02-620 Week 3

Machine Learning for Scientists

Aidan Jan

January 30, 2026

Classification

The goal of classification is to find a functional mapping $f : X \to Y$, where Y is discrete-valued.

- SNPs for X and disease/healthy status for Y
- Gene expression for X and disease/healthy status for Y
- Pathology images for X and tumor/healthy for Y
- Electronic medical records for X and diagnosis for Y
- Genome sequence features for X and transcription factor binding site or not for Y

Training vs. Testing

- In training, the goal is to improve the model using input data and output pairs.
- In testing, the goal is to classify unseen new input data and provide the output.

Different Types of Classifiers

- K-nearest neighbor
  - Non-parametric method: no model, no parameters, no learning (lazy)
- Naive Bayes
  - Parametric method, generative model: model $P(Y, X|\theta)$ to obtain $P(Y|X, \theta)$
- Logistic Regression
  - Parametric method, discriminative model: model $P(Y|X, \theta)$

K-nearest neighbors (KNN) classifier

- Given N training data points $(x_1, y_1), \cdots, (x_N, y_N)$, kNN performs no explicit learning (i.e., no learnable parameters)
- **Inference:** A new data point x_i, is classified by majority vote among its k-nearest neighbors, defined as the k training points with the smallest Euclidean ($l2$) distances $\|x_{i'} - x_i\|_2^2$

How to select k

- Small k: classification is sensitive to noise
- Large k: too much smoothing. (If $k = N$, sample size, all test inputs will receive the same classification.)
- Select k that is not too small and not too large

Computation Time

- **Learning:** No training or parameter learning - cheap!
- **Inference:** When a new data point $x_{i'}$ arrives, kNN must compute the distance between $x_{i'}$ and all N training samples, incurring an $\mathrm{O}(ND)$ computational cost - expensive!

Naive Bayes Classifier

Example: Predicting Cancer from genotype

Individual	Locus 1 X_1	Locus 2 X_2	Locus 3 X_3	Healthy/Cancer Y
1	0	0	1	1
2	1	0	2	1
3	0	2	0	1
4	2	0	0	0
5	2	1	2	0
6	1	2	1	0

Here, the input X represents the allele. 0 = AA (minor allele homozygous), 1 = AT (heterozygous), 2 = TT (major allele homozygous). Y represents healthy (0) or cancer (1). We want to

- learn a classifier, $f : (X_1, X_2, X_3) \to Y$
- learn a probabilistic model for $P(Y|X)$, where Y is discrete

$P(Y|X)$ is given as

Combination	X_1	X_2	X_3	$P(Y=1\|X_1,X_2,X_3)$	$P(Y=0\|X_1,X_2,X_3)$
1	0	0	0	0.01	0.99
2	0	0	1	0.50	0.50
3	0	0	2	0.30	0.70
4	0	1	0	0.25	0.75
5	0	1	1	0.70	0.30
6	0	1	2	0.05	0.95
7	...	...	...	...	...

- How many probability parameters must be specified?
- How can this distribution be learned from data?
- Note that $P(Y=0|X_1,X_2,X_3) = 1 - P(Y=1|X_1,X_2,X_3)$

How many parameters are needed?

- Suppose $X = [X_1, \dots, X_D]$ for D SNPs
  - X_j's: random variables taking values from $\{0, 1, 2\}$
  - Y: binary random variables
- To estimate $P(Y|X_1, X_2, \dots, X_D)$, 3^n quantities need to be estimated!
- If we have 30 SNPs in X: $P(Y|X_1, X_2, \dots, X_{30})$, then we have $3^30 \sim 2 \times 10^{14}$. Too many!
- We need a more compact representation of $P(Y|X_1, X_2, \dots, X_D)$

General Bayesian Inference

- Suppose $X = [X_1, \ldots, X_D]$ for genotypes at D loci and binary health outcome Y. Using Bayes rule,

$$P(Y|X) = \frac{P(X|Y)P(Y)}{P(X)}$$

- How many parameters for $P(X|Y) = P(X_1, \ldots, X_D|Y)$?

$$P(X_1, \cdots, X_D|Y=1) = 3^D - 1$$
$$P(X_1, \cdots, X_D|Y=0) = 3^D - 1$$
$$\therefore P(X_1, \cdots, X_D|Y) = 2(3^D - 1)$$

- How many parameters for $P(Y)$? One.

