10-601 Introduction to Machine Learning

Machine Learning Department
School of Computer Science
Carnegie Mellon University

Bayesian Networks

Matt Gormley
Lecture 21
Apr. 3, 2019

Reminders

- **Midterm Exam 2**
  - **Thu, Apr 4 – evening exam, details announced on Piazza**
- **Homework 7: HMMs**
  - **Out: Fri, Mar 29**
  - **Due: Wed, Apr 10 at 11:59pm**
- **Today's In-Class Poll**
  - **http://p21.mlcourse.org**
- **Midterm Exam 1 Survey**
  - **https://piazza.com/class/jqnuz4ysoi96rm?cid=1806**

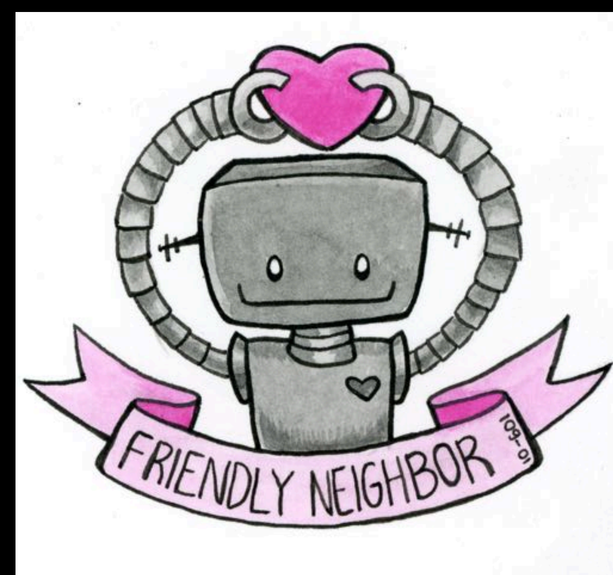
Question 1: Do you prefer chalkboard or digital whiteboard?

Reminders

Congratulations to our top Piazza Question Answerers for Midterm 1!

1. ba98959f457ec10d1272
2. 1465abbd2641a9a32459
3. 7636d8d965fd2e29626e
4. a3e3c79fc6310b5e54f6
5. 6112f1d4b2ad6ec178ff
6. c7b99972d87f77e0288f
7. 927e79510079b78549f4
8. 40ba2f9595a25edf584c
9. 1a2628b684e892154cf4
10. 73ab4e60182a6aa0ee40
11. 305fc04247ce71f3ba06
12. 9094a77492aa4fb6ec94

***Names passed through one-way crytographic hashing function (shake-256 with digest length 10) for FERPA compliance**



Q&A

Bayes Nets Outline

- **Motivation**
  - Structured Prediction
- **Background**
  - Conditional Independence
  - Chain Rule of Probability
- **Directed Graphical Models**
  - Writing Joint Distributions
  - Definition: Bayesian Network
  - Qualitative Specification
  - Quantitative Specification
  - Familiar Models as Bayes Nets
- **Conditional Independence in Bayes Nets**
  - Three case studies
  - D-separation
  - Markov blanket
- **Learning**
  - Fully Observed Bayes Net
  - (Partially Observed Bayes Net)
- **Inference**
  - Background: Marginal Probability
  - Sampling directly from the joint distribution
  - Gibbs Sampling

THE FORWARD-BACKWARD ALGORITHM

Forward-Backward Algorithm

Define: $\alpha_t(k) \triangleq p(x_1, \ldots, x_t, y_t = k)$

$\beta_t(k) \triangleq p(x_{t+1}, \ldots, x_T \mid y_t = k)$

Assume $y_0 = START$

$y_{T+1} = END$

① Initialize $\alpha_0(START) = 1 \qquad \alpha_0(k) = 0 \quad \forall k \neq START$

$\beta_{T+1}(END) = 1 \qquad \beta_{T+1}(k) = 0 \quad \forall k \neq END$

② For $t = 1, \ldots, T$:

For $k = 1, \ldots, K$:

$$\alpha_t(k) = p(x_t \mid y_t = k) \sum_{j=1}^{K} \alpha_{t-1}(j)\, p(y_t = k \mid y_{t-1} = j)$$

the alphas include the emission probabilities so we don't multiply them in separately

③ For $t = T, \ldots, T$:

For $k = 1, \ldots, K$:

$$\beta_t(k) = \sum_{j=1}^{K} p(x_{t+1} \mid y_{t+1} = j)\, \beta_{t+1}(j)\, p(y_{t+1} = j \mid y_t = k)$$

④ Compute $p(\vec{x}) = \alpha_{T+1}(END)$ [Evaluation]

⑤ Compute $p(y_t = k \mid \vec{x}) = \dfrac{\alpha_t(k)\, \beta_t(k)}{p(\vec{x})}$ [Marginals]

Derivation of Forward Algorithm

Definition: $\alpha_t(k) \triangleq p(x_1, \ldots, x_t, y_t = k)$

Derivation:

Herein using "y_T" as shorthand for "$y_T = END$"

$$\begin{aligned}
\alpha_T(END) &= p(x_1, \ldots, x_T, y_T = END) \\
&= p(x_1, \ldots, x_T \mid y_T)\, p(y_T) && \leftarrow \text{by def of joint} \\
&= p(x_T \mid y_T)\, p(x_1, \ldots, x_{T-1} \mid y_T)\, p(y_T) && \leftarrow \text{by cond. indep. of HMM} \\
&= p(x_T \mid y_T)\, p(x_1, \ldots, x_{T-1}, y_T) && \leftarrow \text{by def. of joint} \\
&= p(x_T \mid y_T) \sum_{y_{T-1}} p(x_1, \ldots, x_{T-1}, y_{T-1}, y_T) && \leftarrow \text{by def. of marginal} \\
&= p(x_T \mid y_T) \sum_{y_{T-1}} p(x_1, \ldots, x_{T-1}, y_T \mid y_{T-1})\, p(y_{T-1}) && \leftarrow \text{by def. of joint} \\
&= p(x_T \mid y_T) \sum_{y_{T-1}} p(x_1, \ldots, x_{T-1} \mid y_{T-1})\, p(y_T \mid y_{T-1})\, p(y_{T-1}) && \leftarrow \text{by cond. indep. of HMM} \\
&= p(x_T \mid y_T) \sum_{y_{T-1}} p(x_1, \ldots, x_{T-1}, y_{T-1})\, p(y_T \mid y_{T-1}) && \leftarrow \text{by def. of joint} \\
&= p(x_T \mid y_T) \sum_{y_{T-1}} \alpha_{T-1}(y_{T-1})\, p(y_T \mid y_{T-1}) && \leftarrow \text{by def. of } \alpha_t(k)
\end{aligned}$$

Viterbi Algorithm

Define: $\omega_t(k) \triangleq \max_{y_1,\ldots,y_{t-1}} p(x_1,\ldots,x_t,y_1,\ldots,y_{t-1},y_t=k)$

"backpointers" → $b_t(k) \triangleq \operatorname{argmax}_{y_1,\ldots,y_{t-1}} p(x_1,\ldots,x_t,y_1,\ldots,y_{t-1},y_t=k)$

Assume $y_0 = \text{START}$

① Initialize $\omega_0(\text{START})=1 \quad \omega_0(k)=0 \;\; \forall k \neq \text{START}$

② For $t=1,\ldots,T$:

For $k=1,\ldots,K$:

$$\omega_t(k) = \max_{j\in\{1,\ldots,K\}} p(x_t \mid y_t=k)\, \omega_{k-1}(j)\, p(y_t=k \mid y_{t-1}=j)$$

$$b_t(k) = \operatorname{argmax}_{j\in\{1,\ldots,K\}} p(x_t \mid y_t=k)\, \omega_{k-1}(j)\, p(y_t=k \mid y_{t-1}=j)$$

③ Compute Most Probable Assignment [Decoding]

