Regression and regularization

Matthieu R. Bloch

We now turn our attention to the problem of *regression*, which corresponds to the supervised learning setting when $\mathcal{Y} = \mathbb{R}$. Said differently, we will not attempt to learn a discrete label anymore as in classification but a continuously changing one. Classification is a special case of regression, but the discrete nature of labels lends itself to specific insights and analysis, which is why we studied it separately. Looking at regression will require the introduction of new concepts and will allow us to obtain new insights into the learning problem.

1 From classification to regression

As a refresher, the supervised learning problem we are interested in consists in using a labeled dataset $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$, $\mathbf{x}_i \in \mathbb{R}^d$ to predict the labels of unseen data. In classification, $y_i \in \mathcal{Y} \subset \mathbb{R}$ with $|\mathcal{Y}| \triangleq K < \infty$ while in regresssion $y_i \in \mathcal{Y} = \mathbb{R}$.

Our regression model is that the relation between label and data is of the form $y = f(\mathbf{x}) + n$ with $f \in \mathcal{H}$, where $\mathcal{H}$ is a class of functions (polynomials, splines, kernels, etc.), and n is some random noise.

Definition 1.1 (Linear regression). *Linear regression corresponds to the situation in which $\mathcal{H}$ is the set of affine functions*

$$f(\mathbf{x}) \triangleq \boldsymbol{\beta}^\intercal \mathbf{x} + \beta_0 \text{ with } \boldsymbol{\beta} \triangleq [\beta_1, \cdots, \beta_d]^\intercal \tag{1}$$

Definition 1.2 (Least square regression). *Least square regression corresponds to the situation in which the loss function is* sum of square errors

$$\text{SSE}(\boldsymbol{\beta}, \beta_0) \triangleq \sum_{i=1}^N (y_i - \boldsymbol{\beta}^\intercal \mathbf{x}_i - \beta_0)^2 \tag{2}$$

Linear least square regression is a widely used technique in applied mathematics, which can be traced back to the work of Legendre in *Nouvelles méthodes pour la détermination des orbites des comètes* (1805) and Gauss in *Theoria Motus* (1809, but claim to discovery in 1795).

We will make a change of notation to simplify our analysis moving forward. We set

$$\boldsymbol{\theta} \triangleq \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_d \end{bmatrix} \in \mathbb{R}^{d+1} \qquad \mathbf{y} \triangleq \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix} \in \mathbb{R}^N \qquad \mathbf{X} \triangleq \begin{bmatrix} 1 & -\mathbf{x}_1^\intercal - \\ 1 & -\mathbf{x}_2^\intercal - \\ \vdots & \vdots \\ 1 & -\mathbf{x}_N^\intercal - \end{bmatrix} \in \mathbb{R}^{N \times (d+1)}, \tag{3}$$

which allows us to rewrite the sum of square error as

$$\text{SSE}(\boldsymbol{\theta}) \triangleq \|\mathbf{y} - \mathbf{X}\boldsymbol{\theta}\|_2^2. \tag{4}$$

One of the reason that makes linear least square regression so popular is the existence of a closed form analytical solution.

Lemma 1.3 (Linear least square solution). *If* $\mathbf{X}^\intercal\mathbf{X}$ *is non singular the minimizer of the SSE is*

$$\hat{\boldsymbol{\theta}} = (\mathbf{X}^\intercal\mathbf{X})^{-1}\mathbf{X}^\intercal\mathbf{y} \tag{5}$$

Proof. See annotated slides. ∎

The existence of this solution is a bit misleading because computing $\hat{\boldsymbol{\theta}}$ can be extremely numerically unstable. The matrix $(\mathbf{X}^\intercal\mathbf{X})^{-1}$ could be ill-conditioned.

As for classification, linear methods have their limit, and one can create a non-linear estimator using a non-linear feature map $\Phi : \mathbb{R}^d \to \mathbb{R}^\ell : \mathbf{x} \mapsto \Phi(\mathbf{x})$. The regression model becomes

$$\mathbf{y} = \boldsymbol{\beta}^\intercal\Phi(\mathbf{x}) + \beta_0 \text{ with } \boldsymbol{\beta} \in \mathbb{R}^\ell. \tag{6}$$

