CS-UY 4563: Lecture 7
The Bayesian Perspective cont., Linear Classifiers

NYU Tandon School of Engineering, Prof. Christopher Musco

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

Exercise: With a partner, 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 $x_i \in \{\mathit{Fresh.}, \mathit{Soph.}, \mathit{Jun.}, \mathit{Sen.}\}$ indicating class year. Each $y_1 \in \{0, 1\}$ with zero indicating the student has not taken machine learning, one indicating they have.

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

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:

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.

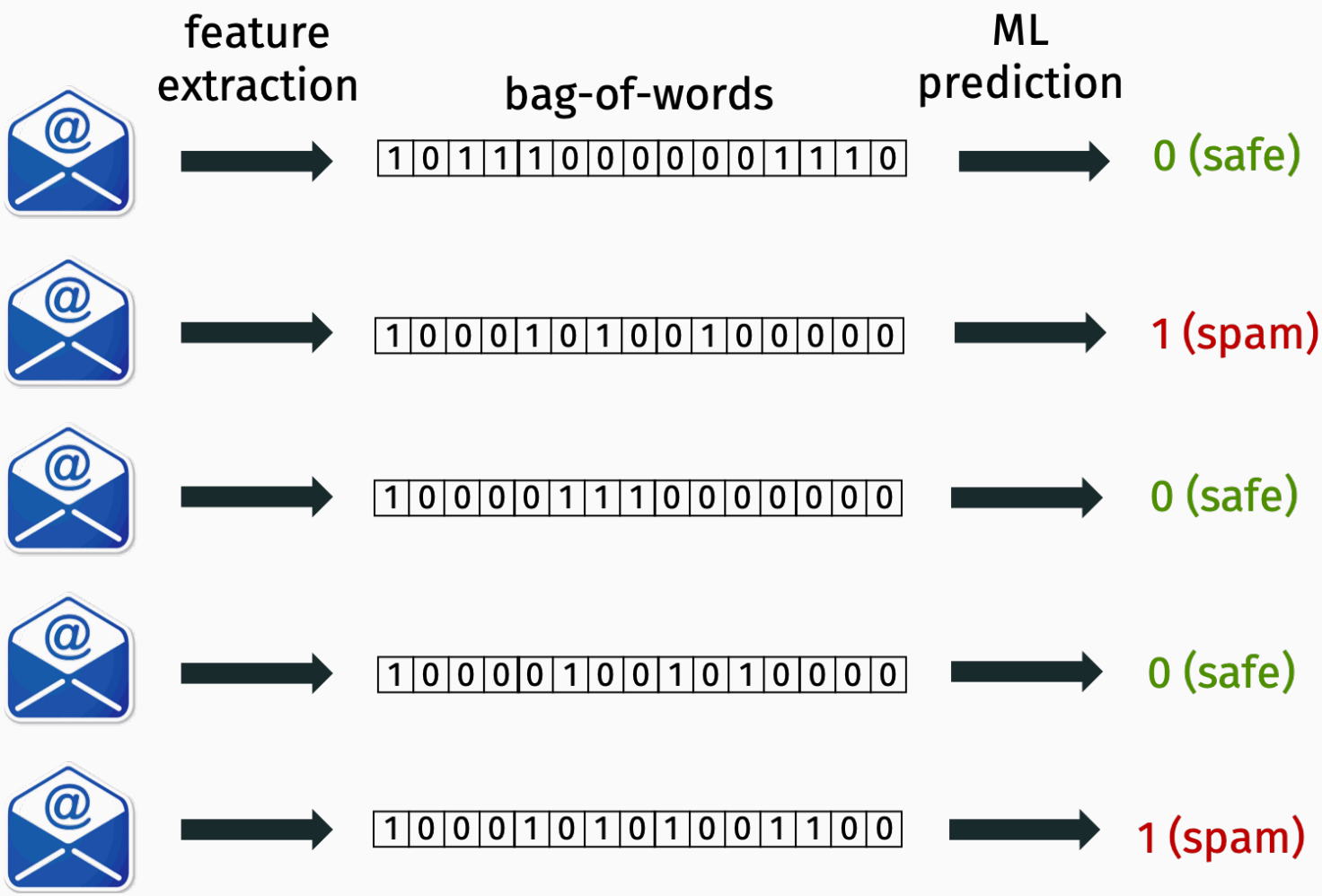
Model:

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



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

Unknown model parameters:

- $p(y = 0), p(y = 1)$,
- $p(x_1 = 1 \mid y = 0), \ldots, p(x_n = 1 \mid y = 0)$.
- $p(x_1 = 1 \mid y = 1), \ldots, p(x_n = 1 \mid y = 1)$.

