CS-GY 6923: Lecture 3
Model Selection + Regularization + Bayesian Perspective

NYU Tandon School of Engineering, Prof. Christopher Musco

COURSE ADMIN

- Homework 1 due tonight.
- New lab will be released tonight, due next Thursday.
- Next problem set will be due a week after that.

LOSS MINIMIZATION

Basic machine learning problem:

- Given model $f_{\boldsymbol{\theta}}$ and loss function $L(f_{\boldsymbol{\theta}})$.
- Choose $\boldsymbol{\theta}^*$ to minimize $L(f_{\boldsymbol{\theta}})$.

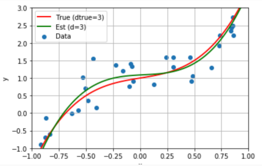
MODEL SELECTION

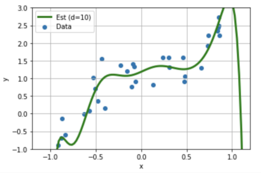
Model selection problem:

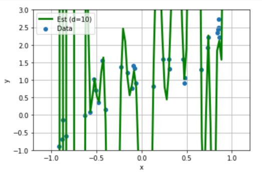
- Given choice of many models $f^{(1)}_{\boldsymbol{\theta}_1}, f^{(2)}_{\boldsymbol{\theta}_2}, \ldots, f^{(q)}_{\boldsymbol{\theta}_q}$.
- Choose $\boldsymbol{\theta}_1^*, \ldots, \boldsymbol{\theta}_q^*$ to minimize $L(f_{\boldsymbol{\theta}_1}), \ldots, L(f_{\boldsymbol{\theta}_q})$.
- Then choose the “best” model for our data.

MODEL SELECTION EXAMPLE

Polynomial regression models with different degree. See `demo_polyfit.ipynb`.



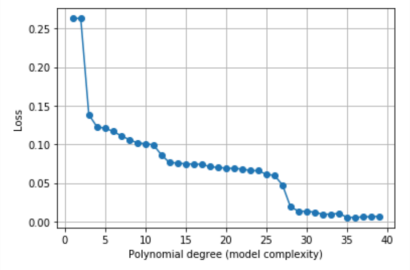




- Model $f^{(1)}_{\boldsymbol{\theta}_1}$: all linear functions.
- Model $f^{(2)}_{\boldsymbol{\theta}_2}$: all quadratic functions.
- Model $f^{(3)}_{\boldsymbol{\theta}_3}$: all cubic functions.
- $\ldots$

MODEL SELECTION

The more **complex** our model class (e.g., the higher degree we allow in polynomial regression) the better our loss:



Training loss alone is not usually a good metric for model selection. Small loss does not imply generalization to new data.

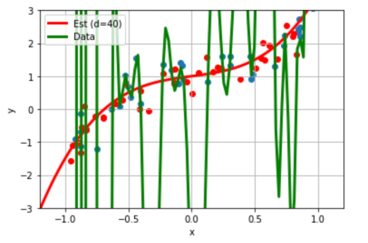
TRAIN-TEST PARADIGM

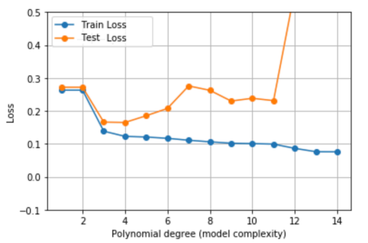
Main approach: Evaluate model on fresh test data which was not used during training.

Test/train split:

- Given data set $(\mathbf{X}, \mathbf{y})$, split into two sets $(\mathbf{X}_{\text{train}}, \mathbf{y}_{\text{train}})$ and $(\mathbf{X}_{\text{test}}, \mathbf{y}_{\text{test}})$.
- Train q models $f^{(1)}, \ldots, f^{(q)}$ by finding parameters which minimize the loss on $(\mathbf{X}_{\text{train}}, \mathbf{y}_{\text{train}})$.
- Evaluate loss of each trained model on $(\mathbf{X}_{\text{test}}, \mathbf{y}_{\text{test}})$.

GENERALIZATION

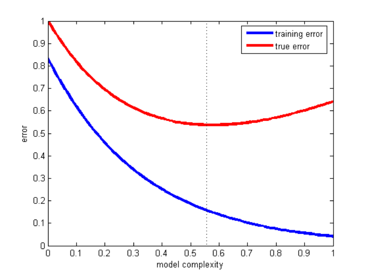




While train error always decreases, we eventually see test error increase with increasing model complexity.

