general point of the previous example is that we can bypass the features x_i entirely and instead think directly of $K = XX^\top$ where an entry K_{ij} represents similarity between the inputs of the i-th and j-th samples. This leads to the notion of a kernel in general.

1.4 Kernels

We will assume our inputs $x_1, \dots, x_n$ live in an abstract space $\mathcal{X}$.

Definition 1.10. A (*positive-definite*) *kernel* is a symmetric map $k\colon \mathcal{X}^2 \to \mathbb{R}$ such that for all $n \in \mathbb{N}$ and all $x_1, \dots, x_n \in \mathcal{X}$, the matrix $K \in \mathbb{R}^{n \times n}$ with $K_{ij} = k(x_i, x_j)$ is positive semi-definite.

Proposition 1.11 (Cauchy-Schwarz for kernels)**.** *Let k be a kernel and $x, x' \in \mathcal{X}$, then*

$$k(x, x')^2 \le k(x, x) k(x', x').$$

Proof. The matrix $\left(\begin{smallmatrix} k(x,x) & k(x,x') \\ k(x',x) & k(x',x') \end{smallmatrix}\right)$ must be positive semi-definite so its determinant must be non-negative. □

In our old models, the data points x_i were vectors in $\mathbb{R}^p$. Now we try to think of them as points in an abstract space with an associated *feature map* $\varphi\colon \mathcal{X} \to \mathcal{H}$ (with $\mathcal{H}$ an inner product space), and a kernel $k(x, x')$ gives a measure of similarity between $\varphi(x)$ and $\varphi(x')$. In this case, we have the following:

Proposition 1.12. *Let $\mathcal{H}$ be an inner product space, $\varphi\colon \mathcal{X} \to \mathcal{H}$ and define $k(x, x') \coloneqq \langle \varphi(x), \varphi(x') \rangle$. Then k is a kernel.*

Proof. We have, for all $x_1, \dots, x_n \in \mathcal{X}$ and $\alpha \in \mathbb{R}^n$ that

$$\alpha^\top K \alpha = \sum_{i,j} K_{ij} \alpha_i \alpha_j = \sum_{i,j} \langle \varphi(x_i), \varphi(x_j) \rangle \alpha_i \alpha_j = \left\| \sum_i \alpha_i \varphi(x_i) \right\|^2 \geq 0.$$

□

The following proposition shows how to make new kernels from old:

Proposition 1.13. *Suppose $k_1, k_2, \dots$ are kernels. Then:*

1. *If $\alpha_1, \alpha_2 \geq 0$ then $\alpha_1 k_1 + \alpha_2 k_2$ is a kernel.*
2. *The pointwise limit of a sequence of kernels is a kernel (if it exists).*
3. *The pointwise product $k_1 k_2$ is a kernel.*

Proof. See Example Sheet 1. □

Example 1.14. Let us consider some examples of kernels:

1. For $\mathcal{X} = \mathbb{R}^p$ we have already seen the *linear kernel* $k(x, x') = x^\top x'$.
2. For $\mathcal{X} = \mathbb{R}^p$, the *polynomial kernel* is defined as $k(x, x') = (1 + x^\top x')^d$. This is a kernel since it is a power of a sum of two kernels.
3. For $\mathcal{X} = \mathbb{R}^p$, the *Gaussian kernel* is defined by

   $$k(x, x') = \exp\left(-\frac{\|x - x'\|_2^2}{2\sigma^2} \right).$$

   To show this is a kernel, write k as the pointwise product $k_1 k_2$ where

   $$k_1(x, x') = \exp\left(-\frac{\|x\|^2}{2\sigma^2} \right) \exp\left(-\frac{\|x'\|^2}{2\sigma^2} \right), \quad k_2(x, x') = \exp\left(\frac{x^\top x'}{\sigma^2} \right).$$

   Clearly k_1 is the kernel induced by the feature map $\varphi(x) = \exp\Big(-\|x\|^2/(2\sigma^2)\Big)$, while k_2 can be seen to be a kernel by using the Taylor expansion, which shows that k_2 is a limit of nonnegative linear combinations of kernels.
4. For $\mathcal{X} = [0, 1]$, define the *Sobolev kernel* $k(x, x') = \min(x, x')$. The proof that this is a kernel is on example sheet 1.
5. For $\mathcal{X} = \mathcal{P}(\{1, \dots, p\})$, define the *Jaccard kernel*

   $$k(x, x') = \frac{|x \cap x'|}{|x \cup x'|} \quad \text{where } 0/0 \coloneqq 1.$$

   The proof that this is a kernel is on example sheet 1.

By proposition 1.12, we see that every feature map $\varphi\colon \mathcal{X} \to \mathcal{H}$ gives rise to a kernel. In the next (important!) theorem, we will see that every kernel is in fact induced by a feature map.

