# Regression recap

Recall that in regression we are given training data

$$(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)$$

where $\mathbf{x}_i \in \mathbb{R}^d$ and $y_i \in \mathbb{R}$

In linear regression we assume that we are trying to estimate a function of the form

$$f(\mathbf{x}) = \boldsymbol{\beta}^T \mathbf{x} + \beta_0$$

where $\boldsymbol{\beta} \in \mathbb{R}^d$, $\beta_0 \in \mathbb{R}$

**Least squares regression:** Select $\boldsymbol{\beta}, \beta_0$ to minimize

$$\mathrm{SSE}(\boldsymbol{\beta}, \beta_0) := \sum_{i=1}^{n} \left(y_i - \boldsymbol{\beta}^T \mathbf{x}_i - \beta_0\right)^2$$

# Least squares

$$\boldsymbol{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} \quad \boldsymbol{A} = \begin{bmatrix} 1 & x_1(1) & \cdots & x_1(d) \\ 1 & x_2(1) & \cdots & x_2(d) \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_n(1) & \cdots & x_n(d) \end{bmatrix} \quad \boldsymbol{\theta} = \begin{bmatrix} \beta_0 \\ \beta(1) \\ \vdots \\ \beta(d) \end{bmatrix}$$

$$\mathsf{SSE}(\boldsymbol{\theta}) = \sum_{i=1}^{n} (y_i - \boldsymbol{\beta}^T \mathbf{x}_i - \beta_0)^2 = \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{\theta}\|_2^2$$

Minimizer given by

$$\widehat{\boldsymbol{\theta}} = \left(\boldsymbol{A}^T \boldsymbol{A}\right)^{-1} \boldsymbol{A}^T \boldsymbol{y}$$

provided that $\boldsymbol{A}^T \boldsymbol{A}$ is ***nonsingular***

# Regularization and regression

Overfitting occurs as the number of features $d$ begins to approach the number of observations $n$

In this regime, we have *too many degrees of freedom*

**Idea:** penalize candidate solutions for using too many features

One candidate regularizer: $r(\boldsymbol{\theta}) = \|\boldsymbol{\theta}\|_2^2$

$$\widehat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{\theta}\|_2^2 + \lambda \|\boldsymbol{\theta}\|_2^2$$

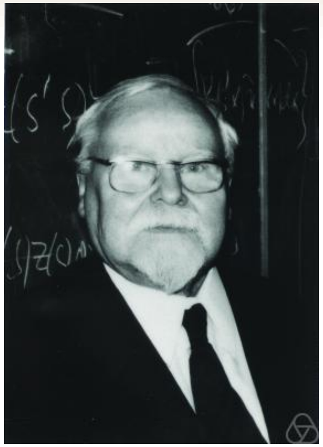
$\lambda > 0$ is a "tuning parameter" that controls the tradeoff between fit and complexity

# Tikhonov regularization

This is one example of a more general technique called ***Tikhonov regularization***

$$\widehat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{\theta}\|_2^2 + \|\boldsymbol{\Gamma}\boldsymbol{\theta}\|_2^2$$

(Note that $\lambda$ has been replaced by the matrix $\boldsymbol{\Gamma}$)



**Solution:**

$$\widehat{\boldsymbol{\theta}} = \left(\boldsymbol{A}^T\boldsymbol{A} + \boldsymbol{\Gamma}^T\boldsymbol{\Gamma}\right)^{-1}\boldsymbol{A}^T\boldsymbol{y}$$

$$\boldsymbol{\Gamma} = \sqrt{\lambda}\boldsymbol{I} \qquad \widehat{\boldsymbol{\theta}} = (\underbrace{\boldsymbol{A}^T\boldsymbol{A} + \lambda\boldsymbol{I}}_{\text{for suitable choice of } \lambda\text{, always well-conditioned}})^{-1}\boldsymbol{A}^T\boldsymbol{y}$$

# Ridge regression

In the context of regression, Tikhonov regularization has a special name: ***ridge regression***

Ridge regression is essentially exactly what we have been talking about, but in the special case where

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

We are penalizing all coefficients in $\beta$ equally, but not penalizing the offset $\beta_0$

# Alternative regularizers

- Akaike information criterion (AIC)
- Bayesian information criterion (BIC)

$$r(\boldsymbol{\theta}) \approx \|\boldsymbol{\theta}\|_0 := |\mathrm{supp}(\boldsymbol{\theta})|$$

- Least absolute shrinkage and selection operator (LASSO)

$$r(\boldsymbol{\theta}) = \|\boldsymbol{\theta}\|_1 = \sum_j |\theta(j)|$$

  - also results in shrinkage, but where all coordinates are shrunk by the same amount (in case of an orthobasis)
  - promotes sparsity
  - can think of $\|\boldsymbol{\theta}\|_1$ as a more computationally tractable replacement for $\|\boldsymbol{\theta}\|_0$

# The LASSO

*LASSO*

$$\widehat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{\theta}\|_2^2 + \lambda \|\boldsymbol{\theta}\|_1$$

Can also be stated in a constrained form

$$\widehat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{\theta}\|_2^2 \quad \text{s.t.} \quad \|\boldsymbol{\theta}\|_1 \leq \tau$$

$$\widehat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \|\boldsymbol{\theta}\|_1 \quad \text{s.t.} \quad \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{\theta}\|_2^2 \leq \sigma$$

For Tikhonov, we have a closed form solution, but LASSO requires solving an optimization problem

