$\LaTeX$ command declarations here.

In [3]:
```
%matplotlib inline
from Lec07 import *
```

EECS 445: Introduction to Machine Learning

Lecture 07: Naive Bayes

- Instructor: **Jacob Abernethy** and **Jia Deng**
- Date: September 28, 2016

Lecture Exposition Credit: Benjamin Bray, Valli Chockalingam

Outline

- Probabilistic Models
  - Generative Models
  - Discriminative Models
- Naive Bayes Classifiers
  - Independence Assumption
  - MLE and MAP Parameter Estimates

Reading List

- Suggested:
  - **[PRML]**, §4.2: Probabilistic Generative Models
  - **[PRML]**, §4.3: Probabilistic Discriminative Models
  - **[MLAPP]**, §3.5: Naive Bayes Classifiers

> In this lecture, we will first talk about the some concepts of probabilistic models for classifiers, especially generative model and discriminant model. And we will introduce Naive Bayes classifier, which assumes independent features give label and is a classical classifier commonly used in spam email classification.

Probablistic Models

Probablistic Models: Generative Models

- **Generative model** learns *class-conditional* $P(X|Y)$ and label densities $P(Y)$ from training data
- Perform prediction using the **posterior** via Bayes' Rule. For some new data $\mathbf{x}^{new}$

$$\begin{aligned} y &= \underset{k\in\{1,\ldots,K\}}{\arg\max} P(Y=k|X=\mathbf{x}^{new}) \\ &= \underset{k\in\{1,\ldots,K\}}{\arg\max} \frac{P(X=\mathbf{x}^{new}|Y=k)P(Y=k)}{P(X=\mathbf{x}^{new})} \\ &= \boxed{\underset{k\in\{1,\ldots,K\}}{\arg\max} P(X=\mathbf{x}^{new}|Y=k)P(Y=k)} \end{aligned}$$

  of which the last equality holds because the denominator $P(X=\mathbf{x}^{new})$ is independent of k.
- Basic idea of prediction is picking the label with largest posterior probability given its features $\mathbf{x}^{new}$.
- Why is this model called **generative**?
  - We learned *class-conditional probabiliy* $P(X|Y)$ from training data.
  - $P(X|Y)$ is distribution of data X given label Y
  - So given some label Y, could **generate**/sample new data X from $P(X|Y)$.
- The *prior* $P(Y)$ encodes beliefs about popularity of each label
- By comparing the synthetic data and real data, we get a sense of how good our generative model is.

Probablistic Models: Discriminative Models

- Conversely, a **discriminative model** learns posterior $P(Y|X)$ directly from training data.
- Goal: select a hypothesis to *discriminates* between class labels.
- The prediction for some new data $\mathbf{x}^{new}$ is

$$y = \underset{k\in\{1,\ldots,K\}}{\arg\max} P(Y=k|X=\mathbf{x}^{new})$$

- Does not (necessarily) provide the ability to **generate** new random examples because unlike generative models, we have no idea what $P(X|Y)$ is.
- Allows us to focus purely on the classification task
- We will discuss the pros and cons of each model later.

Probablistic Models: Discriminative Models—Property

- The discriminative approach will typically
  - make fewer generative assumptions about the data
    - However, reconstruction features from labels may require prior knowledge

Naive Bayes Classifiers

> Follows the approach taken by **[MLAPP]**

Naive Bayes: Problem

- We will use **Naive Bayes** to solve the following classification problem:
  - **Categorical** feature vector $\mathbf{x} = (x_1, x_2, \dots, x_D)$ with length D
    - Each feature $x_d \in \{1, \dots, M\}, \forall d = 1, \dots, D$
  - Predict discrete class label $y \in \{1, 2, \dots, C\}$
- For example, in **Spam Mail Classification**,
  - Predict whether an email is `SPAM` ($y = 1$) or `HAM` ($y = 0$)
  - Use words / metadata in the email as features
  - For simplicity, we can use **bag-of-words** features,
    - Assume fixed vocabulary V of size $|V| = D$
    - Feature x_d, for $d \in \{1, 2, \dots, D\}$, indicates the existence of dth word in the email
    - Eg. $x_d = 1$ if dth word is in the email; $x_d = 0$ otherwise
    - In this case $M = 2$

