Fundamentals of Machine Learning [DSE 2222]

Department of Data Science and Computer Applications

MIT, Manipal

January 2025

Slide Set 3 – Naïve Bayes Classifier

First Approach

- Consider a classification problem in which we want to learn to distinguish between elephants (y = 1) and dogs (y = 0), based on some features of an animal.
  - Given a training set, an algorithm like logistic regression or the perceptron algorithm (basically) tries to find a straight line- that is, a decision boundary- that separates the elephants and dogs.

  - Then, to classify a new animal as either an elephant or a dog, it checks on which side of the decision boundary it falls, and makes its prediction accordingly.

Present approach

- First, looking at elephants, we can build a model of what elephants look like. Then, looking at dogs, we can build a separate model of what dogs look like.

- Finally, to classify a new animal, we can match the new animal against the elephant model, and match it against the dog model, to see whether the new animal looks more like the elephants or more like the dogs we had seen in the training set.

Discriminative and Generative Learning Algorithms

- Algorithms that try to learn p(y|x) directly (such as logistic regression), or algorithms that try to learn mappings directly from the space of inputs X to the labels {0; 1}, (such as the perceptron algorithm) are called discriminative learning algorithms

- Algorithms that try to model p(x|y) (and p(y)) are called generative learning algorithms.
  - For instance, if y indicates whether an example is a dog (0) or an elephant (1), then p(x|y = 0) models the distribution of dogs' features, and p(x|y = 1) models the distribution of elephants' features.

- After modeling p(y) (called the class priors) and p(x|y), algorithm can then use Bayes rule to derive the posterior distribution on y given x:

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

- **Class priors** refer to the prior probabilities of each class in a classification problem. They represent the probability of a class occurring before any evidence (features) is observed.

The class prior P(C) for a class C is defined as:

$$P(C) = \frac{\text{Number of samples in class } C}{\text{Total number of samples}}$$

Prior probability

Consider a dataset where:

- 100 samples belong to **Class A**.
- 50 samples belong to **Class B**.

$$P(\text{Class A}) = \frac{100}{150} = 0.67$$

$$P(\text{Class B}) = \frac{50}{150} = 0.33$$

These priors reflect the imbalance between the two classes.

Posterior Probabilities

- **Posterior probability** is the probability of an event or hypothesis A occurring after observing some evidence B.
- In Bayesian statistics, it is calculated by combining prior knowledge and observed data.

The **posterior probability** is given by Bayes' Theorem:

$$P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)}$$

Conditional Probabilities

- **Conditional probability** is the probability of an event occurring, given that another event has already occurred.
- It measures the likelihood of one event under the assumption that we know another event has occurred.
- **The conditional probability of event A, given that event B has occurred**

$$P(A|B) = \frac{P(A \cap B)}{P(B)}$$

Where:

•**P(A|B)**: The probability of A occurring, given B has occurred.
•**P(A∩B)**: The probability of both A and B occurring (**joint probability**).
•**P(B)**: The probability of B occurring (prior probability, must be >0).

Example:

A: "It rains."

B: "It is cloudy.“

Assume:

- $P(A)=0.3$ (Rain on any given day is 30% likely).
- $P(B)=0.5$ (Cloudiness is 50% likely).
- $P(A \cap B)=0.2$ (Rain and cloudiness together occur 20% of the time).

The conditional probability of rain, given that it is cloudy, is:

$$P(A|B) = \frac{P(A \cap B)}{P(B)} = \frac{0.2}{0.5} = 0.4$$

There's a 40% chance of rain when it’s cloudy

Conditional Probability - Events and Tests

Test:

There is a **test** for liver disease

Event:

Actually having liver disease or not.

Conditional probability is the probability of an event occurring, given that another event has already occurred.

It measures the likelihood of one event under the assumption that we know another event has occurred.

Independence vs. Conditional Independence

Independence: Two events are independent if knowing one does not change the probability of the other.

- **Example**:
  - Tossing two different coins: The result of one coin toss does not affect the other.

Conditional Independence: Two events are independent given a third event.

- **Example**:
  - Suppose it rains , and you bring an umbrella and wear boots .
  - Given that it rains, carrying the umbrella does not depend on wearing boots.