**Note:** Just like in ridge regression, in practice we may just want to penalize the elements of $\boldsymbol{\beta}$ (not $\beta_0$ )

# Sparsity and the LASSO

One can show (see supplemental notes) that if we have a data set of size $n$, then the solution to the LASSO $\widehat{\boldsymbol{\theta}}$ will have at most $n$ nonzeros (for any possible dataset / $\boldsymbol{A}$ )

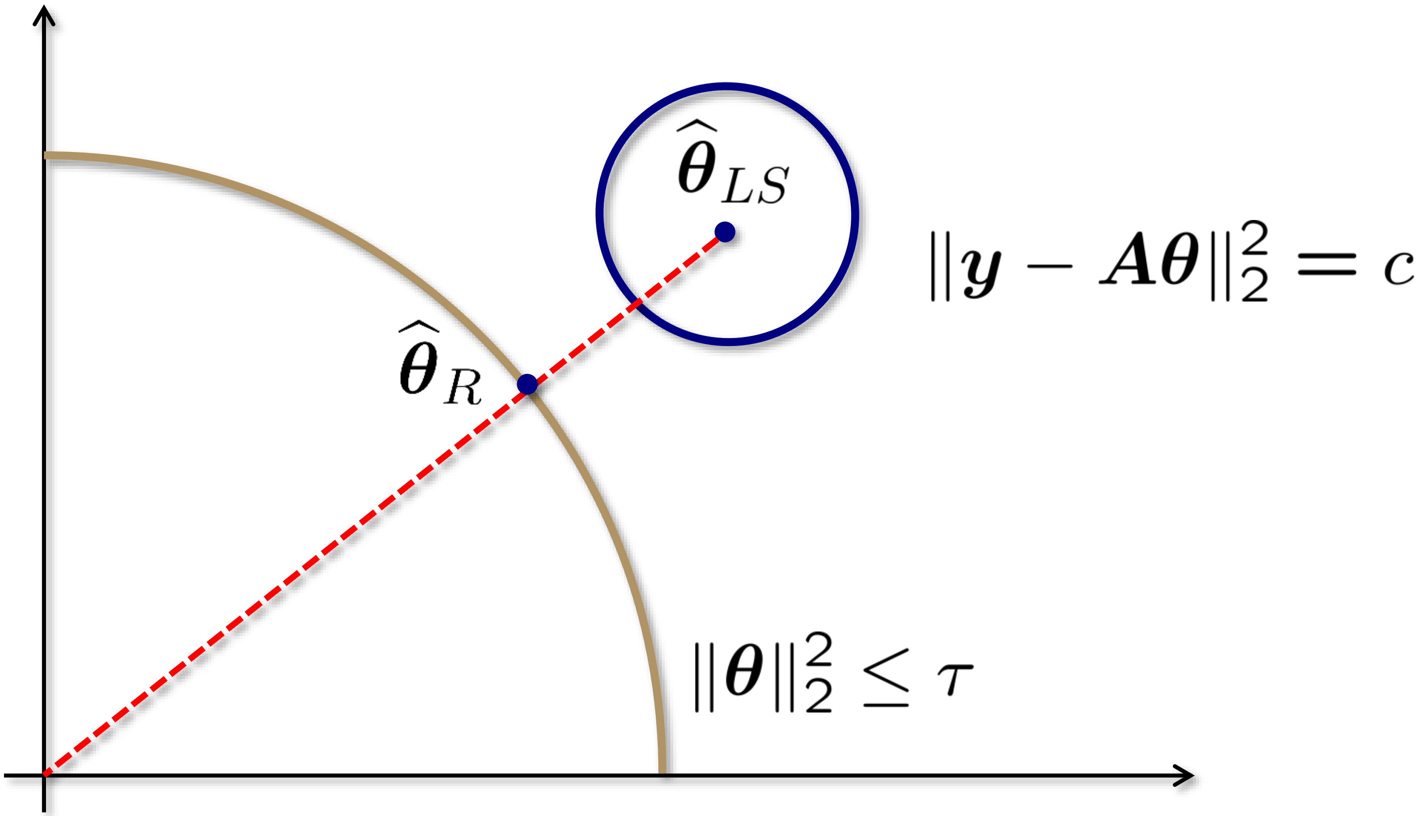
This is a nice property when $n \ll d$, since in this setting we are ***very*** susceptible to overfitting

- fewer observations than unknowns
- $\boldsymbol{A}$ has nontrivial nullspace
- we can achieve $\boldsymbol{y} = \boldsymbol{A\theta}$, with infinitely many different choices of $\boldsymbol{\theta}$ and no obvious way to know which one is best
- limiting the number of nonzeros addresses this problem

