Probabilistic Classification

Tozammel Hossain

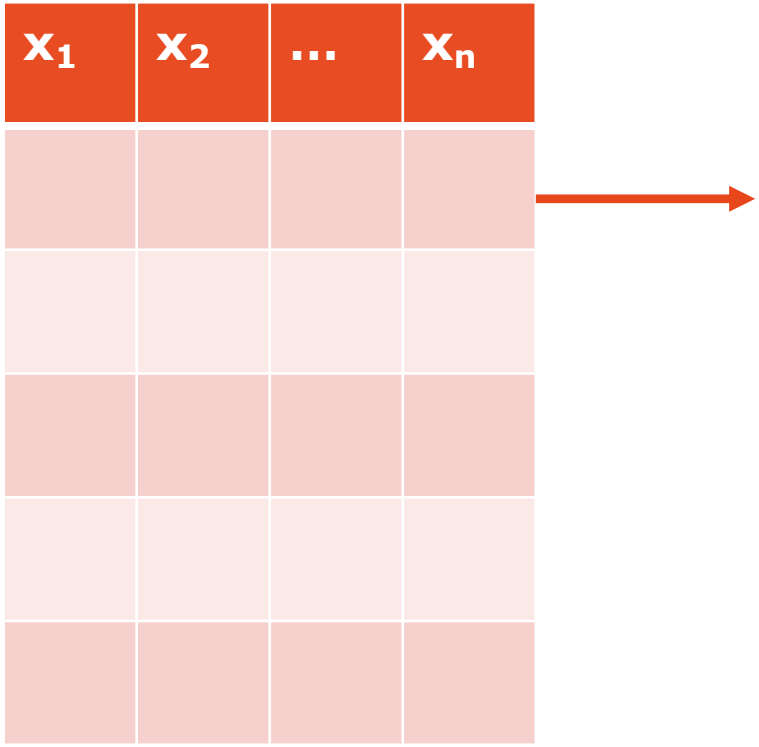
Data Science & Analytics
University of Missouri

Classification Setting Revisited

- **Given a set of rows/instances/examples**
  - Each row is a set of cols/features
  - Learn a function
    - class = f(instance)
  - And classify
    - New instances (test dataset) into a class
- **Across domains there are different names for these components**

x_1	x_2	...	x_n	y
				y_1
				y2
				y2
				Y3
				y1

Alternative Terms for Rows



- **entities**
- **instances**
- **examples**
- **records**
- **transactions**
- **objects**
- **points**
- **feature-vectors**
- **tuples**

Alternative Terms for Columns

x_1	x_2	...	x_n

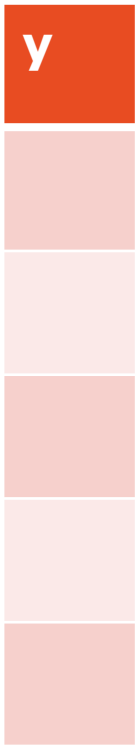
- **attributes**
- **properties**
- **features**
- **dimensions**
- **variables**
- **fields**

Alternative Terms for Input Variables

x_1	x_2	...	x_n

- independent variable
- predictor
- regressor (regression problem)
- covariate
- manipulated variable
- explanatory variable
- exposure variable (reliability theory)
- risk factor (see medical statistics),
- feature (in machine learning and pattern recognition)
- control variable (econometrics)
- exogenous (economics)

Alternative Terms for Output variable



- dependent variable
- response variable
- regressand (regression)
- criterion
- predicted variable
- measured variable
- explained variable
- experimental variable
- outcome variable
- target
- class
- label
- endogenous (economics)

Classification of classifiers

- **Rule-based**
  - Decision tree (M1)
- **Probabilistic**
  - Naïve Bayes classifier (M2)
- **Max-margin classifier**
  - SVM (M5)
- **Neural network (M8)**

Probabilistic Classification

- **Establishing a probabilistic model for classification**
  - Discriminative model

$$P(C \mid \mathbf{X}) \quad C = c_1, \cdots, c_L, \mathbf{X} = (X_1, \cdots, X_n)$$

  - Generative model

$$P(\mathbf{X} \mid C) \quad C = c_1, \cdots, c_L, \mathbf{X} = (X_1, \cdots, X_n)$$

- **MAP classification rule**
  - MAP: Maximum A Posterior
  - Assign x to c^* if

$$P(C = c^* \mid \mathbf{X} = \mathbf{x}) > P(C = c \mid \mathbf{X} = \mathbf{x}) \quad c \neq c^*, \; c = c_1, \cdots, c_L$$