Theorem 1.15. *Let k be a kernel, then there exists an inner product space $\mathcal{H}$ and a feature map $\varphi\colon \mathcal{X} \to \mathcal{H}$ such that*

$$k(x, x') = \langle \varphi(x), \varphi(x') \rangle \quad \text{for all } x, x' \in \mathcal{X}.$$

Proof. We will construct $\mathcal{H}$ and φ explicitly. First we define the function space

$$\mathcal{H} = \left\{ \sum_{i=1}^{n} \alpha_i k(\cdot, x_i) \mid n \in \mathbb{N}, \alpha_i \in \mathbb{R}, x_i \in \mathcal{X} \right\}.$$

Let $f = \sum_{i=1}^{n} \alpha_i k(\cdot, x_i)$ and $g = \sum_{j=1}^{n} \beta_j k(\cdot, x'_j)$, then the inner product on $\mathcal{H}$ is given by

$$\langle f, g \rangle = \left\langle \sum_{i=1}^{n} \alpha_i k(\cdot, x_i), \sum_{j=1}^{m} \beta_j k(\cdot, x'_j) \right\rangle \coloneqq \sum_{i=1}^{n} \sum_{j=1}^{m} \alpha_i \beta_j k(x_i, x'_j).$$

We define $\varphi\colon \mathcal{X} \to \mathcal{H}$ as $\varphi(x) = k(\cdot, x)$.

We must check that the inner product does not depend on the choice of representation of f and g. For this, note that

$$\sum_{i=1}^{n} \sum_{j=1}^{m} \alpha_i \beta_j k(x_i, x'_j) = \sum_{i=1}^{n} \alpha_i g(x_i) = \sum_{j=1}^{m} \beta_j f(x'_j),$$

which holds by symmetry of the kernel. Since $\sum_i \alpha_i g(x_i)$ is independent of the representation of g, while $\sum_j \beta_j f(x'_j)$ is independent of the representation of f, we conclude that the entire expression is independent of both representations.

Secondly, we must verify that the formula $k(x, x') = \langle \varphi(x), \varphi(x') \rangle$ indeed holds. For any $f \in \mathcal{H}, x \in \mathcal{X}$ we have

$$\langle k(\cdot, x), f \rangle = \sum_{i=1}^{n} \alpha_i k(x_i, x) = f(x), \tag{2}$$

i.e., evaluation of a function is a linear functional in $\mathcal{H}$.

In particular, we have

$$\langle \varphi(x), \varphi(x') \rangle = \langle k(\cdot, x), k(\cdot, x') \rangle = k(x, x').$$

Finally, we must check that $\langle \cdot, \cdot \rangle$ is indeed an inner product. Symmetry and bilinearity are clear. Furthermore, we have

$$\langle f, f \rangle = \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j k(x_i, x_j) = \alpha^\top K \alpha \geq 0$$

by the fact that k is a kernel. We must now only show that $f \neq 0 \implies \langle f, f \rangle > 0$. For this, note that $\langle \cdot, \cdot \rangle$ is a kernel on $\mathcal{H}$, so by proposition 1.11 (Cauchy-Schwarz) we have

$$f(x)^2 = \langle k(\cdot, x), f \rangle^2 \leq \langle k(\cdot, x), k(\cdot, x) \rangle \langle f, f \rangle,$$

and therefore if f is nonzero anywhere, $\langle f, f \rangle$ must also be nonzero. □

While $\mathcal{H}$ constructed in the proof is an inner product space, it is not necessarily a Hilbert space. Let $(f_n) \subseteq \mathcal{H}$ be Cauchy, then by Cauchy-Schwarz for kernels we find

$$f_m(x) - f_n(x) = (f_m - f_n)(x) = \langle k(\cdot, x), f_m - f_n \rangle \leq \sqrt{k(x, x)} \| f_n - f_m \|.$$

We can do an analogous computation for $f_n - f_m$ to conclude that $|f_m(x) - f_n(x)| \leq \sqrt{k(x, x)} \| f_n - f_m \|$, and therefore, if (f_n) is Cauchy, then it converges pointwise to some $f^*\colon \mathcal{X} \to \mathbb{R}$. We will not prove the following theorem:

Theorem 1.16. *The inner product space $\mathcal{H}$ constructed in the proof of theorem 1.15 can be extended to a Hilbert space by adding all pointwise limits f^* of Cauchy sequences $(f_n) \subseteq \mathcal{H}$.*

The completion of $\mathcal{H}$ is a special type of Hilbert space:

Definition 1.17. A Hilbert space $\mathcal{B}$ of functions $f\colon \mathcal{X} \to \mathbb{R}$ is called a *reproducing kernel Hilbert space* (RKHS) if for all $x \in \mathcal{X}$, there exists $k_x \in \mathcal{B}$ such that

$$f(x) = \langle k_x, f \rangle,$$

i.e., evaluation of functions is a linear functional.

The function $k(x, x') = \langle k_x, k_{x'} \rangle$ is known as the *reproducing kernel* of $\mathcal{B}$ (induced by the feature map $\varphi(x) = k_x$).

If we start with a kernel k, construct the corresponding RKHS $\mathcal{B}$, then it is easily checked that k is indeed the reproducing kernel of $\mathcal{B}$.

Example 1.18 (Linear kernel)**.** Let $X = \mathbb{R}^p$ and $k(x, x') = x^\top x'$. Then we have

$$\mathcal{H} = \left\{ x \mapsto \sum_{i=1}^{n} \alpha_i x^\top x_i = x^\top \left(\sum_i \alpha_i x_i \right) \mid \alpha_i \in \mathbb{R}, x_i \in \mathbb{R}^p \right\} = \left\{ x \mapsto x^\top \beta \mid \beta \in \mathbb{R}^p \right\},$$

and if $f(x) = x^\top \beta$, $g(x) = x^\top \beta'$, then

$$\langle f, g \rangle = k(\beta, \beta') = \beta^\top \beta' \quad \text{so } \|f\|_{\mathcal{H}} = \|\beta\|_2.$$