In practice, the number of nonzeros is usually much smaller than $n$

# Tikhonov versus least squares

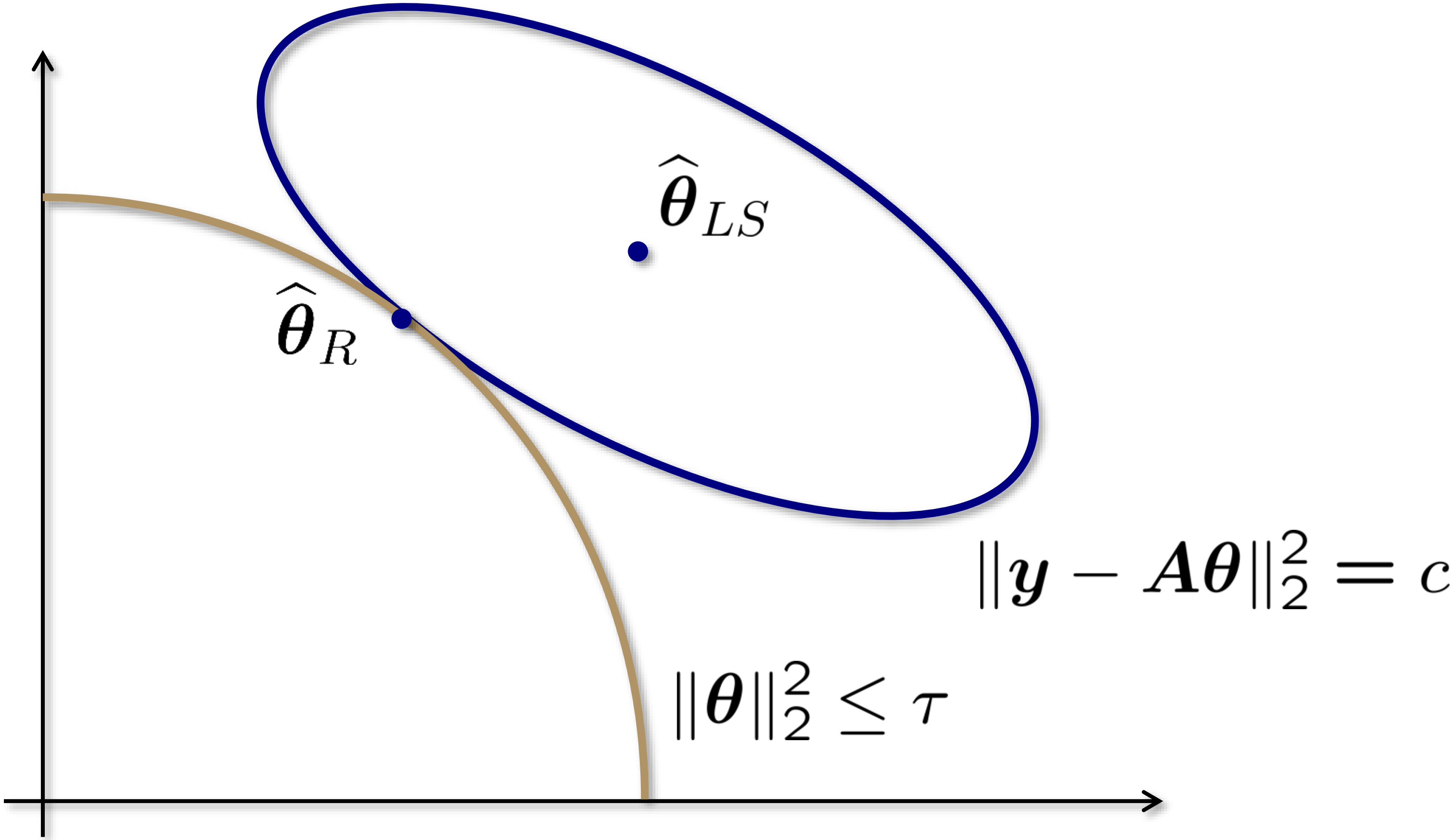
Assume $\boldsymbol{\Gamma} = \boldsymbol{I}$ and that $\boldsymbol{A}$ has orthonormal columns


$\widehat{\boldsymbol{\theta}}_{LS}$
$\|\boldsymbol{y} - \boldsymbol{A\theta}\|_2^2 = c$
$\widehat{\boldsymbol{\theta}}_R$
$\|\boldsymbol{\theta}\|_2^2 \leq \tau$

Tikhonov regularization is equivalent to shrinking the least squares solution towards the origin