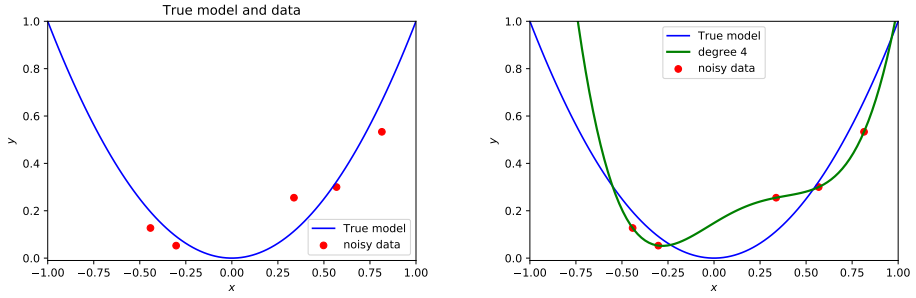
Example 1.4. *To obtain a least square estimate of cubic polynomial* f *with* $d = 1$*, one can use the non linear map*

$$\Phi : \mathbb{R} \to \mathbb{R}^4 : x \mapsto \begin{bmatrix} 1 \\ x \\ x^2 \\ x^3 \end{bmatrix}. \tag{7}$$

2 Overfitting and regularization

Overfitting is the problem that happens when fitting the data well no longer ensures that the out-of-sample error is small, i.e., the underlying model learned generalized poorly. This happens no only when there are too many degrees of freedom in model so that one "learns the noise " but also when the hypothesis set contains simpler functions than the target function f but the number of sample points N is too small. In general, *overfitting occurs as the number of features* d *begins to approach the number of observations* N.

To illustrate this, consider the following example in data is generated as $y = x^2 + n$ with $x \in [-1; 1]$, where $n \sim \mathcal{N}(0, \sigma = 0.1)$. We perform regression with polynomial of degree d. Fig. 1a shows the true underlying model and five samples obtained independently and uniformly at



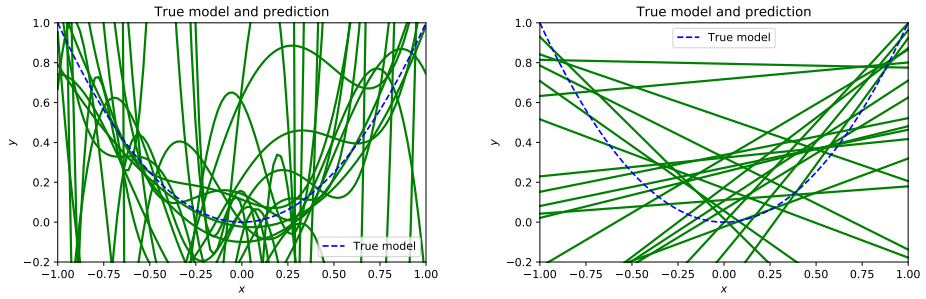
(a) True model and sample points.

(b) Regression fit with $d = 4$.

Figure 1: Illustration of overfitting

random. Fig. 1 shows the resulting predictor obtained by fitting the data to a polynomial of degree $d = 4$. Since we only have five points, there exists a degree four polynomial that predicts exactly the

value of all five training point. This is an example where our regression is effectively learning the noise in the model. To fully appreciate the consequence of overfitting, Fig. 2a shows the regression results for twenty randomly sampled sets of five points.

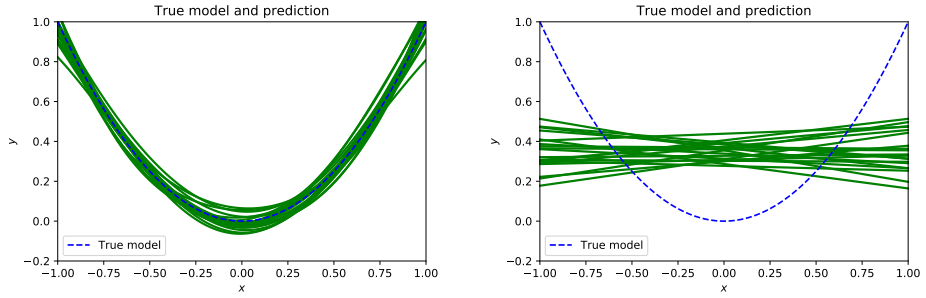


(a) Many regressions with $d = 4$ on five randomly sampled points.

(b) Many regressions with $d = 1$ on five randomly sampled points.

Figure 2: Regression with too few datapoints

As you can see, there is a huge *variance* in the resulting predictor, suggesting that we have an unstable prediction that does not generalize well. Perhaps surprisingly, one observes a similar variance when trying to fig the data to a polynomial of degree $d = 1$. In the latter situation, the degree of the polynomial is one less than the true model so that the model cannot fit the noise; however, the variance stems from the fact that there are few sample points. As shown in Fig. 3a and Fig. 3b, overfitting disappears once we have enough data points.



(a) Many regressions with $d = 4$ on fifty randomly sampled points.

(b) Many regressions with $d = 1$ on fifty randomly sampled points.

Figure 3: Regression with enough data points

In practice though, we are often interested in limiting overfitting even when the number of data points is small. The key solution is a technique called *regularization*.