Naive Bayes Assumption in Spam Detection

- Simplifies spam classification by assuming **conditional independence of words given the class (spam or not spam).**

Why This Matters in Spam Detection

- **Without Naive Bayes Assumption:**
- Complex calculations: Consider all combinations of words.
  - Example: "buy AND price" vs. "buy OR price."
- **With Naive Bayes Assumption:**
  - Simplifies calculations by treating each word separately:

Naive Bayes classifiers

- The **Naive Bayes classification algorithm** is a probabilistic machine learning model based on **Bayes' Theorem**.
- It is widely used for classification tasks, particularly text classification (e.g., spam filtering, sentiment analysis) and medical diagnosis.
- This **generative approach** to classification in which we assume the features are conditionally independent given the class label.
- The "naïve" part of Naïve Bayes assumes that all features $F_1,F_2,\ldots,F_n$ are conditionally independent given the class label C. This means:

$$P(X|C) = P(F_1|C) \cdot P(F_2|C) \cdot \cdots \cdot P(F_n|C)$$

- This is called the *Naive Bayes assumption*.

Bayes' Theorem

- Naïve Bayes algorithm is built on Bayes' Theorem:

- It is a formula for computing the probability distribution over possible values of an unknown (or hidden) quantity H given some observed data Y = y:

$$p(H = h|Y = y) = \frac{p(H = h)p(Y = y|H = h)}{p(Y = y)}$$

- The term p(H) represents what we know about possible values of H before we see any data; this is called the **prior distribution**.
  - **If H has K possible values, then p(H) is a vector of K probabilities, that sum to 1.**

.

Bayes' Theorem

$$p(H = h|Y = y) = \frac{p(H = h)p(Y = y|H = h)}{p(Y = y)}$$

- The term p(Y |H = h) represents the distribution over the possible outcomes Y we expect to see if H = h; this is called the **observation distribution**.
  - When we evaluate this at a point corresponding to the actual observations, y, we get the function p(Y = y|H = h), which is called the **likelihood**
- Multiplying the prior distribution p(H = h) by the likelihood function p(Y = y|H = h) for each h gives the unnormalized joint distribution p(H = h, Y = y).

  (Unnormalized joint distribution: it captures the relationship between H and Y, but its probabilities might not sum to 1 across all hypotheses.)
- We can convert this into a normalized distribution by dividing by p(Y = y), which is known as the **marginal likelihood** (ensures the probabilities across all hypotheses sum to 1)
- Normalizing the joint distribution by computing p(H = h, Y = y)/p(Y = y) for each h gives the ***posterior distribution p(H = h|Y = y);*** this represents our new belief state about the possible values of H.
- We can summarize Bayes rule in words as follows:

posterior ∝ prior × likelihood

Naive Bayes classifiers

- **Naive Bayes Assumption:** The "naive" in Naive Bayes comes from the assumption that all features are **independent** given the class label.
- This simplifies the computation of probabilities, as it assumes:

$$P(x_1, x_2, \ldots, x_n | y) = P(x_1 | y) \cdot P(x_2 | y) \cdot \ldots \cdot P(x_n | y)$$

- Without the independence assumption, calculating p(x|y=c) would require modeling the **joint distribution** of all features:

$$p(x|y=c) = p(x_1, x_2, \ldots, x_D | y = c)$$

- Modeling this joint distribution is computationally expensive, especially for high-dimensional data.

Naive Bayes classifiers

- Given a dataset with n features $(x_1, x_2 \dots x_n)$ and a set of classes $(y_1, y_2, \dots y_k)$, the goal is to find the class y that maximizes the posterior probability:

$$\hat{y} = \arg\max_{y} P(y|x_1, x_2, \dots, x_n)$$

- Using bayes theorem:**posterior ∝ prior × likelihood**

$$P(y|x_1, x_2, \dots, x_n) \propto P(y) \cdot P(x_1, x_2, \dots, x_n|y)$$

- Applying independence assumption:

$$P(x_1, x_2, \dots, x_n|y) = P(x_1|y) \cdot P(x_2|y) \cdot \dots \cdot P(x_n|y)$$

- Thus we get:

$$P(y|x_1, x_2, \dots, x_n) \propto P(y) \cdot \prod_{i=1}^{n} P(x_i|y)$$

Naive Bayes classifiers

- The equation expresses the class-conditional probability p(x|y=c,θ) where:

$$p(x|y=c,\theta)=\prod_{d=1}^{D}p(x_d|y=c,\theta_{dc})$$

- x=(x_1,x_2,…,x_D) is the feature vector (with D features).
- y=c denotes that we are conditioning on a specific class c.
- θ_{dc} represents the parameters for the conditional probability of feature d given the class c.