How would you estimate these parameters?

Reasonable way to set parameters:

- Set $p(y = 0)$ and $p(y = 1)$ to the empirical fraction of not spam/spam emails.
- For each word i, set $p(x_i = 1 \mid y = 0)$ to the empirical probability word i appears in a non-spam email.
- For each word i, set $p(x_i = 1 \mid y = 1)$ to the empirical probability word i appears in a spam email.

Estimating these parameters is the only "training" we will do.

DONE WITH MODELING

ON TO PREDICTION

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!**

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

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

- **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 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.[1]

A more complicated method might take dependencies into account.

[1] 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

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

OTHER APPLICATIONS OF THE BAYESIAN PERSPECTIVE

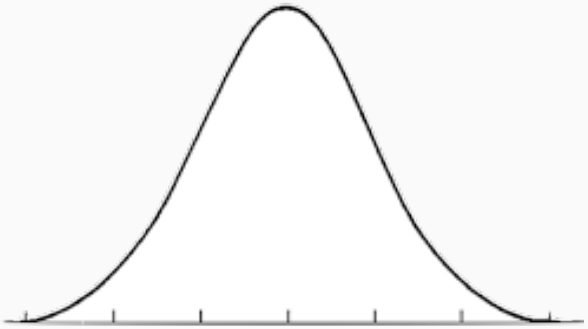
The Bayesian view offers an interesting alternative perspective on many machine learning techniques.

Example: Linear Regression.

Probabilistic model:

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

where the η drawn from $N(0, \sigma^2)$ is **random Gaussian noise**.



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

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

Example: Linear Regression.

Probabilistic model:

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

where the η drawn from $N(0, \sigma^2)$ is **random Gaussian noise**.

If we knew $\boldsymbol{\beta}$ what is the maximum a posterior (MAP) estimate for y_i given observed data $\mathbf{x}_i$?

How should be select $\boldsymbol{\beta}$ for our model?

Bayesian approach: Use MAP estimate again! Now for parameter vector.

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:

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

BAYESIAN REGRESSION

- $y_i = \langle \mathbf{x}_i, \boldsymbol{\beta} \rangle + \eta$
- where $p(\eta = z) \sim e^{-z^2/\sigma^2}$

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

LOG LIKELIHOOD

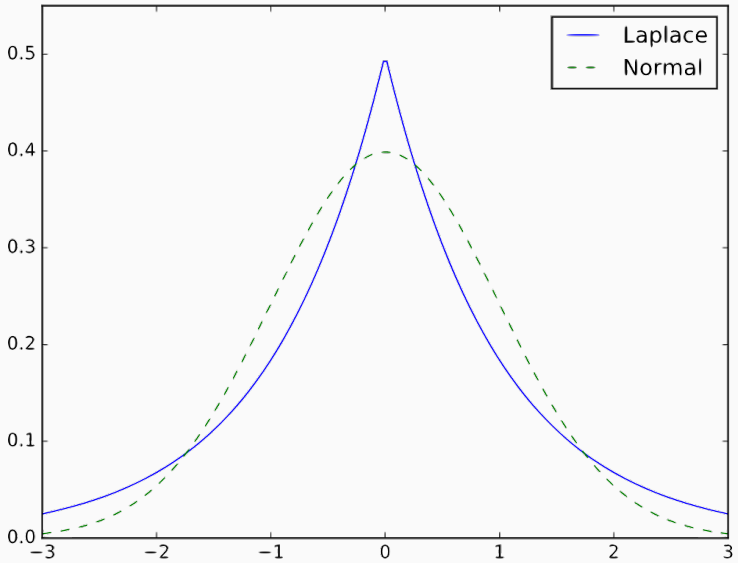
Easier to work with the log likelihood:

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

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 $\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_1$ 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.