# Tikhonov versus least squares

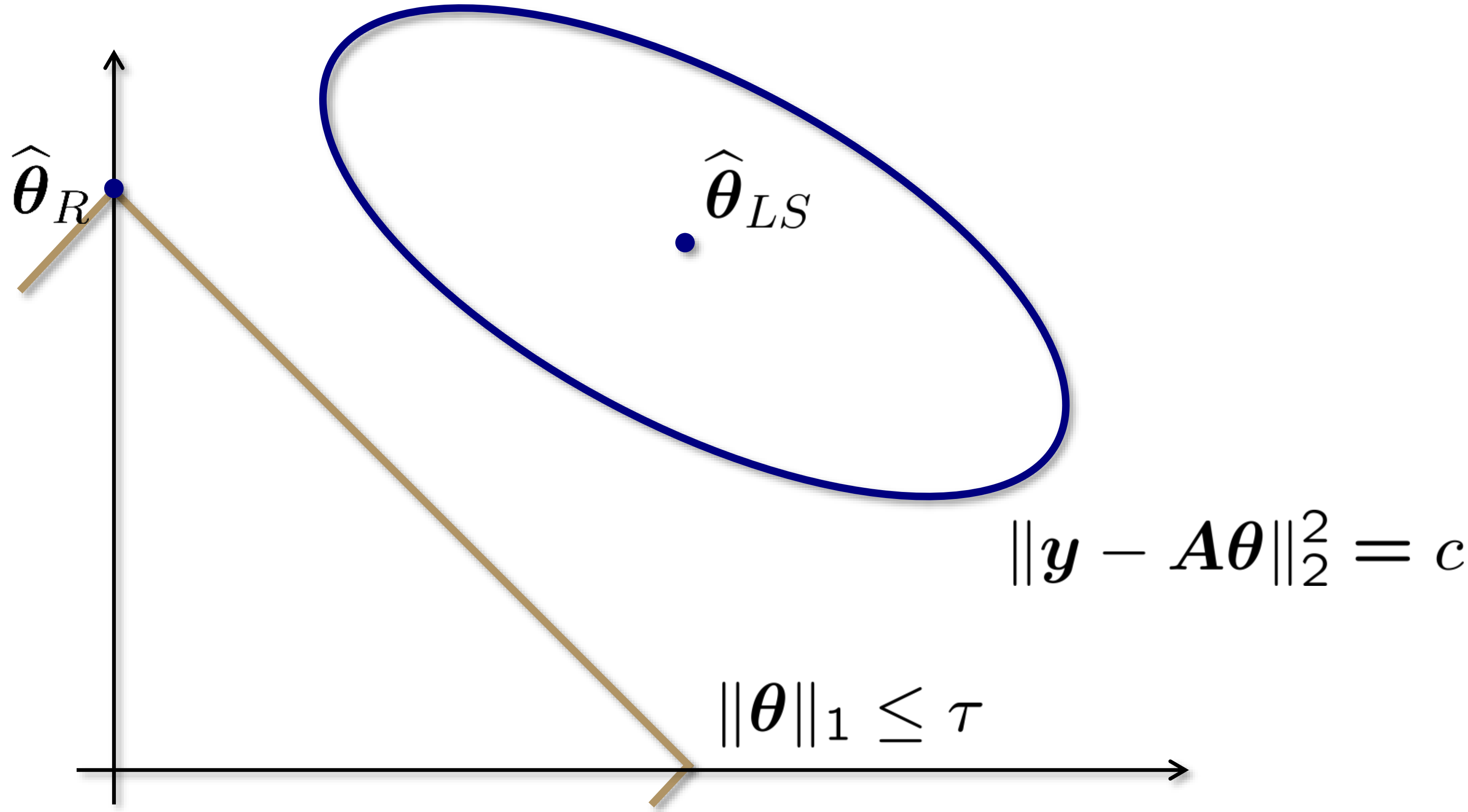
In general, we have this picture


$\widehat{\boldsymbol{\theta}}_{LS}$
$\widehat{\boldsymbol{\theta}}_R$
$\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{\theta}\|_2^2 = c$
$\|\boldsymbol{\theta}\|_2^2 \leq \tau$

Tikhonov regularization still shrinking the least squares solution towards the origin

# Lasso versus least squares

For the LASSO we get something like this...


$\widehat{\boldsymbol{\theta}}_R$
$\widehat{\boldsymbol{\theta}}_{LS}$
$\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{\theta}\|_2^2 = c$
$\|\boldsymbol{\theta}\|_1 \leq \tau$

LASSO still shrinking the least squares solution towards the origin, but now in a way that promotes sparsity

# A general approach to regression

Least squares, ridge regression, and the LASSO call all be viewed as particular instances of the following general approach to regression

$$\widehat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \; L(\boldsymbol{\theta}) + \lambda r(\boldsymbol{\theta})$$

- $L(\boldsymbol{\theta})$, often called the ***loss function***, enforces data fidelity

$$f_{\boldsymbol{\theta}}(\mathbf{x}_i) \approx y_i$$

- $r(\boldsymbol{\theta})$ is a ***regularizer*** which serves to quantify the “complexity” of $\boldsymbol{\theta}$

We have seen some examples of regularizers, what about other loss functions?

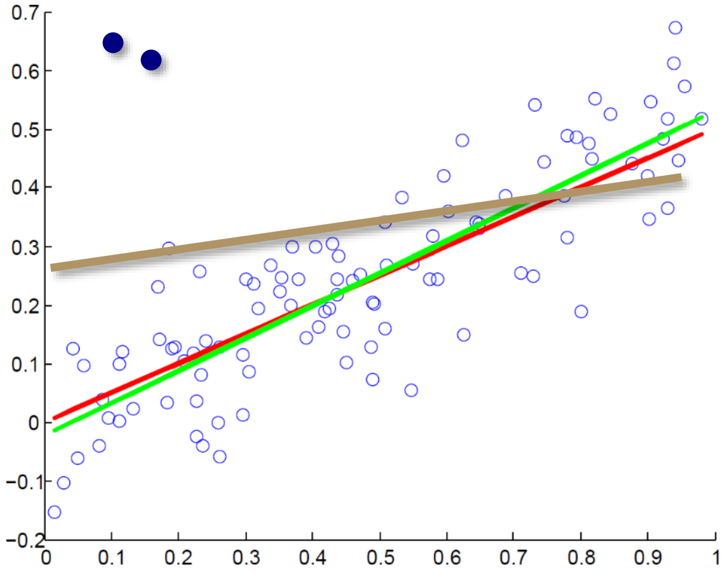
# Outliers in regression

The squared error loss function is sensitive to ***outliers***

If $f(\mathbf{x}_i) - y_i$ is small, then $(f(\mathbf{x}_i) - y_i)^2$ is not too large

But if $f(\mathbf{x}_i) - y_i$ is big, then $(f(\mathbf{x}_i) - y_i)^2$ is ***really*** big

Normally this is not a bad property – we want to penalize big errors – but this can make us very sensitive to large outliers



# Robust regression

What else could we do aside from least squares?