Naive Bayes: Independence Assumption and Full model

- The essence of Naive Bayes is the **conditionally independence assumption**

$$P(\mathbf{x}|y = c) = \prod_{d=1}^{D} P(x_d|y = c)$$

  i.e., given the label, all features are independent.
- The **full generative** model of Naive Bayes is:

$$y \sim \text{Categorical}(\pi)$$
$$x_d|y = c \sim \text{Categorical}(\theta_{cd}) \quad \forall\, d = 1, \dots, D$$

  with parameters:
  - **Class priors** $\pi = (\pi_1, \dots, \pi_C) \in \Delta^C$,
    - i.e. $P(y = c) = \pi_c, \forall c = 1, \dots, C$
    - Δ^C is C-**simplex**. $\pi \in \Delta^C$ is saying that $\sum_{c=1}^{C} \pi_c = 1$ and $\pi_c \geq 0, \forall c = 1, \dots, C$
  - **Class-conditional probabilities** $\theta_{cd} = (\theta_{cd1}, \dots, \theta_{cdM}) \in \Delta^M$
    - i.e. $P(x_d = m|y = c) = \theta_{cdm}$ for every $d = 1, \dots, D, m = 1, \dots, M, c = 1, \dots, C$
- Parameter π and θ are learned from training data.

> **Remark**
>
> - **NOTE** in definition and derivation of this lecture, we assume a more general case $x_d \in \{1, \dots, M\}$ of which $M > 2$. But in spam email classification and the derivation in textbook, binary feature, i.e. $M = 2$, is used. So don't get confused!
> - When $M = 2$, $x_d|y = c$ is also Bernoulli distribution.

Naive Bayes: Prediction

- Given the independence assumption and full model, for some new data $\mathbf{x}^{\text{new}} = (x_1^{\text{new}}, \dots, x_D^{\text{new}})$ we will classify based on

$$\begin{aligned} y &= \underset{c\in\{1,\dots,C\}}{\arg\max} P(y = c|\mathbf{x} = \mathbf{x}^{\text{new}}) \\ &= \underset{c\in\{1,\dots,C\}}{\arg\max} P(\mathbf{x} = \mathbf{x}^{\text{new}}|y = c)P(y = c) \\ &= \underset{c\in\{1,\dots,C\}}{\arg\max} P(y = c) \prod_{d=1}^{D} P(x_d = x_d^{\text{new}}|y = c) \\ &= \boxed{\underset{c\in\{1,\dots,C\}}{\arg\max} \pi_c \prod_{d=1}^{D} \theta_{cdx_d^{\text{new}}}} \end{aligned}$$

- If we assume $x_d^{\text{new}} \in \{1, \dots, M\}, \forall d = 1, \dots, D$, we could also express the above expression equivalently using **indicator function**

$$y = \underset{c\in\{1,\dots,C\}}{\arg\max} \pi_c \prod_{d=1}^{D} \prod_{m=1}^{M} \theta_{cdm}^{\mathbb{I}(m=x_d^{\text{new}})}$$

- So as long as we learned parameter π and θ, we could classify.

> **Remark**
>
> - Indicator function
>
> $$\mathbb{I}(m = x_d^{\text{new}}) = \begin{cases} 1 & \text{if } m = x_d^{\text{new}} \\ 0 & \text{otherwise} \end{cases}$$
>
> - In inner product $\prod_{m=1}^{M} \theta_{cdm}^{\mathbb{I}(m=x_d^{\text{new}})}$, only $\theta_{cdx_d^{\text{new}}}$ is multiplied and all the other multipliers are 1 due to the power of indicator function.
> - One thing to note is that the above classification criterion is the product of a series numbers smaller than 1 which will generate a rather small number. A better way is to take **logarithm** to transform product into summation and then compare.