Reducing Parameters via Conditional Independence

- Suppose $X = [X_1, \ldots, X_D]$ for genotypes at D loci and binary health outcome Y. Using Bayes rule,

$$P(Y|X) = \frac{P(X|Y)P(Y)}{P(X)}$$

- Naive Bayes assumes conditional independence

$$P(X_1, \cdots, X_D|Y) = \prod_{j=1}^{D} P(X_j|Y)$$

  i.e., X_j and $X_{j'}$ conditionally independent given Y, for all $j \neq j'$
  Now, we have $2 \cdot 2D$ parameters.

Naive Bayes Model for Previous Example

Model: Specify $P(Y|X;\theta)$ for discrete output Y.

$$P(Y|X_1, \cdots, X_D) = \frac{P(Y)\prod_{j=1}^{D} P(X_j|Y)}{P(X_1, \cdots, X_D)}$$

- Bernoulli distribution: $P(Y) = \pi^Y(1-\pi)^{(1-Y)}$
- Multinoulli distribution: $P(X_j|Y=k) = \prod_{l=0}^{2} \theta_{jkl}^{I(X_j=l)}$

  where indicator function $I(A) = \begin{cases} 1 & \text{if } A \text{ is true} \\ 0 & \text{otherwise} \end{cases}$

The numerator of our model is:

$P(X_j = 0|Y=k) = \theta_{jk0}^{I(X_j=0)}\theta_{jk1}^{I(X_j=1)}\theta_{jk2}^{I(X_j=2)} = \theta_{jk0}P(X_j = 1|Y=k) \quad = \theta_{jk0}^{I(X_j=0)}\theta_{jk1}^{I(X_j=1)}\theta_{jk2}^{I(X_j=2)} = \theta_{jk1}P(X_j = 2|Y=k)$

The denominator, expanded, is

$$P(X_1, \cdots, X_D) = P(Y=0)\prod_{j=1}^{D} P(X_j|Y) + P(Y=1)\prod_{j=1}^{D} P(X_j|Y)$$

in other words, evaluate the numerator for $Y=1$ and $Y=0$ and sum the results.

In this example, our learnable parameters are

- $\pi = P(Y=1)$
- $\theta_{jkl} = P(X_j = l|Y=k)$, for $j = 1, \ldots, D$, $I = 0, 1, 2$ and $k = 0, 1$

Naive Bayes Model Summary

Without conditional independence (general case)

- Total number of parameters: $2(3^D - 1) + 1$
- 1 parameter for $P(Y)$
- $2(3^D - 1)$ parameters for $P(X_1, \cdots, X_D|Y)$

With conditional independence (Naive Bayes)

- Total number of parameters: $4D + 1$
- 1 parameter for $P(Y)$
- $4D$ parameters for $\prod_{j=1}^{D} P(X_j|Y)$

Naive Bayes Inference

Given the classifier

$$P(Y|X_1, \cdots, X_D) = \frac{P(Y)\prod_{j=1}^{D} P(X_j|Y)}{P(X_1, \cdots, X_D)}$$

Classify a new data point $X_{i'} \in \mathbb{R}^D$

- Step 1: Compute $P(Y = 1|X_1, \cdots, X_D) = P(Y = 1)\prod_{j=1}^{D} P(X_j|Y = 1) = \pi \prod_{j=1}^{D} \theta_{j1X_{i'j}}$
- Step 2: Compute $P(Y = 0|X_1, \cdots, X_D) = P(Y = 0)\prod_{j=1}^{D} P(X_j|Y = 0) = (1 - \pi)\prod_{j=1}^{D} \theta_{j0X_{i'j}}$
- Step 3: Predict the value of Y with the larger posterior probability (value from steps 1-2).

Naive Bayes Learning

Given data

$$\mathcal{D} \; : \; x_1, \cdots, x_N \in \mathbb{R}^D, y_1, \cdots, y_N \in \{0, 1\}$$

Estimate parameters from data

- $\pi \; : \; P(Y = 1)$
- $\theta_{jkl} \; : \; P(X_j = l|Y = k)$ for $j = 1, \ldots, D, l = 0, 1, 2,$ and $k = 0, 1$.