Mean absolute error

$$L_{AE}(r) = |r|$$

Huber loss

$$L_H(r) = \begin{cases} \frac{1}{2}r^2 & \text{if } |r| \leq c \\ c|r| - \frac{c^2}{2} & \text{if } |r| > c \end{cases}$$

$\epsilon$-insensitive loss

$$L_\epsilon(r) = \begin{cases} 0 & \text{if } |r| \leq c \\ |r| - \epsilon & \text{if } |r| > \epsilon \end{cases}$$

# Regularized robust regression

Suppose we combine this loss with an $\ell_2$ regularizer

$$\widehat{\boldsymbol{\beta}}, \beta_0 = \arg\min_{(\boldsymbol{\beta}, \beta_0)} \sum_{i=1}^{n} L_\epsilon(y_i - (\boldsymbol{\beta}^T \mathbf{x}_i + \beta_0)) + \frac{\lambda}{2} \|\boldsymbol{\beta}\|_2^2$$

Note that the $\epsilon$-insensitive loss has no penalty as long as your prediction is within a “margin” of $\epsilon$

This looks like an SVM…

# Support vector regression

The previous problem can also be cast in the following dual form

$$\min_{\boldsymbol{\alpha},\boldsymbol{\alpha}^*} \sum_i ((\epsilon - y_i)\alpha_i^* + (\epsilon + y_i)\alpha_i) + \frac{1}{2}\sum_{i,j}(\alpha_i^* - \alpha_i)(\alpha_j^* - \alpha_j)\mathbf{x}_j^T\mathbf{x}_i$$

subject to $0 \leq \alpha_i^*, \alpha_i \leq \frac{1}{\lambda}$

$$\sum_i (\alpha_i^* - \alpha_i) = 0$$

$$\alpha_i^* \alpha_i = 0$$

The solution has the form $\widehat{f}(\mathbf{x}) = \sum_i (\widehat{\alpha}_i^* - \widehat{\alpha}_i)\mathbf{x}_i^T\mathbf{x} + \beta_0$

# Can we kernelize regression?

In order to "kernelize" an algorithm, the general approach consists of three main steps:

1. Show that the training process only involves the training data via inner products (i.e., $\mathbf{x}_i^T \mathbf{x}_j$ )

2. Show that applying the decision rule to a new $\mathbf{x}$ only involves computing inner products (i.e., $\mathbf{w}^T \mathbf{x}$ )

3. Replace all inner products with evaluations of the kernel function $k(\cdot, \cdot)$

This approach extends well beyond SVMs

# Kernel support vector regression

$$\min_{\boldsymbol{\alpha},\boldsymbol{\alpha}^*} \sum_i ((\epsilon - y_i)\alpha_i^* + (\epsilon + y_i)\alpha_i) + \frac{1}{2}\sum_{i,j}(\alpha_i^* - \alpha_i)(\alpha_j^* - \alpha_j)\mathbf{x}_j^T\mathbf{x}_i$$

subject to $0 \leq \alpha_i^*, \alpha_i \leq \frac{1}{\lambda}$

$$\sum_i (\alpha_i^* - \alpha_i) = 0$$

$$\alpha_i^* \alpha_i = 0$$

$$\widehat{f}(\mathbf{x}) = \sum_i (\widehat{\alpha}_i^* - \widehat{\alpha}_i)\mathbf{x}_i^T\mathbf{x} + \beta_0$$

Straightforward to kernelize

# Kernelized LASSO?

Can we kernelize this?

$$\widehat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{\theta}\|_2^2 + \lambda\|\boldsymbol{\theta}\|_1$$

Not exactly...

Nevertheless, we can assert (with no justification) that

$$\widehat{\boldsymbol{\theta}} = \sum_i \alpha_i \mathbf{x}_i$$

and then replace $\|\boldsymbol{\theta}\|_1$ with $\|\boldsymbol{\alpha}\|_1$

This yields an algorithm that can be easily kernelized, although it is really something different than the LASSO

- promotes sparsity in $\alpha$, not $\theta$

# Ridge regression revisited

Ridge regression: Given $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$ , $\mathbf{x}_i \in \mathbb{R}^d, y_i \in \mathbb{R}$

$$(\widehat{\boldsymbol{\beta}}, \widehat{\beta}_0) = \arg\min_{\boldsymbol{\beta}, \beta_0} \sum_{i=1}^n (y_i - \boldsymbol{\beta}^T \mathbf{x}_i - \beta_0)^2 + \lambda \|\boldsymbol{\beta}\|_2^2$$

Solution: $$\frac{\partial}{\partial \beta_0} = -2 \sum_{i=1}^n (y_i - \boldsymbol{\beta}^T \mathbf{x}_i - \beta_0) = 0$$

$$\widehat{\beta}_0 = \frac{1}{n} \sum_{i=1}^n y_i - \widehat{\boldsymbol{\beta}}^T \mathbf{x}_i$$

$$= \bar{y} - \widehat{\boldsymbol{\beta}}^T \bar{\mathbf{x}}$$

$$\bar{y} = \frac{1}{n} \sum_i y_i$$

$$\bar{\mathbf{x}} = \frac{1}{n} \sum_i \mathbf{x}_i$$

# Ridge regression revisited

Plugging this back in we are left to minimize

$$\sum_{i=1}^{n}(y_i - \bar{y} - \boldsymbol{\beta}^T(\mathbf{x}_i - \bar{\mathbf{x}}))^2 + \lambda\|\boldsymbol{\beta}\|_2^2$$

with respect to $\boldsymbol{\beta}$

$$\widehat{\boldsymbol{\beta}} = (\mathbf{A}^T\mathbf{A} + \lambda\mathbf{I})^{-1}\mathbf{A}^T\widetilde{\mathbf{y}}$$

$$\mathbf{A} = \begin{bmatrix}(\mathbf{x}_1 - \bar{\mathbf{x}})^T \\ \vdots \\ (\mathbf{x}_n - \bar{\mathbf{x}})^T\end{bmatrix} \qquad \widetilde{\mathbf{y}} = \begin{bmatrix}y_1 - \bar{y} \\ \vdots \\ y_n - \bar{y}\end{bmatrix}$$

