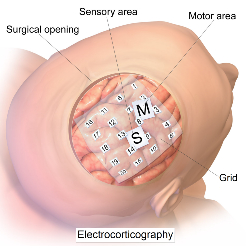
piazza poll about office hours

CS-UY 4563: Lecture 6
Naive Bayes, the Bayesian Perspective

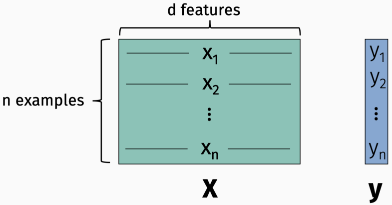
NYU Tandon School of Engineering, Prof. Christopher Musco

Lab 3, due **Next Thursday**.



- Predict hand motion based on electrical measurements of a monkeys brain activity.
- Experience working with sequential (time series) data.
- First lab where computation actually matters (solving regression problems with 40k examples, 1500 features)

If you have enough features, even most basic model will overfit in practice.



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

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 (ℓ_1): $Reg(\boldsymbol{\beta}) = \lambda \|\boldsymbol{\beta}\|_1$
- Elastic net: $Reg(\boldsymbol{\beta}) = \lambda_1 \|\boldsymbol{\beta}\|_1 + \lambda_2 \|\boldsymbol{\beta}\|_2^2$

$\lambda > 0$

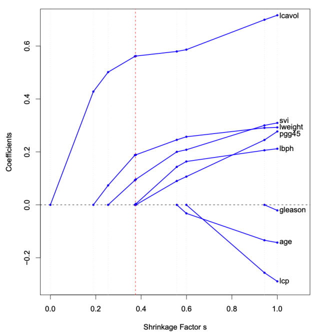
Ridge regression: $\min_{\beta} \|\mathbf{X}\beta - \mathbf{y}\|_2^2 + \lambda\|\beta\|_2^2$.

- Minimized at $\beta = (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{y}$.
- Let $\beta^* = \arg\min_{\beta} L(\beta)$ and $\beta_R^* = \arg\min_{\beta} L(\beta) + Reg(\beta)$.
- Always have $\|\beta_R^*\|_2^2 < \|\beta^*\|_2^2$ and $\|\mathbf{X}\beta_R^* - \mathbf{y}\|_2^2 > \|\mathbf{X}\beta^* - \mathbf{y}\|_2^2$.

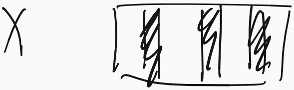
Feature selection methods attempt to set many coordinates in β to 0. Regularization encourages coordinates to be small.

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

- Similarly encourages coordinates in $\boldsymbol{\beta}$ to be small.
- Often the optimal $\boldsymbol{\beta}_R^*$ will have subset of coordinates equal to zero, in contrast to ridge regularization.



$= \sum_{i=1}^{d} |\beta_i|$



Pros:

- Simpler, more interpretable model.

Cons:

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

CLASSIFICATION

- **Data Examples:** $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathbb{R}^d$
- **Target:** $y_1, \ldots, y_n \in \{0, 1, 2, \ldots, q-1\}$ 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.

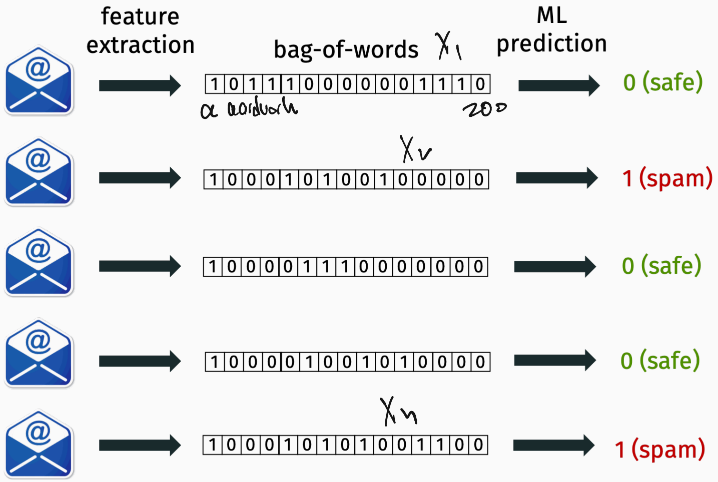
Running example today: Email Spam Classification.

Today: ML from a **Probabilistic/Bayesian Perspective**.

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

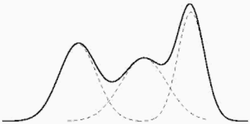
We won't see that today! We're going to use classification as a window into another way of thinking about machine learning.

SPAM PREDICTION



Both target labels and data vectors are binary.

First Goal: Model data $(\mathbf{x}, y)$ – in our case emails – as a simple probabilistic process. **Probabilistic Modeling.**


✓


X


X

How would you randomly create a set of email feature vectors and labels (from scratch) that looks like a typical inbox?

