Empirical Risk Minimization

Sanjay Lall and Stephen Boyd

EE104
Stanford University

Loss and empirical risk

Parametrized predictors

- many predictors have the form $\hat{y} = g(x, \theta)$ (also written as $g_\theta(x)$)
- the function g fixes the *structure* or *form* of the predictor
- θ is a set of *parameters*, which can be a vector, matrix, or other structure
- example: *linear regression model* for scalar y
  - $\hat{y} = g_\theta(x) = \theta_1 x_1 + \cdots + \theta_d x_d$
  - here $\theta \in \mathbf{R}^d$ is a vector
- example: *linear regression model* for vector $y \in \mathbf{R}^m$
  - $\hat{y} = g_\theta(x) = \theta_1 x_1 + \cdots + \theta_d x_d$
  - here θ is a collection of m-vectors $\theta_1, \ldots, \theta_d \in \mathbf{R}^m$
  - usually organized as a $d \times m$ matrix θ with rows θ_i^T
- for a tree prediction model, θ encodes the tree, thresholds, and leaf values

Training a predictor

- choosing a particular θ given some *training data*

$$x^1, \ldots, x^n, \quad y^1, \ldots, y^n$$

  is called *training* or *fitting* the model (to the data)

- example: linear regression model for scalar y can be trained using *least squares*, *i.e.*, choose θ to minimize

$$\sum_{i=1}^{n} (\hat{y}^i - y^i)^2 = \sum_{i=1}^{n} (g_\theta(x^i) - y^i)^2$$

- this lecture covers a general and effective method to train a predictor, *empirical risk minimization* (ERM)
- ERM is a generalization of least squares

Loss function

- a *loss* function $\ell : \mathbf{R}^m \times \mathbf{R}^m \to \mathbf{R}$ quantifies how well (more accurately, how badly) $\hat{y}$ approximates y
  - smaller values of $\ell(\hat{y}, y)$ indicate that $\hat{y}$ is a good approximation of y
  - typically (but not always) $\ell(y, y) = 0$ and $\ell(\hat{y}, y) \geq 0$ for all $\hat{y}$, y
- examples
  - *quadratic loss*: $\ell(\hat{y}, y) = (\hat{y} - y)^2$ (for scalar y); $\ell(\hat{y}, y) = ||\hat{y} - y||_2^2$ (for vector y)
  - *absolute loss*: $\ell(\hat{y}, y) = |\hat{y} - y|$ (for scalar y)
  - *fractional loss* or *relative loss* (for scalar, positive y),

$$l(\hat{y}, y) = \max\left\{\frac{\hat{y}}{y} - 1, \frac{y}{\hat{y}} - 1\right\} = \exp\Big(\big|\log \hat{y} - \log y\big|\Big) - 1$$

    (often scaled by 100 to become *percentage error*)

Empirical risk

- the *empirical risk* is the average loss over the data points,

$$\mathcal{L}(\theta) = \frac{1}{n}\sum_{i=1}^{n} \ell(\hat{y}^i, y^i) = \frac{1}{n}\sum_{i=1}^{n} \ell(g_\theta(x^i), y^i)$$

- if $\mathcal{L}(\theta)$ is small, the predictor predicts or fits the given data well (according to the loss ℓ)

- empirical risk and performance metric are closely related
  - performance metric is used to *judge* a prediction model
  - empirical risk is used to *train* a (parametrized) prediction model
- empirical risk and performance metric are often, but not always, the same; we'll see why later

Examples

(for scalar y)

- for quadratic loss, $\mathcal{L}(\theta)$ is *mean-square-error* (MSE)

$$\mathcal{L}(\theta) = \frac{1}{n}\sum_{i=1}^{n}(g_\theta(x^i) - y^i)^2$$

- for absolute loss, $\mathcal{L}(\theta)$ is *mean absolute error* (MAE)

$$\mathcal{L}(\theta) = \frac{1}{n}\sum_{i=1}^{n}|g_\theta(x^i) - y^i|$$

Empirical risk minimization

Empirical risk minimization

- *empirical risk minimization* (ERM) is a general method for choosing θ, *i.e.*, fitting a parametrized predictor
- ERM: *choose θ to minimize empirical risk $\mathcal{L}(\theta)$*
- ERM chooses θ by attempting to match given data set well, as measured by the loss ℓ
- in some cases, *e.g.*, square loss, we can solve this minimization problem analytically
- in most cases, there is no analytic solution to this minimization problem, so we use *numerical optimization* to find θ that minimizes (or approximately minimizes) $\mathcal{L}(\theta)$; more on this topic later
- the predictor found by ERM depends on the loss you choose
- we use validation (with the performance metric) to choose from among candidate losses