$\hat{y}_T = b_{T+1}(\text{END})$

For $t=T-1,\ldots,1$

$\hat{y}_t = b_{t+1}(\hat{y}_{t+1})$

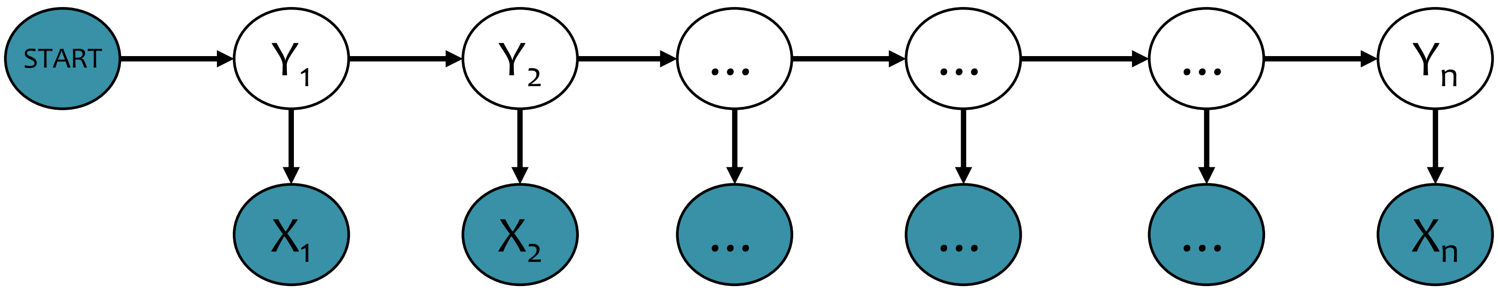
] follow the "backpointers"

Inference in HMMs

What is the **computational complexity** of inference for HMMs?

- The **naïve** (brute force) computations for *Evaluation*, *Decoding*, and *Marginals* take **exponential time**, $O(K^T)$
- The **forward-backward** algorithm and **Viterbi** algorithm run in **polynomial time**, $O(T*K^2)$
  - Thanks to dynamic programming!

Shortcomings of Hidden Markov Models



- HMM models capture dependences between each state and only its corresponding observation
  - NLP example: In a sentence segmentation task, each segmental state may depend not just on a single word (and the adjacent segmental stages), but also on the (non-local) features of the whole line such as line length, indentation, amount of white space, etc.
- Mismatch between learning objective function and prediction objective function
  - HMM learns a joint distribution of states and observations P(**Y**, **X**), but in a prediction task, we need the conditional probability P(**Y**|**X**)

MBR DECODING

Inference for HMMs

Four

– ~~Three~~ Inference Problems for an HMM

1. Evaluation: Compute the probability of a given sequence of observations
2. Viterbi Decoding: Find the most-likely sequence of hidden states, given a sequence of observations
3. Marginals: Compute the marginal distribution for a hidden state, given a sequence of observations
4. MBR Decoding: Find the lowest loss sequence of hidden states, given a sequence of observations (Viterbi decoding is a special case)

Minimum Bayes Risk Decoding

- Suppose we given a loss function $l(\boldsymbol{y}', \boldsymbol{y})$ and are asked for a single tagging
- How should we choose just one from our probability distribution $p(\boldsymbol{y}|\boldsymbol{x})$?
- A minimum Bayes risk (MBR) decoder $h(\boldsymbol{x})$ returns the variable assignment with minimum **expected** loss under the model's distribution

$$h_{\boldsymbol{\theta}}(\boldsymbol{x}) = \underset{\hat{\boldsymbol{y}}}{\operatorname{argmin}} \; \mathbb{E}_{\boldsymbol{y} \sim p_{\boldsymbol{\theta}}(\cdot|\boldsymbol{x})}[\ell(\hat{\boldsymbol{y}}, \boldsymbol{y})]$$

$$= \underset{\hat{\boldsymbol{y}}}{\operatorname{argmin}} \; \sum_{\boldsymbol{y}} p_{\boldsymbol{\theta}}(\boldsymbol{y} \mid \boldsymbol{x}) \ell(\hat{\boldsymbol{y}}, \boldsymbol{y})$$

Minimum Bayes Risk Decoding

$$h_{\boldsymbol{\theta}}(\boldsymbol{x}) = \underset{\hat{\boldsymbol{y}}}{\operatorname{argmin}} \; \mathbb{E}_{\boldsymbol{y} \sim p_{\boldsymbol{\theta}}(\cdot|\boldsymbol{x})}[\ell(\hat{\boldsymbol{y}}, \boldsymbol{y})]$$

Consider some example loss functions:

The ***0-1* loss function** returns *1* only if the two assignments are identical and *0* otherwise:

$$\ell(\hat{\boldsymbol{y}}, \boldsymbol{y}) = 1 - \mathbb{I}(\hat{\boldsymbol{y}}, \boldsymbol{y})$$

The MBR decoder is:

$$h_{\boldsymbol{\theta}}(\boldsymbol{x}) = \underset{\hat{\boldsymbol{y}}}{\operatorname{argmin}} \; \sum_{\boldsymbol{y}} p_{\boldsymbol{\theta}}(\boldsymbol{y} \mid \boldsymbol{x})(1 - \mathbb{I}(\hat{\boldsymbol{y}}, \boldsymbol{y}))$$

$$= \underset{\hat{\boldsymbol{y}}}{\operatorname{argmax}} \; p_{\boldsymbol{\theta}}(\hat{\boldsymbol{y}} \mid \boldsymbol{x})$$

which is exactly the Viterbi decoding problem!