THE FUNDAMENTAL CURVE OF ML

The above trend is fairly representative of what we tend to see across the board:



TRAIN-TEST INTUITION

Is “test error” the end goal though? Don’t we care about “future” error?

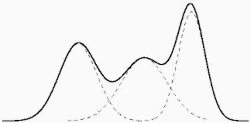
Intuition: Models which perform better on the test set will generalize better to future data.

Goal: Introduce a little bit of formalism to better understand what this means. What is “future” data?

STATISTICAL LEARNING MODEL

Statistical Learning Model:

- Assume each data example is randomly drawn from some distribution $(\mathbf{x}, y) \sim \mathcal{D}$.



✓

✓

✓

E.g. $x_1, \ldots, x_d$ are Gaussian random variables with parameters $\mu_1, \sigma_1, \ldots, \mu_d, \sigma_d$.

This is not a simplifying assumptions! The distribution could be arbitrarily complicated.

RISK

Statistical Learning Model:

- Assume each data example is randomly drawn from some distribution $(\mathbf{x}, y) \sim \mathcal{D}$.
- Define the Risk of a model/parameters:

$$R(f, \boldsymbol{\theta}) = \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\left[L\left(f(\mathbf{x}, \boldsymbol{\theta}), y\right)\right]$$

here L is our loss function (e.g. $L(z, y) = |z - y|$ or $L(z, y) = (z - y)^2$).

Goal: Find model $f \in \{f^{(1)}, \ldots, f^{(q)}\}$ and parameter vector $\boldsymbol{\theta}$ to minimize the $R(f, \boldsymbol{\theta})$.

RISK

- (Population) Risk:

$$R(f, \boldsymbol{\theta}) = \mathbb{E}_{(\mathsf{x}, y) \sim \mathcal{D}} \left[L \left(f(\mathsf{x}, \boldsymbol{\theta}), y \right) \right]$$

- **Empirical Risk**: Draw $(\mathsf{x}_1, y_1), \ldots, (\mathsf{x}_n, y_n) \sim \mathcal{D}$

$$R_E(f, \boldsymbol{\theta}) = \frac{1}{n} \sum_{i=1}^{n} L \left(f(\mathsf{x}, \boldsymbol{\theta}), y \right)$$

EMPIRICAL RISK

For any fixed model f and parameters $\boldsymbol{\theta}$,

$$\mathbb{E}\left[R_E(f, \boldsymbol{\theta})\right] = R(f, \boldsymbol{\theta}).$$

Only true if f and $\boldsymbol{\theta}$ are chosen *without looking at the data used to compute the empirical risk.*

MODEL SELECTION

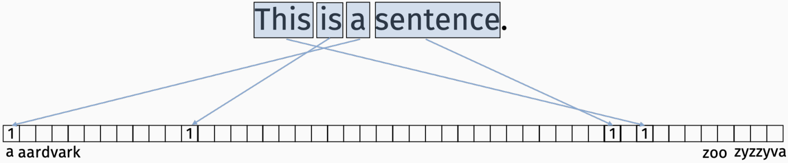
- Train q models $(f^{(1)}, \boldsymbol{\theta}_1^*), \ldots, (f^{(q)}, \boldsymbol{\theta}_q^*)$.
- For each model, compute empirical risk $R_E(f^{(i)}, \boldsymbol{\theta}_i^*)$ using test data.
- Since we assume our original dataset was drawn independently from $\mathcal{D}$, so is the random test subset.

No matter how our models were trained or how complex they are, $R_E(f^{(i)}, \boldsymbol{\theta}_i^*)$ is an unbiased estimate of the true risk $R(f^{(i)}, \boldsymbol{\theta}_i^*)$ for every i. Can use it to distinguish between models.

MODEL SELECTION EXAMPLE

bag-of-words models and **n-grams**

Common way to represent documents (emails, webpages, books) as numerical data. The ultimate example of 1-hot encoding.

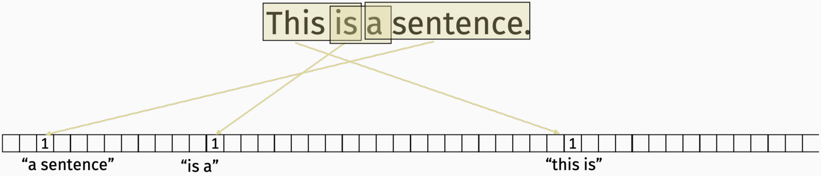


bag-of-words

MODEL SELECTION EXAMPLE

bag-of-words models and **n-grams**

Common way to represent documents (emails, webpages, books) as numerical data. The ultimate example of 1-hot encoding.