Taxonomy of Learning Methods

Learning: Estimating Parameters θ

- $\mathcal{D}$: data / evidence (e.g., (X, y))

Maximum Likelihood Estimation (MLE) Probabilistic instantiation of ERM with deterministic θ

$$\hat{\theta} = \arg\max_{\theta} P(\mathcal{D}; \theta)$$

Maximum a Posteriori (MAP) Θ is random.

$$\begin{aligned}
\hat{\theta} &= \arg\max_{\theta} P(\Theta|\mathcal{D}) \\
&= \arg\max_{\theta} \left[\log P(\Theta = \theta) + \sum_{i=1}^{N} \log P(Y_i = y_i, X_i = x_i|\Theta = \theta) \right]
\end{aligned}$$

Naive Bayes Classifier: Learning via MLE

$$
\begin{aligned}
\arg\max_{\pi,\theta} P(\mathcal{D};\theta) &= \arg\max_{\pi,\theta} \log \prod_{i=1}^{N} P(Y_i, X_i) \\
&= \arg\max_{\pi,\theta} \log \prod_{i=1}^{N} P(Y_i)P(X_i|Y_i) \\
&= \arg\max_{\pi,\theta} \log \prod_{i=1}^{N} P(Y_i)P(X_{i1},\cdots,X_{iD}|Y_i) \\
&= \arg\max_{\pi,\theta} \sum_{i=1}^{N} \log \left[P(Y_i) \prod_{j=1}^{D} P(X_{ij}|Y_i) \right] \\
&= \arg\max_{\pi,\theta} \sum_{i=1}^{N} \log P(Y_i) + \sum_{i=1}^{N} \log P(X_{i1}|Y_i) + \cdots + \sum_{i=1}^{N} \log P(X_{iD}|Y_i)
\end{aligned}
$$

Notice that the first summation in the series is only related to π.

$$
\begin{aligned}
\therefore \hat{\pi} &= \arg\max_{\pi} \sum_{i=1}^{N} \log P(Y_i) \\
&= \arg\max_{\pi} \sum_{i=1}^{N} \log(\pi^{Y_i}(1-\pi)^{1-Y_i})
\end{aligned}
$$

Additionally, notice that each summation term after the first is only related to one θ. $P(X_{ij}|Y_i)$ is only related to θ_j.

$$
\begin{aligned}
\therefore \hat{\theta}_j &= \arg\max_{\theta_j} \sum_{i=1}^{N} \log P(X_{ij}|Y_i) \\
&= \arg\max_{\theta_j} \sum_{i=1}^{N} \log \prod_{l=0}^{2} \theta_{jY_il}^{I(X_{ij}=l)}
\end{aligned}
$$

where $\theta_j = \{\theta_{jkl}\}_{k=0,1,l=0,1,2}$

Naive Bayes Classifier: Learning

$$\hat{\pi} = \arg\max_{\pi} \sum_{i=1}^{N} \log P(Y_i) = \arg\max_{\pi} \sum_{i=1}^{N} \log(\pi^{Y_i}(1-\pi)^{1-Y_i})$$

- $\hat{\pi}$ is the MLE of the Bernoulli mean $\pi = P(Y=1)$
- Thus, $\hat{\pi} = \frac{1}{N}\sum_{i=1}^{N} Y_i$

$$\hat{\theta}_j = \arg\max_{\theta_j} \sum_{i=1}^{N} \log P(X_{ij}|Y_i) = \arg\max_{\theta_j} \sum_{i=1}^{N} \log \prod_{l=0}^{2} \theta_{jY_il}^{I(X_{ij}=l)}$$

- $\hat{\theta}_{jkl}$ is the MLE of the multinoulli parameter $\theta_{jkl} = P(X_j = l|Y=k)$
- $\hat{\theta}_{jkl} = \frac{\sum_{i=1}^{N} I(Y_i=k)I(X_{ij}=l)}{\sum_{i=1}^{N} I(Y_i=k)}$

Naive Bayes Subtlety 1

- If unlucky, our MLE estimate for $P(X_j|Y)$ might be zero.
  - e.g., no ddata points has $X_1 = 1$ and $Y = 0$, then $P(X_1 = 1|Y = 0) = 0$
- Why worry about just one parameter out of many?

$$P(Y|X_1, \cdots, X_D) = \frac{P(Y)\prod_{j=1}^{D} P(X_j|Y)}{P(X_1, \cdots, X_D)}$$

- If one of the terms are zero, the entire probability is zero since the terms are multiplied together.
- What can be done to avoid this?

