

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.

+Alex Search Images Maps Play YouTube News

Google

Gmail ▾ C More ▾

COMPOSE

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

+Alex Search Images Maps Play YouTube News

Google

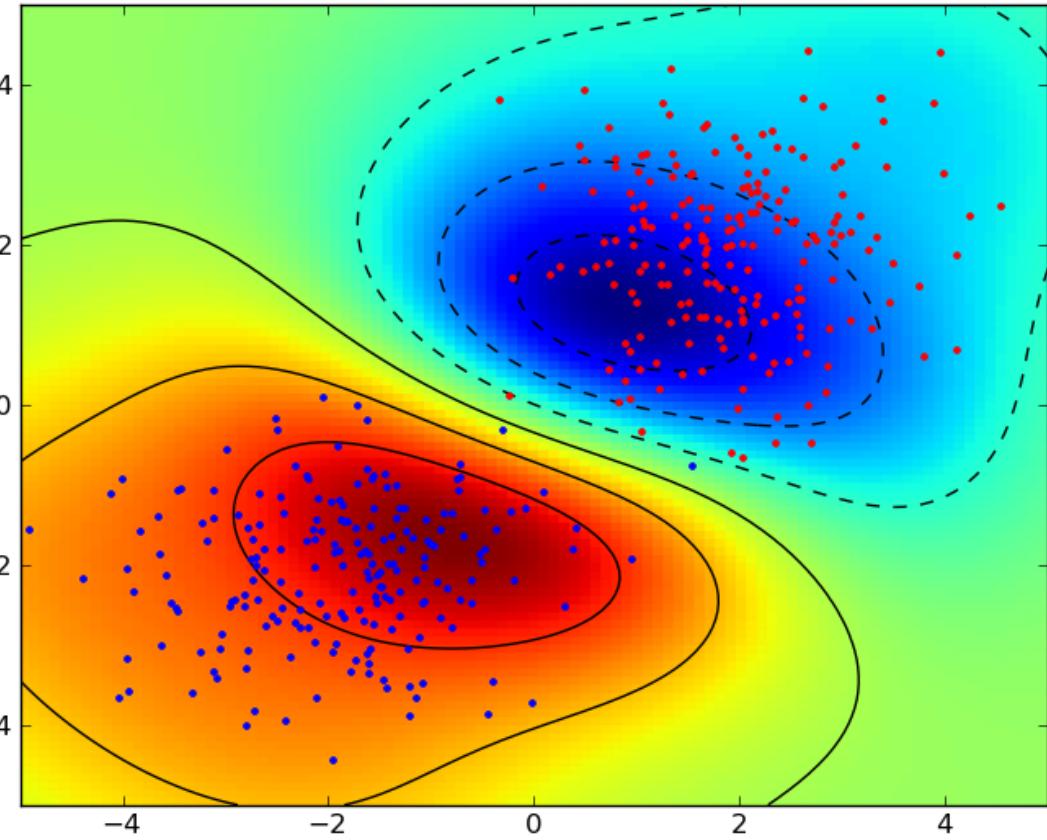
Gmail ▾ C More ▾

COMPOSE

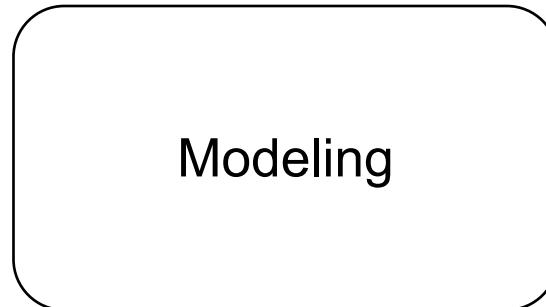
Inbox (7,180)
Important
Sent Mail
Drafts (61)
All Mail

▶ Circles

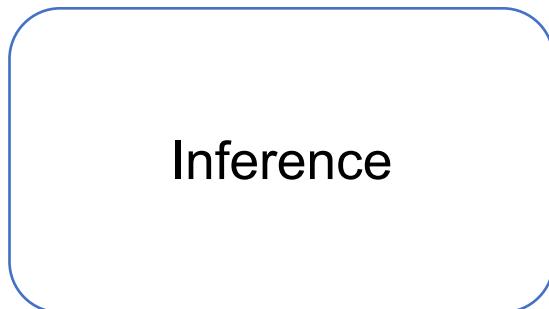
◀ [Gmail]
Done (1,006)
[Imap]/Drafts
[Imap]/Sent
alex.smola@yahoo...