3 Tikhonov regularization

The key idea behind regularization is to introduce a penalty term to "regularize" the vector $\boldsymbol{\theta}$:

$$\boldsymbol{\theta} = \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \|\mathbf{y} - \mathbf{X}\boldsymbol{\theta}\|_2^2 + \|\boldsymbol{\Gamma}\boldsymbol{\theta}\|_2^2 \tag{8}$$

where $\boldsymbol{\Gamma} \in \mathbb{R}^{(d+1)\times(d+1)}$

Lemma 3.1 (Tikhonov regularization solution)**.** *The minimizer of the least-square problem with Tikhonov regularization is*

$$\hat{\boldsymbol{\theta}} = (\mathbf{X}^\intercal\mathbf{X} + \boldsymbol{\Gamma}^\intercal\boldsymbol{\Gamma})^{-1}\mathbf{X}^\intercal\mathbf{y} \tag{9}$$

Proof. See annotated slides. ∎

For the special case $\boldsymbol{\Gamma} = \sqrt{\lambda}\mathbf{I}$ for some $\lambda > 0$, we obtain

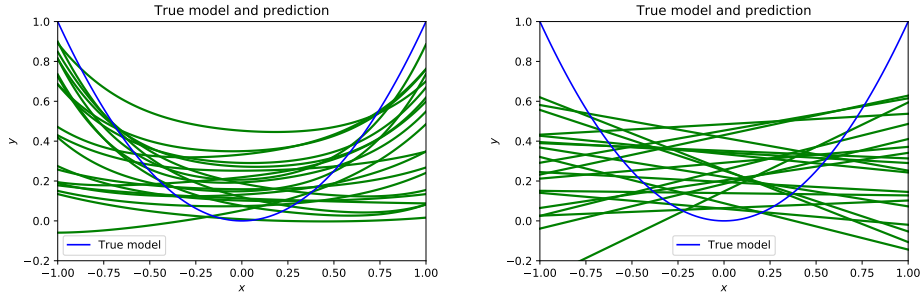
$$\hat{\boldsymbol{\theta}} = (\mathbf{X}^\intercal\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^\intercal\mathbf{y} \tag{10}$$

This simple change has many benefits, including improving numerical stability when computing $\hat{\boldsymbol{\theta}}$ since $\mathbf{X}^\intercal\mathbf{X} + \lambda\mathbf{I}$ is better conditioned than $\mathbf{X}^\intercal\mathbf{X}$.

Ridge regression is a slight variant of the above that does not penalize β_0 and corresponds to

$$\boldsymbol{\Gamma} = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ 0 & \sqrt{\lambda} & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & \cdots & \sqrt{\lambda} \end{bmatrix} \tag{11}$$

To appreciate the effect of regularization. Fig. 4 shows the resulting regressions with $\lambda = 1$ in the same situation as earlier. Notice how the variance of the regression is substantially reduced.



(a) Many ridge regressions with $d = 4$ on five randomly sampled points.

(b) Many ridge regressions with $d = 1$ on five randomly sampled points.

Figure 4: Ridge regression

It is also useful to understand Tikhonov regularization as a constrained optimization problem.

Lemma 3.2 (Tikhonov regularization solution revisited). *The minimizer of the least-square problem with Tikhonov regularization is the solution of*

$$\underset{\boldsymbol{\theta}}{argmin}\, \|\mathbf{y} - \mathbf{X}\boldsymbol{\theta}\|_2^2 \text{ such that } \|\boldsymbol{\Gamma}\boldsymbol{\theta}\|_2^2 \leqslant \tau \tag{12}$$

for some $\tau > 0$

Proof. See annotated slides. ∎

Fig. 5 illustrates the effect of Tikhonov regularization in $\mathbb{R}^2$ assuming that $\boldsymbol{\Gamma} = \mathbf{I}$. The Tikhonov solution is shrinked towards the zero vector to satisfy the constraint. Intuitively, the regularized solution corresponds to the point where the level set of $\|y - \mathbf{X}\boldsymbol{\theta}\|_2$ first intersects the feasible region $\|\boldsymbol{\theta}\|_2^2 = \tau$.

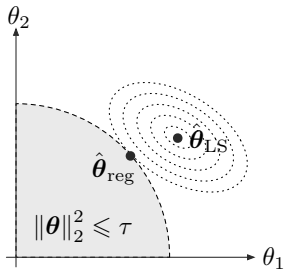


Figure 5: Illustration of Tikhonov regularization

4 Shrinkage estimators

The Tikhonov regularization previously introduced is a *shrinkage* estimator, in the sense that its shrinks a naive estimate towards some guess. As illustrated in Fig. 5, one can think of the regularization as shrinking the least-square estimate $\boldsymbol{\theta}_{\text{LS}}$ towards zero.