Naive Bayes: Parameter Estimation

- **Goal:** Given training data $\mathcal{D} = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_N, y_N)\}$, estimate **class-conditional probabilities** θ and **class priors** π.
- We will discuss the **MLE** and **MAP** parameter estimates.

Naive Bayes: Maximum Likelihood

- The **likelihood** for a single data case $(\mathbf{x}_n, y_n = c)$ is

$$\begin{aligned} & P((\mathbf{x}_n, y_n)|\pi, \theta) \\ &= P(y_n) \prod_{d=1}^{D} P(x_{nd}|y_n) \\ &= \prod_{c=1}^{C} P(y_n = c)^{\mathbb{I}(y_n=c)} \cdot \prod_{c=1}^{C} \prod_{d=1}^{D} \prod_{m=1}^{M} P(x_{nd} = m|y_n = c)^{\mathbb{I}(x_{nd}=m)\mathbb{I}(y_n=c)} \\ &= \prod_{c=1}^{C} \pi_c^{\mathbb{I}(y_n=c)} \cdot \prod_{c=1}^{C} \prod_{d=1}^{D} \prod_{m=1}^{M} \theta_{cdm}^{\mathbb{I}(x_{nd}=m)\mathbb{I}(y_n=c)} \end{aligned}$$

- Therefore, the **log-likelihood** is

$$\begin{aligned} & \log P((\mathbf{x}_n, y_n)|\pi, \theta) \\ &= \sum_{c=1}^{C} \mathbb{I}(y_n = c) \log \pi_c + \sum_{c=1}^{C} \sum_{d=1}^{D} \sum_{m=1}^{M} \mathbb{I}(x_{nd} = m)\mathbb{I}(y_n = c) \log \theta_{cdm} \end{aligned}$$

- The **log-likelihood** for all training data $\mathcal{D} = \{(\mathbf{x}_n, y_n)\}_{n=1}^{N}$ is

$$\begin{aligned} & \log P(\mathcal{D}|\pi, \theta) \\ &= \log \prod_{n=1}^{N} P((\mathbf{x}_n, y_n)|\pi, \theta) = \sum_{n=1}^{N} \log P((\mathbf{x}_n, y_n)|\pi, \theta) \\ &= \boxed{\sum_{n=1}^{N} \sum_{c=1}^{C} \mathbb{I}(y_n = c) \log \pi_c + \sum_{n=1}^{N} \sum_{c=1}^{C} \sum_{d=1}^{D} \sum_{m=1}^{M} \mathbb{I}(x_{nd} = m)\mathbb{I}(y_n = c) \log \theta_{cdm}} \end{aligned}$$

Naive Bayes: Maximum Likelihood

- With the constraints $\sum_{c=1}^{C} \pi_c = 1$ and $\sum_{m=1}^{M} \theta_{cdm} = 1$, we could maximize log-likelihood function $\log P(\mathcal{D}|\pi, \theta)$ using *Lagrange multiplier*. (Derivation is in the notes!)
- By maximizing log-likelihood function, we could have maximum likelihood estimators:

$$\hat{\pi}_c = \frac{N_c}{N} \quad \hat{\theta}_{cdm} = \frac{N_{cdm}}{N_c}$$

and

$$\hat{\pi} = (\hat{\pi}_1, \ldots, \hat{\pi}_c, \ldots, \hat{\pi}_C); \hat{\theta}_{cd} = (\hat{\theta}_{cd1}, \ldots, \hat{\theta}_{cdm}, \ldots, \hat{\theta}_{cdM})$$

  - N = Number of examples in $\mathcal{D}$
  - N_c = Number of examples in class c in $\mathcal{D}$
  - N_{cdm} = Number of examples in class c with $x_d = m$ in $\mathcal{D}$
- Intuitive Interpretation
  - The class prior π is obtained from the density of each class $\{1, \ldots, C\}$ in $\mathcal{D}$
  - The class-conditional probability θ_{cd} is obtained from the density of $x_d \in \{1, \ldots, M\}$ among all examples in class c