Remember that the maximum likelihood estimates are:

- $\hat{\pi} = P(Y = 1) = \frac{1}{N}\sum_{i=1}^{N} Y_i$
- $\hat{\theta}_{jkl} = P(X_j = l|Y = k) = \frac{\sum_{i=1}^{N} I(Y_i=k)I(X_{ij}=l)}{\sum_{i=1}^{N} I(Y_i=k)}$

MAP estimates (Beta, Dirichlet priors):

- $\hat{\pi} = P(Y = 1) = \frac{\alpha_0 + \sum_{i=1}^{N} Y_i}{\alpha_0 + \beta_0 + N}$
- $\hat{\theta}_{jkl} = P(X_j = l|Y = k) = \frac{\alpha_{jkl0} + \sum_{i=1}^{N} I(Y_i=k)I(X_{ij}=l)}{\alpha_{jkl0} + \beta_{jkl0} + \sum_{i=1}^{N} I(Y_i=k)}$
- The only difference here is "imaginary" examples.

Naive Bayes Subtlety 2

- Often X_j's are not actually conditionally independent given Y
- We use Naive Bayes in many cases anyways, and it often works pretty well.
  - Often the right classification, even when not the right probability (see [Domingos and Pazzani, 1996])
- What is the effect on estimated $P(Y|X)$?
  - Special case: what if we add two copies: $X_j = X_{j'}$

What if we have continuous X_j?

For example, image classification: X_j is real-valued j-th pixel.

Given input images X:

- Classify whether this is from a normal or schizophrenic brain
- Classify which tasks he/she is performing?
- Classify which word he/she is reading?

Naive Bayes requires $P(X_j|Y = k)$, but X_j is real (continuous).

$$P(Y|X_1, \cdots, X_D) = \frac{P(Y)\prod_{j=1}^{D} P(X_j|Y)}{P(X_1, \cdots, X_D)}$$

Common approach: assume that $P(X_j|Y_k)$ follows a continuous distribution (e.g., Normal).

Questions for thought

- Can you use Naive Bayes for a combination of discrete and real-valued X_j
- How can we easily model just 2 of D features as dependent?

$$P(X_j, X_{j'}|Y)$$

- How many parameters must we estimate for Gaussian Naive Bayes if Y has K possible values, $X = [X_1, \ldots, X_D]$?
  - $P(Y)$: $K-1$ parameters
  - $P(X_j|Y=k)$: 2 parameters, $2KD$ in total for $k = 1, \ldots, K$ and $j = 1, \ldots, D$

Naive Bayes Classifier Summary

- Model: $P(Y|X_1, \cdots, X_D) = \frac{P(Y)\prod_{j=1}^{D} P(X_j|Y)}{P(X_1, \cdots, X_D)}$
- Learning
  - $\hat{\pi} = P(Y=1) = \frac{1}{N}\sum_{i=1}^{N} Y_i$
  - $\hat{\theta}_{jkl} = P(X_j = l|Y=k) = \frac{\sum_{i=1}^{N} I(Y_i=k)I(X_{ij}=l)}{\sum_{i=1}^{N} I(Y_i=k)}$
- Inference:
  - Step 1: Compute $P(Y=1)\prod_{j=1}^{D} P(X_j|Y=1) = \pi \prod_{j=1}^{D} \theta_{j1X_{i'j}}$
  - Step 2: Compute $P(Y=0)\prod_{j=1}^{D} P(X_j|Y=1) = (1-\pi)\prod_{j=1}^{D} \theta_{j0X_{i'j}}$
  - Step 3: Predict the value of Y with the larger value from steps 1-2.