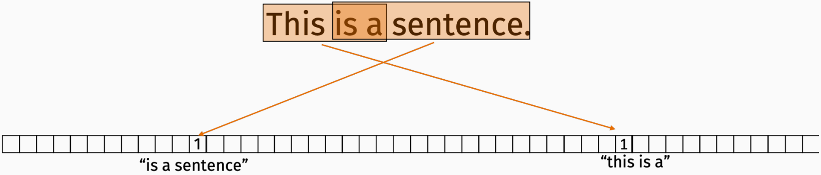


bi-grams

MODEL SELECTION EXAMPLE

bag-of-words models and **n-grams**

Common way to represent documents (emails, webpages, books) as numerical data. The ultimate example of 1-hot encoding.



tri-grams

model selection example

Models of increasing order:

- Model $f^{(1)}_{\boldsymbol{\theta}_1}$: spam filter that looks at **single words**.
- Model $f^{(2)}_{\boldsymbol{\theta}_2}$: spam filter that looks at **bi-grams**.
- Model $f^{(3)}_{\boldsymbol{\theta}_3}$: spam filter that looks at **tri-grams**.
- $\ldots$

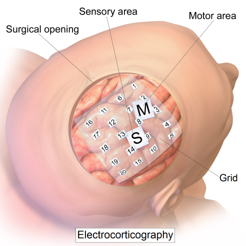
“interest” “low interest” “low interest loan”

Increased length of **n-gram** means more expressive power.

MODEL SELECTION EXAMPLE

Electrocorticography ECoG (upcoming lab):

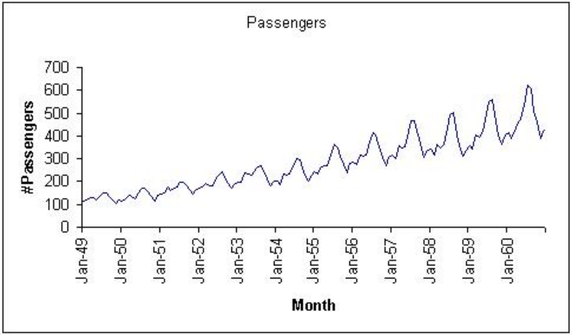
- Implant grid of electrodes on surface of the brain to measure electrical activity in different regions.



- Predict hand motion based on ECoG measurements.
- **Model order:** predict movement at time t using brain signals at time $t, t-1, \ldots, t-q$ for varying values of q.

AUTOREGRESSIVE MODEL

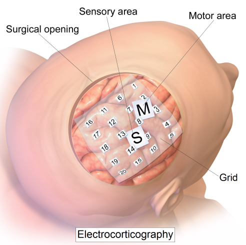
Predicting time t based on a linear function of the signals at time $t, t-1, \ldots, t-q$ is <u>not the same</u> as fitting a line to the time series. It's much more expressive.



Predecessor of modern "recurrent neural networks".

MODEL SELECTION LAB TIP

Electrocorticography ECoG lab:



First lab where computation actually matters (solving regression problems with $\sim 40k$ examples, ~ 1500 features)

Makes sense to test and debug code using a subset of the data.

ADAPTIVE DATA ANALYSIS

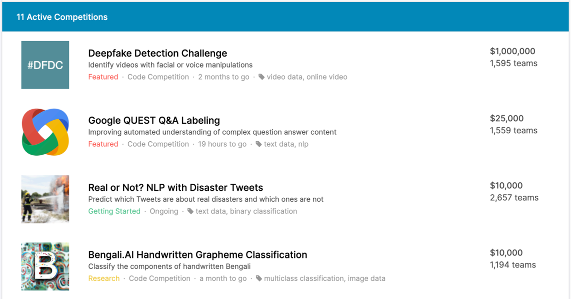
Slight caveat: This is typically not how machine learning or scientific discovery works in practice!

Typical workflow:

- Train a class of models.
- Test.
- Adjust class of models.
- Test.
- Adjust class of models.
- Cont...

Final model implicitly depends on test set because performance on the test set guided how we changed our model.

Popularity of ML benchmarks and competitions leads to adaptivity at a massive scale.