Remark

- Derivation of **maximum likelihood estimator** $\hat{\pi}_c$
  - We have the following problem

$$\begin{cases} \max & \log P(\mathcal{D}|\pi,\theta) \\ \text{s.t.} & \sum_{c=1}^{C}\pi_c = 1 \end{cases} \quad \overset{\text{equivalent to}}{\Longrightarrow} \quad \begin{cases} \max & \sum_{n=1}^{N}\sum_{c=1}^{C}\mathbb{I}(y_n = c)\log\pi_c \\ \text{s.t.} & \sum_{c=1}^{C}\pi_c = 1 \end{cases}$$

    We drop the second term in $\log P(\mathcal{D}|\pi,\theta)$ because it doesn't depend on π_c
  - The lagragian is

$$L(\pi,\lambda) = \sum_{n=1}^{N}\sum_{c=1}^{C}\mathbb{I}(y_n = c)\log\pi_c - \lambda\sum_{c=1}^{C}\pi_c - \lambda$$

  - Setting partial derivative with respect to π_c to 0, we have

$$\frac{\partial L(\pi,\lambda)}{\partial\pi_c} = 0 \quad \Rightarrow \quad \sum_{n=1}^{N}\mathbb{I}(y_n = c)\frac{1}{\pi_c} - \lambda = 0 \quad \Rightarrow$$

$$\pi_c = \frac{1}{\lambda}\sum_{n=1}^{N}\mathbb{I}(y_n = c)$$

  - Plug π_c back into the constraint $\sum_{c=1}^{C}\pi_c = 1$, we have

$$\frac{1}{\lambda}\sum_{c=1}^{C}\sum_{n=1}^{N}\mathbb{I}(y_n = c) = 1 \quad \Rightarrow \quad \lambda = \sum_{c=1}^{C}\sum_{n=1}^{N}\mathbb{I}(y_n = c)$$

  - Plug λ into $\pi_c = \frac{1}{\lambda}\sum_{n=1}^{N}\mathbb{I}(y_n = c)$, we have

$$\hat{\pi}_c = \frac{\sum_{n=1}^{N}\mathbb{I}(y_n = c)}{\sum_{n=1}^{N}\sum_{c=1}^{C}\mathbb{I}(y_n = c)} = \frac{N_c}{N}$$

- Derivation of maximum likelihood estimator $\hat{\theta}_{cdm}$
  - With the constraint $\sum_{m=1}^{M}\theta_{cdm} = 1$, using similar approach, we could have

$$\hat{\theta}_{cdm} = \frac{\sum_{n=1}^{N}\mathbb{I}(x_{nd} = m)\mathbb{I}(y_n = c)}{\sum_{n=1}^{N}\sum_{m=1}^{M}\mathbb{I}(x_{nd} = m)\mathbb{I}(y_n = c)} = \frac{N_{cdm}}{N_c}$$

    - Details are left as an exercise XD

Naive Bayes: Sparse Features

- **Problem:** When working with text, features are **sparse**:
  - In training, we only see a *small, small* fraction of words in the vocabulary
  - Moreover, we won't see all words exhibited across all classes
- This causes overfitting!
  - What if a word (e.g. "`subject:`") occurs in every training example of both classes?
  - Then if we encounter a new email without this word, our algorithm will crash.
  - What happens if that word never appears in testing? (*Black Swan Paradox*)

Naive Bayes: Priors

- **Solution:** Place Dirichlet priors on π and θ_{cd} to *smooth out* unknowns:

$$\begin{aligned} \pi &\sim \text{Dirichlet}(\alpha_1, \dots, \alpha_C) \\ \theta_{cd} &\sim \text{Dirichlet}(\beta_{cd1}, \dots, \beta_{cdM}) && \forall\, c = 1, \dots, C; d = 1, \dots D \\ y &\sim \text{Categorical}(\pi) \\ x_d | y = c &\sim \text{Categorical}(\theta_{cd}) && \forall\, d = 1, \dots, D \end{aligned}$$