Recall goal is to maximize over $\boldsymbol{\beta}$:

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

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

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

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

Goal: choose $\boldsymbol{\beta}$ to maximize:

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

- 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**:

$$\begin{aligned}
&\arg\max_{\boldsymbol{\beta}} \Pr(\mathbf{X}, \mathbf{y} \mid \boldsymbol{\beta}) \cdot \Pr(\boldsymbol{\beta}) \\
&= \arg\max_{\boldsymbol{\beta}} \prod_{i=1}^{n} e^{-(y_i - \langle \mathbf{x}_i, \boldsymbol{\beta} \rangle)^2 / \sigma^2} \cdot \prod_{i=1}^{n} e^{-(\beta_i)^2 / \gamma^2} \\
&= \arg\max_{\boldsymbol{\beta}} \sum_{i=1}^{n} -(y_i - \langle \mathbf{x}_i, \boldsymbol{\beta} \rangle)^2 / \sigma^2 + \sum_{i=1}^{d} -(\beta_i)^2 / \gamma^2 \\
&= \arg\min_{\boldsymbol{\beta}} \sum_{i=1}^{n} (y_i - \langle \mathbf{x}_i, \boldsymbol{\beta} \rangle)^2 + \frac{\sigma^2}{\gamma^2} \sum_{i=1}^{d} (\beta_i)^2 / \sigma^2.
\end{aligned}$$

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

Completely different justification for ridge regularization!

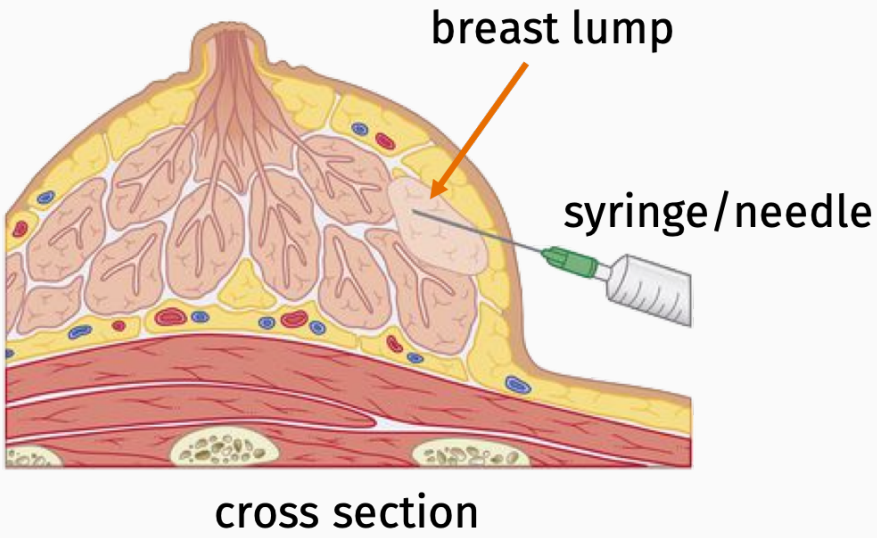
Test your intuition: What modeling assumption justifies LASSO regularization: $\min \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda\|\boldsymbol{\beta}\|_1$?

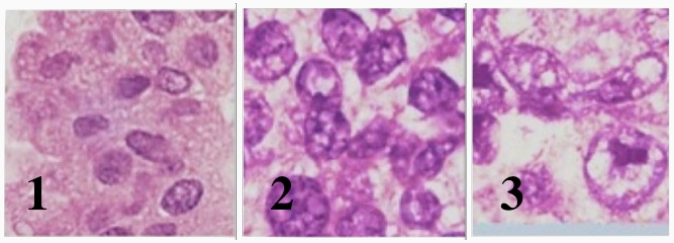
LINEAR CLASSIFICATION

MOTIVATING PROBLEM

Breast Cancer Biopsy: Determine if a breast lump in a patient is malignant (cancerous) or benign (safe).

- Collect cells from lump using fine needle biopsy.
- Stain and examine cells under microscope.
- Based on certain characteristics (shape, size, cohesion) determine if likely malignant or not).