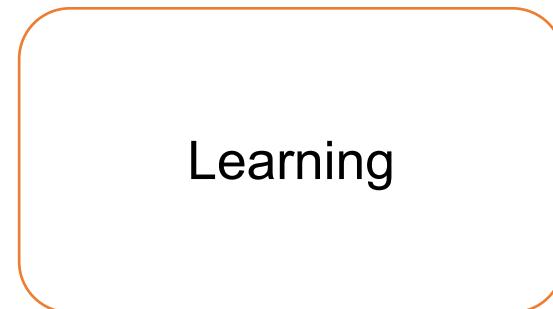
Recap: Modeling-Learning-inference in a machine learning workflow



- 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: $\mathcal{Y} = \{0, 1\} = \{\text{non-spam, spam}\}$
- A classifier (hypothesis): $h : \mathcal{X} \rightarrow \mathcal{Y}$
- A hypothesis class: \mathcal{H}
- Data: $(x_1, y_1), \dots, (x_n, y_n) \in \mathcal{X} \times \mathcal{Y}$
- Learning task: Find $h \in \mathcal{H}$ 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)
 - How many possible feature vectors are there?
 - How many classifiers are there (without restrictions)?

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

Notations from probability

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

$\mathbb{P}_{\mathcal{D}}$ [Event]

$f_{X \sim \mathcal{D}}(x)$

$F_{X \sim \mathcal{D}}(x)$

Conditional expectation / conditional probability / density

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

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

$f(x|y)$

Notations from linear algebra

- Matrices and vectors

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

- Transpose and inverse

$$A^T \in \mathbb{R}^{n \times m} \quad A^{-1} A = I$$

- Inner product / dot product

$$\mathbf{x}^\top \mathbf{y} = \sum_{i=1}^n x_i y_i.$$

Other useful notations

$B = (\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3)$	(Ordered) tuple
$\mathbf{B} = [\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3]$	Matrix of column vectors stacked horizontally
$\mathcal{B} = \{\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3\}$	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

$$[n] := \{1, 2, 3, \dots, n\}$$

Indicator (one-zero) function:

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

Conventions and typical meaning of specific variables in machine learning

- x : input
- y : output
- 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

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

- Risk function

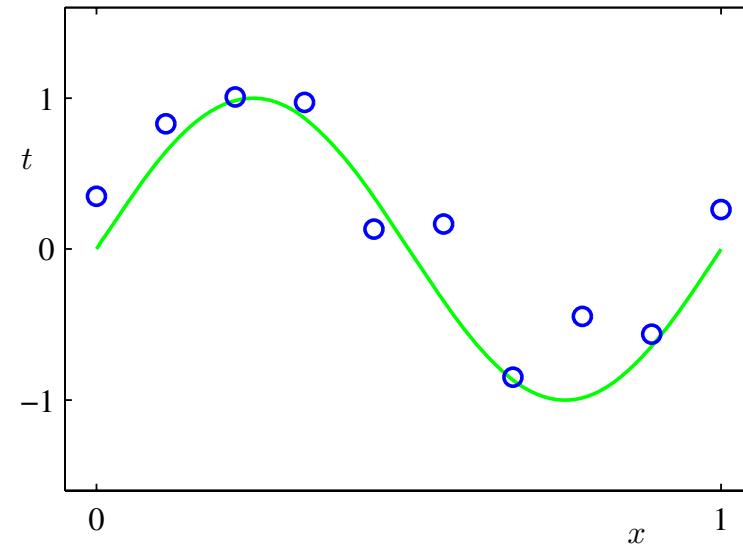
$$R(h, \mathcal{D}) = \mathbb{E}_{\mathcal{D}}[\ell(h, (x_i, y_i))]$$

- Empirical risk

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

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.