Regularized empirical risk minimization

Sensitivity of a predictor

- an important attribute of a predictor g_θ: *sensitivity* or *continuity*
- g_θ is *insensitive* if for x near $\tilde{x}$, $g_\theta(x)$ is near $g_\theta(\tilde{x})$
- *i.e.*, if the features are close, the predictions are close
- there are many ways to make this more precise
- insensitive predictors often generalize well, especially when you don't have a lot of training data
- so insensitivity is a good attribute for a predictor to have

Regularizers

- a *regularizer* is a function $r : \mathbf{R}^p \to \mathbf{R}$ that measures the sensitivity of g_θ
- *i.e.*, $r(\theta)$ is small when g_θ is insensitive, and larger when g_θ is sensitive

- for linear regression model $g_\theta(x) = \theta^\mathsf{T} x$, small sensitivity is associated with small θ
- by Cauchy-Schwarz inequality,

$$||g_\theta(x) - g_\theta(\tilde{x})||_2 = ||\theta^\mathsf{T}(x - \tilde{x})||_2 \leq ||\theta||_F ||x - \tilde{x}||_2$$

  where $||A||_F^2 = \sum_{i,j} A_{ij}^2$ is the Frobenius norm squared
- suggests regularizer $r(\theta) = ||\theta||_F^2$

Ridge and ℓ_1 regularizers

- the most common regularizer for scalar y is *ℓ_2* or *square* or *ridge* regularization,

$$r(\theta) = ||\theta||_2^2 = \theta_1^2 + \cdots + \theta_d^2$$

- for vector y, we use $r(\theta) = ||\theta||_F^2 = \sum_{i=1}^d \sum_{j=1}^m \theta_{ij}^2$

- another popular regularizer is the *ℓ_1 regularizer*

$$r(\theta) = ||\theta||_1 = |\theta_1| + \cdots + |\theta_d|$$

  for scalar y; for vector y we use $r(\theta) = \sum_{i=1}^d \sum_{j=1}^m |\theta_{ij}|$

- we will see other regularizers later

Regularizers when there is a constant feature

- suppose $x_1 = 1$, *i.e.*, the first feature is constant
- with linear predictor, this means

$$g_\theta(x) = \theta^\top x = \theta_{1,:}^\top + \theta_{2:d,:}^\top x_{2:d}$$

  where $\theta_{1,:}$ is the first row of θ and $\theta_{2:d,:}$ are the remaining $d-1$ rows of θ
- $\theta_{1,:}$ does not affect sensitivity, since

$$||g_\theta(x) - g_\theta(\tilde{x})||_2 = ||\theta_{2:d,:}^\top (x - \tilde{x})||_2$$

- so there is no need to regularize first row of θ when x_1 is constant
- suggests that regularizer can be function of $\theta_{2:d,:}$, *e.g.*, $r(\theta) = ||\theta_{2:d,:}||_F^2 = \sum_{i=2}^d \sum_{j=1}^m \theta_{ij}^2$

Regularized empirical risk minimization

- regularized ERM is a method to trade off
  - good predictor fit on the training data, *i.e.*, $\mathcal{L}(\theta)$ small
  - insensitivity of g_θ, *i.e.*, $r(\theta)$ small
- *regularized* ERM (RERM): choose θ to minimize weighted sum $\mathcal{L}(\theta) + \lambda r(\theta)$
- $\lambda \geq 0$ is a parameter, called the *regularization hyper-parameter*
- when $\lambda = 0$, RERM reduces to ERM
- in most cases there is no analytic solution to this minimization problem, so we use *numerical optimization* to find θ that minimizes (or approximately minimizes) $\mathcal{L}(\theta) + \lambda r(\theta)$

Regularized versus unregularized ERM

- with ERM, you choose the model parameter θ that minimizes $\mathcal{L}(\theta)$
- with RERM, you choose a model parameter θ that *does not* minimize $\mathcal{L}(\theta)$
- but it is *less sensitive* than the ERM predictor
- and therefore often generalizes better, *i.e.*, makes better predictions on new, unseen data