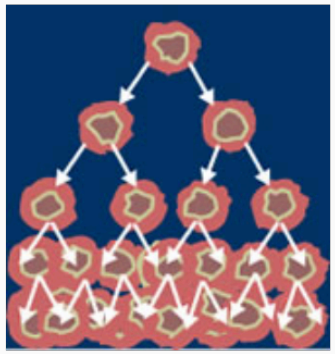
Demo: `demo_breast_cancer.ipynb`

Data: UCI machine learning repository

Breast Cancer Wisconsin (Original) Data Set

Download: Data Folder, Data Set Description

Abstract: Original Wisconsin Breast Cancer Database



Data Set Characteristics:	Multivariate	**Number of Instances:**	699	**Area:**	Life
Attribute Characteristics:	Integer	**Number of Attributes:**	10	**Date Donated**	1992-07-15
Associated Tasks:	Classification	**Missing Values?**	Yes	**Number of Web Hits:**	564320

`https://archive.ics.uci.edu/ml/datasets/breast+cancer+wisconsin+(original)`

Features: 10 numerical scores about cell characteristics (Clump Thickness, Uniformity, Marginal Adhesion, etc.)

Data: $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)$.

$\mathbf{x}_i = [1, 5, 4 \ldots, 2]$ contains score values.

Label $y_i \in \{0, 1\}$ is 0 if benign cells, 1 if malignant cells.

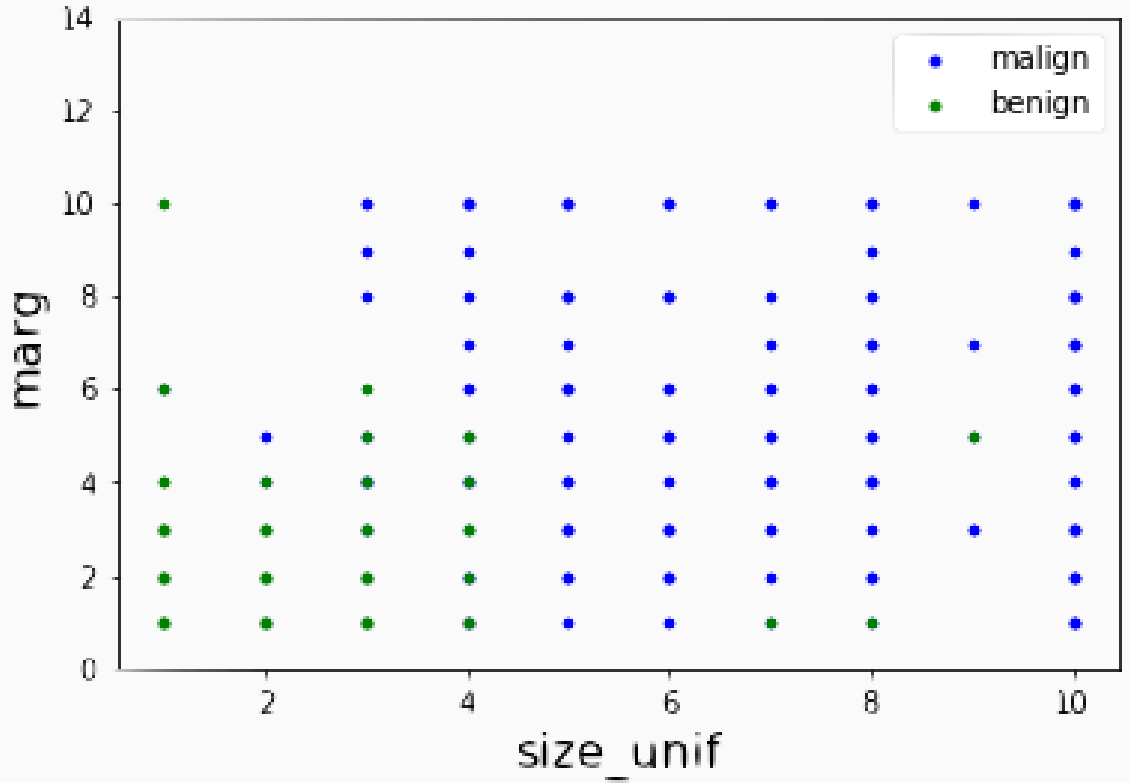
Goal: Based on scores (which would be collected manually, or even learned on their own using an ML algorithm) predict if a sample of cells is malignant or benign.