- **Dirichlet distribution** $\pi \sim \text{Dirichlet}(\alpha_1, \dots, \alpha_C)$ defines the distribution of C-simplex $\pi = (\pi_1, \dots, \pi_C)$ such that
  - $\pi_1, \dots, \pi_C \geq 0$
  - $\pi_1 + \cdots + \pi_C = 1$
  - PDF $f(\pi_1, \dots, \pi_C) = \frac{1}{B(\alpha)} \prod_{c=1}^{C} \pi_c^{\alpha_c - 1}$, where the quantity $B(\alpha)$ is to normalize the distribution
- When $M = 2$, θ_{cd} reduces to **Beta** distribution

Naive Bayes: MAP Estimate

- The **MAP parameter** estimates with priors

$$\pi \sim \text{Dirichlet}(\alpha_1, \dots, \alpha_C) \qquad \theta_{cd} \sim \text{Dirichlet}(\beta_{cd1}, \dots, \beta_{cdM})$$

  are

$$\hat{\pi}_c = \frac{N_c + \alpha_c - 1}{N + \sum_{c'=1}^{C} (\alpha_{c'} - 1)} \qquad \hat{\theta}_{cdm} = \frac{N_{cdm} + \beta_{cdm} - 1}{N_c + \sum_{m'=1}^{M} (\beta_{cdm'} - 1)}$$

- Proof is in the notes!
- The Dirichlet α and β_{cd} parameters turn out to be **pseudocounts**!
  - We assume we've seen α_c examples of class c beforehand
  - and β_{cdm} examples with $x_d = m$ in class c.
- The choice $\alpha_c = \beta_{cdm} = 1$ is referred to as **Laplace Smoothing**

Naive Bayes: Mean Estimate

- Note that the posterior of parameters still have **Dirichlet** distributions!

$$\pi|\mathcal{D} \sim \text{Dirichlet}(N_1 + \alpha_1, \dots, N_C + \alpha_C)$$
$$\theta_{cd}|\mathcal{D} \sim \text{Dirichlet}(N_{cd1} + \beta_{cd1}, \dots, N_{cdM} + \beta_{cdM})$$

  Proof for $\pi|\mathcal{D}$ is in the notes! Proof for $\theta_{cd}|\mathcal{D}$ is left as an exercise.
- If we pick the **mean** of posteriors as parameter estimate, we could get a slightly different result:

$$\bar{\pi}_c = \frac{N_c + \alpha_c}{N + \sum_{c'=1}^{C} \alpha_{c'}} \qquad \bar{\theta}_{cdm} = \frac{N_{cdm} + \beta_{cdm}}{N_c + \sum_{m'=1}^{M} \beta_{cdm'}}$$

  -1 terms no longer exist!
- You might see this version of estimate in **[MLAPP]** and some online materials
- Advantage of using mean estimate
  - Since posteriors still have Dirichlet distribution, we could use posterior as the prior for the next learning phase!
  - This can be helpful in sequential learning!

Remark

- Posterior also has Dirichlet distribution!
  - **Prior:**

$$\pi \sim \text{Dirichlet}(\alpha_1, \dots, \alpha_C) \qquad f(\pi) = \frac{1}{B(\alpha)} \prod_{c=1}^{C} \pi_c^{\alpha_c - 1}$$

  - **Likelihood:**

$$P(\mathcal{D}|\pi) = \prod_{n=1}^{N} \prod_{c=1}^{C} \pi_c^{\mathbb{I}(y_n = c)} = \prod_{c=1}^{C} \pi_c^{\sum_{n=1}^{N} \mathbb{I}(y_n = c)} = \prod_{c=1}^{C} \pi_c^{N_c}$$