A Bayesian Classifier

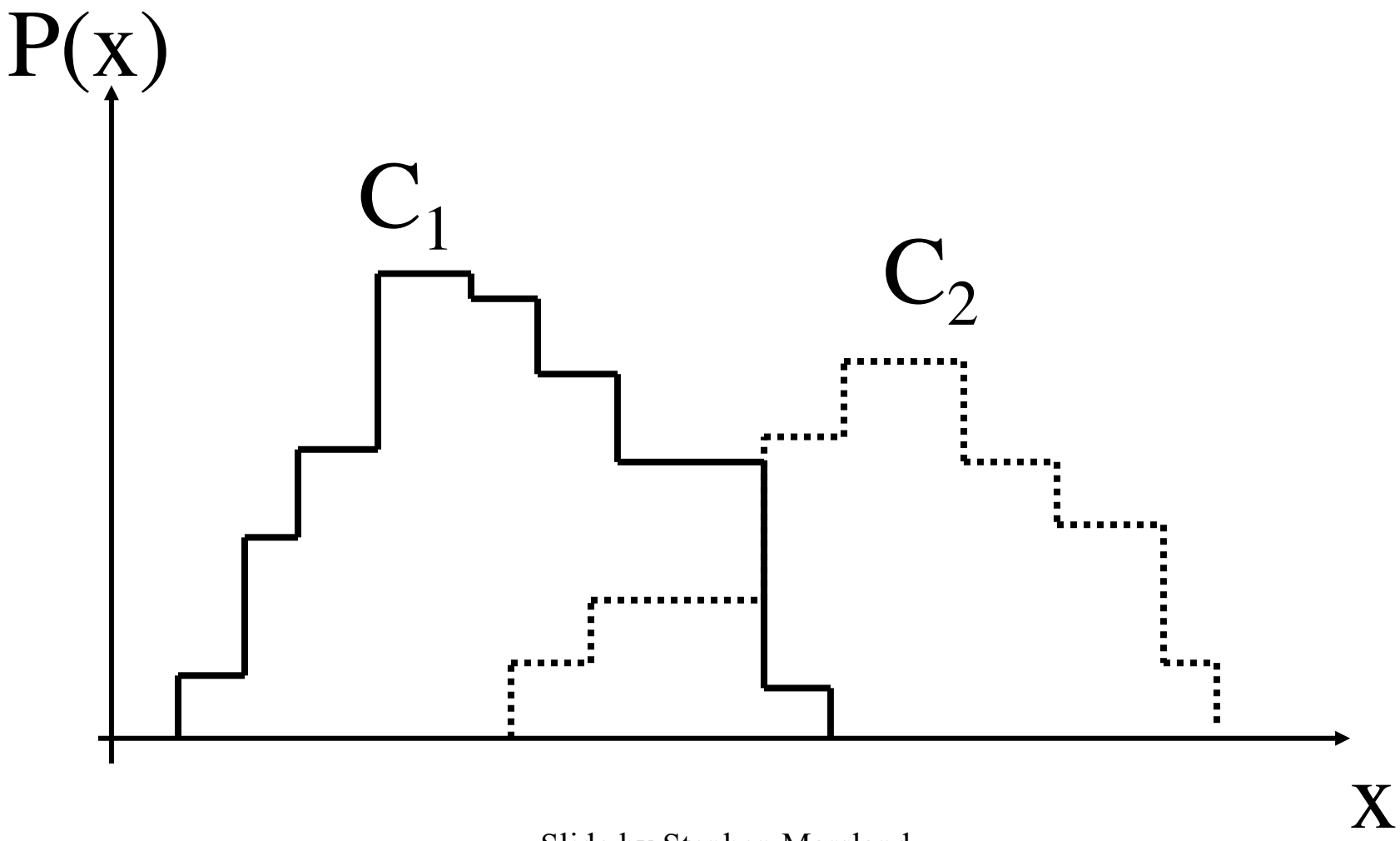
- **Generative classification with the MAP rule**
  - Apply Bayesian rule to convert

$$P(C \mid \mathbf{X}) = \frac{P(\mathbf{X} \mid C)P(C)}{P(\mathbf{X})} \propto P(\mathbf{X} \mid C)P(C)$$