Should have some spam emails, and some regular emails.

spam words : (wire transfer, student loan, .. credit card)
$, 1, 2, .. 4)

not spam words : (meeting, question, calendar)

{0 1 0 0 1 1 0 0 0} → {0,1}

Random model for generating data example $(\mathbf{x}, y)$:

- 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 probability $p_i^{(0)}$.
- If $y = 1$, for each i, set $x_i = 1$ with probability $p_i^{(1)}$.

x = [0 1 0 0 1 0 1 _ _]

Each index i corresponds to a different word. For what words would we expect $p_i^{(1)} > p_i^{(0)}$? $p_i^{(0)} > p_i^{(1)}$?

word$_i$ more likely in spam

- **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.

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

BAYES THEOREM/RULE

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

So:

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

Random model for generating data example $(\mathbf{x}, y)$:

- Set $y = 0$ with probability $p(C_0)$, $y = 1$ with probability $p(C_1) = 1 - p(C_0)$.
    - $p(C_0)$ is probability an email is not spam (e.g. 99%).
    - $p(C_1)$ is probability an email is spam (e.g. 1%).
- If $y = 0$, for each i, set $x_i = 1$ with probability $p(x_i = 1 \mid C_0)$.
- If $y = 1$, for each i, set $x_i = 1$ with probability $p(x_i = 1 \mid C_1)$.

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

maximum a posterior probability (MAP) estimate

Bayesian Classification Algorithm:

Compute:

- $p(C_0|\mathbf{x})$: probability $y = 0$ given observed data vector $\mathbf{x}$.
- $p(C_1|\mathbf{x})$: probability $y = 1$ given observed data vector $\mathbf{x}$.

Output: C_0 or C_1 depending on which probability is larger.

$p(C_0|\mathbf{x})$ and $p(C_1|\mathbf{x})$ are called posterior probabilities.

How to compute the posterior? Bayes rule!

$$p(C_0|\mathrm{x}) = \frac{p(\mathrm{x} \mid C_0)p(C_0)}{p(\mathrm{x})} \quad (1)$$

$$\text{posterior} = \frac{\text{likelihood} \times \text{prior}}{\text{evidence}} \quad (2)$$

- **Prior:** Probability in class C_0 prior to seeing any data.
- **Posterior:** Probability in class C_0 after seeing the data.

Goal is to determine which is larger:

$$p(C_0|\mathbf{x}) = \frac{p(\mathbf{x} \mid C_0)p(C_0)}{p(\mathbf{x})} \quad \text{vs.} \quad p(C_1|\mathbf{x}) = \frac{p(\mathbf{x} \mid C_1)p(C_1)}{p(\mathbf{x})}$$

.99 .01

We can ignore evidence $p(\mathbf{x})$ since it is the same for both sides.

Estimate all of the other terms from the labeled data set:

- $p(C_0)$ = fraction of emails in data which are not spam.
- $p(C_1)$ = fraction of emails in data which are spam.
- $p(\mathbf{x} \mid C_0) = ?$

"Naive" Bayes Classifier: Approximate $p(\mathbf{x} \mid C_0)$ by assuming independence:

$$p(\mathbf{x} \mid C_0) = p(x_1 \mid C_0) \cdot p(x_2 \mid C_0) \cdot \ldots \cdot p(x_n \mid C_0)$$

- $p(x_i \mid C_0)$ 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

Using data set compute:

- $p(C_0), p(C_1)$
- For all i:
  - Compute $p(0 \text{ at position } i \mid C_0), p(1 \text{ at position } i \mid C_0)$
  - Compute $p(0 \text{ at position } i \mid C_1), p(1 \text{ at position } i \mid C_1)$

For prediction:

- For all i:
  - Compute $p(\mathbf{x} \mid C_0) = \prod_i p(x_i \mid C_0)$
  - Compute $p(\mathbf{x} \mid C_1) = \prod_i p(x_i \mid C_1)$
- Return

$$\arg\max \left[p\left(\mathbf{x} \mid C_0\right) p\left(C_0\right), p\left(\mathbf{x} \mid C_1\right) p\left(C_1\right)\right].$$

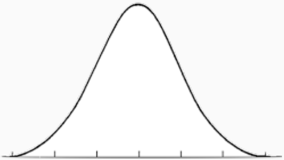
The Bayesian view offers an interesting alternative perspective on many machine learning techniques.

Example: Linear Regression.

Probabilistic model:

$$y_i = \langle \mathbf{x}_i, \boldsymbol{\beta} \rangle + \eta$$

where the $\eta \sim N(0, \sigma^2)$ is **random Gaussian noise**.



$$Pr(\eta = z) \sim$$

The symbol $\sim$ means "is proportional to".

Bayesian Goal: Choose $\boldsymbol{\beta}$ to maximize:

$$\Pr(\boldsymbol{\beta} \mid (\mathbf{X}, \mathbf{y})) = \frac{\Pr((\mathbf{X}, \mathbf{y}) \mid \boldsymbol{\beta}) \Pr(\boldsymbol{\beta})}{\Pr((\mathbf{X}, \mathbf{y}))}.$$

Assume all $\boldsymbol{\beta}$'s are equally likely, so we only care about $\Pr((\mathbf{X}, \mathbf{y}) \mid \boldsymbol{\beta})$ when maximizing.

