Hello !

Lecture 2: Supervised Learning

Instructor: Lei Li, **Yu-Xiang Wang**

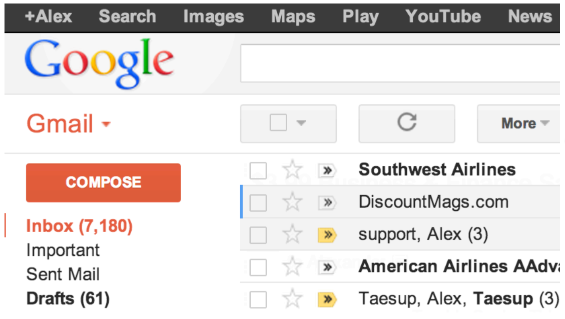
Announcement

- Thank you for sharing your motivation and goals for taking the course!
    - Please keep providing feedback during the course.
- HW0 due date on Thursday instead.
- Late days policy: 4 late days in total.

Recap: Last lecture

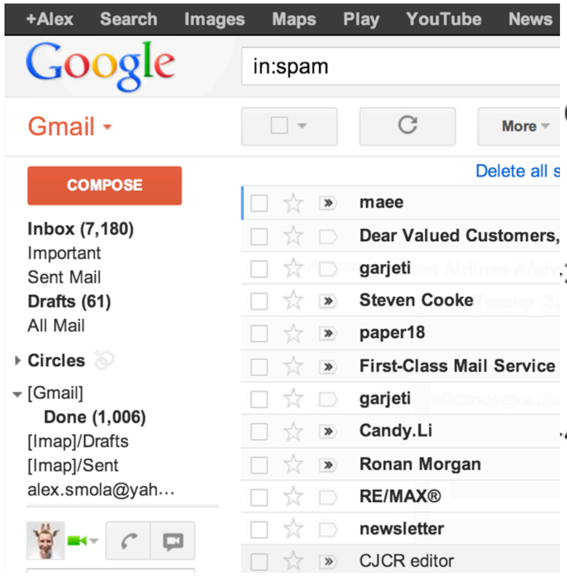
- Machine learning overview
- Supervised learning: Spam filtering as an example
  - Features, feature extraction
  - Models, hypothesis class
    - Free parameters of a hypothesis class
  - Choosing an appropriate hypothesis class
  - Performance metric
  - Overfitting and generalization

Recap: **Supervised learning** is about predicting label y using feature x by learning from labeled examples.



Inbox (7,180)
Important
Sent Mail
Drafts (61)

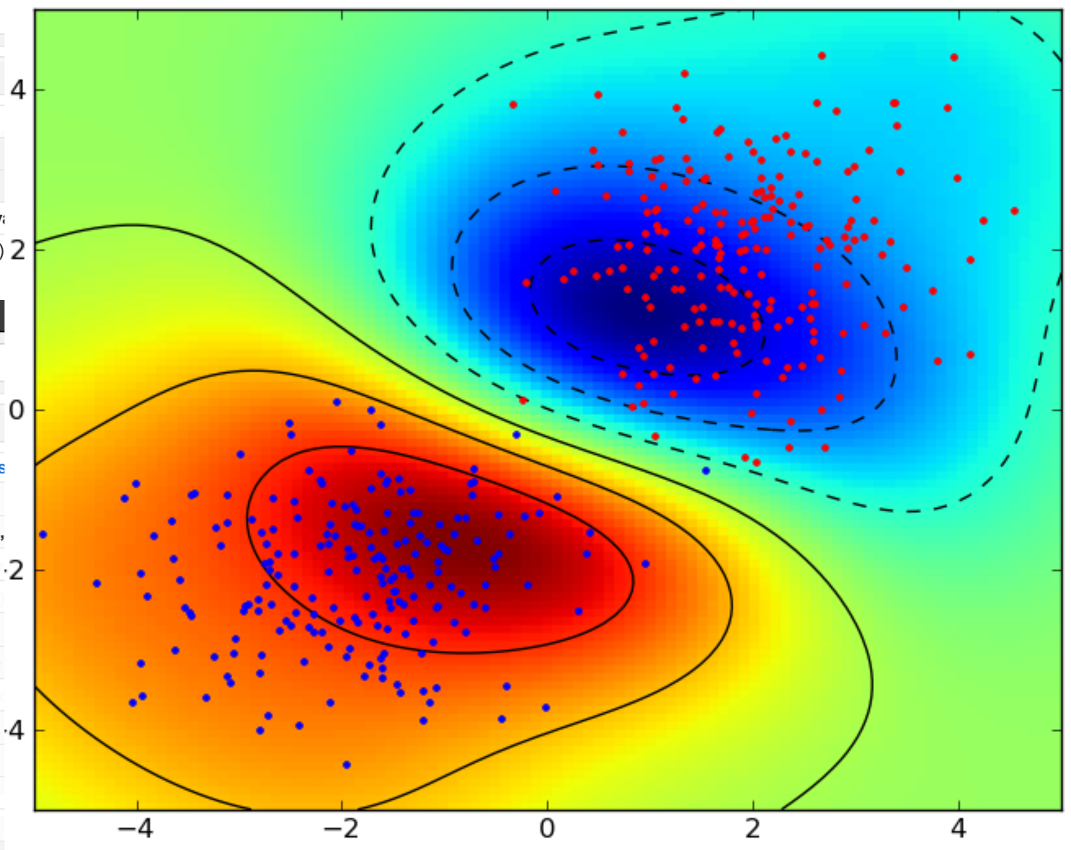
Southwest Airlines
DiscountMags.com
support, Alex (3)
American Airlines AAdv
Taesup, Alex, Taesup (3)