Approach:

- Naive Bayes Classifier can be extended to $\mathbf{x}$ with numerical values (instead of binary values as seen before). Will see on homework.
- **Today:** Learn a different type of classifier.

BEGIN BY PLOTTING DATA

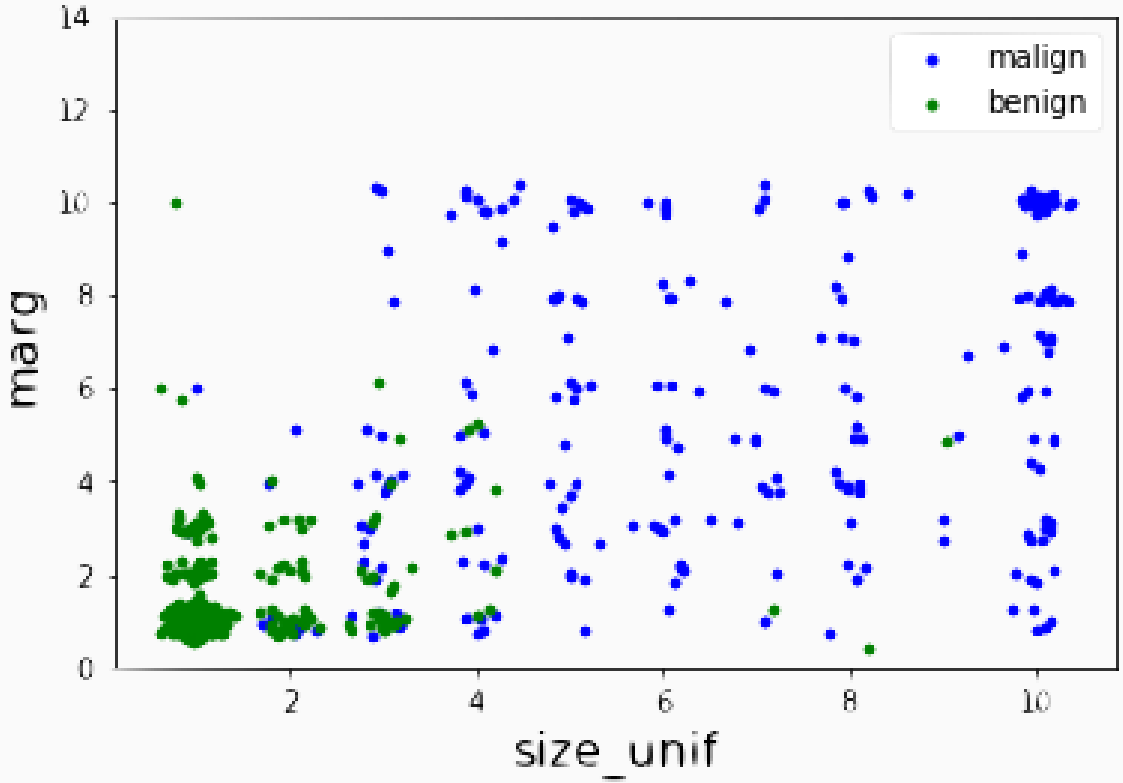
We pick two variables, Margin Adhesion and Size Uniformity and plot a scatter plot. Points with label 1 (malignant) are plotted in blue, those with label 2 (benign) are plotted in green.



Lots of overlapping points! Hard to get a sense of the data.