- What are the feature space, label space?
- What is a reasonable hypothesis class to use and its free-parameter?

Examples of hypothesis classes for this problem

- Polynomials

$$h(x, \mathbf{w}) = w_0 + w_1x + w_2x^2 + \dots + w_Mx^M = \sum_{j=0}^M w_jx^j$$

- Sine function

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

- Anything else?

What are some reasonable loss functions for regression problems?

- Square error loss function
- Absolute deviation loss function
- Huber loss function
- epsilon-sensitive loss function
 - aka support vector regression

Learning is often achieved by solving the Empirical Risk Minimization

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

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

$$\hat{h} = \arg \min_{h \in \mathcal{H}} \hat{R}(h, \text{Data}) + g(h)$$

Polynomial regression under square loss

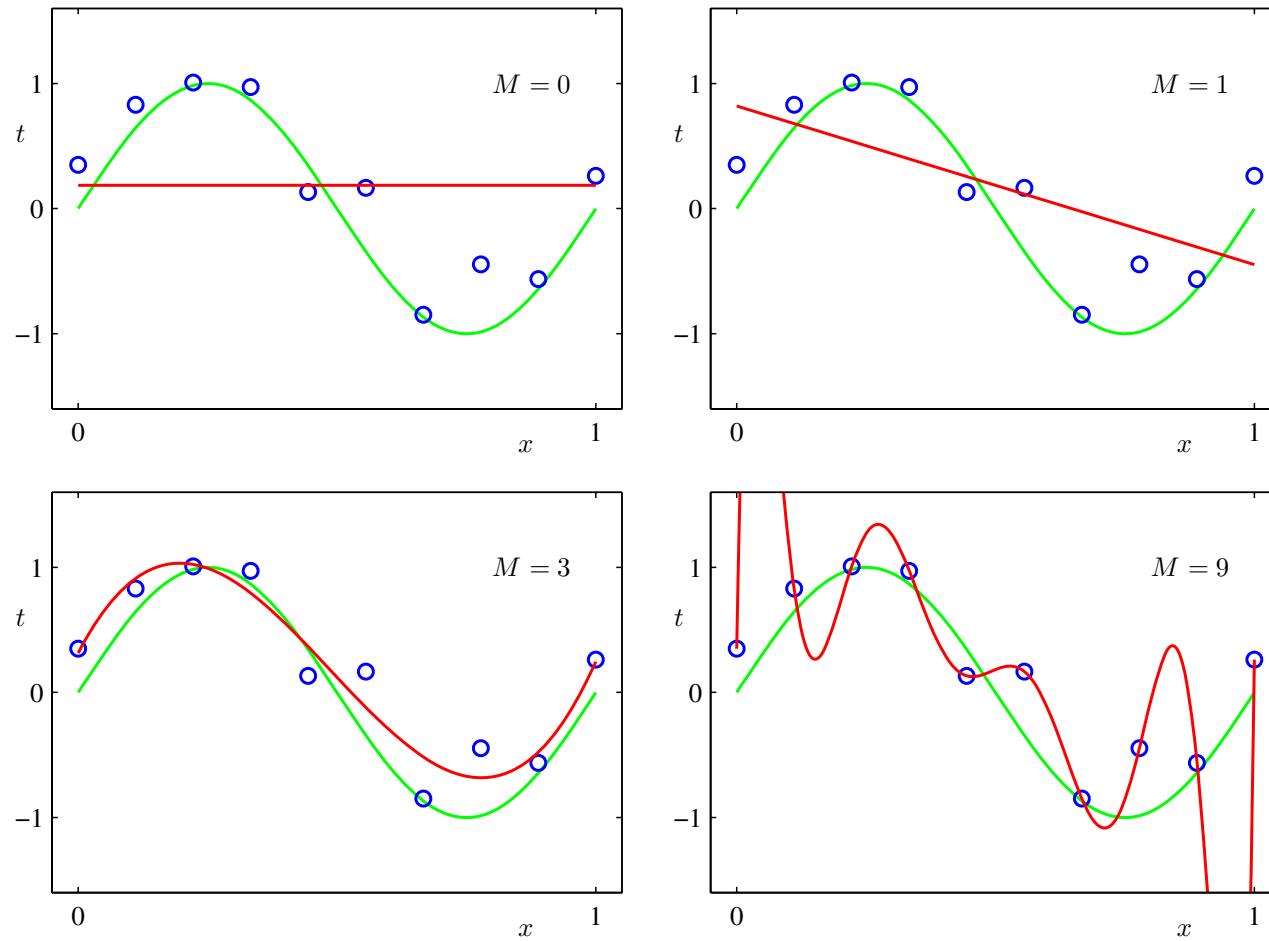
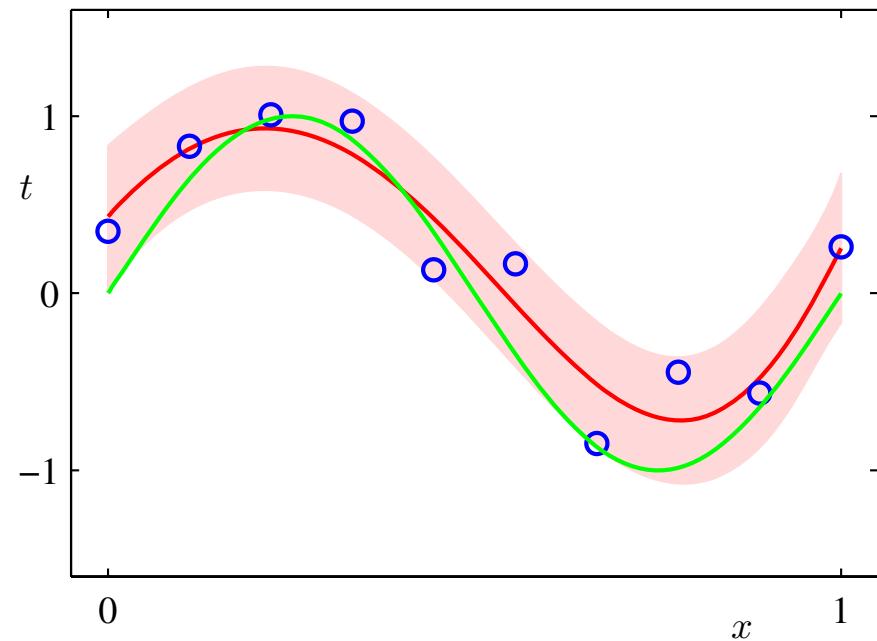