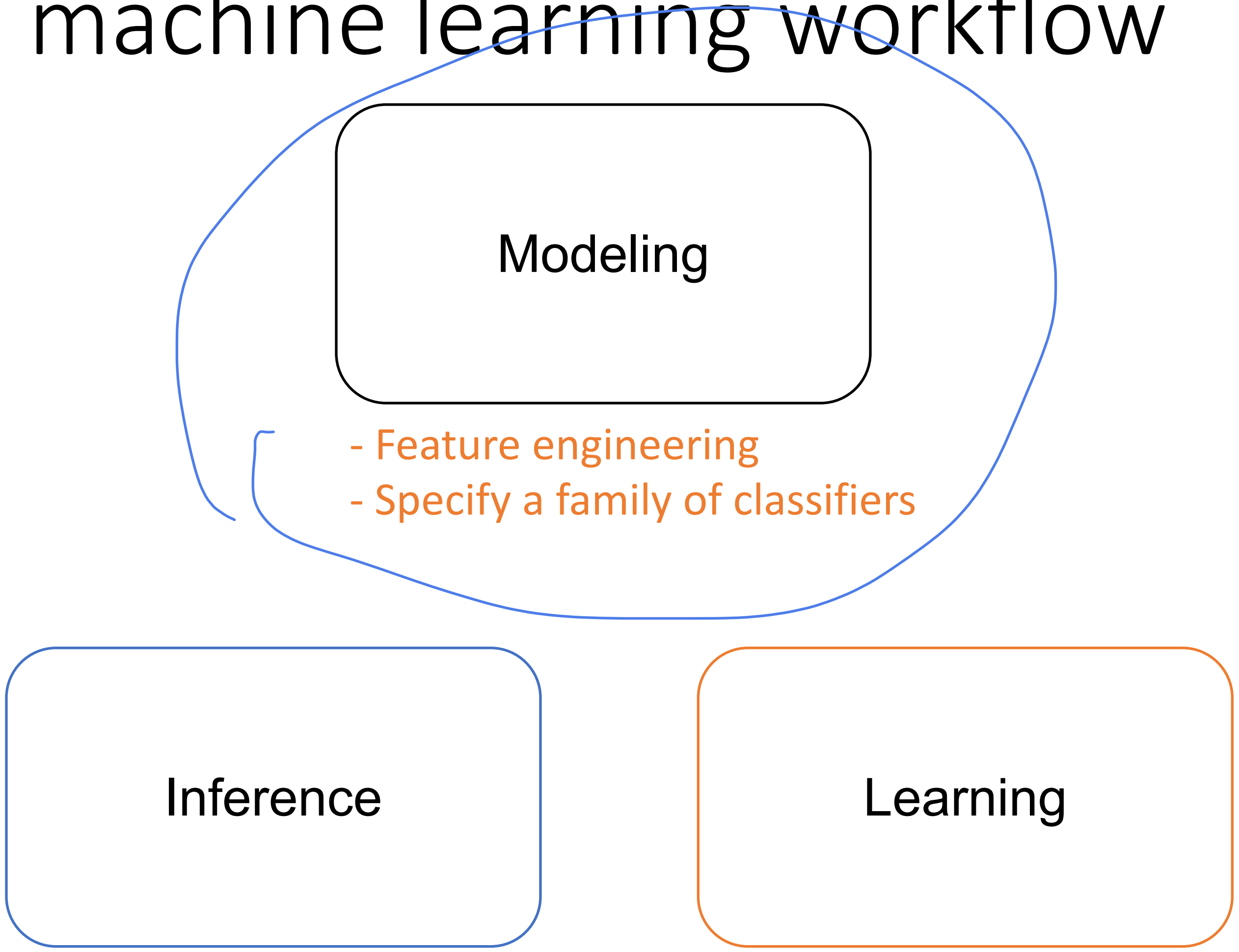
in:spam

Inbox (7,180)
Important
Sent Mail
Drafts (61)
All Mail
Circles
[Gmail]
Done (1,006)
[Imap]/Drafts
[Imap]/Sent
alex.smola@yah...

maee
Dear Valued Customers,
garjeti
Steven Cooke
paper18
First-Class Mail Service
garjeti
Candy.Li
Ronan Morgan
RE/MAX®
newsletter
CJCR editor



Recap: Modeling-Learning-inference in a machine learning workflow



- Feature engineering
- Specify a family of classifiers

Deployment to email client

Learning the best performing classifier

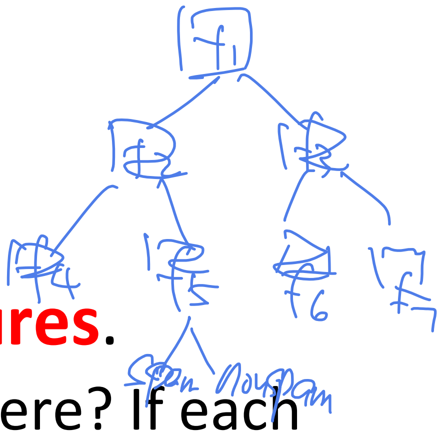
Recap: Mathematically defining the supervised learning problem

- Feature space: $\mathcal{X} = \mathbb{R}^d$
- Label space: $\mathcal{Y} = \{0, 1\} = \{\text{non-spam}, \text{spam}\}$
- A classifier (hypothesis): $h : \underline{\mathcal{X}} \to \underline{\mathcal{Y}}$
- A hypothesis class: $\mathcal{H}$
- Data: $(x_1, y_1), \ldots, (x_n, y_n) \in \mathcal{X} \times \mathcal{Y}$
- Learning task: Find $\underline{h} \in \underline{\mathcal{H}}$ that "works well".

Recap: The “free parameters” of the two hypothesis classes we learned

- Decision trees
  - “Which feature to use when branching?”
  - “The threshold parameter”
  - “Which label to assign at the leaf node”
  - …
- Linear classifiers
  - “Coefficient vector of the score function”
  - a (d+1) dimensional vector.

Answers for the quiz



- Consider a problem with **4 binary features**.
  - How many decision trees of **3 layers** are there? If each decision uses only one feature? (you may repeat features)

$$4^7 \cdot 2^8 = \underline{2^{22}}$$

  - How many possible feature vectors are there?

$$2^4$$

  - How many classifiers are there (without restrictions)?

$$2^{2^4} = 2^{16} = 65536$$

Recap: What do we mean by "working well"?

- What's the "Performance measure" for a classifier agent?

  - Really the **average error rate** on **new** data points.
  - But all we have is a training dataset.
  - Training error: (empirical) error rate on the training data.

  - When does the learned classifier ***generalize***?
  - How to know it if it does not?

This lecture

- Supervised learning:
  - formal notations and problem setup
  - Loss function, Risk, Empirical Risk
  - Examples
- Theory of supervised learning
  - Risk bounds for ‘fixed design’ linear regression model
  - Risk bounds for a general supervised learning problem
- Model selection

Mathematically defining the supervised learning problem

- Feature space: $\mathcal{X}$
- Label space: $\mathcal{Y}$
- A classifier (hypothesis): $h : \mathcal{X} \to \mathcal{Y}$
- A hypothesis class: $\mathcal{H}$
- Data: $(x_1, y_1), \ldots, (x_n, y_n) \in \mathcal{X} \times \mathcal{Y}$
- Learning task: Find $h \in \mathcal{H}$ that “works well”.

Notations from probability

R.V. $X \sim D$

$\underline{\mathbb{E}_{\mathcal{D}}}$ [Function of an r.v. X]

$\underline{\mathbb{P}_{\mathcal{D}}}$ [Event] $\quad \mathbb{P}_D[X \geq 5]$

$f_{X\sim\mathcal{D}}(x)$ or. $p(x) = \lim_{\varepsilon \to 0} \frac{\mathbb{P}[x \leq X \leq x+\varepsilon]}{\varepsilon}$

$\underline{F}_{X\sim\mathcal{D}}(x) = \mathbb{P}[X \leq x]$

Conditional expectation / conditional probability / density

$\mathbb{E}[\underline{\text{Func}(X,Y)}|\overset{}{\underset{\uparrow}{Y}}]$

$\mathbb{P}[\text{Event\_of}(X,Y)|Y]$

$f(x|y)$

$p(x|y)$

Notations from linear algebra

- Matrices and vectors

$$\boldsymbol{A} = m\left.\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}\right.^{\overbrace{}^{n}} \in \mathbb{R}^{m \times n}, \quad a_{ij} \in \mathbb{R}\,. \qquad \boldsymbol{a} = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} \in \mathbb{R}^3$$

- Transpose and inverse

$$A^T \in \mathbb{R}^{n \times m} \quad A[i,j] = A^T[j,i] \qquad \underline{A^{-1}}A = \underline{I} = \begin{bmatrix} 1 & & 0 \\ & 1 & \\ 0 & & \ddots \\ & & & 1 \end{bmatrix}$$

- Inner product / dot product

$$\langle x, y \rangle = x \cdot y = \underline{\boldsymbol{x}^\top \boldsymbol{y}} = \sum_{i=1}^{n} x_i y_i\,.$$

- <u>Norms.</u> $\|x\|_2 = \sqrt{\sum_{i=1}^{n} x_i^2}$ $\qquad \|x\|_p = \left(\sum_{i=1}^{n} (x_i)^p\right)^{\frac{1}{p}}$

Other useful notations

$B = (\boldsymbol{b}_1, \boldsymbol{b}_2, \boldsymbol{b}_3)$	(Ordered) tuple
$\rightarrow \boldsymbol{B} = [\boldsymbol{b}_1, \boldsymbol{b}_2, \boldsymbol{b}_3]$	Matrix of column vectors stacked horizontally
$\mathcal{B} = \{\boldsymbol{b}_1, \boldsymbol{b}_2, \boldsymbol{b}_3\}$	Set of vectors (unordered)
$\mathbb{Z}, \mathbb{N}$	Integers and natural numbers, respectively
$\mathbb{R}, \mathbb{C}$	Real and complex numbers, respectively
$\mathbb{R}^n$	n-dimensional vector space of real numbers
$\forall x$	Universal quantifier: for all x
$\exists x$	Existential quantifier: there exists x