PLOTTING WITH JITTER

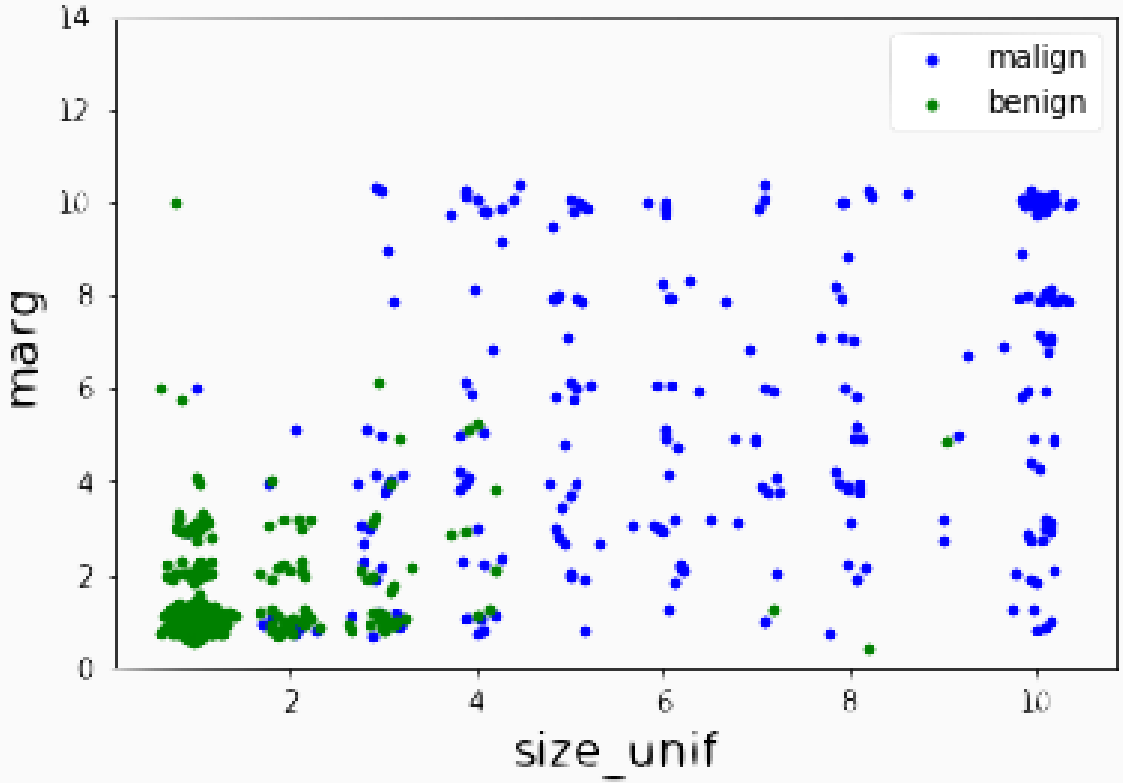
Simple + Useful Trick: data jittering. Add tiny random noise (using e.g. `np.random.randn`) to data to prevent overlap.



Noise is only for plotting. It is not added to the data for training, testing, etc.

