EECE 5644: Naïve Bayes Classifier

Mark Zolotas

E-mail: m.zolotas@northeastern.edu
Webpage: https://coe.northeastern.edu/people/zolotas-mark/

Tentative Course Outline (Wks. 3-4)

Topics	Dates	Assignments	Additional Reading
Naïve Bayes Classifier & *Homework 0 Practice Lab*	07/18	**Homework 2** released on Canvas on 07/22 **Due 08/01**	N/A
Model Fitting/Training: Bayesian Parameter Estimation	07/19-20		Chpts. 4.1-4.3, 8.7.2-3 Murphy 2022
Logistic Regression	07/21		Chpt. 10 Murphy 2022
Model Selection: Hyperparameter Tuning, k-fold Cross-Validation	07/25	**Homework 3** released on Canvas on 07/29 **Due 08/08**	Chpts. 4.5, 5.2, 5.4.3 Murphy 2022
Regularization, Ridge and Lasso Regression	07/26		Chpts. 4.5, 11.1-11.4 Murphy 2022
Neural Networks: Multilayer Perceptrons & Backpropagation	07/27-28		Chpts. 13.1-13.5 Murphy 2022

Recap: Decision Rules (1)

- Notation:
  - Let $L = \{1, \dots, C\}$ be the finite set of **labels**
  - Let $D = \{1, \dots, A\}$ be the finite set of possible **actions/decisions**
  - Then $\boldsymbol{\Lambda}$ is the **loss matrix** such that λ_{ij} is the loss/cost associated with deciding action i when the true label is j

- Maximum Likelihood (ML):

$$D(\mathbf{x}) = \arg\max_{j \in \{1,\dots,C\}} p(\mathbf{x} \mid l_j)$$

If $p(L)$ is uniform, then MAP reduces to an ML classifier

- Minimum Probability of Error/Maximum a Posteriori (MAP):

$$D(\mathbf{x}) = \arg\max_{j \in \{1,\dots,C\}} p(l_j \mid \mathbf{x}) = \arg\max_{j \in \{1,\dots,C\}} p(\mathbf{x} \mid l_j) \underline{p(l_j)}$$

Recap: Decision Rules (2)

- Minimum Probability of Error/Maximum a Posteriori (MAP):

$$D(\mathbf{x}) = \underset{j \in \{1,\dots,C\}}{\arg\max}\; p(l_j \,|\, \mathbf{x}) = \underset{j \in \{1,\dots,C\}}{\arg\max}\; p(\mathbf{x} \,|\, l_j) p(l_j)$$

If 0-1 loss, then ERM reduces to an MAP classifier
$\lambda_{ij} = 1 - \delta_{ij}$

- Empirical Risk Minimization (ERM):

$$D(\mathbf{x}) = \underset{i \in \{1,\dots,A\}}{\arg\min}\; R(d_i \,|\, \mathbf{x}) = \underset{i \in \{1,\dots,A\}}{\arg\min} \sum_{j=1}^{C} \underline{\lambda_{ij}} p(\mathbf{x} \,|\, l_j) p(l_j)$$

$$\begin{bmatrix} R(d_1 \,|\, \mathbf{x}) \\ \cdots \\ R(d_A \,|\, \mathbf{x}) \end{bmatrix} = \mathbf{\Lambda} \begin{bmatrix} p(l_1 \,|\, \mathbf{x}) \\ \cdots \\ p(l_C \,|\, \mathbf{x}) \end{bmatrix} = \begin{bmatrix} \lambda_{11} & \cdots & \lambda_{1C} \\ \vdots & \ddots & \vdots \\ \lambda_{A1} & \cdots & \lambda_{AC} \end{bmatrix} \operatorname{diag}\big(p(l_1), \dots, p(l_C)\big) \begin{bmatrix} p(\mathbf{x} \,|\, l_1) \\ \cdots \\ p(\mathbf{x} \,|\, l_C) \end{bmatrix}$$

Important Side Note: Log-Likelihoods

Let $\mathcal{D} = \{\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(N)}\}$ be our dataset of N random vector samples $\mathbf{x}^{(i)} \in \mathbb{R}^n$

- Assume samples are **iid**:

$$p(\mathcal{D}) = p_X(\mathbf{x}^{(1)}) \cdots p_X(\mathbf{x}^{(N)}) = \boxed{\prod_{i=1}^{N}} p(\mathbf{x}^{(i)})$$

Joint likelihood of entire dataset

Product will underflow

- Often take **log likelihoods** instead:

$$\ln p(\mathcal{D}) = \ln p_X(\mathbf{x}^{(1)}) + \cdots + \ln p_X(\mathbf{x}^{(N)}) = \sum_{i=1}^{N} \ln p(\mathbf{x}^{(i)})$$

Avoids the problem of very small dataset likelihoods

Bayesian Decision Theory vs Parameter Estimation

- **Bayesian Decision Theory** assumes the probability model for each category is known perfectly, i.e., $p(L)$ and $p(\mathbf{x} \mid L)$
- Bayes rule to infer **posterior:**

Easy to estimate from data

Impractical but necessary

$$p(L = j \mid \mathbf{x}) = \frac{\boxed{p(L = j)}\boxed{p(\mathbf{x} \mid L = j)}}{\boxed{p(\mathbf{x})}}$$

Unnecessary in practice to compute

- Rarely do we know true distributions
- **Bayesian Parameter Estimation** is concerned with estimating these pdfs from data, notably acquiring $\boldsymbol{\theta}$

Sources Of Error

- **Bayes or Irreducible Error**
  - Lowest possible error rate, cannot be eliminated
  - E.g., error due to overlapping densities for different classes
- **Model Error**
  - Due to having an incorrect model, e.g., assume $N(\mu, 1)$ when really $N(\mu, 10)$
  - Eliminated if designer picks true model based on domain knowledge
- **Estimation Error**
  - Error arising from estimating parameters based on finite samples (overfitting)
  - Can be reduced by increasing training data N

Naïve Bayes Classifier – Motivation

- Class-conditional likelihoods not easy to compute:

$$p(x_1, x_2, \cdots, x_n \,|\, L = j)$$

- Assume Bernoulli variables for inputs and labels, how many parameters do we need to describe $p(\mathbf{x} \,|\, L = j)$?

$$p(\mathbf{x}) = \prod_{k=1}^{n} \theta_k^{x_k} (1 - \theta_k)^{(1 - x_k)}$$

- **Exponential growth** with n: $2(2^n - 1)$ parameters

- How to reduce parameters (**sparsity**)? **Conditional Independence**

$$p(x_1, x_2, \cdots, x_n \,|\, L = j) = \prod_{k=1}^{n} p(x_k \,|\, L = j)$$

Only $2n$ *parameters in Bernoulli case*

Naïve Bayes Classifier – Key Idea

- Assume x_k are conditionally independent (**uncorrelated**) given a label

$$p(L = j \,|\, \mathbf{x}) = \frac{p(L = j) \prod_{k=1}^{n} p(x_k \,|\, L = j)}{p(\mathbf{x})}$$

- Same inference technique as before:

$$\underset{j \in \{1,\ldots,C\}}{\arg\max}\; p(L = j \,|\, \mathbf{x}) = \underset{j \in \{1,\ldots,C\}}{\arg\max}\; p(L = j) \prod_{k=1}^{n} p(x_k \,|\, L = j)$$

- Usually features are NOT conditionally independent given labels L
- Naïve Bayes still widely used to **reduce complexity** of PDFs

Practice Homework