# Kernel ridge regression

$$\widehat{\boldsymbol{\beta}} = (\mathbf{A}^T\mathbf{A} + \lambda\mathbf{I})^{-1}\mathbf{A}^T\widetilde{\mathbf{y}}$$

$$\mathbf{A} = \begin{bmatrix} (\mathbf{x}_1 - \bar{\mathbf{x}})^T \\ \vdots \\ (\mathbf{x}_n - \bar{\mathbf{x}})^T \end{bmatrix} \qquad \widetilde{\mathbf{y}} = \begin{bmatrix} y_1 - \bar{y} \\ \vdots \\ y_n - \bar{y} \end{bmatrix}$$

$$\widehat{f}(\mathbf{x}) = \bar{y} + \widehat{\boldsymbol{\beta}}^T(\mathbf{x} - \bar{\mathbf{x}})$$

Can we express ridge regression in terms of inner products $\langle \mathbf{x}_i, \mathbf{x}_j \rangle$ and $\langle \mathbf{x}_i, \mathbf{x} \rangle$ ?

Not immediately. $[\boldsymbol{A}^T\boldsymbol{A}](i,j) \neq \mathbf{x}_i^T\mathbf{x}_j$

# Woodbury matrix inversion identity

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

$$\mathbf{P}=\lambda\mathbf{I}\qquad\mathbf{Q}=\mathbf{A}^T\qquad\mathbf{R}=\mathbf{I}\qquad\mathbf{S}=\mathbf{A}$$

$$\begin{aligned}(\lambda\mathbf{I}+\mathbf{A}^T\mathbf{A})^{-1}&=\frac{1}{\lambda}\mathbf{I}-\frac{1}{\lambda}\mathbf{I}\mathbf{A}^T\left(\mathbf{I}+\frac{1}{\lambda}\mathbf{A}\mathbf{A}^T\right)^{-1}\mathbf{A}\frac{1}{\lambda}\\&=\frac{1}{\lambda}\left[\mathbf{I}-\mathbf{A}^T(\lambda\mathbf{I}+\mathbf{A}\mathbf{A}^T)^{-1}\mathbf{A}\right]\end{aligned}$$

$$(\lambda\mathbf{I}+\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\widetilde{\mathbf{y}}=\frac{1}{\lambda}\left[\mathbf{A}^T-\mathbf{A}^T(\lambda\mathbf{I}+\mathbf{A}\mathbf{A}^T)^{-1}\mathbf{A}\mathbf{A}^T\right]\widetilde{\mathbf{y}}$$

# Kernelizing ridge regression

$$(\lambda \mathbf{I} + \mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \widetilde{\mathbf{y}} = \frac{1}{\lambda} \left[ \mathbf{A}^T - \mathbf{A}^T (\lambda \mathbf{I} + \mathbf{A}\mathbf{A}^T)^{-1} \mathbf{A}\mathbf{A}^T \right] \widetilde{\mathbf{y}}$$

$$\begin{aligned} K(i,j) &= (\mathbf{x}_i - \bar{\mathbf{x}})^T (\mathbf{x}_j - \bar{\mathbf{x}}) \\ &= \mathbf{x}_i^T \mathbf{x}_j - \frac{1}{n} \sum_{r=1}^{n} \mathbf{x}_i^T \mathbf{x}_r - \frac{1}{n} \sum_{s=1}^{n} \mathbf{x}_s^T \mathbf{x}_j + \frac{1}{n^2} \sum_{r,s=1}^{n} \mathbf{x}_r^T \mathbf{x}_s \end{aligned}$$

$$(\lambda \mathbf{I} + \mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \widetilde{\mathbf{y}} = \frac{1}{\lambda} \left[ \mathbf{A}^T - \mathbf{A}^T (\lambda \mathbf{I} + \mathbf{K})^{-1} \mathbf{K} \right] \widetilde{\mathbf{y}}$$

# Kernelizing ridge regression

What about the remaining $\mathbf{A}^T$?

$$\begin{aligned}\widehat{f}(\mathbf{x}) &= \bar{y} + \widehat{\boldsymbol{\beta}}^T(\mathbf{x} - \bar{\mathbf{x}}) \\ &= \bar{\mathbf{y}} + \frac{1}{\lambda}\widetilde{\mathbf{y}}^T\left[\mathbf{A} - \mathbf{K}(\lambda\mathbf{I} + \mathbf{K})^{-1}\mathbf{A}\right](\mathbf{x} - \bar{\mathbf{x}}) \\ &= \bar{\mathbf{y}} + \frac{1}{\lambda}\widetilde{\mathbf{y}}^T\left[\mathbf{I} - \mathbf{K}(\lambda\mathbf{I} + \mathbf{K})^{-1}\mathbf{I}\right]\mathbf{k}(\mathbf{x})\end{aligned}$$

where $\mathbf{k}(\mathbf{x}) = \begin{bmatrix}(\mathbf{x}_1 - \bar{\mathbf{x}})^T(\mathbf{x} - \bar{\mathbf{x}}) \\ \vdots \\ (\mathbf{x}_n - \bar{\mathbf{x}})^T(\mathbf{x} - \bar{\mathbf{x}})\end{bmatrix}$

$$[k(\mathbf{x})](i) = \mathbf{x}_i^T\mathbf{x} - \frac{1}{n}\sum_{j=1}^{n}\mathbf{x}_i^T\mathbf{x}_j - \frac{1}{n}\sum_{j=1}^{n}\mathbf{x}^T\mathbf{x}_j + \frac{1}{n^2}\sum_{j,k=1}^{n}\mathbf{x}_j^T\mathbf{x}_k$$