Shrinkage estimators are arguably a bit strange, especially because it may not be clear priori how biasing an estimate towards a guess would bring any benefit. The intuition you should have is that the shrinkage often leads to a lower variance of the estimator, perhaps at the expense of an increase in the bias. The example below illustrates this idea in a simple situation.

Example 4.1 (Estimating the variance). *Let* $\{x_i\}_{i=1}^N$ *be iid samples drawn according to unknown distribution with variance* σ^2*. Consider two estimators of the variance*

$$\hat{\sigma}^2_{\textit{biased}} \triangleq \frac{1}{N}\sum_{i=1}^{N}(\mathbf{x}_i - \hat{\mu})^2 \qquad \hat{\sigma}^2_{\textit{unbiased}} \triangleq \frac{1}{N-1}\sum_{i=1}^{N}(\mathbf{x}_i - \hat{\mu})^2 \tag{13}$$

As the names suggest, it is not hard to show that

$$\mathbb{E}\big(\hat{\sigma}^2_{\textit{biased}}\big) = \frac{N-1}{N}\sigma^2 \qquad \mathbb{E}\big(\hat{\sigma}^2_{\textit{unbiased}}\big) = \sigma^2 \tag{14}$$

Perhaps surprisingly one can also show that the biased estimate has a lower variance than the unbiased one.

Tykhonov regularization is not the only regularizer that one can use, and statisticians have developed many alternative regularizers, such as

- Akaike information criterion (AIC): penalty that is an increasing function of the number of estimated parameters;
- Bayesian information criterion (BIC): also an increasing function of the number of estimated parameters.

One regularize that has become increasingly popular is the Least Absolute Shrinkage and Selection Operator (LASSO) estimator

$$\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \|\mathbf{y} - \mathbf{X}\boldsymbol{\theta}\|_2^2 + \lambda \|\boldsymbol{\theta}\|_1. \tag{15}$$

With LASSO, coordinates are shrunk by the same amount. The main benefit of LASSO is that it promotes sparsity, i.e., solutions tend to have a small number of non-zero components. The problem is also much more computationally tractable than directly enforcing sparsity through a $\|\boldsymbol{\theta}\|_0$ constraint. In constrained form, the LASSO estimation problem becomes

$$\boldsymbol{\theta} = \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \|\mathbf{y} - \mathbf{X}\boldsymbol{\theta}\|_2^2 \text{ such that } \|\boldsymbol{\theta}\|_1 \leqslant \tau, \tag{16}$$

for which there is no closed form solution but which is very powerful when$n \ll d$ (susceptible to overfitting) or $\mathbf{X}$ has non trivial null space and there is no obvious way to find the best solution. The intuition behind LASSO is illustrated in Fig. 6, the norm 1 ball $\{\mathbf{x} : \|\mathbf{x}\|_1 \leqslant \tau\}$ is more "pointy" than the norm 2 ball, which increase the likelihood of the regularization to yield an extreme point with few non-zero components.

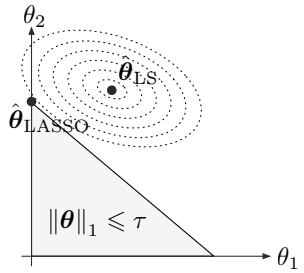


Figure 6: Illustration of LASSO regularization.

5 General approach to regression

Consider now a general approach to regression, in which we define our estimate as

$$\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\operatorname{argmin}}\, L(\boldsymbol{\theta}, \mathbf{X}, \mathbf{y}) + \lambda r(\boldsymbol{\theta}) \tag{17}$$

where $L : \mathbb{R}^{d+1} \times \mathbb{R}^{N\times(d+1)} \times \mathbb{R}^N \to \mathbb{R}$ is a *loss function* and $r : \mathbb{R}^{d+1} \to \mathbb{R}$ is a *regularizer*. Intuitively, the role of the loss function it to ensure data fidelity while the role of the regularizer is to limit the complexity of the solution.

Among the many possible choices of loss functions, we highlight three that are fairly popular:

- the mean absolute error $L_{\text{AE}}(r) \triangleq |r|$;
- the Huber loss $L_{\text{H}}(r) = \begin{cases} \frac{1}{2}r^2 \text{ if } |r| \leqslant c \\ c\,|r| - \frac{c^2}{2} \text{ else} \end{cases}$;
- the ϵ-insensitive loss $L_\epsilon(r) = \begin{cases} 0 \text{ if } |r| \leqslant \epsilon \\ |r| - \epsilon \text{ else} \end{cases}$.

These losses are illustrated in Fig. 7.