  - MAP Classification Rule

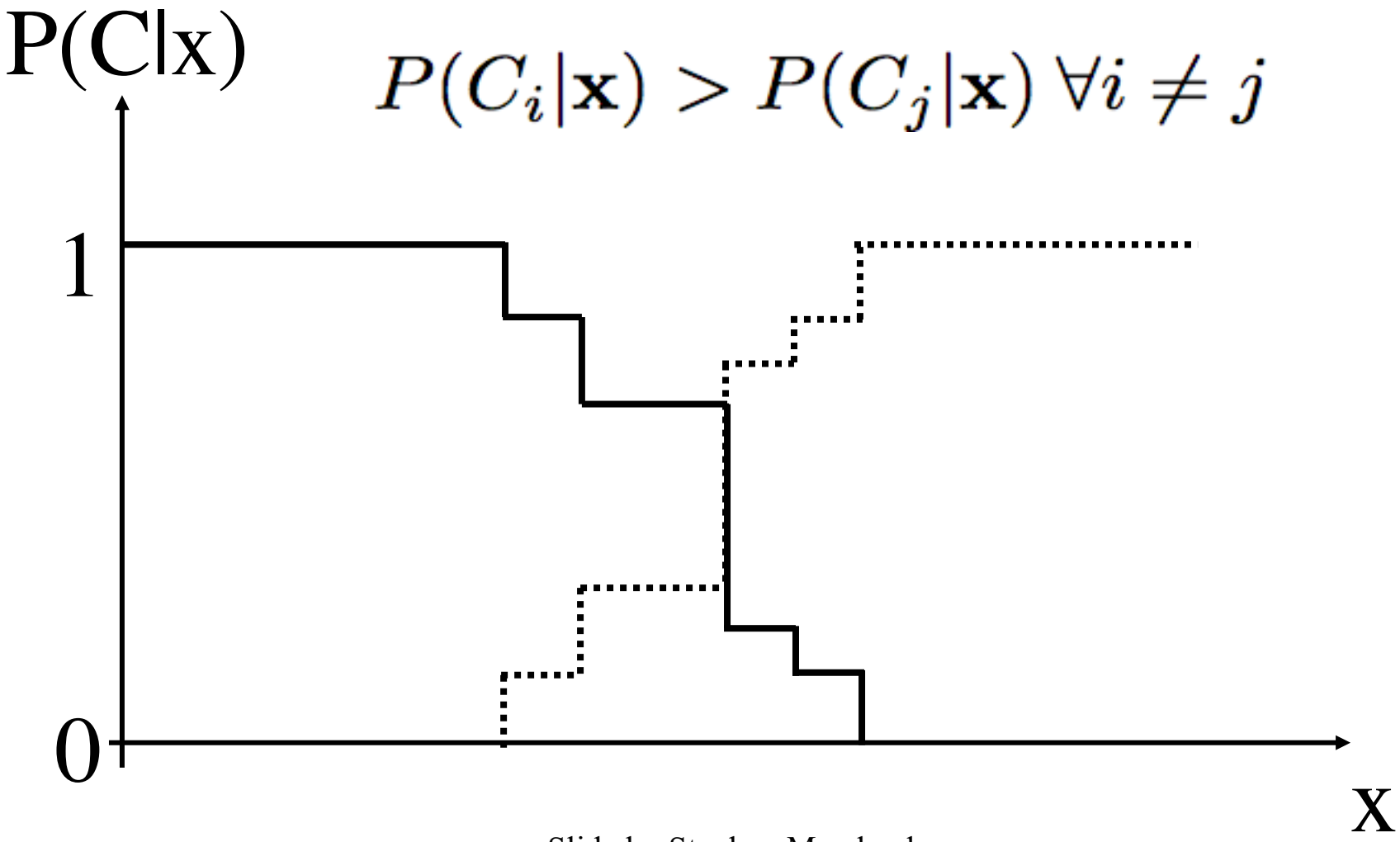
$$P(C = c^* \mid \mathbf{X} = \mathbf{x}) > P(C = c \mid \mathbf{X} = \mathbf{x}) \quad c \neq c^*,\ c = c_1, \cdots, c_L$$

Example: A Feature Histograms



Slide by Stephen Marsland

Example: Posterior Probability



Slide by Stephen Marsland

A Bayesian Classifier (Contd.)

$$P(C \mid \mathbf{X}) \propto P(\mathbf{X} \mid C)P(C) = P(X_1, \cdots, X_n \mid C)P(C)$$

- **Why is this not feasible in practice?**
  - Consider 3 classes, 10 features, each with 6 possible discrete instantiations
  - We need $3*6^{10} + 3$ parameters to be estimated ≈181 million!