Minimum Bayes Risk Decoding

$$h_{\boldsymbol{\theta}}(\boldsymbol{x}) = \underset{\hat{\boldsymbol{y}}}{\text{argmin}} \; \mathbb{E}_{\boldsymbol{y} \sim p_{\boldsymbol{\theta}}(\cdot | \boldsymbol{x})}[\ell(\hat{\boldsymbol{y}}, \boldsymbol{y})]$$

Consider some example loss functions:

The **Hamming loss** corresponds to accuracy and returns the number of incorrect variable assignments:

$$\ell(\hat{\boldsymbol{y}}, \boldsymbol{y}) = \sum_{i=1}^{V} (1 - \mathbb{I}(\hat{y}_i, y_i))$$

The MBR decoder is:

$$\hat{y}_i = h_{\boldsymbol{\theta}}(\boldsymbol{x})_i = \underset{\hat{y}_i}{\text{argmax}} \; p_{\boldsymbol{\theta}}(\hat{y}_i \mid \boldsymbol{x})$$

This decomposes across variables and requires the variable marginals.

Learning Objectives

Hidden Markov Models

You should be able to…

1. Show that structured prediction problems yield high-computation inference problems
2. Define the first order Markov assumption
3. Draw a Finite State Machine depicting a first order Markov assumption
4. Derive the MLE parameters of an HMM
5. Define the three key problems for an HMM: evaluation, decoding, and marginal computation
6. Derive a dynamic programming algorithm for computing the marginal probabilities of an HMM
7. Interpret the forward-backward algorithm as a message passing algorithm
8. Implement supervised learning for an HMM
9. Implement the forward-backward algorithm for an HMM
10. Implement the Viterbi algorithm for an HMM
11. Implement a minimum Bayes risk decoder with Hamming loss for an HMM

Bayesian Networks

DIRECTED GRAPHICAL MODELS

Example: Tornado Alarms

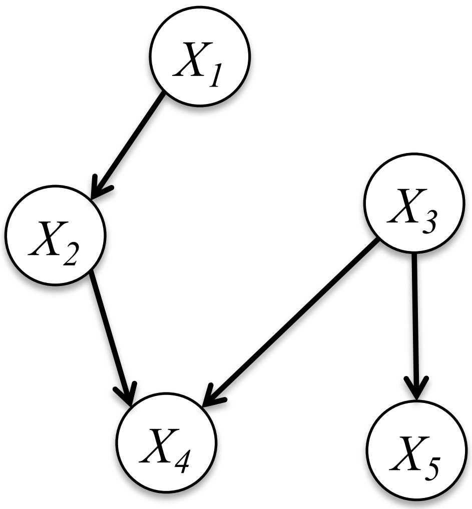
1. Imagine that you work at the 911 call center in Dallas
2. You receive six calls informing you that the Emergency Weather Sirens are going off
3. What do you conclude?

Directed Graphical Models (Bayes Nets)

Whiteboard

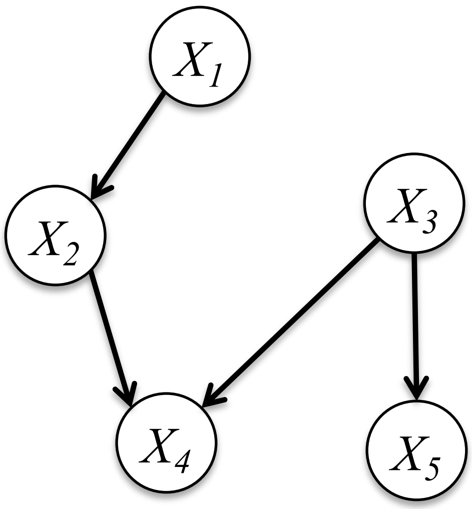
- Example: Tornado Alarms
- Writing Joint Distributions
  - Idea #1: Giant Table
  - Idea #2: Rewrite using chain rule
  - Idea #3: Assume full independence
  - Idea #4: Drop variables from RHS of conditionals
- Definition: Bayesian Network