11 Active Competitions

Deepfake Detection Challenge
Identify videos with facial or voice manipulations
Featured · Code Competition · 2 months to go · video data, online video
$1,000,000
1,595 teams

Google QUEST Q&A Labeling
Improving automated understanding of complex question answer content
Featured · Code Competition · 19 hours to go · text data, nlp
$25,000
1,559 teams

Real or Not? NLP with Disaster Tweets
Predict which Tweets are about real disasters and which ones are not
Getting Started · Ongoing · text data, binary classification
$10,000
2,657 teams

Bengali.AI Handwritten Grapheme Classification
Classify the components of handwritten Bengali
Research · Code Competition · a month to go · multiclass classification, image data
$10,000
1,194 teams

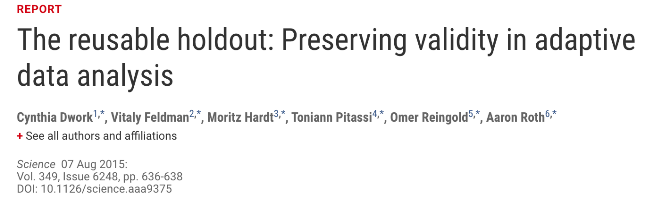
Kaggle (various competitions)

14,197,122 images, 21841 synsets indexed

Imagenet (image classification and categorization)

ADAPTIVE DATA ANALYSIS

Is adaptivity a problem? Does it lead to over-fitting? How much? How can we prevent it? All current research.



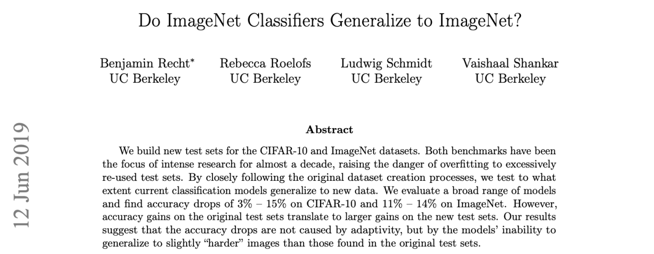
REPORT

The reusable holdout: Preserving validity in adaptive data analysis

Cynthia Dwork[1,*], Vitaly Feldman[2,*], Moritz Hardt[3,*], Toniann Pitassi[4,*], Omer Reingold[5,*], Aaron Roth[6,*]

+ See all authors and affiliations

Science 07 Aug 2015:
Vol. 349, Issue 6248, pp. 636-638
DOI: 10.1126/science.aaa9375



Do ImageNet Classifiers Generalize to ImageNet?

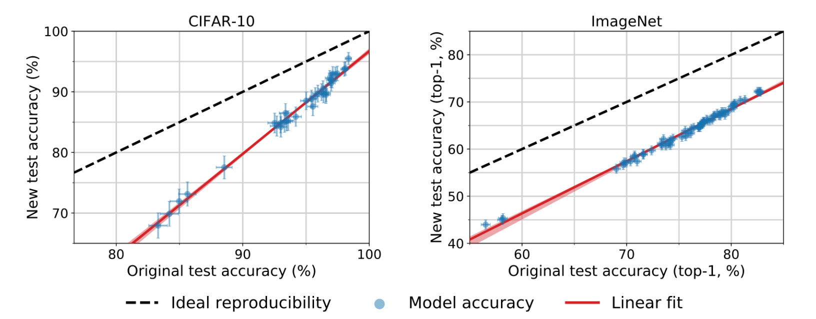
Benjamin Recht* UC Berkeley, Rebecca Roelofs UC Berkeley, Ludwig Schmidt UC Berkeley, Vaishaal Shankar UC Berkeley

12 Jun 2019

Abstract

We build new test sets for the CIFAR-10 and ImageNet datasets. Both benchmarks have been the focus of intense research for almost a decade, raising the danger of overfitting to excessively re-used test sets. By closely following the original dataset creation processes, we test to what extent current classification models generalize to new data. We evaluate a broad range of models and find accuracy drops of 3% – 15% on CIFAR-10 and 11% – 14% on ImageNet. However, accuracy gains on the original test sets translate to larger gains on the new test sets. Our results suggest that the accuracy drops are not caused by adaptivity, but by the models' inability to generalize to slightly "harder" images than those found in the original test sets.

Do ImageNet Classifiers Generalized to ImageNet?



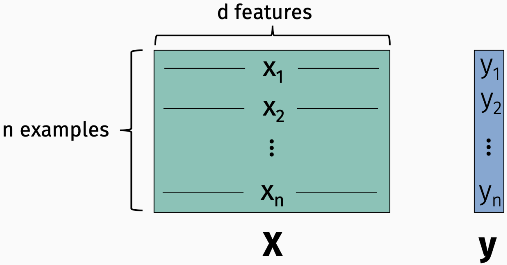
Interestingly, when comparing popular vision models on "fresh" data, while performance dropped across the board, the relative rank of model performance did not change significantly.