Figure 1.4 Plots of polynomials having various orders M , shown as red curves, fitted to the data set shown in Figure 1.2.

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.



Regularization prevents overfitting!

Example 2: Linear regression

- Feature space
- Label space
- Hypothesis space
- Loss function

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

- Q1: When hypothesis class is polynomial?
- Q2: When the hypothesis class is sine function with parameter t?

Empirical risk minimization for linear regression under square loss

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

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

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

- Example: p-norm regularized least square

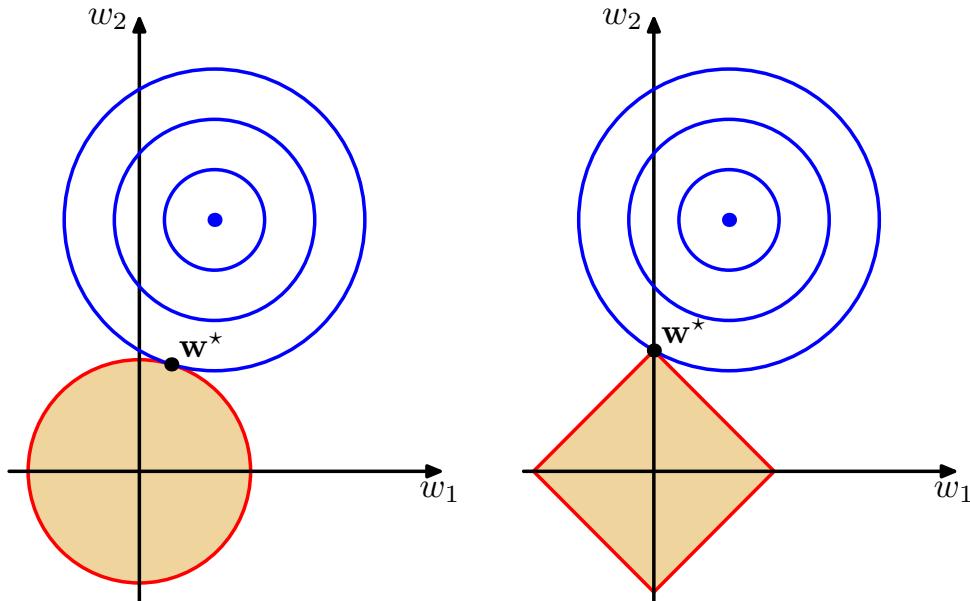
$$\hat{\theta} = \arg \min_{\theta} \frac{1}{n} \|X\theta - y\|_2^2 + \lambda \|\theta\|_p^p$$

- 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.

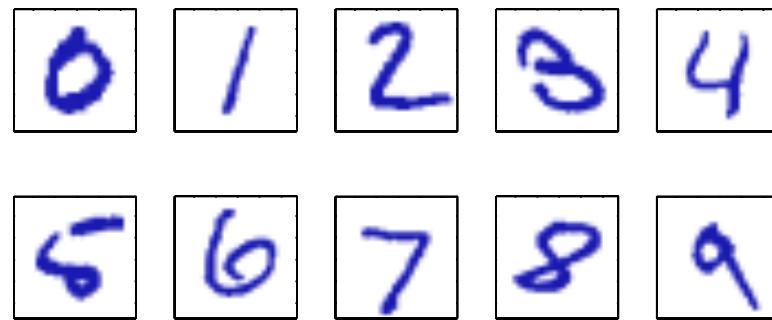
- Ridge regression induces solutions that are small but dense.
- Lasso induces solutions that are “sparse”.

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 w is denoted by w^* . The lasso gives a sparse solution in which $w_1^* = 0$.