$[n] \underset{\text{def}}{:=} \{1, 2, 3, ..., n\}$ − range(n)+1

$|S|$ cardinality of S

e.g. $|[n]| = ?$: n

Indicator (one-zero) function:

Eg $\int_5^{\infty} P(x)\,dx$

$$\mathbb{I}[\text{condition}] \overset{\text{def}}{=} \begin{cases} 1 & \text{if condition is true} \\ 0 & \text{otherwise.} \end{cases}$$

$$\int \mathbb{I}(x \geq 5)\, p(x)\, dx = \mathbb{E}[\mathbb{I}(X \geq 5)] = \underline{\mathbb{P}(X \geq 5)}$$

Conventions and typical meaning of specific variables in machine learning

- x: input
- y: output
- $(x,y) =$ z: input-output pair
- d: dimensionality
- n: number of examples

The "hat" notation, e.g.: $\hat{h}, \hat{f}, \hat{\theta}, \hat{\mathbb{E}}$ associated with being an estimate, computed as a function of the data

The "star" notation, e.g.,: $h^*, f^*, \theta^*, p^*, R^*$ associated with being "optimal"

Loss, Risk, Empirical Risk: What do we mean by working well?

- Loss function

$$\ell(h, (x, y))$$

e.g. $\mathbb{I}(h(x) \neq y)$

$(h(x) - y)^2$

$(x, y) \sim D$

- Risk function

$$R(h, \mathcal{D}) = \mathbb{E}_{\mathcal{D}}[\ell(h, (x, y))]$$

classifier Distribution $(x, y) \sim D$

$h : X \to Y$

- Empirical risk

$$\hat{R}(h, \text{Data}) = \underline{\frac{1}{n} \sum_{i=1}^{n} \ell(h, (\underline{x_i, y_i}))}$$

Example 1: Regression

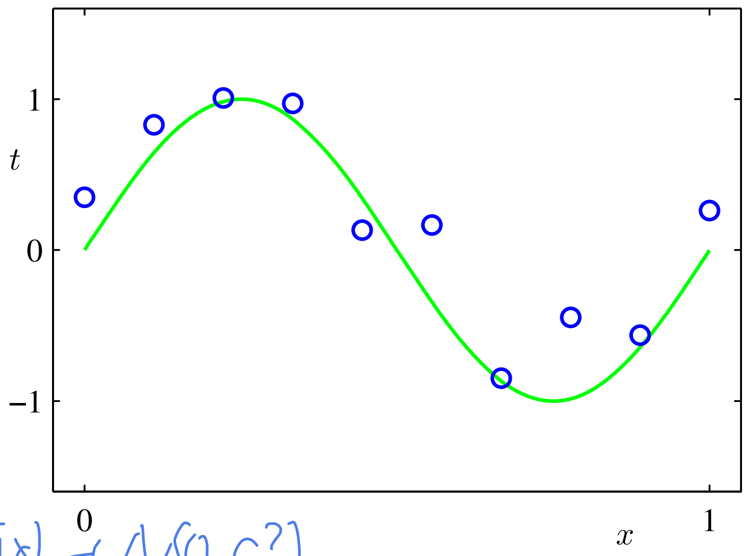


Figure 1.2 Plot of a training data set of $N = 10$ points, shown as blue circles, each comprising an observation of the input variable x along with the corresponding target variable t. The green curve shows the function $\sin(2\pi x)$ used to generate the data. Our goal is to predict the value of t for some new value of x, without knowledge of the green curve.

$y = \sin(2\pi x) + \mathcal{N}(0, \sigma^2)$

- What are the feature space, label space?

$X = [0,1]$ $\quad Y = \mathbb{R}$ or $[-1,1]$

- What is a reasonable hypothesis class to use and its free-parameter?

Examples of hypothesis classes for this problem

- Polynomials

$$\underline{\mathcal{H} = \{ h(x,w) \mid \underline{w \in \mathbb{R}^{M+1}} \}}$$

$$h(x, \mathbf{w}) = w_0 + w_1 x + w_2 x^2 + \ldots + w_M x^M = \sum_{j=0}^{M} w_j x^j$$

- Sine function

$$h(x, t) = \sin(2\pi \overset{x}{t})$$

$$\mathcal{H} = \{ \sin(2\pi \overset{x}{t}) \mid \underline{t \in \mathbb{R}_+} \}$$

- Anything else?

What are some reasonable loss functions for regression problems?

- Square error loss function

$\ell(h,(x,y)) = (h(x)-y)^2$

- Absolute deviation loss function

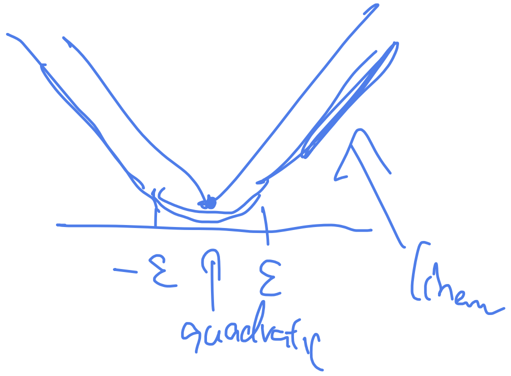
$|h(x)-y|$

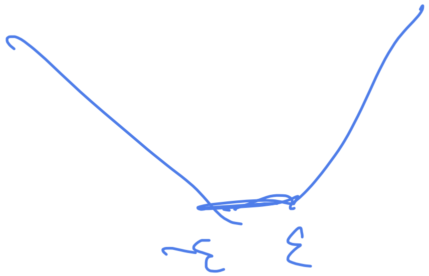
- Huber loss function

$a = h(x)-y \quad \ell_\varepsilon(a) = \begin{cases} \frac{a^2}{2} & |a| \le \varepsilon \\ \frac{\varepsilon}{2}|a-\varepsilon| & \text{otherwise} \end{cases}$

- epsilon-sensitive loss function
  - aka support vector regression

$\ell_\varepsilon(a) = \begin{cases} 0 & |a| \le \varepsilon \\ |a-\varepsilon| & \text{otherwise} \end{cases}$





Learning is often achieved by solving the **Empirical Risk Minimization**

$$\hat{h} = \arg\min_{h\in\mathcal{H}} \underset{\overset{||}{\frac{1}{n}\sum \ell(h,(x_i,y_i))}}{\hat{R}}(h, \{(x_i, y_i) | i \in [n]\})$$

- Sometimes with an additional **regularization functional** (also known as a penalty term)

$$\hat{h} = \arg\min_{h\in\mathcal{H}} \underset{\overset{\Uparrow}{\text{loss}}}{\hat{R}}(h, \text{Data}) + \underset{\overset{\Uparrow}{\text{Regularization}}}{\underline{g(h)}}$$