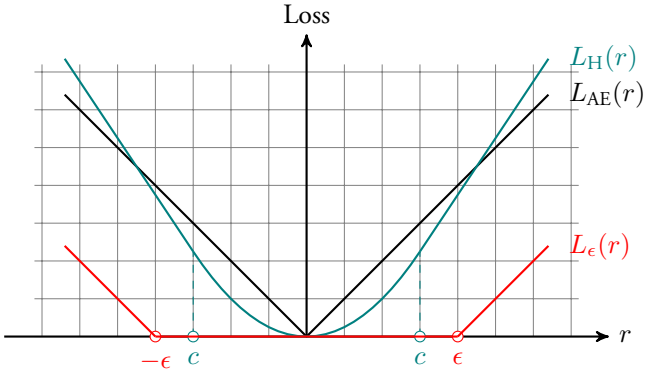


Figure 7: Illustration of loss functions

Let us now investigate one specific example called the *regularized robust regression*, which combines the ϵ-insensitive loss with an ℓ_2 regularizer as

$$\hat{\boldsymbol{\beta}}, \hat{\beta}_0 = \underset{\boldsymbol{\beta}, \beta_0}{\text{argmin}} \sum_{i=1}^{N} L_\epsilon(y_i - (\boldsymbol{\beta}^\intercal \mathbf{x}_i + \beta_0)) + \frac{\lambda}{2}\|\boldsymbol{\beta}\|_2^2 \tag{18}$$

This ϵ-insensitive loss does not incur a penalty as long as prediction is within a margin of ϵ of the true value. This is suspiciously similar to what the problem we were solving when looking at Support Vector Machines (SVMs), and one can therefore wonder if we could kernelize the regression problem. The answer turns out to be affirmative as shown by the characterization of the dual problem of (18).

Lemma 5.1 (Dual of regularized robust regression). *Robust regression is the solution of*

$$\underset{\boldsymbol{\alpha}, \boldsymbol{\alpha}^*}{\textit{argmin}} \sum_{i=1}^{N} ((\epsilon - y_i)\alpha_i + (\epsilon + y_i)\alpha_i^*) + \frac{1}{2}\sum_{i,j}(\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*)\mathbf{x}_j^\intercal \mathbf{x}_i \tag{19}$$

such that

$$0 \leqslant \alpha_i^*, \alpha_i \leqslant \frac{1}{\lambda}, \quad \sum_{i=1}^{N}(\alpha_i^* - \alpha_i) = 0, \quad \alpha_i^* \alpha_i = 0 \tag{20}$$

The solution is then $\hat{y} = \sum_{i=1}^{N}(\alpha_i - \alpha_i^*)\mathbf{x}_i^\intercal \mathbf{x} - \beta_0$.

Proof. See for yourselves! ∎

The previous kernelization of the ϵ-robust regression through the formulation of the dual problem is a bit misleading because the kernelization is in general not straightforward. We highlight next two examples for which the kernelization can be worked out.

6 Kernelized LASSO

The LASSO regression is given by

$$\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\text{argmin}} \, \|\mathbf{y} - \mathbf{X}\boldsymbol{\theta}\|_2^2 + \|\boldsymbol{\theta}\|_1,$$

which is not easy to kernerlize directly. Instead, we modify the problem and and enforce the kernelization by looking for a *specific* $\boldsymbol{\theta}$ of the form $\tilde{\boldsymbol{\theta}} \triangleq \sum_i \alpha_i \mathbf{x}_i$ for $\{\alpha_i\}_{i=1}^d$ and solve the optimization problem

$$\hat{\boldsymbol{\alpha}} = \underset{\boldsymbol{\alpha}}{\text{argmin}} \, \|\mathbf{y} - \mathbf{X}\tilde{\boldsymbol{\theta}}\|_2^2 + \|\boldsymbol{\alpha}\|_1$$

This can be kernelized but promotes sparsity in $\boldsymbol{\alpha}$ instead of $\boldsymbol{\theta}$, which is not quite the original LASSO problem.

7 Kernel ridge regression

One example we can explore in much more details is kernel ridge regression. The first step towards kernelization consists in reformulating the objective of ridge regression as follows.

Lemma 7.1. *The ridge regression problem is equivalent to*

$$\hat{\boldsymbol{\beta}}, \hat{\beta}_0 = \underset{\boldsymbol{\beta}, \beta_0}{\textit{argmin}} \sum_{i=1}^N \|y_i - \bar{y} - \boldsymbol{\beta}^\mathsf{T}(\mathbf{x}_i - \bar{\mathbf{x}})\|_2^2 + \lambda \|\boldsymbol{\beta}\|_2^2 \tag{21}$$

with $\bar{\mathbf{x}} = \frac{1}{N}\sum_i \mathbf{x}_i$ *and* $\bar{y} = \frac{1}{N}\sum_i y_i$. *The prediction for an unseen* $\mathbf{x}$ *is then computed as*

$$\hat{y} = \bar{y} + \hat{\boldsymbol{\beta}}^\mathsf{T}(\mathbf{x} - \bar{\mathbf{x}}) \tag{22}$$