REGULARIZATION

OVER-PARAMETERIZED MODELS

In all the model selection examples we discussed we had full control over the complexity of the model: could range from underfitting to overfitting.

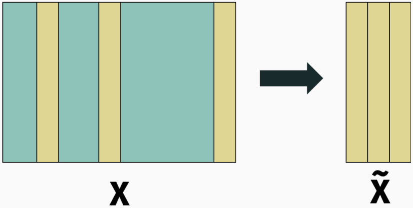
In practice, you often don't have this freedom. Even the most basic model might lead to overfitting.



Example: Linear regression model where $d \geq n$. Can always find β so that $\mathbf{X}\beta = \mathbf{y}$ exactly.

FEATURE SELECTION

Select some subset of features to use in model:



Filter method: Compute some metric for each feature, and select features with highest score.

- Example: compute loss or R^2 value when each feature in $\mathbf{X}$ is used in single variate regression.

Any potential limitations of this approach?

FEATURE SELECTION

Exhaustive approach: Pick best subset of q features.

Faster approach: Greedily select q features.

Stepwise Regression:

- **Forward:** Step 1: pick single feature that gives lowest loss. Step k: pick feature that when combined with previous $k-1$ chosen features gives lowest loss.
- **Backward:** Start with all of the features. Greedily eliminate those which have least impact on model performance.

Feature selection deserves more than two slides, but we won't go into too much more detail!

ALTERNATIVE APPROACH

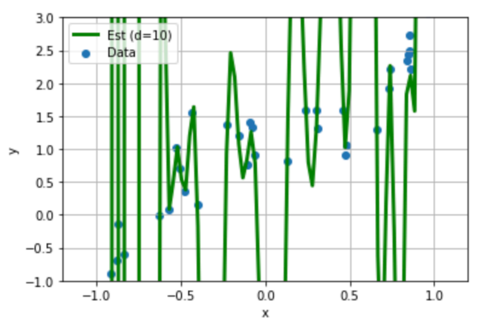
Regularization: Explicitly discourage overfitting by adding a regularization penalty to the loss minimization problem.

$$\min_{\boldsymbol{\theta}} \left[L(\boldsymbol{\theta}) + Reg(\boldsymbol{\theta}) \right].$$

Example: Least squares regression. $L(\boldsymbol{\beta}) = \|\mathbf{X}\boldsymbol{\beta} - \mathbf{y}\|_2^2$.

- Ridge regression (ℓ_2): $Reg(\boldsymbol{\beta}) = \lambda \|\boldsymbol{\beta}\|_2^2$
- LASSO (least absolute shrinkage and selection operator) (ℓ_1): $Reg(\boldsymbol{\beta}) = \lambda \|\boldsymbol{\beta}\|_1$
- Elastic net: $Reg(\boldsymbol{\beta}) = \lambda_1 \|\boldsymbol{\beta}\|_1 + \lambda_2 \|\boldsymbol{\beta}\|_2^2$

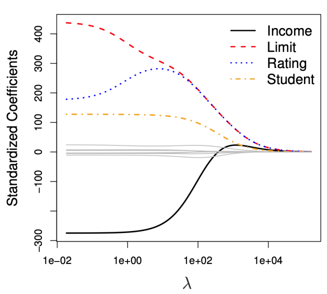
REGULARIZATION



RIDGE REGULARIZATION

Ridge regression: $\min_{\beta} \|\mathbf{X}\beta - \mathbf{y}\|_2^2 + \lambda\|\beta\|_2^2$.

- As $\lambda \to \infty$, we expect $\|\beta\|_2^2 \to 0$ and $\|\mathbf{X}\beta - \mathbf{y}\|_2^2 \to \|\mathbf{y}\|_2^2$.
- Feature selection methods attempt to set many coordinates in β to 0. Ridge regularizations encourages coordinates to be small.



DUALITY WITH CONSTRAINED REGRESSION

Ridge regression: $\min_{\boldsymbol{\beta}} \|\mathbf{X}\boldsymbol{\beta} - \mathbf{y}\|_2^2 + \lambda\|\boldsymbol{\beta}\|_2^2$.

- Can be viewed as shrinking the size of our model class. Relaxed version of $\min_{\boldsymbol{\beta}:\|\boldsymbol{\beta}\|_2^2 < c} \|\mathbf{X}\boldsymbol{\beta} - \mathbf{y}\|_2^2$.