Polynomial regression under square loss

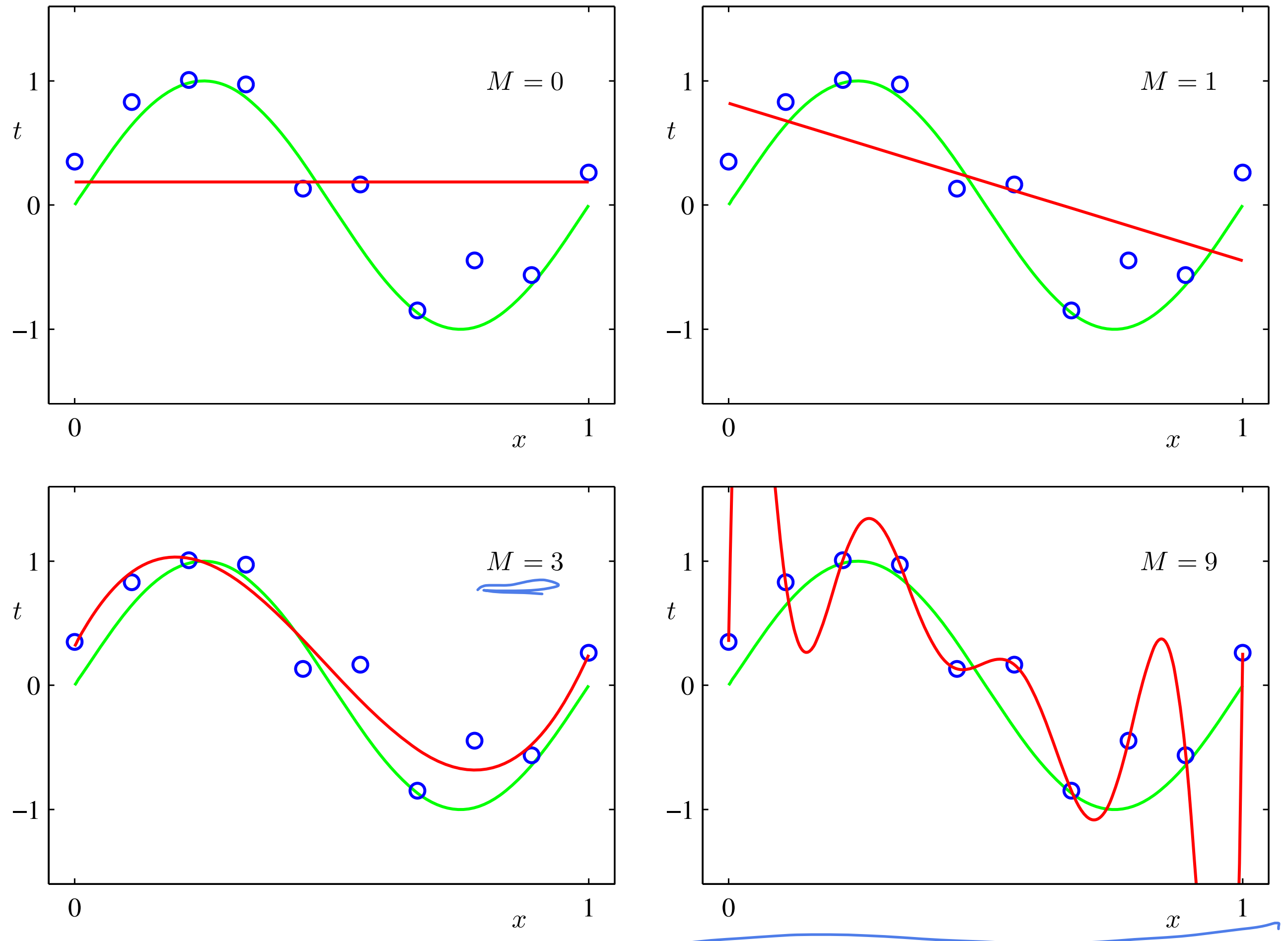
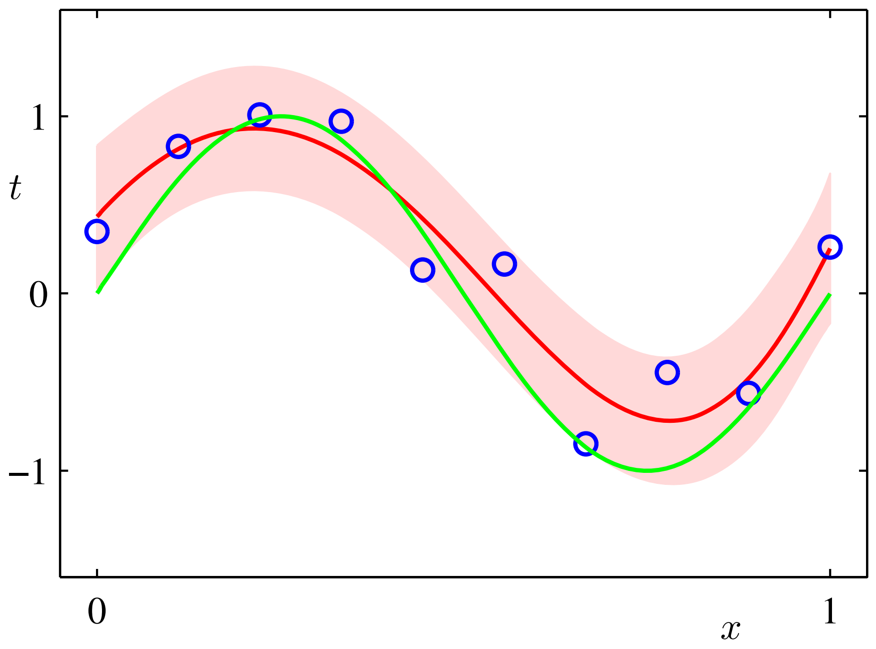


Figure 1.4 Plots of polynomials having various orders M, shown as red curves, fitted to the data set shown in Figure 1.2.

Appropriately regularized fit of a 9th order polynomial.

Figure 1.17 The predictive distribution resulting from a Bayesian treatment of polynomial curve fitting using an $M = 9$ polynomial, with the fixed parameters $\alpha = 5 \times 10^{-3}$ and $\beta = 11.1$ (corresponding to the known noise variance), in which the red curve denotes the mean of the predictive distribution and the red region corresponds to ± 1 standard deviation around the mean.



$+\lambda \|w\|^2$

Regularization prevents overfitting!

Example 2: Linear regression

- Feature space $X \in \mathbb{R}^d$ or $[0,1]^d$
- Label space $Y = \mathbb{R}$ or $[-B, B]$ or $[0,1]$
- Hypothesis space $w \in \mathbb{R}^d$

  $h(w, x) = \sum_{i=1}^{d} w_i x_i = \vec{w}^T \vec{x}$
- Loss function $(w^T x - y)^2$

  $w, \theta \in \mathbb{R}^d$

Quiz 1: Can we reformulate Example 1 as a linear regression task?

- Q1: When hypothesis class is polynomial?

$\mathcal{X} = [0,1]$ $\quad \phi(x) \in \mathbb{R}^{M+1}$

$\mathcal{X} = \{ (1, x, x^2, \dots, x^M) \mid x \in [0,1] \}$

$h(w,x) = w_0 + w_1 x + w_2 x^2 + \dots + w_m x^m = w^T \phi(x)$

- Q2: When the hypothesis class is sine function with parameter t?

$h(t,x) = \sin(t x \pi) = \int_{t=0}^{\infty} w(t) \sin(t x \pi) = \langle w, \sin(\cdot \, x \pi) \rangle$

$\mathcal{H} = \{ \langle w, \sin(\cdot \pi x) \rangle \mid \|w\|_1 \le 1, \ w \ge 0 \}$

Empirical risk minimization for linear regression under square loss

$$\hat{\theta} = \arg\min_{\theta} \underline{\frac{1}{n} \sum_{i \in [n]} (x_i^T \theta - y_i)^2}$$

- aka: Ordinary Least square (OLS), MLE under Gaussian noise
- A convenient form using linear algebra

$$\frac{1}{n} \| X \theta - \vec{y} \|^2$$

$$\begin{array}{c|c|l} & d & \\ \hline n & \begin{matrix} \text{———} \\ \text{———} \\ \text{———} \\ \text{———} \end{matrix} & \begin{matrix} x_1^T \\ \\ x_n^T \end{matrix} \end{array} \qquad \begin{bmatrix} \theta \end{bmatrix} \in \mathbb{R}^d$$

$$\hat{\theta} = (X^T X)^{-1} X^T \vec{y} \qquad \text{OLS}$$

Regularization helps to **reduce overfitting** and induce **structures** in the solution.