Regularization hyper-parameter search

- we choose regularizer r and regularization parameter λ using validation, with the performance metric
- choosing a value of λ is called *regularization hyper-parameter search*
- typical regularization hyper-parameter search:
  - choose a set of values of λ, typically a few tens of values, log-spaced
  - find $\theta(\lambda)$ for each λ ($\theta(\lambda)$ is called the *regularization path*)
  - for each λ, evaluate the test set performance of $g_{\theta(\lambda)}$
  - choose the value of λ that gives the best test performance

Least squares and ridge regression

ERM via least squares

- with square loss and linear prediction model, we can solve the ERM problem exactly
- for model $g_\theta(x) = \theta^\mathsf{T} x$ and data $x^1, \dots, x^n \in \mathbf{R}^d$, and $y^1, \dots, y^n \in \mathbf{R}^m$,
- express empirical risk in matrix notation as

$$\mathcal{L}(\theta) = \frac{1}{n} \sum_{i=1}^{n} (\theta^\mathsf{T} x^i - y^i)^2 = \frac{1}{n} \|X\theta - Y\|_F^2$$

- $X \in \mathbf{R}^{n \times d}$ and $Y \in \mathbf{R}^{n \times m}$ are the feature and outcome data matrices

$$X = \begin{bmatrix} (x^1)^\mathsf{T} \\ \vdots \\ (x^n)^\mathsf{T} \end{bmatrix} \qquad Y = \begin{bmatrix} (y^1)^\mathsf{T} \\ \vdots \\ (y^n)^\mathsf{T} \end{bmatrix}$$

Least squares regression

- the minimizing θ is
$$\theta = X^\dagger Y = (X^\mathsf{T} X)^{-1} X^\mathsf{T} Y$$
(assuming columns of data matrix X are independent)
- called *least squares regression*

Ridge regression

- with square loss and regularization, and linear predictor, we can solve the RERM problem exactly
- called *ridge regression*
- RERM objective function is

$$\mathcal{L}(\theta) + \lambda||\theta||_F^2 = \frac{1}{n}||X\theta - Y||_F^2 + \lambda||\theta||_F^2 = \frac{1}{n}\left\|\begin{bmatrix} X \\ \sqrt{n\lambda}I \end{bmatrix}\theta - \begin{bmatrix} Y \\ 0 \end{bmatrix}\right\|_F^2$$

- solution is

$$\theta = (X^\mathsf{T}X + n\lambda I)^{-1}X^\mathsf{T}Y$$

(for $\lambda > 0$, the inverse always exsts)

Julia implementation

```
using LinearAlgebra
function ridgeregression(X,Y,lambda)
n,d = size(X)
m = size(Y,2)
A = [X; sqrt(lambda*n)*I(d)]
B = [Y; zeros(d,m)]
theta = A\B
end
```

Ridge regression with a constant feature

- when $x_1 = 1$, we don't regularize first row of θ
- we use regularizer $||\tilde{\theta}||_F^2$, where $\tilde{\theta} = \theta_{2:d,:} \in \mathbf{R}^{(d-1)\times m}$ is θ with its first row removed
- RERM objective function is

$$\mathcal{L}(\theta) + \lambda||\tilde{\theta}||_F^2 = \frac{1}{n}||X\theta - Y||_F^2 + \lambda||E\theta||_F^2 = \frac{1}{n}\left\| \begin{bmatrix} X \\ \sqrt{n\lambda}E \end{bmatrix} \theta - \begin{bmatrix} Y \\ 0 \end{bmatrix} \right\|_F^2$$

  where $E = \begin{bmatrix} 0 & I_{d-1} \end{bmatrix}$
- solution is

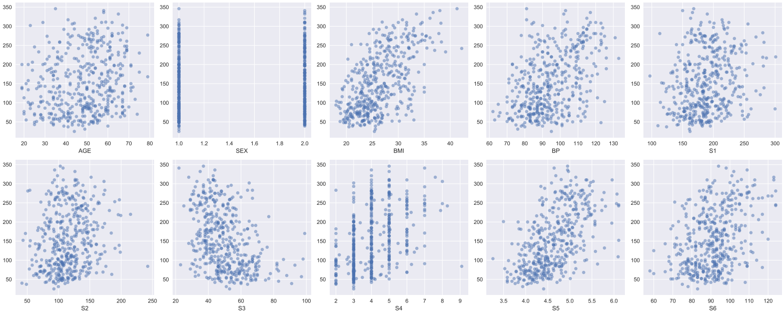
$$\theta = (X^\mathsf{T} X + n\lambda E^\mathsf{T} E)^{-1} X^\mathsf{T} Y$$