Claim: For any λ, let $\boldsymbol{\beta}_\lambda^* = \arg\min_{\boldsymbol{\beta}} \|\mathbf{X}\boldsymbol{\beta} - \mathbf{y}\|_2^2 + \lambda\|\boldsymbol{\beta}\|_2^2$. Then there is some $c(\lambda)$ such that:

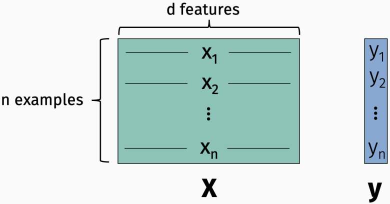
$$\boldsymbol{\beta}_\lambda^* = \arg\min_{\boldsymbol{\beta}:\|\boldsymbol{\beta}\|_2^2 < c(\lambda)} \|\mathbf{X}\boldsymbol{\beta} - \mathbf{y}\|_2^2.$$

Moreover, we have the for $\lambda' > \lambda$, $c(\lambda') < c(\lambda)$.

RIDGE REGULARIZATION

Ridge regression: $\min_{\boldsymbol{\beta}} \|\mathbf{X}\boldsymbol{\beta} - \mathbf{y}\|_2^2 + \lambda\|\boldsymbol{\beta}\|_2^2$.

- $\min_{\boldsymbol{\beta}:\|\boldsymbol{\beta}\|_2^2<c} \|\mathbf{X}\boldsymbol{\beta} - \mathbf{y}\|_2^2$ won't have a solution at zero for all $\mathbf{y}$, even when over-parameterized.



- Regularization methods are not invariant to data scaling. Typically when using regularization we mean center and scale columns to have unit variance.

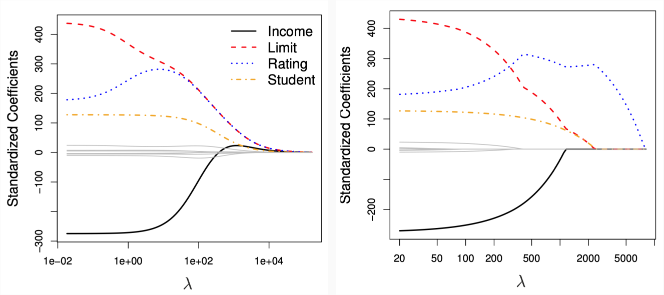
RIDGE REGULARIZATION

How do we minimize: $L_R(\beta) = \|X\beta - y\|_2^2 + \lambda\|\beta\|_2^2$?

LASSO REGULARIZATION

Lasso regularization: $\min_{\beta} \|\mathbf{X}\beta - \mathbf{y}\|_2^2 + \lambda\|\beta\|_1$.

- As $\lambda \to \infty$, we expect $\|\beta\|_1 \to 0$ and $\|\mathbf{X}\beta - \mathbf{y}\|_2^2 \to \|\mathbf{y}\|_2^2$.
- Typically encourages subset of β_i's to go to zero, in contrast to ridge regularization.



LASSO REGULARIZATION

Pros:

- Simpler, more interpretable model.
- More intuitive reduction in model order.

Cons:

- No closed form solution because $\|\beta\|_1$ is not differentiable.
- Can be solved with iterative methods, but generally not as quickly as ridge regression.

REGULARIZATION

Notes:

- Model selection/cross validation used to choose optimal scaling λ on $\lambda\|\boldsymbol{\beta}\|_2^2$ or $\lambda\|\boldsymbol{\beta}\|_1$.
- Often grid search for best parameters is performed in “log space”. E.g. consider $[\lambda_1, \ldots, \lambda_q] = 1.5^{[-4,-3,-2,-1,-0,1,2,3,4]}$.

THE BAYESIAN/PROBABILISTIC MODELING PERSPECTIVE

CLASSIFICATION SETUP

- **Data Examples:** $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathbb{R}^d$
- **Target:** $y_1, \ldots, y_n \in \{0, 2, \ldots, q-1\}$ when there are q classes.
  - Binary Classification: $q = 2$, so each $y_i \in \{0, 1\}$.
  - Multi-class Classification: $q > 2$. [1]

[1] Note that there is also multi-label classification where each data example maybe belong to more than one class.

CLASSIFICATION EXAMPLES

- Medical diagnosis from MRI: 2 classes.
- MNIST digits: 10 classes.
- Full Optical Character Regonition: 100s of classes.
- ImageNet challenge: 21,000 classes.

Running example today: Email Spam Classification.

CLASSIFICATION

Classification can (and often is) solved using the same **loss-minimization framework** we saw for regression.

We won’t see that today! We’re going to use classification as a window into another way of thinking about machine learning.

Will give new an interesting justifications for tools like regularization.

Today: ML from a Probabilistic Modeling/Bayesian Perspective.