Proof. Recall that the original ridge regression problem is

$$\hat{\boldsymbol{\beta}}, \hat{\beta}_0 = \underset{\boldsymbol{\beta}, \beta_0}{\text{argmin}} \underbrace{\frac{1}{N} \sum_{i=1}^N \|y_i - \boldsymbol{\beta}^\mathsf{T}\mathbf{x}_i - \beta_0\|_2^2 + \lambda \|\boldsymbol{\beta}\|_2^2}_{\triangleq \mathcal{L}} \tag{23}$$

Applying the stationarity condition of the Karush-Kuhn Tucker (KKT) conditions to this problem, we must have

$$\frac{\partial \mathcal{L}}{\partial \beta_0} = 0 \Leftrightarrow -\frac{2}{N}\sum_{i=1}^N (y_i - \boldsymbol{\beta}^\mathsf{T}\mathbf{x}_i - \beta_0) = 0 \tag{24}$$

so that

$$\beta_0 \triangleq \frac{1}{N}\sum_{i=1}^{N} y_i - \boldsymbol{\beta}^\mathsf{T}\left(\frac{1}{N}\sum_{i=1}^{N}\mathbf{x}_i\right) \triangleq \bar{y} - \boldsymbol{\beta}^\mathsf{T}\bar{\mathbf{x}}. \tag{25}$$

Substituting this optimal value of β_0 in (25) into the optimization problem (23), we obtain (21). The evaluation $y = \boldsymbol{\beta}^\mathsf{T}\mathbf{x} + \beta_0$ with (25) yields (22). ∎

Note that the optimization problem in Lemma 7.1 is a ridge regression problem for a centered data set, in which the value y_i are replaced by $y_i - \bar{y}$ and in which the feature vectors $\mathbf{x}_i$ are replaced by $\mathbf{x}_i - \bar{\mathbf{x}}$. Using the known analytical solution for ridge regression developed in a previous lecture, we immediately obtain

$$\hat{\boldsymbol{\beta}} = (\mathbf{A}^\mathsf{T}\mathbf{A} + \lambda\mathbf{I})^{-1}\mathbf{A}^\mathsf{T}\tilde{\mathbf{y}}$$

where

$$\mathbf{A} \triangleq \begin{bmatrix} (\mathbf{x}_1 - \bar{\mathbf{x}})^\mathsf{T} \\ (\mathbf{x}_2 - \bar{\mathbf{x}})^\mathsf{T} \\ \vdots \\ (\mathbf{x}_N - \bar{\mathbf{x}})^\mathsf{T} \end{bmatrix} \qquad \tilde{\mathbf{y}} \triangleq \begin{bmatrix} y_1 - \bar{y} \\ y_2 - \bar{y} \\ \vdots \\ y_N - \bar{y} \end{bmatrix}.$$

Unfortunately, one cannot directly kernelize this solution because the matrix $\mathbf{A}^\mathsf{T}\mathbf{A}$ does not contain inner products between feature vectors. One can, however, rewrite the solution using the Woodbury matrix inversion identity given below.

Lemma 7.2 (Woodbury matrix inversion identity)**.**

$$(\mathbf{P} + \mathbf{QRS})^{-1} = \mathbf{P}^{-1} - \mathbf{P}^{-1}\mathbf{Q}(\mathbf{R}^{-1} + \mathbf{SP}^{-1}\mathbf{Q})^{-1}\mathbf{SP}^{-1}$$

The main benefit of this identity is to switch the order in which the matrix are multiplied. In our context, we use this to introduce inner products between feature vector in the expression of the solution for $\boldsymbol{\beta}$.

Lemma 7.3.

$$\hat{\boldsymbol{\beta}} = \frac{1}{\lambda}(\mathbf{A}^\mathsf{T} - \mathbf{A}^\mathsf{T}(\lambda\mathbf{I} + \mathbf{K})^{-1}\mathbf{K})\tilde{\mathbf{y}} \text{ with } \mathbf{K} \triangleq \mathbf{AA}^\mathsf{T} = [(\mathbf{x}_i - \bar{\mathbf{x}})^\mathsf{T}(\mathbf{x}_j - \bar{\mathbf{x}})]_{i,j}$$

and

$$\hat{\mathbf{y}} = \bar{\mathbf{y}} + \frac{1}{\lambda}\tilde{\mathbf{y}}^\mathsf{T}(\mathbf{I} - \mathbf{K}(\lambda\mathbf{I} + \mathbf{K})^{-1}\mathbf{I})\mathbf{k}(\mathbf{x}) \text{ with } \mathbf{k}(\mathbf{x}) = \begin{bmatrix} (\mathbf{x}_1 - \bar{\mathbf{x}})^\mathsf{T}(\mathbf{x} - \bar{\mathbf{x}}) \\ \vdots \\ (\mathbf{x}_N - \bar{\mathbf{x}})^\mathsf{T}(\mathbf{x} - \bar{\mathbf{x}}) \end{bmatrix}$$