Bayesian Network



$$p(X_1, X_2, X_3, X_4, X_5) =$$
$$p(X_5|X_3)p(X_4|X_2, X_3)$$
$$p(X_3)p(X_2|X_1)p(X_1)$$

Bayesian Network



Definition:

$$P(X_1 \ldots X_n) = \prod_{i=1}^{n} P(X_i \mid parents(X_i))$$

- A Bayesian Network is a **directed graphical model**
- It consists of a graph **G** and the conditional probabilities **P**
- These two parts full specify the distribution:
  - Qualitative Specification: **G**
  - Quantitative Specification: **P**

Qualitative Specification

- Where does the qualitative specification come from?
  - Prior knowledge of causal relationships
  - Prior knowledge of modular relationships
  - Assessment from experts
  - Learning from data (i.e. structure learning)
  - We simply prefer a certain architecture (e.g. a layered graph)
  - …

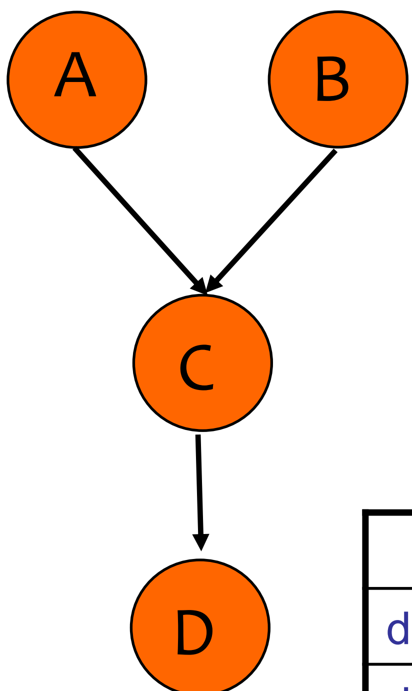
Quantitative Specification

Example: Conditional probability tables (CPTs) for discrete random variables

$$P(a,b,c.d) = P(a)P(b)P(c|a,b)P(d|c)$$

a^0	0.75
a^1	0.25

b^0	0.33
b^1	0.67



	a^0b^0	a^0b^1	a^1b^0	a^1b^1
c^0	0.45	1	0.9	0.7
c^1	0.55	0	0.1	0.3

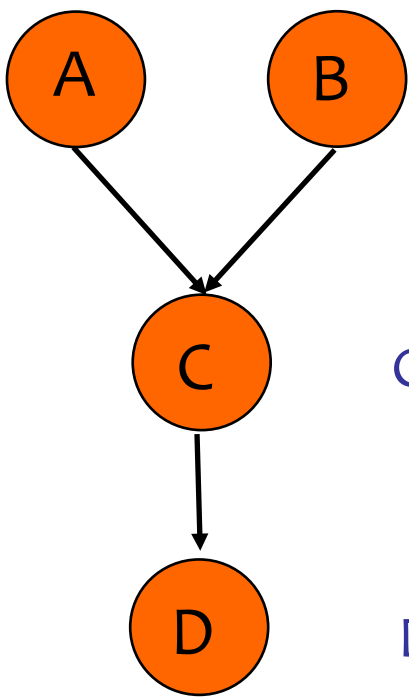
	c^0	c^1
d^0	0.3	0.5
d^1	07	0.5

Quantitative Specification

Example: Conditional probability density functions (CPDs) for continuous random variables