- Example: p-norm regularized least square

$$\hat{\theta} = \arg\min_{\theta} \frac{1}{n}\|X\theta - y\|_2^2 + \lambda\underline{\|\theta\|_p^p}$$

  - when p=2, this is called "Ridge Regression"
  - when p=1, this is called "Lasso"
  - when p=0, this is called "Best subset selection"

Regularization helps to **reduce overfitting** and induce **structures** in the solution.

- Ridge regression induces solutions that are small but dense.
- Lasso induces solutions that are "sparse".

$$\hat{\theta} = \underset{\theta}{\operatorname{argmin}} \|X\theta - y\|^2$$

$$\text{s.t. } \|\theta\|_p \leq \underline{u}$$

u is a func of λ, X, y

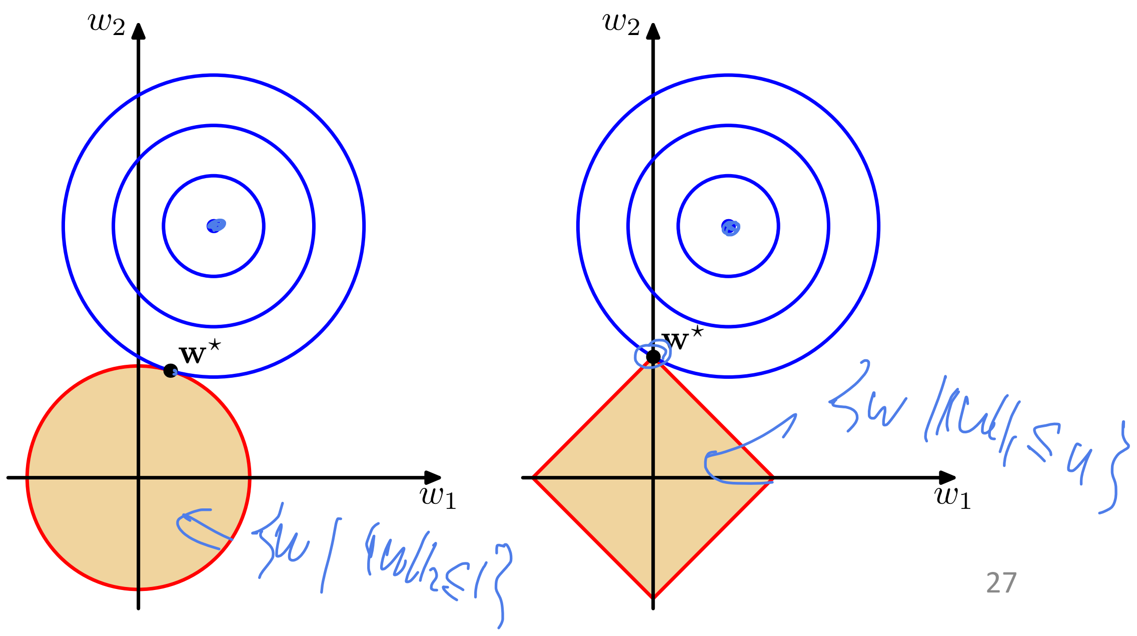
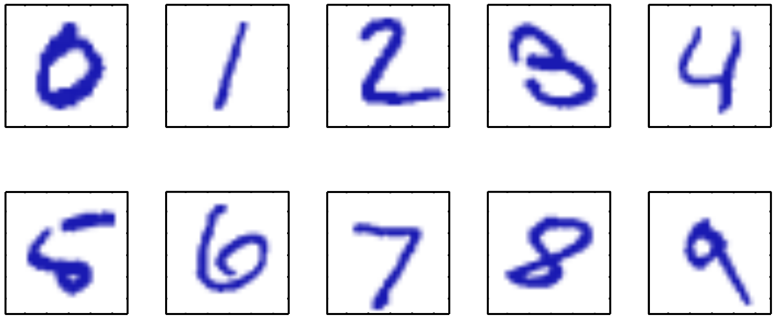


Figure 3.4 Plot of the contours of the unregularized error function (blue) along with the constraint region (3.30) for the quadratic regularizer $q = 2$ on the left and the lasso regularizer $q = 1$ on the right, in which the optimum value for the parameter vector $\mathbf{w}$ is denoted by $\mathbf{w}^\star$. The lasso gives a sparse solution in which $w_1^\star = 0$.

Example 3: Multi-class classification

Figure 1.1 Examples of hand-written digits taken from US zip codes.



- What are the feature space, label space?

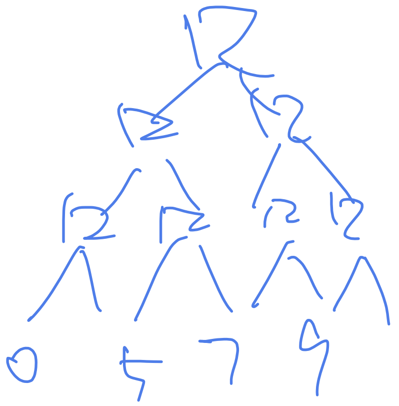
$X = ~~\cdot~~ [0,1]^{m \times n}$

$x \in \mathbb{R}^{m \cdot n \times 1}$

$Y = \{0, 1, 2, \cdots, 9\}$

Hypothesis class for multi-class classification problems

- Decision trees



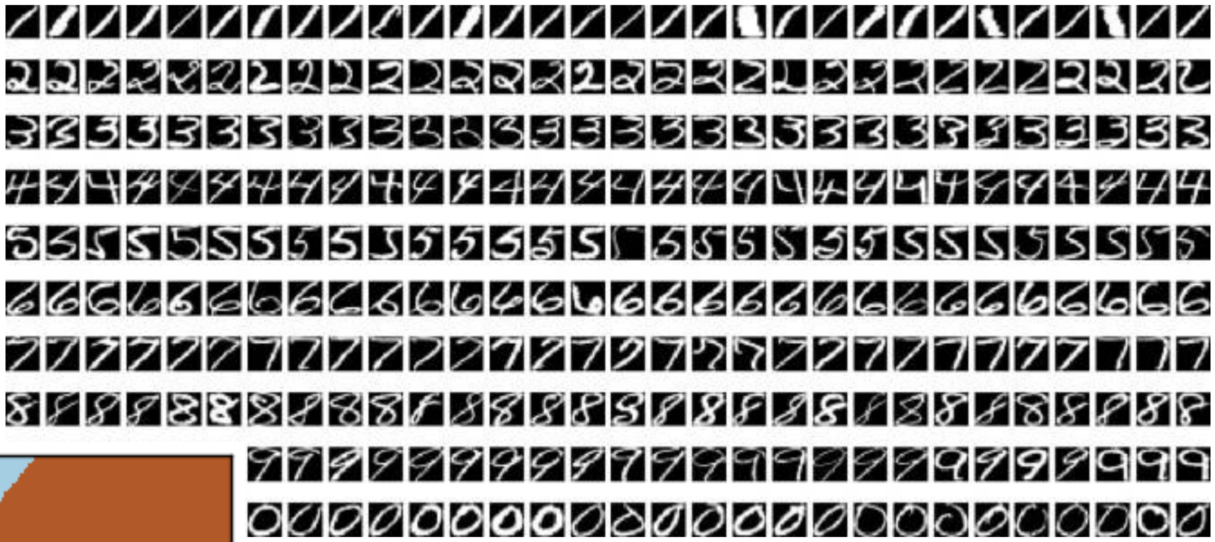
- Linear classifier

$$\text{score}_w(x) = w^T x$$

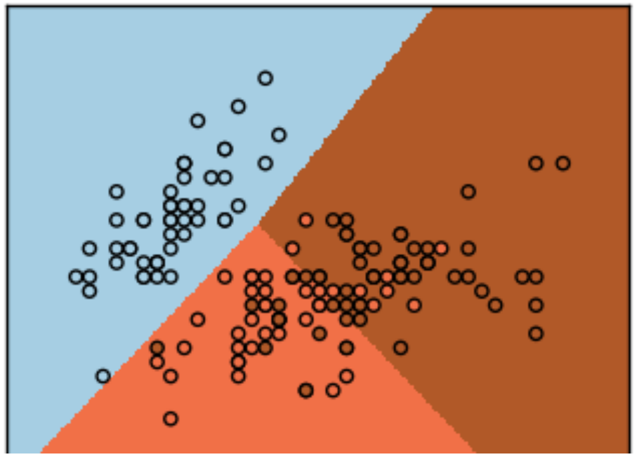
$$w_0, w_2, \ldots, w_9 \in \mathbb{R}^d$$