# Homogenous kernel ridge regression

For many kernels, $\Phi(\mathbf{x})$ already contains a constant component, in which case we often omit $\beta_0$

- inhomogenous polynomial kernel
- Gaussian kernel does not seem to require a constant

In this case, the kernel ridge regression solution becomes

$$\widehat{f}(\mathbf{x}) = \mathbf{y}^T (\mathbf{K} + \lambda \mathbf{I})^{-1} \mathbf{k}(\mathbf{x})$$

where $\mathbf{y} = \left[y_1, \cdots, y_n\right]^T$

$$\mathbf{k}(\mathbf{x}) = \left[k(\mathbf{x}_1, \mathbf{x}), \cdots, k(\mathbf{x}_n, \mathbf{x})\right]^T$$

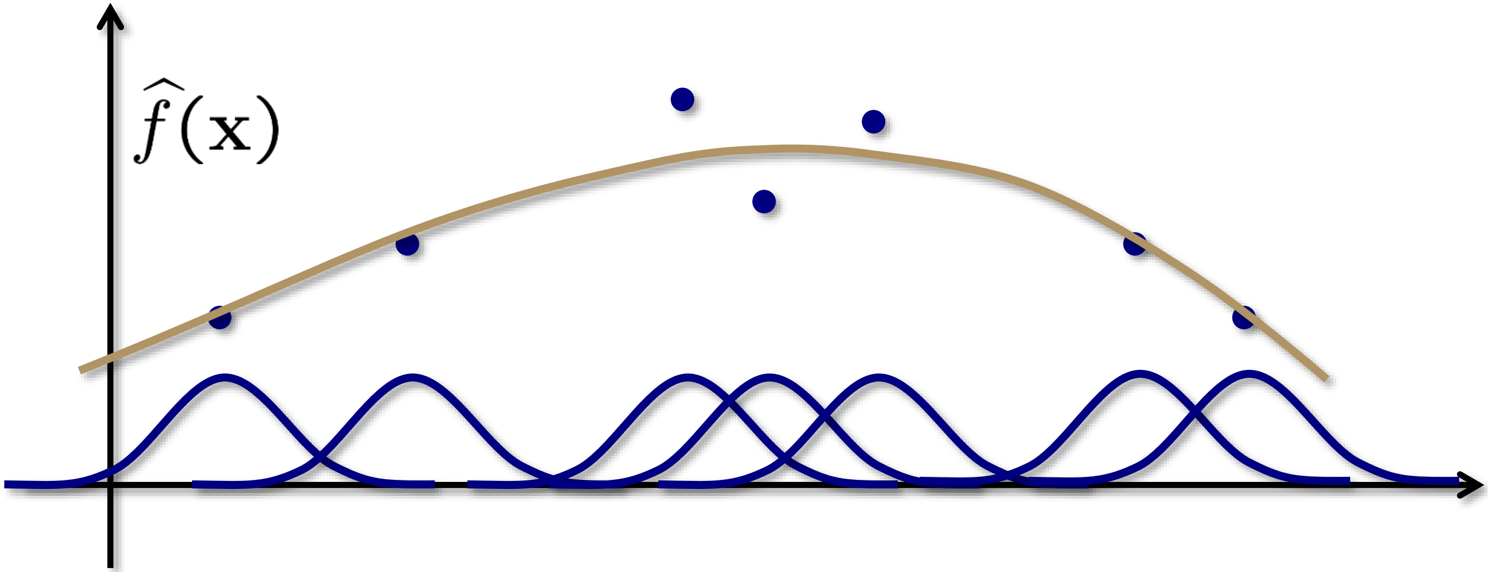
$$\mathbf{K} = \begin{bmatrix} k(\mathbf{x}_1, \mathbf{x}_1) & \cdots & k(\mathbf{x}_1, \mathbf{x}_n) \\ \vdots & \ddots & \vdots \\ k(\mathbf{x}_n, \mathbf{x}_1) & \cdots & k(\mathbf{x}_n, \mathbf{x}_n) \end{bmatrix}$$

# Example: Gaussian kernel

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

If we omit $\beta_0$, then $\widehat{f}(\mathbf{x}) = \mathbf{y}^T(\mathbf{K}+\lambda\mathbf{I})^{-1}\mathbf{k}(\mathbf{x})$

$$= \boldsymbol{\alpha}^T\mathbf{k}(\mathbf{x}) = \sum_{i=1}^{n} \alpha(i)k(\mathbf{x},\mathbf{x}_i)$$



# Loss functions and regularization

We have talked about a whole range of algorithms for regression that can be viewed through the lens of minimizing a loss function plus a regularization term

$$\widehat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \; L(\boldsymbol{\theta}) + \lambda r(\boldsymbol{\theta})$$

Does this viewpoint also apply to classification?