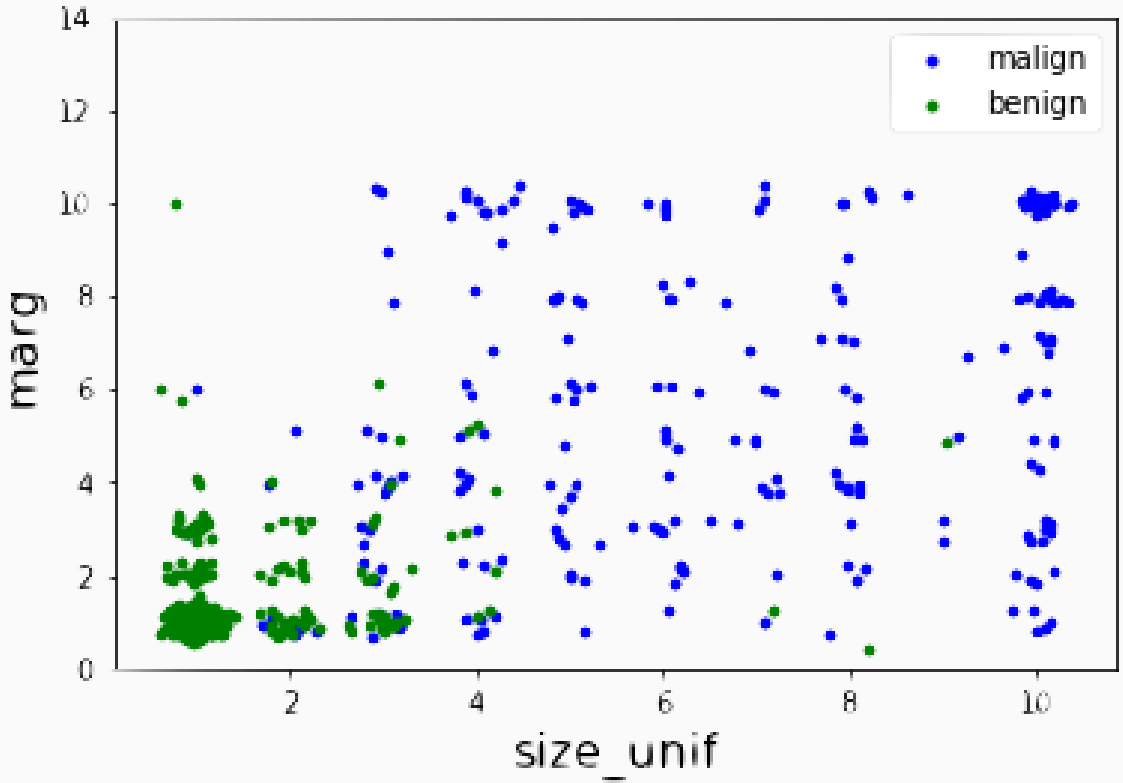
PLOTTING WITH JITTER

Simple + Useful Trick: data jittering. Add tiny random noise (using e.g. `np.random.randn`) to data to prevent overlap.



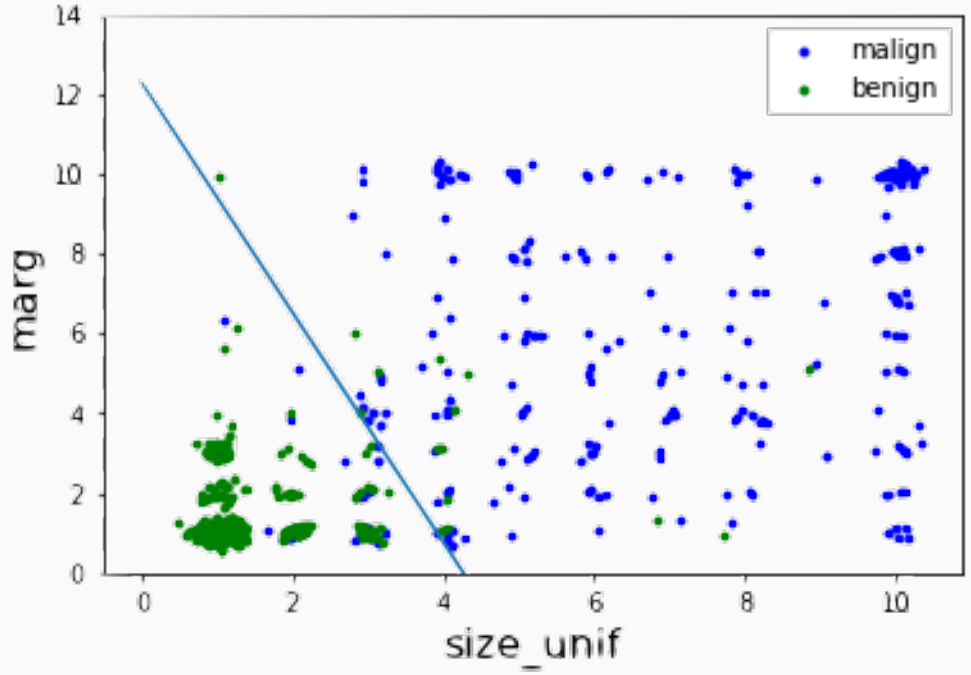
Noise is only for plotting. It is not added to the data for training, testing, etc.

Any ideas for possible classification rules for this data?



Given vector of predictors $\mathbf{x}_i \in \mathbb{R}^d$ (here $d = 2$) find a parameter vector $\boldsymbol{\beta} \in \mathbb{R}^d$ and threshold λ.

- Predict $y_i = 0$ if $\langle \mathbf{x}_i, \boldsymbol{\beta} \rangle \leq \lambda$.
- Predict $y_i = 1$ if $\langle \mathbf{x}_i, \boldsymbol{\beta} \rangle > \lambda$



Line has equation $\langle \mathbf{x}_i, \boldsymbol{\beta} \rangle = \lambda$.

Next class: How do we find a good linear separator?