Naive Bayes classifiers

- The equation expresses the posterior probability of a class y=c given the feature vector x:

$$p(y=c|x,\theta)=\frac{p(y=c|\pi)\prod_{d=1}^{D}p(x_d|y=c,\theta_{dc})}{\sum_{c'}p(y=c'|\pi)\prod_{d=1}^{D}p(x_d|y=c',\theta_{dc'})}$$

- p(y=c|x,θ): Posterior probability of class c given the feature vector x.
- $p(y{=}c|\pi)$: Prior probability of class c, often denoted as π_c

Types of Naïve Bayes Classifiers

- **Gaussian Naïve Bayes**: Handles continuous features with a normal distribution.
- **Multinomial Naïve Bayes**: Suitable for text data and discrete features (e.g., word counts).
- **Bernoulli Naïve Bayes**: Works with binary/Boolean features (e.g., word presence/absence).

General Problem Solving

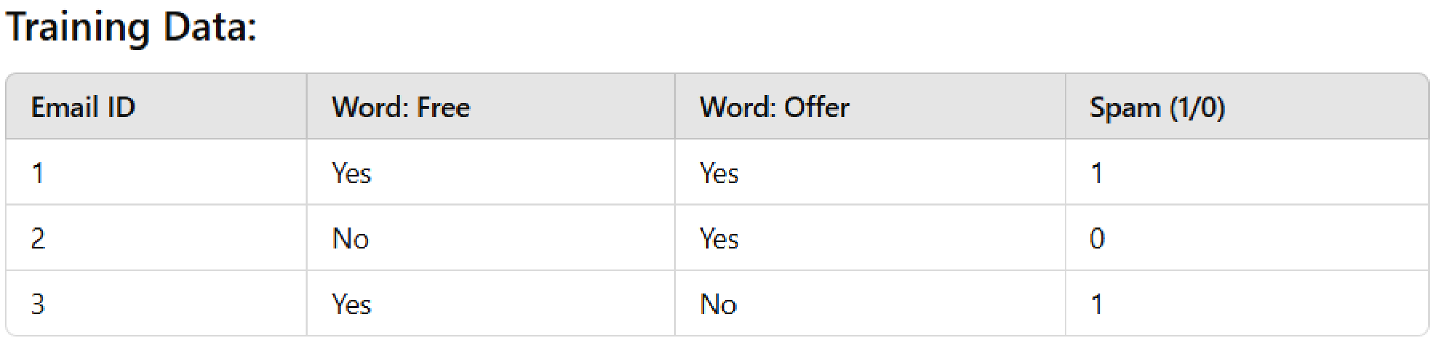
- **Data Preparation**: Collect and preprocess the dataset. Represent in tabular columns
- **Calculate Priors** (P(C) :For each class C, compute the proportion of samples that belong to C
- **Compute Likelihood ($P(F_i|C)$) :**For each feature F_i and class C, compute probabilities
- **Apply Bayes' Theorem :** Combine priors and likelihoods for a given class C to compute the posterior probability

$$P(y|x_1, x_2, \ldots, x_n) \propto P(y) \cdot \prod_{i=1}^{n} P(x_i|y)$$

- **Classification:** Assign the sample to the class C with the highest posterior probability $C_{\text{predicted}} = \arg\max_{C} P(C|X)$

NB Solved Example:

- Classify an email as "spam" or "not spam" based on features like the presence of specific words (e.g., "free," "offer").



Training Data:

Email ID	Word: Free	Word: Offer	Spam (1/0)
1	Yes	Yes	1
2	No	Yes	0
3	Yes	No	1

- Compute Prior Probabilities:
  - P(Spam)=2/3,
  - P(Not Spam)=1/3

NB Solved Example:

- **Compute Likelihood Probabilities:**

$$P(\text{Free} \mid \text{Spam}) = \frac{2}{2}, \quad P(\text{Offer} \mid \text{Spam}) = \frac{1}{2}$$

2 out of 2 Spam emails contain "free"
1 out of 2 Spam emails contain “Offer"

$$P(Free|Not\ Spam) = \frac{0}{1} = 0 \quad P(Offer|Not\ Spam) = \frac{1}{1} = 1$$

Bayes rule :
posterior ∝ prior × likelihood

- **Compute Posterior Probabilities:**
- Given a new email with Free = Yes, Offer = Yes

$$P(\text{Spam} \mid \text{Free, Offer}) \propto P(\text{Spam}) \cdot P(\text{Free} \mid \text{Spam}) \cdot P(\text{Offer} \mid \text{Spam})$$

$$P(\text{Not Spam} \mid \text{Free, Offer}) \propto P(\text{Not Spam}) \cdot P(\text{Free} \mid \text{Not Spam}) \cdot P(\text{Offer} \mid \text{Not Spam})$$

- Classify : $C_{\text{predicted}} = \arg\max_{C} P(C|X)$

END