Example 3: Multi-class classification

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

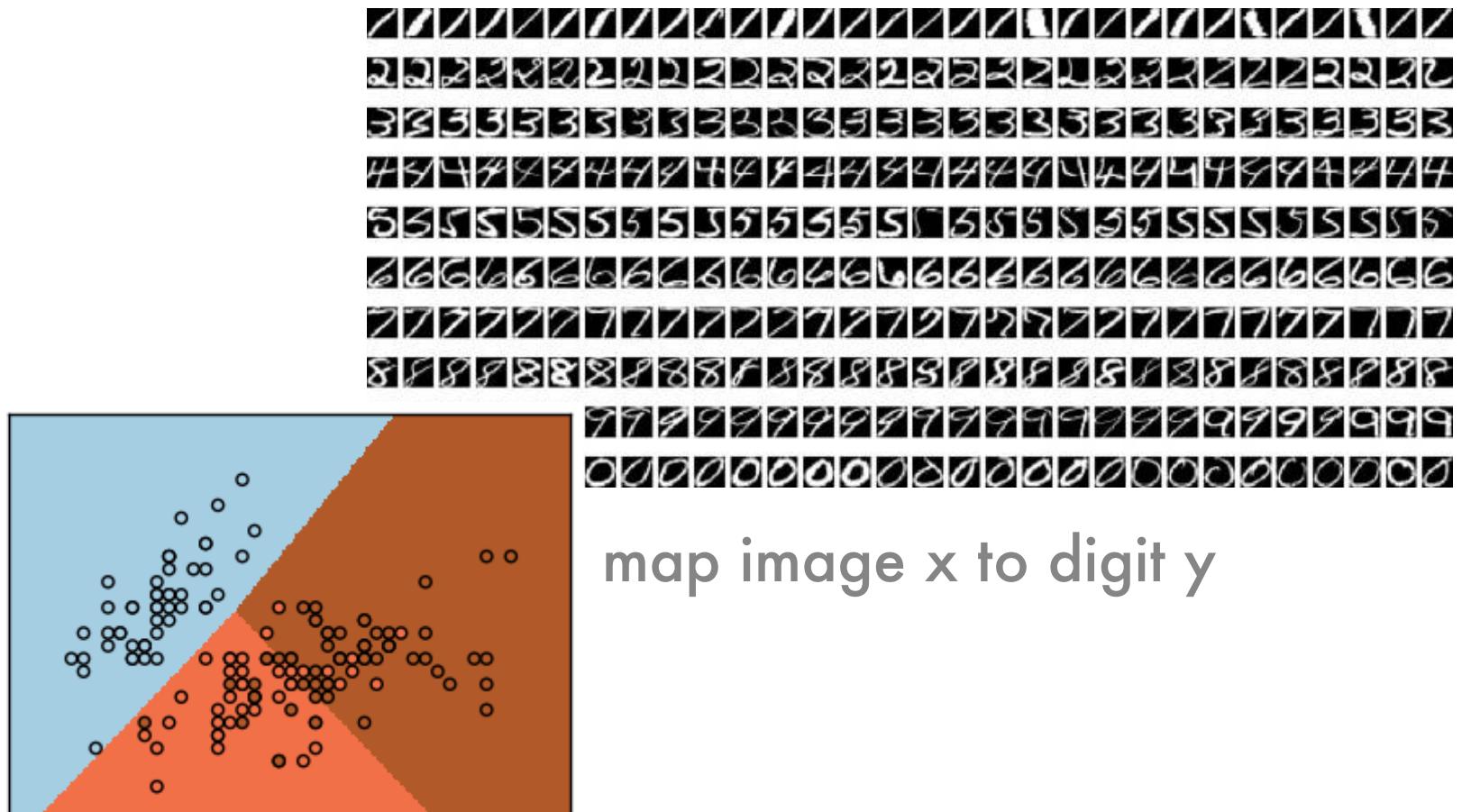


- What are the feature space, label space?

Hypothesis class for multi-class classification problems

- Decision trees
- Linear classifier

Illustration of the decision boundary in multi-class linear classification



Loss functions for classification tasks when the predictions are discrete

- 0-1 loss
- Cost-sensitive loss

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)

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

- Soft-argmax transform

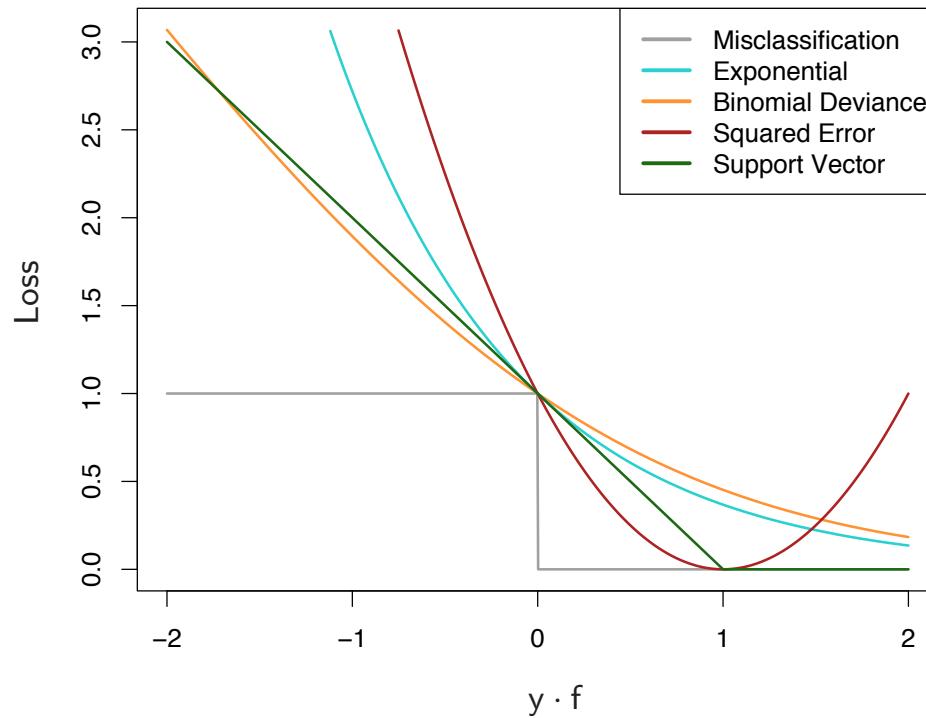
(Compare this to argmax in One-Hot representation)