PROBABILISTIC MODELING

In a Bayesian or Probabilistic approach to machine learning we always start by conjecturing a

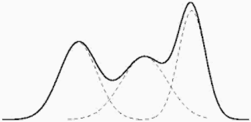
probabilistic model

that plausibly could have generated our data.

- The model guides how we make predictions.
- The model typically has unknown parameters $\vec{\theta}$ and we try to find the most reasonable parameters based on observed data (more on this later in lecture).

PROBABILISTIC MODELING

Typically we try to keep things simple!



✓

X

X

Exercise: Come up with a probabilistic model for any one of the following data sets $(x_1, y_1), \ldots, (x_n, y_n)$.

1. For n **people**: each $x_i \in \{0, 1\}$ with zero indicating male, one indicating female. Each y_i is the height of the person in inches.
2. For n **NYC apartments**: each x_i is the size of the apartment in square feet. Each y_i is the monthly rent in dollars.
3. For n **students**: each $x_i \in \{Fresh., Soph., Jun., Sen.\}$ indicating class year. Each $y_i \in \{0, 1\}$ with zero indicating the student has not taken machine learning, one indicating they have.

What are the unknown parameters of your model. What would be a guess for their values? How would you confirm or refine this guess using data?

PROBABILISTIC MODELING

Dataset: $(x_1, y_1), \ldots, (x_n, y_n)$

Description: For n **people**: each $x_i \in \{0, 1\}$ with zero indicating male, one indicating female. Each y_i is the height of the person in inches.

Model:

Dataset: $(x_1, y_1), \ldots, (x_n, y_n)$

Description: For n **NYC apartments**: each x_i is the size of the apartment in square feet. Each y_i is the monthly rent in dollars.

Model:

probabilistic modeling

Dataset: $(x_1, y_1), \ldots, (x_n, y_n)$

Description: For n **students**: each $x_i \in \{Fresh., Soph., Jun., Sen.\}$ indicating class year. Each $y_1 \in \{0, 1\}$ with zero indicating the student has not taken machine learning, one indicating they have.

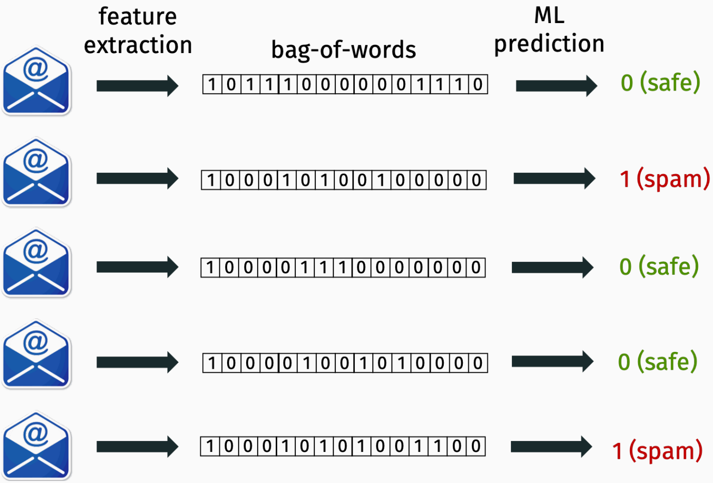
Model:

NAIVE BAYES CLASSIFIER

Goal:

- Build a probabilistic model for a binary classification problem.
- Estimate parameters of the model.
- From the model derive a classification rule for future predictions (the Naive Bayes Classifier).

SPAM PREDICTION



feature extraction

bag-of-words

ML prediction

1	0	1	1	1	0	0	0	0	0	0	1	1	1	0
1	0	0	0	1	0	1	0	0	1	0	0	0	0	0
1	0	0	0	0	1	1	1	0	0	0	0	0	0	0
1	0	0	0	0	1	0	0	1	0	1	0	0	0	0
1	0	0	0	1	0	1	0	1	0	0	1	1	0	0

0 (safe)

1 (spam)

0 (safe)

0 (safe)

1 (spam)

Both target labels and data vectors are binary.

PROBABILISTIC MODEL FOR EMAIL

Probabilistic model for (bag-of-words, label) pair $(\mathbf{x}, y)$:

- Set $y = 0$ with probability p_0, $y = 1$ with probability $p_1 = 1 - p_0$.
  - p_0 is probability an email is not spam (e.g. 99%).
  - p_1 is probability an email is spam (e.g. 1%).
- If $y = 0$, for each i, set $x_i = 1$ with prob. p_{i0}.
- If $y = 1$, for each i, set $x_i = 1$ with prob. p_{i1}.