Proof. We use Lemma 7.2 with the choice of matrices

$$\mathbf{P} \triangleq \lambda\mathbf{I} \quad \mathbf{Q} \triangleq \mathbf{A}^\mathsf{T} \quad \mathbf{R} = \mathbf{I} \quad \mathbf{S} = \mathbf{A}, \tag{26}$$

for which we obtain

$$(\lambda\mathbf{I} + \mathbf{A}^\mathsf{T}\mathbf{A})^{-1} = \frac{1}{\lambda}\mathbf{I} - \frac{1}{\lambda}\mathbf{I}\mathbf{A}^\mathsf{T}(\mathbf{I} + \mathbf{A}\frac{1}{\lambda}\mathbf{A}^\mathsf{T})^{-1}\mathbf{A}\frac{1}{\lambda}\mathbf{I} \tag{27}$$

$$= \frac{1}{\lambda}\left[\mathbf{I} - \mathbf{A}^\mathsf{T}(\lambda\mathbf{I} + \mathbf{AA}^\mathsf{T})^{-1}\mathbf{A}\right]. \tag{28}$$

Consequently,

$$(\lambda \mathbf{I} + \mathbf{A}^\intercal \mathbf{A})^{-1} \mathbf{A}^\intercal \tilde{\mathbf{y}} = \frac{1}{\lambda} \left(\mathbf{A}^\intercal - \mathbf{A}^\intercal (\lambda \mathbf{I} + \mathbf{A}\mathbf{A}^\intercal)^{-1} \mathbf{A}\mathbf{A}^\intercal \right) \tilde{\mathbf{y}} \tag{29}$$

$$= \frac{1}{\lambda} \mathbf{A}^\intercal \left(\mathbf{I} - (\lambda \mathbf{I} + \mathbf{K})^{-1} \mathbf{K} \right) \tilde{\mathbf{y}}, \tag{30}$$

and the evaluation $\hat{y} - \bar{y} = \boldsymbol{\beta}^\intercal (\mathbf{x} - \bar{\mathbf{x}})$ from (22) yields

$$\hat{y} - \bar{y} = \tilde{\mathbf{y}}^\intercal \left(\mathbf{I} - \mathbf{K} (\lambda \mathbf{I} + \mathbf{K})^{-1} \right) \mathbf{A} (\mathbf{x} - \bar{\mathbf{x}}), \tag{31}$$

where we have used the fact that $\mathbf{K}^\intercal = \mathbf{K}$. The result follows from the definition of $\mathbf{k}(\mathbf{x})$. ∎

Notice that the evaluation of $\hat{y}$ in Lemma 7.3 only involves inner products between centered feature vectors in $\mathbf{K}$ and $\mathbf{k}(\mathbf{x})$. One kernelize the expression by substituting every inner product $(\mathbf{x}_j - \bar{\mathbf{x}})^\intercal (\mathbf{x} - \bar{\mathbf{x}})$ with $k(\mathbf{x}_j - \bar{\mathbf{x}}, \mathbf{x} - \bar{\mathbf{x}})$ for some inner product kernel $k(\cdot, \cdot)$.

Remark 7.4. *For many kernels, the map $\Phi(\mathbf{x})$ already contains a constant component so that we often omit β_0. The expression of kernel ridge regression without offset then simplifies to*

$$\hat{\mathbf{y}} = \mathbf{y}^\intercal (\lambda \mathbf{I} + \mathbf{K})^{-1} \mathbf{k}(\mathbf{x})$$

where

$$\mathbf{k}(\mathbf{x}) = \begin{bmatrix} k(\mathbf{x}_1, \mathbf{x}) & \cdots & k(\mathbf{x}_N, \mathbf{x}) \end{bmatrix}^\intercal \qquad \mathbf{K}_{i,j} = k(\mathbf{x}_i, \mathbf{x}_j).$$

Example 7.5 (Gaussian kernel). *A widely used kernel is the Gaussian kernel defined as*

$$k(\mathbf{u}, \mathbf{v}) \triangleq \exp\left(-\frac{\|\mathbf{u} - \mathbf{v}\|_2^2}{2\sigma^2} \right). \tag{32}$$

The parameter σ is called the width of the kernel and can be thought of as controlling how smooth we want the learned function to be. To visualize this, note that $\mathbf{y}^\intercal (\lambda \mathbf{I} + \mathbf{K})^{-1}$ is merely a row vector $\boldsymbol{\alpha}^\intercal = [\alpha_1, \cdots, \alpha_N]$, so that $\hat{y} = \sum_{i=1}^N \alpha_i k(\mathbf{x}_i, \mathbf{x})$. One can interpret this as writing our approximation $\hat{y}$ as a weighted sum of bell-shaped curves placed around each data point as shown in Fig. 8.