- $E^\mathsf{T} E = \mathrm{diag}(0, \mathbf{1}_{d-1}) = \begin{bmatrix} 0 & 0 \\ 0 & I_{d-1} \end{bmatrix}$

Julia implementation

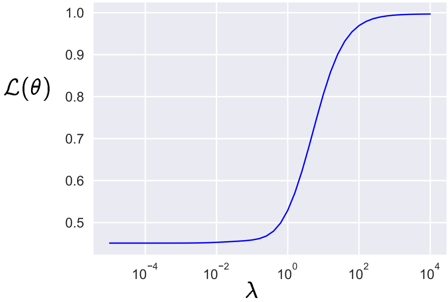
```
using LinearAlgebra
function ridgeregressionconstfeature(X,Y,lambda)
n,d = size(X)
m = size(Y,2)
E = [zeros(d-1,1)  I(d-1)]
A = [X; sqrt(lambda*n)*E]
B = [Y; zeros(d-1,m)]
theta = A\B
end
```

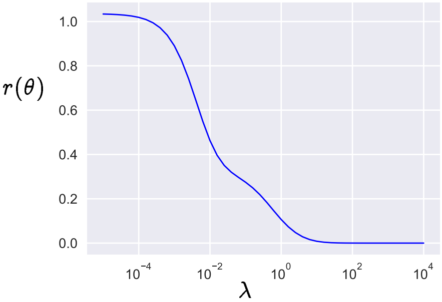
Example: Diabetes



- target is diabetes progression over a year
- 10 explanatory variables (age, bmi,. . .), standardized, plus constant feature
- data from 442 individuals, split 80% for training, 20% for validation
- we fit models using ridge regression with λ ranging from 10^{-5} to 10^4

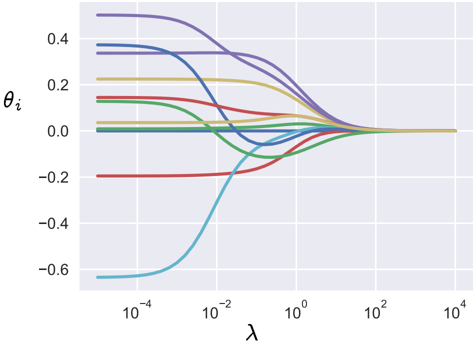
Empirical risk versus sensitivity





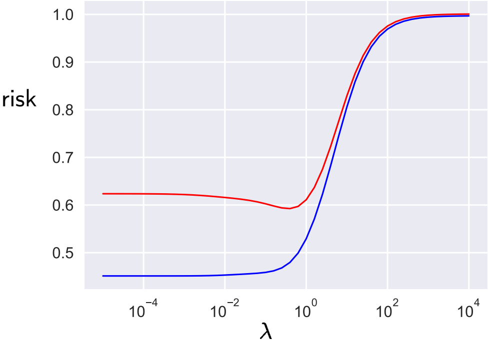
- as λ increases, empirical risk $\mathcal{L}(\theta)$ increases and sensitivity $r(\theta)$ decreases

Regularization path



- plot shows regularization path, *i.e.*, $d = 11$ components of θ versus λ
- as λ increases, model parameters (generally) get smaller
- explains why regularization is also called *shrinkage*

Validation results



- performance metric (mean square error) on training data (blue) and test data (red)
- a reasonable choice of λ is 0.3
- in this example regularization only improved model performance a little bit

Summary

Summary

- empirical risk is a function of the parameter θ that measures the fit on the training data set
- it is often but not always the same as the performance metric
- ERM chooses θ to minimize the empirical risk
- regularized ERM trades off two objectives:
  - small empirical risk (*i.e.*, good fit on the training data)
  - predictor insensitivity
- we choose the loss (and regularizer) by validation, using our performance metric
- for quadratic loss and regularizers we can find the parameters by least squares
- in other cases we use numerical optimization, covered later