Choose $\boldsymbol{\beta}$ to maximize:

$$\Pr((\mathbf{X}, \mathbf{y}) \mid \boldsymbol{\beta}) \sim$$

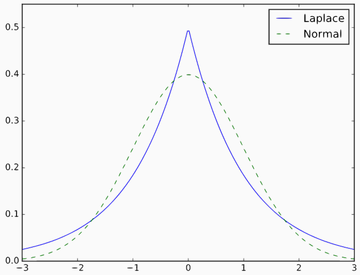
Easier to work with the log likelihood:

$$\begin{aligned}
&\arg\max_{\boldsymbol{\beta}} \prod_{i=1}^{n} e^{-(y_i - \langle \mathbf{x}_i, \boldsymbol{\beta}\rangle)^2/\sigma^2} \\
&= \arg\max_{\boldsymbol{\beta}} \ \log\left(\prod_{i=1}^{n} e^{-(y_i - \langle \mathbf{x}_i, \boldsymbol{\beta}\rangle)^2/\sigma^2}\right) \\
&= \arg\max_{\boldsymbol{\beta}} \sum_{i=1}^{n} -(y_i - \langle \mathbf{x}_i, \boldsymbol{\beta}\rangle)^2/\sigma^2 \\
&= \arg\min_{\boldsymbol{\beta}} \sum_{i=1}^{n} (y_i - \langle \mathbf{x}_i, \boldsymbol{\beta}\rangle)^2.
\end{aligned}$$

Choose $\boldsymbol{\beta}$ to minimize $\sum_{i=1}^{n}(y_i - \langle \mathbf{x}_i, \boldsymbol{\beta}\rangle)^2 = \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2$!

This is a completely different justification for squared loss.

If we had modeled our noise η as Laplace noise, we would have found that minimizing $\|\mathbf{y} - \mathbf{X}\beta\|_1$ was optimal.



$$Pr(\eta = z) \sim$$

Laplace noise has “heavier tails”, meaning that it results in more outliers.

This is a completely different justification for ℓ_1 loss.

~~assume all $\boldsymbol{\beta}$'s equally likely~~

Bayesian view: Assume values in $\boldsymbol{\beta} = [\beta_1, \ldots, \beta_d]$ come from some distribution.

- **Common model:** $\beta_i \sim N(0, \gamma^2)$, i.e. normally distributed, independent.
- Encodes a belief that we are unlikely to see models with very large coefficients.

Recall: want to choose $\boldsymbol{\beta}$ to maximize:

$$\Pr(\boldsymbol{\beta} \mid (\mathbf{X}, \mathbf{y})) = \frac{\Pr((\mathbf{X}, \mathbf{y}) \mid \boldsymbol{\beta}) \Pr(\boldsymbol{\beta})}{\Pr((\mathbf{X}, \mathbf{y}))}.$$

- We can still ignore the "evidence" term $\Pr((\mathbf{X}, \mathbf{y}))$ since it is a constant that does not depend on $\boldsymbol{\beta}$.
- $\Pr(\boldsymbol{\beta}) = \Pr(\beta_1) \cdot \Pr(\beta_2) \cdot \ldots \cdot \Pr(\beta_d)$
- $\Pr(\boldsymbol{\beta}) \sim$

Easier to work with the **log likelihood**:

$$\arg\max_{\boldsymbol{\beta}} \Pr((\mathbf{X}, \mathbf{y}) \mid \boldsymbol{\beta}) \cdot \Pr(\boldsymbol{\beta})$$

$$= \arg\max_{\boldsymbol{\beta}} \prod_{i=1}^{n} e^{-(y_i - \langle \mathbf{x}_i, \boldsymbol{\beta} \rangle)^2/\sigma^2} \cdot \prod_{i=1}^{n} e^{-(\beta_i)^2/\gamma^2}$$

$$= \arg\max_{\boldsymbol{\beta}} \sum_{i=1}^{n} -(y_i - \langle \mathbf{x}_i, \boldsymbol{\beta} \rangle)^2/\sigma^2 + \sum_{i=1}^{d} -(\beta_i)^2/\gamma^2$$

$$= \arg\min_{\boldsymbol{\beta}} \sum_{i=1}^{n} (y_i - \langle \mathbf{x}_i, \boldsymbol{\beta} \rangle)^2 + \frac{\sigma^2}{\gamma^2} \sum_{i=1}^{d} (\beta_i)^2/\sigma^2.$$

Choose $\boldsymbol{\beta}$ to minimize $\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \frac{\sigma^2}{\gamma^2}\|\boldsymbol{\beta}\|_2^2$!

This is a completely different justification for ridge regularization.

Test your intuition: What modeling assumption justifies LASSO regularization: $\min \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda\|\boldsymbol{\beta}\|_1$.