$$h(x) = \underset{y \in Y}{\arg\max}\ w_y^T x$$

Illustration of the decision boundary in multi-class linear classification



map image x to digit y



Loss functions for classification tasks when the predictions are discrete

- 0-1 loss

$$\mathbb{I}(h(x) \neq y)$$

- Cost-sensitive loss

$$\begin{array}{cc|cc} & & \multicolumn{2}{c}{\hat{y}} \\ & & 0 & 1 \\ \hline y & 0 & 0 & 0.01 \\ & 1 & 100 & 0 \end{array}$$

$$\text{loss}(h, (x, y)) = C(y, h(x))$$

Cost matrix

Soft-(arg)max transform helps to convert real-valued predictions to a probability distribution

- Softmax function

(You should've seen from HW0 for why this is soft-max)

$$f(x_1, \ldots, x_n) = \log \sum_{i=1}^{n} \exp(x_i).$$

- Soft-argmax transform

(Compare this to argmax in One-Hot representation)

$$F(x_1, \ldots, x_n) = \frac{[e^{x_1}, \ldots, e^{x_n}]}{\underline{\sum_{i=1}^{n} e^{x_i}}} \approx \begin{bmatrix} \text{small} \\ \text{big} \\ \text{small} \\ \vdots \\ \text{small} \end{bmatrix} \approx \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

Loss functions for classification tasks for soft-predictions

$\hat{p}$

or $\text{Score}_\theta(x)$

$\in \mathbb{R}^{|Y|}$

- log-loss

$$-\sum_{i\in Y} \log(\hat{p}(i))\, \mathbb{1}(y=i)$$

- Cross entropy loss

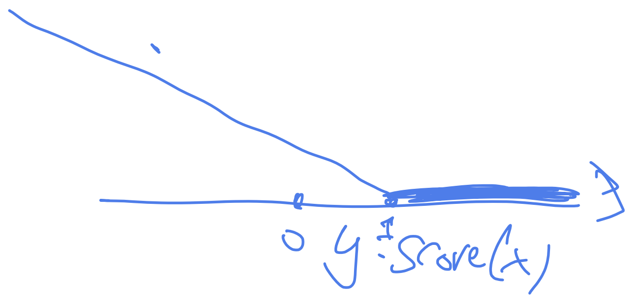
$$-\sum_{i\in Y} \log(\hat{p}(i)) \cdot \underset{\uparrow \text{true label}}{p(i)}$$

- Logistic loss in the binary case

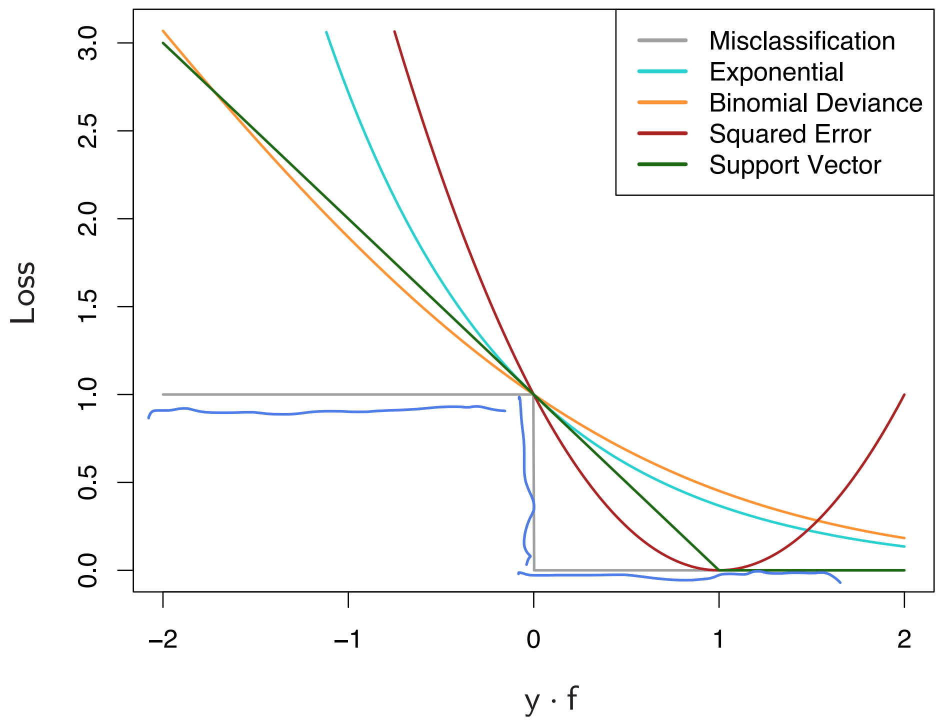
$Y=\{-1,1\}$

$$\log\left(1+e^{-y\cdot \text{Score}(x)}\right)$$

- Hinge loss



Visualization of the loss functions for classification



** "Binomial deviance" is the "logistic loss" from the previous slide.

FIGURE 10.4. *Loss functions for two-class classification. The response is* $y = \pm 1$*; the prediction is* f*, with class prediction* $\operatorname{sign}(f)$*. The losses are misclassification:* $I(\operatorname{sign}(f) \neq y)$*; exponential:* $\exp(-yf)$*; binomial deviance:* $\log(1 + \exp(-2yf))$*; squared error:* $(y - f)^2$*; and support vector:* $(1 - yf)_+$ *(see Section 12.3). Each function has been scaled so that it passes through the point* $(0, 1)$.

(Section 10.4 of "Elements of Statistical Learning")

Empirical risk minimization for multi-class classification

$$\hat{h} = \arg\min_{h \in \mathcal{H}} \underline{\hat{R}}(h, \{(x_i, y_i) | i \in [n]\})$$

Computation-approximation tradeoff in choosing loss functions

	0-1 loss / cost-sensitive loss	**Log loss / cross-entropy loss**
Computation	NP-hard in general	More efficient
Approximation	No approximation	Used as a surrogate

Also, depends on the choice of hypothesis class.
We will see more of this tradeoff later.

Loss function is often domain-specific. It is often part of the design of an ML workflow

- Discussion: Loss function for stock price prediction
  - Square loss?
  - 0-1 loss?