Enter Naïve Bayes

- **Naïve Bayes assumption: Features are conditionally independent given the class**

$$
\begin{aligned}
P(X_1, X_2, \cdots, X_n \mid C) &= P(X_1 \mid X_2, \cdots, X_n; C) P(X_2, \cdots, X_n \mid C) \\
&= P(X_1 \mid C) P(X_2, \cdots, X_n \mid C) \\
&= P(X_1 \mid C) P(X_2 \mid C) \cdots P(X_n \mid C)
\end{aligned}
$$

Disclaimer: By all accounts, Rev. Thomas Bayes was pretty smart. Naivette is on our part, not Bayes!

Why is this a big deal?

- **Back to: 3 classes, 10 features, each with 6 possible discrete instantiations**
- **Requires 181 million parameters:**

$$C_{MAP} = argmax_{c \in C} P(C = c|\boldsymbol{X})$$
$$= argmax_{c \in C} P(X_1, X_2, \ldots, Xn|C = c)P(C = c)$$

- **Requires 3*60 + 3 = 183 parameters!**

$$C_{NB} = argmax_{c \in C} P(C = c) \prod_{i=1}^{n} P(Xi|C = c)$$

Example: Play Tennis Data

PlayTennis: training examples

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

How many parameters are required for a naïve Bayes classifier?

= (3 + 3 + 2 + 2) * 2 + 2
= 22

Learning Phase

P(Outlook=*o* | Play=*b*)

Outlook	Play=*Yes*	Play=*No*
Sunny	2/9	3/5
Overcast	4/9	0/5
Rain	3/9	2/5

P(Temperature=*t* | Play=*b*)

Temperature	Play=*Yes*	Play=*No*
Hot	2/9	2/5
Mild	4/9	2/5
Cool	3/9	1/5

P(Humidity=*h* | Play=*b*)

Humidity	Play=*Yes*	Play=No
High	3/9	4/5
Normal	6/9	1/5

P(Wind=*w* | Play=*b*)

Wind	Play=*Yes*	Play=*No*
Strong	3/9	3/5
Weak	6/9	2/5

P(Play=*Yes*) = 9/14

P(Play=*No*) = 5/14

Example

- **Test Phase**
  - Given a new instance, **x**′=(Outlook=*Sunny*, Temperature=*Cool*, Humidity=*High*, Wind=*Strong*)
  - Look up tables

P(Outlook=*Sunny*|Play=*Yes*) = 2/9

P(Temperature=*Cool*|Play=*Yes*) = 3/9

P(Huminity=*High*|Play=*Yes*) = 3/9

P(Wind=*Strong*|Play=*Yes*) = 3/9

P(Play=*Yes*) = 9/14

P(Outlook=S*unny*|Play=*No*) = 3/5

P(Temperature=*Cool*|Play==*No*) = 1/5

P(Huminity=*High*|Play=*No*) = 4/5

P(Wind=*Strong*|Play=*No*) = 3/5

P(Play=*No*) = 5/14

Example

- **MAP Rule**

P(*Yes*|**x**'): [P(*Sunny*|Yes)P(*Cool*|*Yes*)P(*High*|Yes)P(*Strong*|*Yes*)]P(Play=*Yes*) = 0.0053

P(*No*|**x**'): [P(*Sunny*|No) P(*Cool*|No)P(*High*|*No*)P(*Strong*|*No*)]P(Play=*No*) = 0.0206

Given the fact P(*Yes*|**x**') < P(*No*|**x**'), we label **x**' to be "*No*".

Applying NBC in practice

- **Classical example: text classification**
  - Instances are text samples
    - emails
    - paragraphs
    - sentences
    - documents
    - tweets
    - posts
    - comments
    - reviews
  - Classes are {spam, ~spam}

Sklearn naïve Bayes

- https://scikit-learn.org/stable/modules/naive_bayes.html

naive_bayes.BernoulliNB(*[, alpha, ...])	Naive Bayes classifier for multivariate Bernoulli models.
naive_bayes.CategoricalNB(*[, alpha, ...])	Naive Bayes classifier for categorical features
naive_bayes.ComplementNB(*[, alpha, ...])	The Complement Naive Bayes classifier described in Rennie et al.
naive_bayes.GaussianNB(*[, priors, ...])	Gaussian Naive Bayes (GaussianNB)
naive_bayes.MultinomialNB(*[, alpha, ...])	Naive Bayes classifier for multinomial models

Conclusions

- **Naïve Bayes based on the independence assumption**
  - Training is very easy and fast; just requiring considering each attribute in each class separately
  - Test is straightforward; just looking up tables or calculating conditional probabilities with normal distributions
- **A popular generative model**
  - Performance competitive to most of state-of-the-art classifiers even in presence of violating independence assumption
  - Many successful applications, e.g., spam mail filtering
  - Apart from classification, naïve Bayes can do more…