# Regularized logistic regression

Everything we have said so far about least squares regression can be extended to many classification problems

For example, in logistic regression we can replace

$$\min_{\boldsymbol{\theta}} \quad -\ell(\boldsymbol{\theta})$$

with

$$\min_{\boldsymbol{\theta}} \quad -\ell(\boldsymbol{\theta}) + \lambda \|\boldsymbol{\theta}\|_2^2$$

Has a similar interpretation to least squares regularization

- makes the Hessian matrix well conditioned
- super useful when the number of observations is small
- also helpful when data is separable

# Regularization for linear classification

The kinds of regularization we have talked about can also give us a new way to think about designing linear classifiers

**Goal (ideal):** Find $(\mathbf{w}, b)$ minimizing

$$\frac{1}{n}\sum_{i=1}^{n} 1_{\{y_i(\mathbf{w}^T\mathbf{x}_i+b)<0\}}$$

This is actually much harder than it sounds and is not computationally tractable for large problems

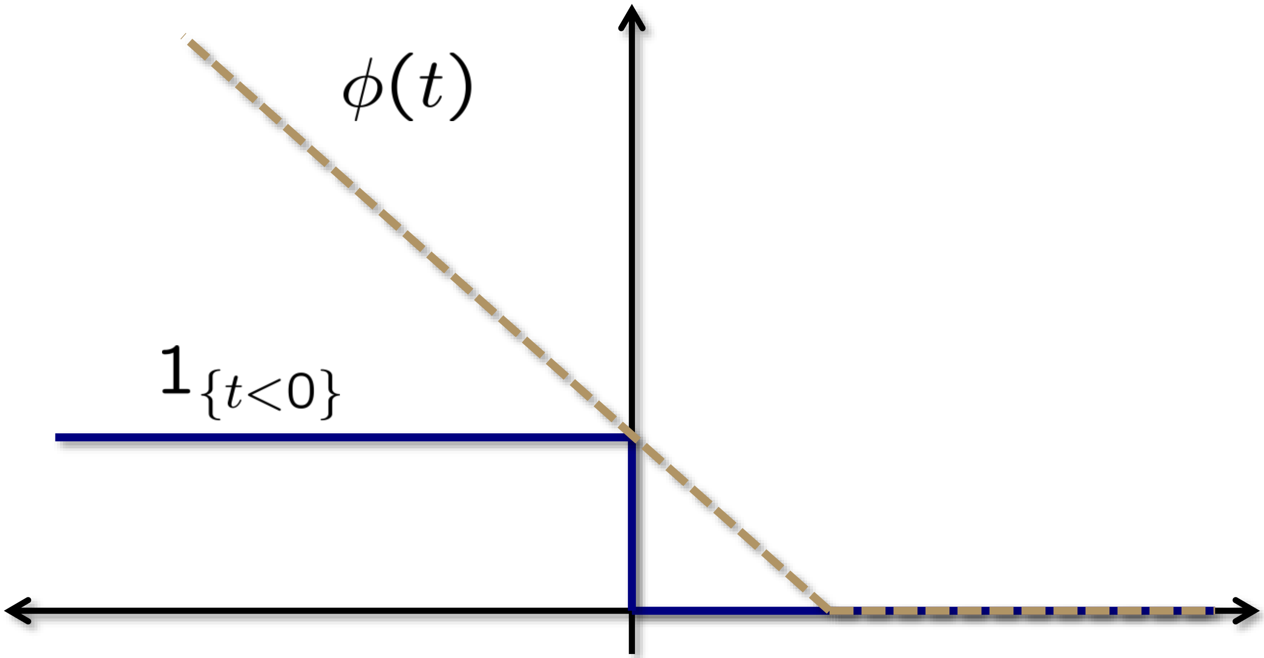
Instead, we can consider replacing this with

$$\frac{1}{n}\sum_{i=1}^{n} \phi(y_i(\mathbf{w}^T\mathbf{x}_i + b))$$

where $\phi(t)$ is some upper bound on $1_{\{t<0\}}$

# Hinge loss

Let's take $\phi(t) = \max\{0, 1 - t\} =: (1 - t)_+$



# Adding regularization

Let’s try to minimize

$$\frac{1}{n}\sum_{i=1}^{n}(1 - y_i(\mathbf{w}^T\mathbf{x}_i + b))_+$$

but to prevent overfitting, let’s add a regularization penalty on $\mathbf{w}$

$$\min_{\mathbf{w},b}\ \frac{1}{n}\sum_{i=1}^{n}(1 - y_i(\mathbf{w}^T\mathbf{x}_i + b))_+ + \frac{\lambda}{2}||\mathbf{w}||_2^2$$

# Soft-margin hyperplane

Compare this to the optimization problem we considered previously for the optimal soft-margin hyperplane:

$$\min_{\mathbf{w},b} \quad \frac{1}{n}\sum_{i=1}^{n}(1 - y_i(\mathbf{w}^T\mathbf{x}_i + b))_+ + \frac{\lambda}{2}\|\mathbf{w}\|_2^2$$

vs

$$\min_{\mathbf{w},b,\boldsymbol{\xi}} \quad \frac{1}{2}\|\mathbf{w}\|_2^2 + \frac{C}{n}\sum_{i=1}^{n}\xi_i$$

$$\text{s.t.} \quad y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 - \xi_i \qquad i = 1, \ldots, n$$

$$\xi_i \geq 0 \qquad i = 1, \ldots, n$$

As you all just showed, these are equivalent!