0.9 w.p. 0.1 $\hat{\Delta} = -100$ $\hat{p} = 1$

w.p 0.9 $\Delta = 0.1$ $\hat{p} = 1$

$$\boxed{C_i(x) \cdot \mathbb{I}(\hat{p} \neq y)}$$

Checkpoint: Supervised learning

- Formal problem setup
  - Feature space, label space, hypothesis class, loss function, risk function
- Examples:
  - Regression, Linear regression, multi-class classification
  - Regularization
- Choices of loss functions

Remainder of this lecture

- Supervised learning:
  - formal notations and problem setup
  - Loss function, Risk, Empirical Risk
  - Examples
- Theory of supervised learning
  - Risk bounds for ‘fixed design’ linear regression model
  - Risk bounds for a general supervised learning problem
- Model selection

Theory of linear regression

- What are the assumptions?
  - A1. Linear model + iid noise

$$\underset{\sim}{y_i} = x_i \cdot \underline{\theta^*} + \underline{\epsilon_i}, \qquad \mathbb{E}[\epsilon_i] \;=\; 0 \quad \underline{\mathrm{Var}[\epsilon_i] = \sigma^2}$$

  - A2. Fixed design matrix with full rank

$$\mathbb{R}^{n\times d} \ni \overline{X = [x_1, \dots, x_n]}^T \text{ fixed}$$

- Risk function in this case

$$R(\theta) = \mathbb{E}\left[\ell(\theta,(x,y))\right] = \frac{1}{n}\sum_{i=1}^{n}\mathbb{E}\left[(\theta^T x_i - y_i)^2\right]$$

$$= \frac{1}{n}\sum_{i=1}^{n}\mathbb{E}\left[(\theta^T x_i - \theta^{*T} x_i - \epsilon_i)^2\right]$$

$$= \underline{\frac{1}{n}\sum_{i=1}^{n}\mathbb{E}((\theta-\theta^*)^T x_i)^2} + \underbrace{\mathbb{E}[\epsilon_i^T(\dots)]}_{0} + \mathbb{E}[\epsilon_i^2]$$

$$\mathbb{E}[\epsilon_i^2] = \sigma^2$$

$$\underline{\hat{\theta}} \qquad \theta$$

What are we hoping to achieve?

$$R(\theta^*) = \mathbb{E}[\epsilon_i^2] = \sigma^2$$

- **Excess risk** --- the difference between the the performance of the learner and that of the oracle.

$$R(\theta) - \min_{\theta} R(\theta) = R(\theta) - \underset{R^*}{\underset{\uparrow}{R(\theta^*)}} = \frac{1}{n}\sum_{i=1}^{n}\mathbb{E}\left[\left(x_i^T(\theta-\theta^*)\right)^2\right]$$

$$\rightarrow = \underline{\frac{1}{n}\mathbb{E}\left[(\theta-\theta^*)^T X^T X(\theta-\theta^*)\right]}$$

$$\theta \leftarrow \hat{\theta} = (X^TX)^{-1}X^T\vec{y}$$

Recall the empirical risk minimizer here for this problem.

$$\hat{\theta} = \arg\min_{\theta} \frac{1}{n} \sum_{i \in [n]} (x_i^T \theta - y_i)^2$$

- aka: Ordinary Least square (OLS), MLE under Gaussian noise
- A convenient form using linear algebra

$$\hat{\theta} = \arg\min_{\theta} \frac{1}{n} \|X\theta - y\|_2^2$$

- A closed-form solution

$$\underline{\hat{\theta} = (X^T X)^{-1} X^T y}$$

Deriving an *expected* excess risk bound for the ERM estimator

def $\|\cdot\|_A^2 = (\cdot)^T A (\cdot)$

1. Working out the excess risk

$$R(\hat\theta) - R^* = \frac{1}{n}(\hat\theta - \theta^*)^T X^T X(\hat\theta - \theta^*) = \frac{1}{n}\|\hat\theta - \theta^*\|^2_{X^TX}$$

closed form $\hat\theta$ $\rightarrow$

$$= \frac{1}{n}\|(X^TX)^{-1}X^T\vec{y} - \theta^*\|^2_{X^TX}$$

assumption on the distribution of $\vec{y}$

$$= \frac{1}{n}\|(X^TX)^{-1}X^T(X\theta^* + \vec{\varepsilon}) - \theta^*\|^2_{X^TX}$$

$$= \frac{1}{n}\|\cancel{(X^TX)^{-1}X^TX}\theta^* - \theta^* + (X^TX)^{-1}X^T\vec{\varepsilon}\|^2_{X^TX}$$

What is random?

$$= \frac{1}{n}\vec{\varepsilon}^T X(X^TX)^{-1}X^TX(X^TX)^{-1}X^T\vec{\varepsilon} = \frac{1}{n}\vec{\varepsilon}^T \underbrace{X(X^TX)^{-1}X^T}\varepsilon$$

Π: an orthogonal projection to the subspace $\mathrm{col}(X)$

$\uparrow$ $\mathrm{span}(\{x_1, \cdots, x_n\})$

$$= \frac{1}{n}\mathrm{tr}(\varepsilon^T\Pi\varepsilon) = \frac{1}{n}\mathrm{tr}(\Pi \cdot \varepsilon\varepsilon^T)$$

$\uparrow$ Trace $\quad\quad$ $\uparrow$ $\mathrm{Trace}(ABC) = \mathrm{Trace}(BCA)$

orthogonal projection to a d-dimensional space

$\Pi \in \mathbb{R}^{n\times n}$ $\quad$ $\mathrm{rank}(\Pi) = d$

@ Eigen Decomposition

$\Pi = Q\Lambda Q^T$ where $\mathrm{diag}(\Lambda) = [\underbrace{1, 1, 1, \cdots, 1}_{d \text{ of these}}, 0, \cdots, 0]$

2. Take expectation

$$\mathbb{E}[R(\hat\theta)] - R^* = \frac{1}{n}\mathrm{tr}[\Pi \cdot \mathbb{E}[\varepsilon\varepsilon^T]]$$

$\boxed{\text{independent, zero mean, } \mathrm{Var} = \sigma^2}$ $\rightarrow$

$$= \frac{1}{n}\mathrm{tr}[\Pi \cdot \sigma^2 I_n]$$

$$= \frac{\sigma^2}{n}\mathrm{tr}[Q\Lambda Q^T \cdot I_n] = \frac{\sigma^2}{n}\sum_{i=1}^n \Lambda_{ii} = \frac{d\sigma^2}{n}$$

$\square$

Theorem for (fixed design) linear regression

Theorem: Assume (A1) and (A2), the ordinary least square estimator for linear regression satisfies:

$$\mathbb{E}[R(\hat{\theta})] - R(\theta^*) \leq \frac{d\sigma^2}{n} = O\left(\frac{1}{n}\right)$$

(*) X can be arbitrarily ill-conditioned,

The result relies on strong assumptions on how the data is generated

- e.g., it does NOT apply to the case for fitting a polynomial to a noisy sine function we gave earlier!

- The **statistical learning problem**:
  - Assumption B1: iid samples

    $(x_i, y_i) \overset{iid}{\sim} D$ unknown distribution

  - Assumption B2: Bounded loss function

    $\sup_{x,y,h} |\ell(h,(x,y))| \leq B$ $\quad 0 \leq \ell(h,(x,y)) \leq B$

  - Assumption B3: Finite hypothesis class

    $|\mathcal{H}| < +\infty$

The goal again is to bound the **excess risk** . This time we want a high probability bound.

- With probability at least $1 - \delta$

$$R(\hat{h}) - R(h^*) \leq \epsilon$$

- Parameterize ϵ as a function of
  - Number of data points
  - Size of the hypothesis class
  - Boundedness of the loss
  - Failure probability

Introducing two powerful “hammers”: Hammer 1. Hoeffding’s inequality

Theorem D.2 (Hoeffding’s inequality) *Let $X_1, \ldots, X_m$ be independent random variables with X_i taking values in $[a_i, b_i]$ for all $i \in [m]$. Then, for any $\epsilon > 0$, the following inequalities hold for $S_m = \sum_{i=1}^{m} X_i$:*

$$\mathbb{P}[S_m - \mathbb{E}[S_m] \geq \epsilon] \leq e^{-2\epsilon^2 / \sum_{i=1}^{m} (b_i - a_i)^2} \tag{D.4}$$

$$\mathbb{P}[S_m - \mathbb{E}[S_m] \leq -\epsilon] \leq e^{-2\epsilon^2 / \sum_{i=1}^{m} (b_i - a_i)^2} . \tag{D.5}$$

(see Appendix D.1 of FML textbook for a proof)

Roughly saying that the **empirical averages** of independent random variable converges to the **mean** at a O(1/sqrt(n)) rate, with high probability.