Unknown model parameters:

- p_0, p_1,
- $p_{10}, p_{20}, \ldots p_{n0}$, one for each of the n vocabulary words.
- $p_{11}, p_{21}, \ldots p_{n1}$, one for each of the n vocabulary words.

How would you estimate these parameters?

PARAMETER ESTIMATION

Reasonable way to set parameters:

- Set p_0 and p_1 to the empirical fraction of not spam/spam emails.
- For each word i, set p_{i0} to the empirical probability word i appears in a non-spam email.
- For each word i, set p_{i1} to the empirical probability word i appears in a spam email.

Estimating these parameters from previous data examples is the only “training” we will do.

DONE WITH MODELING

ON TO PREDICTION

PROBABILITY REVIEW

- **Probability:** p(x) – the probability event x happens.
- **Joint probability:** p(x,y) – the probability that event x and event y happen.
- **Conditional Probability** $p(x \mid y)$ – the probability x happens given that y happens.

$$p(x|y) =$$

bayes theorem/rule

$$p(x|y) = \frac{p(y|x)p(x)}{p(y)}$$

Proof:

CLASSIFICATION RULE

Given unlabeled input $(\mathbf{x}, \_\_\_)$, choose the label $y \in \{0, 1\}$ which is most likely given the data. Recall $\mathbf{x} = [0, 0, 1, \ldots, 1, 0]$.

Classification rule: maximum a posterior prob. (MAP) estimate.

Step 1. Compute:

- $p(y = 0 \mid \mathbf{x})$: prob. $y = 0$ given observed data vector $\mathbf{x}$.
- $p(y = 1 \mid \mathbf{x})$: prob. $y = 1$ given observed data vector $\mathbf{x}$.

Step 2. Output: 0 or 1 depending on which probability is larger.

$p(y = 0 \mid \mathbf{x})$ and $p(y = 1 \mid \mathbf{x})$ are called **posterior** probabilities.

EVALUATING THE POSTERIOR

How to compute the posterior? **Bayes rule!**

$$p(y = 0 \mid \mathsf{x}) = \frac{p(\mathsf{x} \mid y = 0)p(y = 0)}{p(\mathsf{x})} \tag{1}$$

$$\text{posterior} = \frac{\text{likelihood} \times \text{prior}}{\text{evidence}} \tag{2}$$

- **Prior:** Probability in class 0 prior to seeing any data.
- **Posterior:** Probability in class 0 after seeing the data.

EVALUATING THE POSTERIOR

Goal is to determine which is larger:

$$p(y = 0 \mid \mathbf{x}) = \frac{p(\mathbf{x} \mid y = 0)p(y = 0)}{p(\mathbf{x})} \qquad \text{vs.}$$

$$p(y = 1 \mid \mathbf{x}) = \frac{p(\mathbf{x} \mid y = 1)p(y = 1)}{p(\mathbf{x})}$$

How to compute posteriors:

- Ignore evidence $p(\mathbf{x})$ since it is the same for both sides.
- $p(y = 0)$ and $p(y = 1)$ already known (computed from training data).
- $p(\mathbf{x} \mid y = 0) = ?$ $p(\mathbf{x} \mid y = 1) = ?$

NAIVE BAYES

"Naive" Bayes Rule: Compute $p(\mathbf{x} \mid y = 0)$ by assuming independence:

$$p(\mathbf{x} \mid y = 0) = p(x_1 \mid y = 0) \cdot p(x_2 \mid y = 0) \cdot \ldots \cdot p(x_n \mid y = 0)$$

- $p(x_i \mid y = 0)$ is the probability you observe x_i given that an email is not spam.[2]

A more complicated method might take dependencies into account.

[2] Recall, x_i is either 0 when word i is not present, or 1 when word i is present.

Final Naive Bayes Classifier

Training/Modeling: Use existing data to compute:

- $p(y = 0), p(y = 1)$
- For all i:
    - Compute $p(0 \text{ at position } i \mid y = 0), p(1 \text{ at position } i \mid y_0)$
    - Compute $p(0 \text{ at position } i \mid y = 1), p(1 \text{ at position } i \mid y = 1)$

Prediction:

- For all i:
    - Compute $p(\mathbf{x} \mid y = 0) = \prod_i p(x_i \mid y = 0)$
    - Compute $p(\mathbf{x} \mid y = 1) = \prod_i p(x_i \mid y = 1)$
- Return

$$\arg\max \left[p\left(\mathbf{x} \mid y = 0\right) \cdot p\left(y = 0\right), p\left(\mathbf{x} \mid y = 1\right) \cdot p\left(y = 1\right)\right].$$