  - **Posterior**

$$\begin{aligned} f(\pi|\mathcal{D}) &= \frac{P(\mathcal{D}|\pi) f(\pi)}{P(\mathcal{D})} = \frac{1}{P(\mathcal{D})} \prod_{c=1}^{C} \pi_c^{N_c} \cdot \frac{1}{B(\alpha)} \prod_{c=1}^{C} \pi_c^{\alpha_c - 1} \\ &= \frac{1}{P(\mathcal{D}) B(\alpha)} \prod_{c=1}^{C} \pi_c^{N_c + \alpha_c - 1} \\ &= \frac{1}{B'(\alpha)} \prod_{c=1}^{C} \pi_c^{N_c + \alpha_c - 1} \end{aligned}$$

    of which $B'(\alpha) = P(\mathcal{D}) B(\alpha)$.
  - From the expression of $f(\pi|\mathcal{D})$, we could see posterior also has **Dirichlet** distribution

$$\pi|\mathcal{D} \sim \text{Dirichlet}(N_1 + \alpha_1, \dots, N_C + \alpha_C)$$

- Conjugate Prior
  - If posterior and prior are in the same distribution family with respect to some likelihood, we could call this distribution **conjugate prior** for that likelihood.
  - This is useful because we could take the posterior as the prior for next learning phase, which enables us to do sequential Bayesian learning.
  - In our case, we have shown Dirichlet distribution is the conjugate prior of the multinomial distribution!

- Derivation of **MAP estimate** $\hat{\pi}_{c\,MAP}$
  - MAP estimate is obtained by maximizing the posterior $f(\pi|\mathcal{D})$

$$\begin{cases} \max & \prod_{c=1}^{C} \pi_c^{N_c+\alpha_c-1} \\ \text{s.t.} & \sum_{c=1}^{C} \pi_c = 1 \end{cases} \overset{\text{equivalent to}}{\Longrightarrow} \begin{cases} \max & \sum_{c=1}^{C} (N_c + \alpha_c - 1) \log \pi_c \\ \text{s.t.} & \sum_{c=1}^{C} \pi_c = 1 \end{cases}$$

  - Recall when deriving maximum likelihood estimator, we solved the following problem

$$\begin{cases} \max & \sum_{c=1}^{C} \sum_{n=1}^{N} \mathbb{I}(y_n = c) \log \pi_c \\ \text{s.t.} & \sum_{c=1}^{C} \pi_c = 1 \end{cases} \longrightarrow \hat{\pi}_{c\,MLE}$$

$$= \frac{\sum_{n=1}^{N} \mathbb{I}(y_n = c)}{\sum_{n=1}^{N} \sum_{c=1}^{C} \mathbb{I}(y_n = c)} = \frac{N_c}{N}$$

  - So the solution of our current problem can be easily read off

$$\hat{\pi}_{c\,MAP} = \frac{N_c + \alpha_c - 1}{\sum_{c'=1}^{C} \left(N_c + \alpha_c' - 1\right)} = \frac{N_c + \alpha_c - 1}{N + \sum_{c'=1}^{C} \left(\alpha_c' - 1\right)}$$

- Derivation of **MAP estimator** $\hat{\theta}_{cdm}$ is left as an exercise! Approach is exactly the same as deriving $\hat{\pi}_c$.

Naive Bayes: Is Independence Justified?

- Naive Bayes assumes features contribute *independently* to the class label.
  - This is the *simplest possible* generative model... and an **extreme** assumption...
- This model is *naive* because we would never expect features to be independent!
  - We are completely ignoring correlations between variables!
- It seems not to matter that independence is often false...
  - Naive Bayes performs surprisingly well on real-world data
  - Naive Bayes is often used as a baseline

Summary of Classifiers

- **Logistic Regression**
  - Provides model for $P(y|\mathbf{x})$ using sigmoid
  - No explicit model for $P(\mathbf{x}|y)$
- **Naive Bayes**
  - Provides a full model for $P(\mathbf{x}|y)$ and $P(y)$
  - Assumes independence between features *conditioned on* target y
  - Typically requires discrete data (can generalize to continuous spaces)
  - ML estimates are pretty straightforward