Introducing two powerful "hammers": Hammer 2. Union bound

Lemma (Union bound): For any probability distribution and any event E1, E2:

$$\mathbb{P}[E_1 \cup E_2] \leq \mathbb{P}[E_1] + \mathbb{P}[E_2]$$

Now let’s apply these two hammers to solve statistical learning

1. For each hypothesis h, apply Hoeffding

2. Union bound over all hypothesis

Now let’s apply these two hammers to solve statistical learning

Theorem: Assume (B1),(B2) and (B3), with probability at least $1-\delta$ (over the distribution of the data), ERM satisfies

$$R(\hat{h}) - R(h^*) = O\left(\sqrt{\frac{\log|\mathcal{H}| + \log(1/\delta)}{n}}\right)$$

Quiz 2: Application to decision tree classifier

- **d**-dimensional discrete feature (**L**-levels for each)
- **H**-layer decision tree, binary decision in one layer
- **K** Labels
- *Upper bound of the size of hypothesis class?*

Quiz 3: Application to generic classification (no restriction on the hypothesis class)

- **d**-dimensional discrete feature (**L**-levels for each)
- **K** labels
- ***Total number of unique classifiers?***

Computation-approximation tradeoff in the choice of hypothesis class

	model $p^*(y\|x)$	**Linear learners**	**Neural networks**
Computation	Depends on how complex p* is	Efficient	Not efficient in the worst case, but…
Approximation	No approximation	Large approx. error	Small approx. error
Statistical efficiency	Depends on how complex p* is	Need less data	Need more data

“All models are wrong, but some are useful.”

George Box
(1919 - 2013)

Checkpoint: Theory of supervised learning

- Risk bounds for linear regression model

$$\mathbb{E}[R(\hat{\theta})] - R(\theta^*) \leq \frac{d\sigma^2}{n}$$

- Risk bounds for a general supervised learning

$$R(\hat{h}) - R(h^*) = O\left(\sqrt{\frac{\log|\mathcal{H}| + \log(1/\delta)}{n}}\right)$$

- Observations:
  - Not directly comparable for several reasons
  - Strong assumption => Strong results
  - Weak assumption => Weak results

Remainder of this lecture

- Supervised learning:
  - formal notations and problem setup
  - Loss function, Risk, Empirical Risk
  - Examples

- Theory of supervised learning
  - Risk bounds for ‘fixed design’ linear regression model
  - Risk bounds for a general supervised learning problem

- Model selection

Typical problems in model selection

- Choosing hypothesis class
  - Decision tree?   Linear classifier? Or neural networks?
- Choose hyperparameters
  - Depth of decision tree
  - Regularization weights for Ridge / Lasso
- Choose which set of features of include

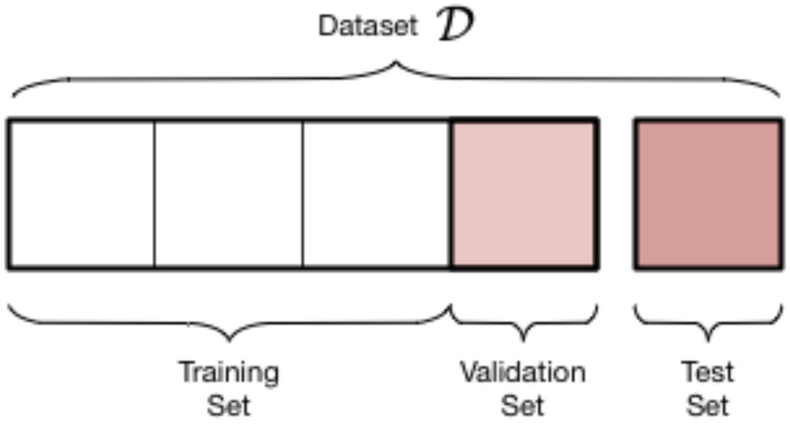
Model selection is challenging because we do not observe the actual *risk*!

- Empirical risk is often a poor surrogate due to the optimization bias
  - Example: 1-Nearest Neighbor classifier

- Two ideas for estimating the risk
  - Calculate or bound the actual risk in theory

  - Simulate the actual risk on a dataset not used for training.

Empirically measuring the *Risk* by splitting the data into: Training, Test, and Validation Sets



Validation set is used for model-selection:

- choosing decision tree vs. linear classifier
- Select features, tune hyperparameters

Test set is used only once to report the final results.

Cross-validation

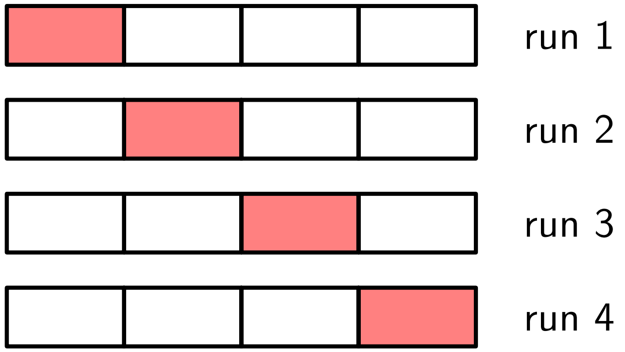


Figure 1.18 The technique of S-fold cross-validation, illustrated here for the case of $S = 4$, involves taking the available data and partitioning it into S groups (in the simplest case these are of equal size). Then $S - 1$ of the groups are used to train a set of models that are then evaluated on the remaining group. This procedure is then repeated for all S possible choices for the held-out group, indicated here by the red blocks, and the performance scores from the S runs are then averaged.

- Pros:
  - No assumption on the data generating distributions, except iid.
  - Do not waste data, comparing to holdout.

- Cons:
  - It evaluates the model applying to (S-1)/S fraction of the data
  - Computation cost = O(S * number of models to select from)

Other approaches for model selection

- AIC (Akaike Information Criteria) / BIC (Bayesian information criteria)
  - (see PRML Section 1.3 and 4.4.1)

- Effective degree of freedom
  - Measuring the effective number of parameters
  - For fixed-design regression with square loss + Gaussian noise, any estimator:

$$R(\hat{h}) - \mathbb{E}[\hat{R}(\hat{h})] = \frac{2\sigma^2}{n} df(\hat{h})$$

So if one can estimate df, then can use it for model selection

Effective degree of freedom for Regularized Linear Regression

- Ridge regression

$$df(X\hat{\theta}) = \mathrm{tr}(X(X^TX + \lambda I)^{-1}X^T)$$

  - Number of parameters, if no regularization
  - Independent to data y, can be computed ahead of time

- Lasso

$$df(X\hat{\theta}) = \mathbb{E}\left[\sum_{j\in[d]} \mathbb{I}(\hat{\theta}_j \neq 0)\right]$$

  - Expected number of non-zero weights -- Sparsity.
  - This is truly remarkable that we get this via L1-regularization

See e.g. : https://www.stat.cmu.edu/~ryantibs/papers/lassodf.pdf

Checkpoint: model selection

- Three approaches for model selection

  - Holdout
  - Cross validation
  - Penalize information criteria

- Cross validation is what is most commonly used in practice.

Next two lectures

- Unsupervised learning
  - Thursday
- Optimization methods for machine learning
  - Next Tuesday