$$F(x_1, \dots, x_n) = \frac{[e^{x_1}, \dots, e^{x_n}]}{\sum_{i=1}^n e^{x_i}}$$

Loss functions for classification tasks for soft-predictions

- log-loss
- Cross entropy loss
- Logistic loss in the binary case
- 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 $\text{sign}(f)$. The losses are misclassification: $I(\text{sign}(f) \neq y)$; 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

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

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?

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

$$y_i = x_i \cdot \theta^* + \epsilon_i, \quad \mathbb{E}[\epsilon_i] = 0 \quad \text{Var}[\epsilon_i] = \sigma^2$$

- A2. Fixed design matrix with full rank
- Risk function in this case

$$R(\theta) =$$

What are we hoping to achieve?

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

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

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

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

1. Working out the excess risk

2. Take expectation

Theorem for (fixed design) linear regression

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

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

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
 - Assumption B2: Bounded loss function
 - Assumption B3: Finite hypothesis class

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, \dots, 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 $S_m = \sum_{i=1}^m X_i$:

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

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

(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 E_1, E_2 :

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

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

- \mathbf{d} -dimensional discrete feature (\mathbf{L} -levels for each)
- \mathbf{H} -layer decision tree, binary decision in one layer
- \mathbf{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

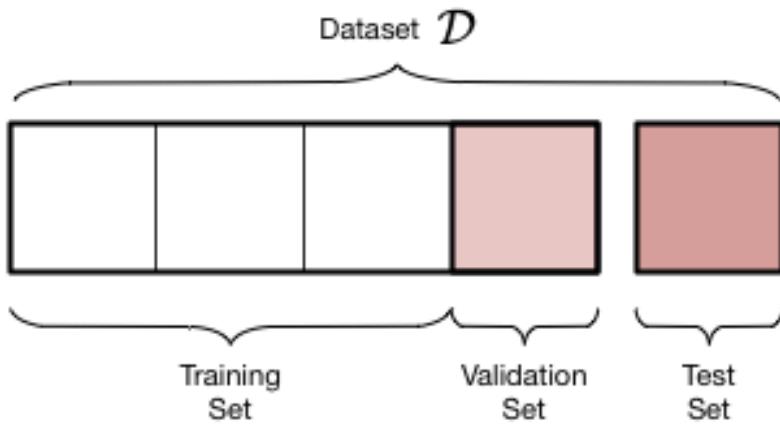
Typical problems in model selection

- 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



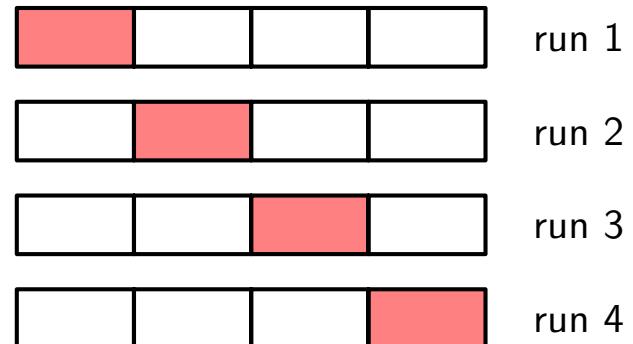
Validation set is used for model-selection:

- 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 = $O(S * \text{number of models to select from})$

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

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

- 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