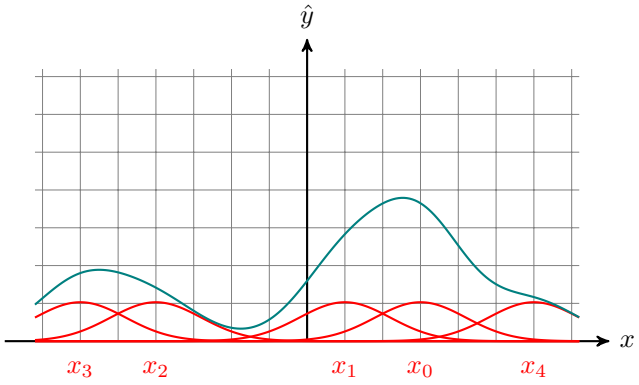


Figure 8: Illustration of Gaussian kernel ridge regression.

8 Regularization and classification

Since classification is a special case of regression, the viewpoint of loss function and regularization also applies to classification. As a simple application, the logistic regression $\min_\theta -\ell(\theta)$ can be regularized as $\min_\theta -\ell(\theta) + \lambda\|\theta\|_2^2$ to make the Hessian matrix well conditioned in a Newton method. This is also very useful when the number of observations is small and the data not separable.

Most importantly, the loss function and regularization framework offers a new way to think about classification. In fact, when trying to separate two classes with a hyperplane, we would ideally like to solve the optimization problem

$$\operatorname*{argmin}_{\mathbf{w},b} \frac{1}{N}\sum_{i=1}^{N} \mathbb{1}\{y_i(\mathbf{w}^\intercal \mathbf{x}_i + b) < 0\},$$

which assigns a penalty of one to every misclassified point and zero otherwise. Unfortunately, this is a hard problem to solve not only analytically but also numerically because the indicator function $\mathbb{1}\{\cdot\}$ is not well behaved. Viewing the indicator function as a $\{0,1\}$-loss, we now know that we can use other loss functions; in particular, we can choose a loss function $\phi(t) \geqslant \mathbb{1}\{t<0\}$ and a regularization to solve a relaxed but well-behaved modified optimization

$$\operatorname*{argmin}_{\mathbf{w},b} \frac{1}{N}\sum_{i=1}^{N} \phi\left(y_i(\mathbf{w}^\intercal \mathbf{x}_i + b)\right) + \lambda\|\mathbf{w}\|_2^2.$$

There are many possible choices, but a popular choice is the *hinge loss* $\phi(t) \triangleq \max(0, 1-t) \triangleq (1-t)_+$ illustrated in Fig 9.

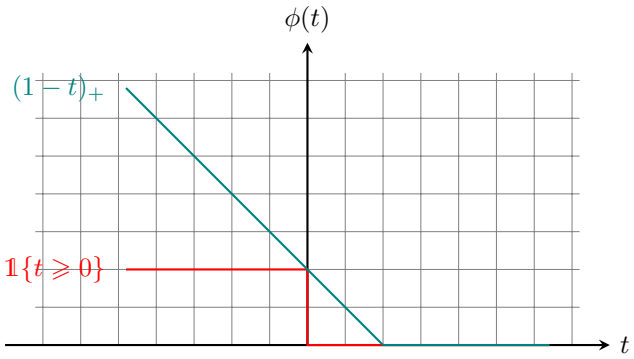


Figure 9: Hinge loss

The optimization problem with the hinge loss then becomes

$$\operatorname*{argmin}_{\mathbf{w},b} \frac{1}{N}\sum_{i=1}^{N} (1 - y_i(\mathbf{w}^\intercal \mathbf{x}_i + b))_+ + \lambda\|\mathbf{w}\|_2^2.$$

We can introduce slack variables $\xi_i = 1 - y_i(\mathbf{w}^\intercal \mathbf{x}_i + b)$ to rewrite the problem equivalently as

$$\operatorname*{argmin}_{\mathbf{w},b,\boldsymbol{\xi}\geqslant\mathbf{0}} \frac{1}{N}\sum_{i=1}^{N} \xi_i + \lambda\|\mathbf{w}\|_2^2 \text{ such that } \forall i \in [\![1,N]\!]\; y_i(\mathbf{w}^\intercal \mathbf{x}_i + b) \geqslant 1 - \xi_i, \tag{33}$$

which is exactly the soft-margin hyperplane seen in previous lectures.

References