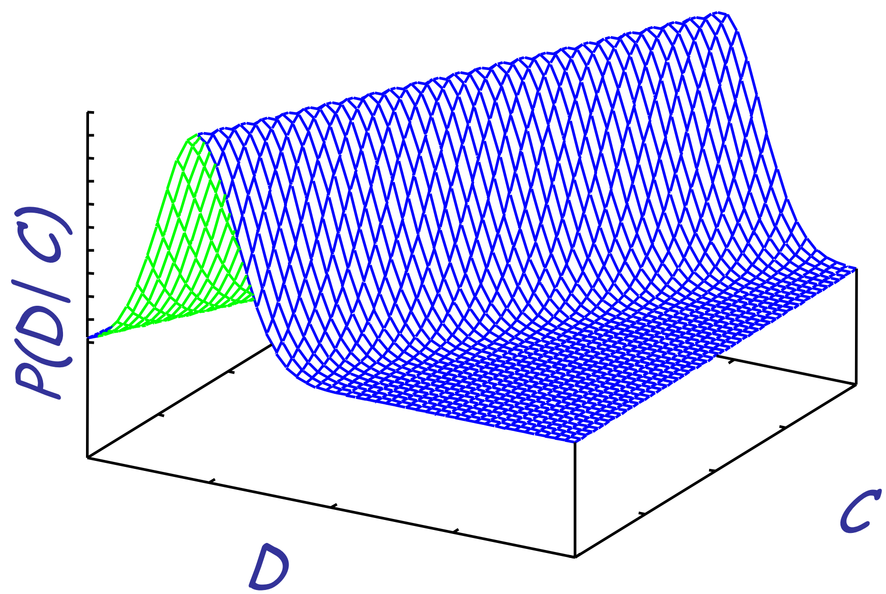
$A \sim N(\mu_a, \Sigma_a)$ $B \sim N(\mu_b, \Sigma_b)$

$$P(a,b,c.d) = P(a)P(b)P(c|a,b)P(d|c)$$



$C \sim N(A+B, \Sigma_c)$

$D \sim N(\mu_d + C, \Sigma_d)$



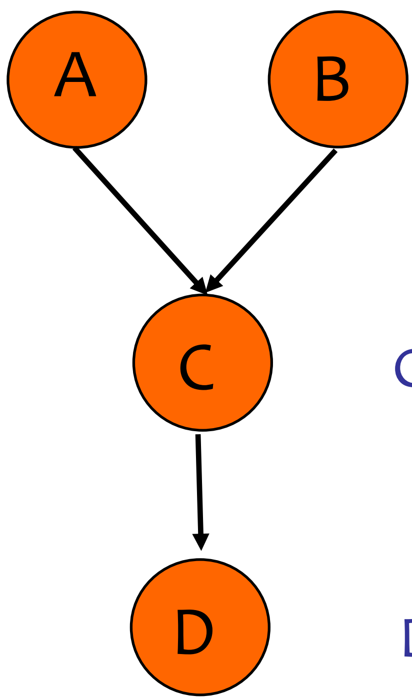
Quantitative Specification

Example: Combination of CPTs and CPDs
for a mix of discrete and continuous variables

a^0	0.75
a^1	0.25

b^0	0.33
b^1	0.67

$$P(a,b,c.d) = P(a)P(b)P(c|a,b)P(d|c)$$



$C \sim N(A+B, \Sigma_c)$

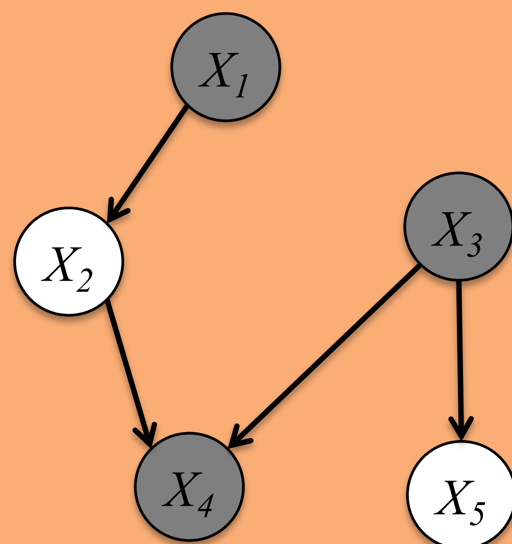
$D \sim N(\mu_d + C, \Sigma_d)$

Observed Variables

- In a graphical model, **shaded nodes** are "**observed**", i.e. their values are given

Example:

$$P(X_2, X_5 \mid X_1 = 0, X_3 = 1, X_4 = 1)$$



Familiar Models as Bayesian Networks

Question:

Match the model name to the corresponding Bayesian Network

1. Logistic Regression
2. Linear Regression
3. Bernoulli Naïve Bayes
4. Gaussian Naïve Bayes
5. 1D Gaussian

Answer: