CSC 580 Principles of Machine Learning

10 Probabilistic graphical models: Naïve Bayes

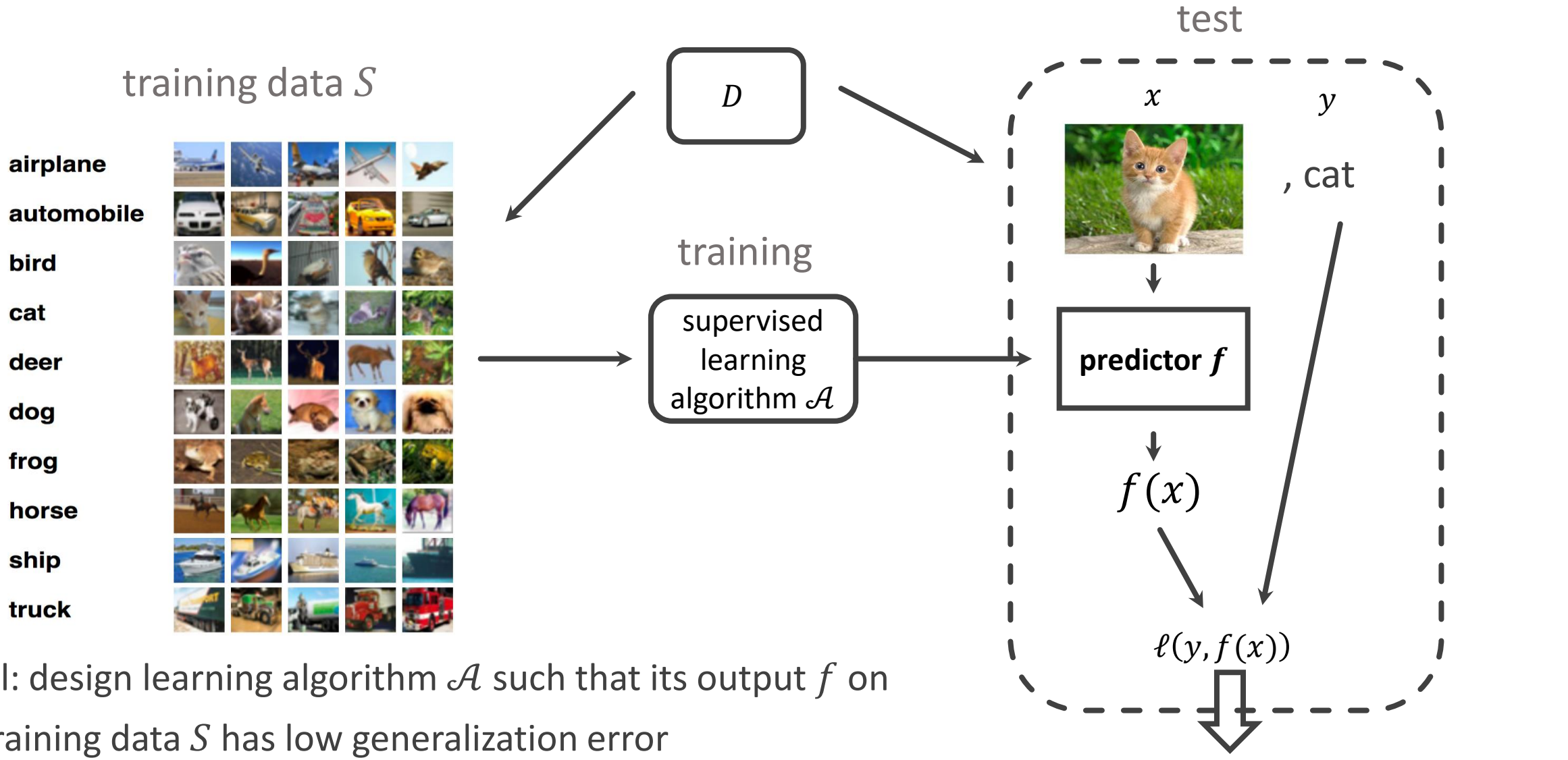
Chicheng Zhang

Department of Computer Science

THE UNIVERSITY OF ARIZONA

*slides credit: built upon CSC 580 Fall 2021 lecture slides by Kwang-Sung Jun

Recap: supervised learning setup

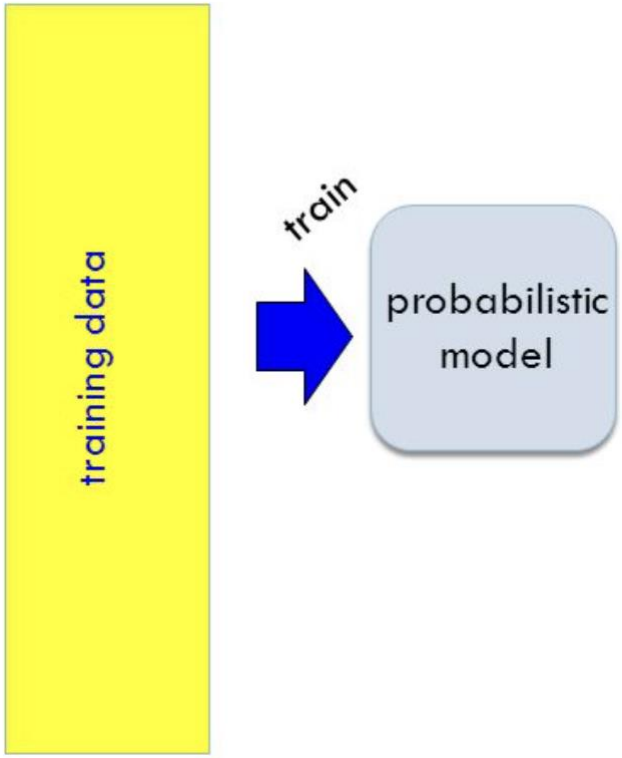


- Goal: design learning algorithm $\mathcal{A}$ such that its output f on iid training data S has low generalization error

Generalization error: $L_D(f) = \mathrm{E}_{(x,y)\sim D}\, \ell(y, f(x))$

Probabilistic modeling

- A systematic approach for machine learning
- Steps:
  1. Model how the data is generated by probabilistic models, but with parameters unspecified (modeling assumption / generative story)
     - Each example $z \sim P(z; \theta)$ for some $\theta \in \Theta$
       - For $z = (x, y)$ => supervised learning
       - For $z = x$ => unsupervised learning
  2. (Training) Learn the model parameter $\hat{\theta}$
     - Important example: maximum likelihood estimation (MLE), i.e., $\text{maximize}_{\theta \in \Theta} \log P(z_1, \ldots, z_n; \theta)$
  3. (Test) Make prediction / decision based on the learned model $P(z; \hat{\theta})$
     - Important example: predict using the Bayes classifier of $P(z; \hat{\theta})$ (for supervised learning)



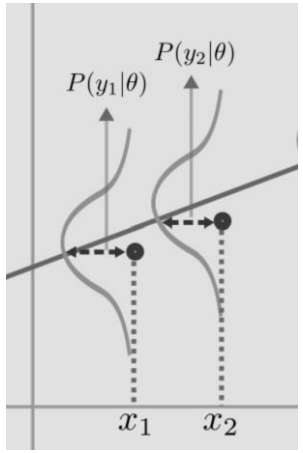
http://slideplayer.com/slide/4527958/

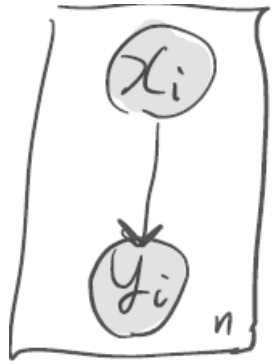
Probabilistic modeling (cont'd)

- Why probabilistic modeling?
  - Right thing to do if the model is correct
  - If not…
    - "All models are wrong, but some are useful" -- George E. P. Box
  - Interpretability
  - A view taken by classical statistics

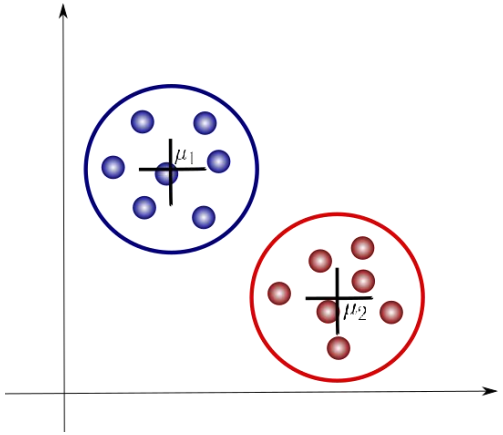
Discriminative vs Generative modeling

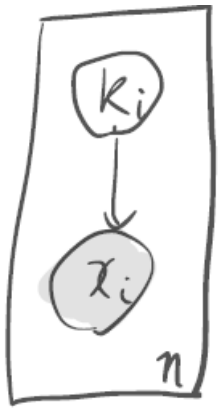
- Discriminative model: models only $P(y \mid x)$
  - Recall linear regression: $y \mid x; \theta \sim N(x^\top \theta, \sigma^2)$
  - Logistic regression: $y \mid x; \theta \sim \text{Bernoulli}(\sigma(x^\top \theta))$





- Generative model for unsupervised learning: models $P(x)$
  - e.g., Gaussian mixture model (GMM)
    - $\theta = (\pi_k, \mu_k, \Sigma_k)_{k=1}^K$
    - $k \sim \text{Categorical}(\pi)$ (*hidden*), i.e. $P(k = l) = \pi_l$
    - $x \mid k \sim N(\mu_k, \Sigma_k)$

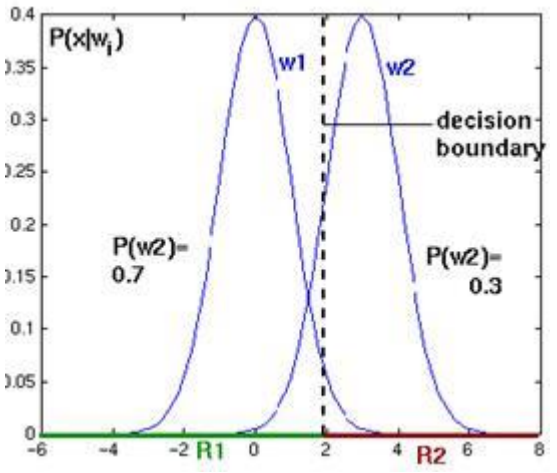


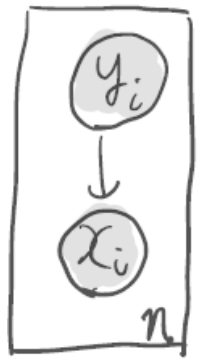


https://towardsdatascience.com/gaussian-mixture-models-explained-6986aaf5a95
https://suriyadeepan.github.io/2017-01-22-mle-linear-regression/

Discriminative vs Generative modeling (cont'd)

- Generative model for supervised learning:
  - models 'class-conditional' distribution $P(x \mid y)$
  - E.g. $\theta = (\pi_c, \mu_c, \Sigma_c)_{c=1}^{C}$,
  - $y \sim \text{Categorical}(\pi)$, $x \mid y = c \sim N(x; \mu_c, \Sigma_c)$



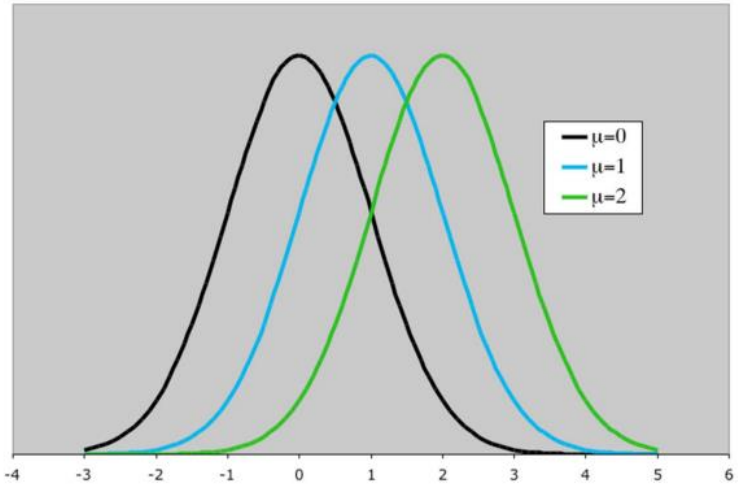


- Given test data from $P(x, y; \theta)$, what is the optimal classification rule?
  - $f_{BO,\theta}(x) = \text{argmax}_c P(y = c \mid x; \theta)$
  - Recall Bayes formula: $P(y = c \mid x) = \frac{P(x|y=c)P(y=c)}{P(x)}$, where $P(x) = \sum_c P(x \mid y = c)P(y = c)$
  - Therefore, $f_{BO,\theta}(x) = \text{argmax}_c \, P(x \mid y = c; \theta)P(y = c; \theta)$

https://www.byclb.com/TR/Tutorials/neural_networks/ch4_1.htm

Generative model: basic example I

- **Wait time at the barbershop**: Suppose you go to a barbershop at every last Friday of the month. You want to be able to predict the waiting time. You have collected 12 data points (i.e., how long it took to be served) from the last year: $S = \{x_1, \dots, x_{12}\}$
- 1. Modeling assumption: $x_i \sim$ Gaussian distribution $N(\mu, 1)$
  - $p(x; \mu) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2}\right)$
  - Observation: this distribution has mean μ



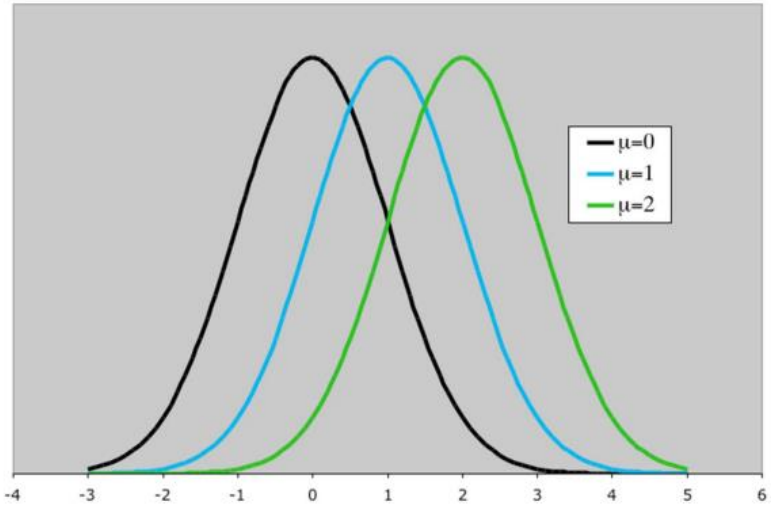
- 2. Find the MLE $\hat{\mu}$ from data S
  - (2.1) write down the neg. log likelihood of the sample (we'd like to minimize)

$$L_n(\mu) = -\ln P(x_1, \dots, x_n; \mu) = 12 \ln\sqrt{2\pi} + \frac{1}{2}\sum_{i=1}^{12}(x_i - \mu)^2$$

Generative model: basic example I (cont'd)

- 2. Find the MLE $\hat{\mu}$ from data S
  - (2.2) compute the first derivative, set it to 0, solve for λ (be sure to check convexity)

$$L'_n(\mu) = \sum_{i=1}^{12}(x_i - \mu) = 0 \Rightarrow \mu = \frac{x_1 + \cdots x_{12}}{12}$$



- 3. The learned model $N(\hat{\mu}, 1)$ is yours!
  - Simple prediction: e.g., predict the next wait time by $\mathbb{E}_{X \sim N(\hat{\mu},1)}[X]$
  - which is $\hat{\mu} = \frac{x_1 + \cdots x_{12}}{12}$

Generative modeling: basic example II

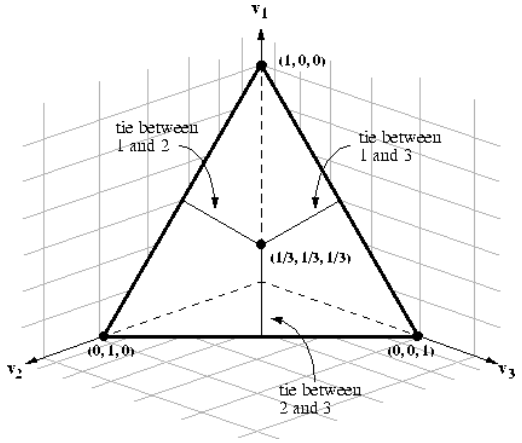
- [Data] $S = \{y_i\}_{i=1}^n$, where $y_i \in \{1, \dots, C\}$

- [Generative story]

  $y \sim \text{Categorical}(\pi)$, where $\pi = (\pi_1, \dots, \pi_C) \in \Delta^{C-1}$ ($\pi_c \geq 0$ and $\pi_1 + \cdots + \pi_C = 1$)

  e.g. y_i = the color of i-th ball drawn randomly from a bin (with replacement)

  $p(y; \pi) = \pi_y \left(= \prod_{c=1}^C \pi_c^{I(y=c)} \right)$



- [Training]

  (2.1) $L_n(\pi) = -\ln P(y_1, \dots, y_n; \pi) = \sum_{i=1}^n -\ln \pi_{y_i} = -\sum_{c=1}^C n_c \ln \pi_c$,

  where $n_c = \#\{i: y_i = c\} = \sum_{i=1}^n I(y_i = c)$

Generative modeling: basic example II (cont'd)

- [Training]

  (2.2) $\text{minimize}_{\pi \in \Delta^{C-1}} L_n(\pi) \coloneqq -\sum_{c=1}^{C} n_c \ln \pi_c$

  Constrained maximization problem; solve by Lagrange multipliers

$$\frac{\partial}{\partial \pi}\left(-\sum_{c=1}^{C} n_c \ln \pi_c - \lambda\left(\sum_{c=1}^{C} \pi_c - 1\right)\right) = -\frac{n_c}{\pi_c} - \lambda = 0 \Rightarrow \pi_c = -\frac{n_c}{\lambda}$$

Combined with the constraint that $\pi_1 + \cdots + \pi_C = 1 \Rightarrow \hat{\pi}_c = \frac{n_c}{n}$, for all c

- [Test] predict label $\text{argmax}_c P(y = c; \hat{\pi}) = \text{argmax}_c \hat{\pi}_c$

Naïve Bayes for supervised learning

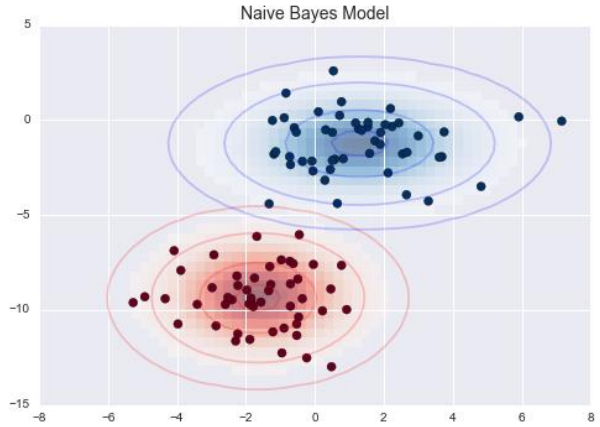
- Motivation: supervised learning for classification
- high-dimensional $x = (x(1), \dots, x(F))$, modeling $P(x \mid y)$ can be tricky
- In general, $P(x \mid y) = P(x(1) \mid y) \cdot P(x(2) \mid x(1), y) \cdot \dots \cdot P(x(F) \mid x(1), \dots, x(F-1), y)$

- A modeling assumption: $x(1), \dots, x(F)$ are conditionally independent given y

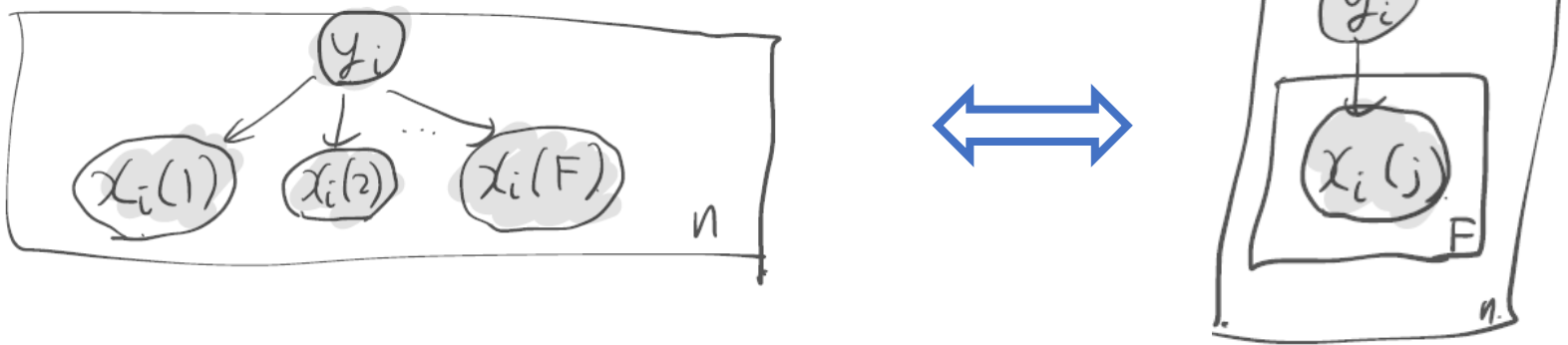
  i.e. for all i

$$x(i) \perp\!\!\!\perp \big(x(1), \dots, x(i-1), x(i+1), \dots, x(F)\big) \mid y$$

  (Conditional independence notation: $A \perp\!\!\!\perp B \mid C$)



- Equivalently $P(x \mid y) = P(x(1) \mid y) \cdot \dots P(x(F) \mid y)$



Naïve Bayes: binary-valued features

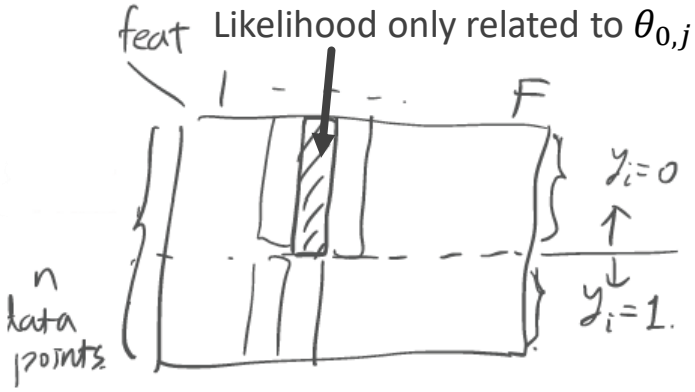
(recall the course recommendation example)

- [Data] $S = \{(x_i, y_i)\}_{i=1}^n$, $x_i \in \{0,1\}^F$ $y_i \in \{0,1\}$
- [Generative story]

  $y \sim \text{Bernoulli}(\pi)$; for all $j \in [F]$, $x(j) \mid y = c \sim \text{Bernoulli}(\theta_{c,j})$

  #parameters = $1 + 2F$

  [Training] (denote by $\theta = \{\theta_{c,j}\}$)



$$\max_{\pi,\theta} \sum_{i=1}^n \ln P(x_i, y_i; \pi, \theta) = \sum_{i=1}^n \ln P(y_i; \pi) + \sum_{i=1}^n \ln P(x_i \mid y_i; \theta)$$

$$= \max_{\pi} \sum_{i=1}^n \ln P(y_i; \pi) + \max_{\{\theta_{0,j}\}} \sum_{i:y_i=0} \ln P(x_i \mid y_i; \theta) + \max_{\{\theta_{1,j}\}} \sum_{i:y_i=1} \ln P(x_i \mid y_i; \theta)$$

Key observation: optimal π, optimal $\{\theta_{0,j}\}$, optimal $\{\theta_{1,j}\}$ can be found separately

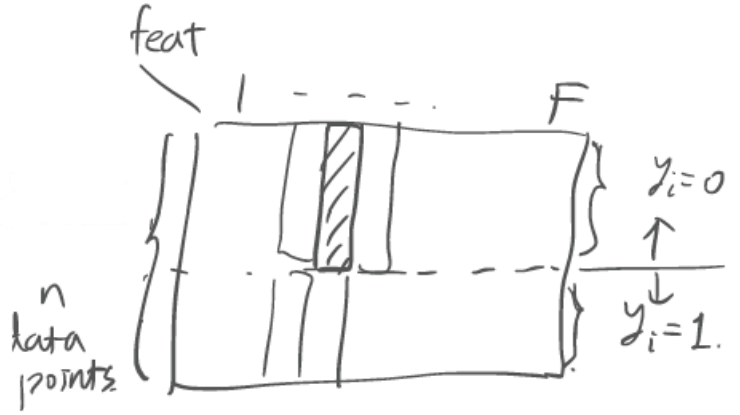
Optimal π: $\max_{\pi} \sum_{i=1}^n \ln P(y_i; \pi) = \max_{\pi} n_0 \ln(1-\pi) + n_1 \ln(\pi)$ => $\hat{\pi} = \frac{n_1}{n}$

Naïve Bayes: binary-valued features (cont'd)

- By the Naïve Bayes modeling assumption,

$$\max_{\{\theta_{0,j}\}} \sum_{i:y_i=0} \ln P(x_i \mid y_i;\theta) = \max_{\{\theta_{0,j}\}} \sum_{j=1}^{F} \sum_{i:y_i=0} \ln P\big(x_i(j) \mid y_i;\theta_{0,j}\big)$$

$$= \sum_{j=1}^{F} \max_{\theta_{0,j}} \sum_{i:y_i=0} \ln P\big(x_i(j) \mid y_i;\theta_{0,j}\big)$$



- Again, can optimize each $\theta_{0,j}$ separately

- Optimal $\theta_{0,j}$:

$$\max_{\theta_{0,j}} \sum_{i:y_i=0,\, x_i(j)=1} \ln \theta_{0,j} + \sum_{i:y_i=0,\, x_i(j)=0} \ln (1-\theta_{0,j})$$

- $\hat{\theta}_{0,j} = \frac{\#\{i:\, y_i=0,\, x_i(j)=1\}}{\#\{i:y_i=0\}}$; similarly, $\hat{\theta}_{1,j} = \frac{\#\{i:\, y_i=1,\, x_i(j)=1\}}{\#\{i:y_i=1\}}$

Naïve Bayes: binary-valued features (cont'd)

[Test] Given $\hat{\pi}$, $\{\hat{\theta}_{c,j}\}$, Bayes optimal classifier

$$\hat{f}_{BO}(x) = \text{argmax}_y \, P(x, y; \hat{\pi}, \{\hat{\theta}_{c,j}\}) = \text{argmax}_y \log P(x, y; \hat{\pi}, \{\hat{\theta}_{c,j}\})$$

- $\log P(x, y = 0; \pi, \{\theta_{c,j}\}) = \ln(1 - \pi) + \sum_{j=1}^{F} \ln P\big(x(j) \mid y; \theta_{0,j}\big)$

  $= \ln(1 - \pi) + \sum_{j=1}^{F} \ln\big(1 - \theta_{0,j}\big) I(x(j) = 0) + \ln\big(\theta_{0,j}\big) I(x(j) = 1)$

  $= \ln(1 - \pi) + \sum_{j=1}^{F} \ln(1 - \theta_{0,j}) + \sum_{j=1}^{F} x(j) \ln \frac{\theta_{0,j}}{1-\theta_{0,j}}$

- Similarly, $\log P(x, y = 1; \pi, \{\theta_{c,j}\}) = \ln(\pi) + \sum_{j=1}^{F} \ln(1 - \theta_{1,j}) + \sum_{j=1}^{F} x(j) \ln \frac{\theta_{1,j}}{1-\theta_{1,j}}$
- Therefore, $\hat{f}_{BO}(x) = 1 \Leftrightarrow \underbrace{\ln\left(\frac{\pi}{1-\pi}\right) + \sum_{j=1}^{F} \ln\left(\frac{1-\theta_{1,j}}{1-\theta_{0,j}}\right)}_{b} + \sum_{j=1}^{F} x(j) \underbrace{\left(\ln \frac{\theta_{1,j}}{1-\theta_{1,j}} - \ln \frac{\theta_{0,j}}{1-\theta_{0,j}}\right)}_{w(j)} \geq 0$
- I.e. Bayes classifier is linear

Naïve Bayes: Discrete (Categorical-valued features)

- [Data] $S = \{(x_i, y_i)\}_{i=1}^n$, $x_i \in [W]^F$ $y_i \in \{0,1\}$
- [Generative story]

  $y \sim \text{Bernoulli}(\pi)$; for all $j \in [F]$, $x(j) \mid y = c \sim \text{Categorical}(\theta_c)$ $(\theta_c \in \Delta^{W-1})$

  #parameters = $1 + 2W$

  Note: θ_c *shared across all features* in this example!
- [Training]

  Optimal π the same as before

  Optimal θ_c:

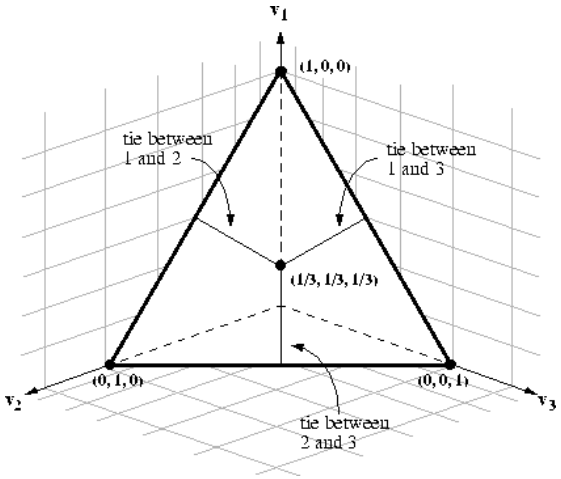
$$\max_{\theta_0} \sum_{i:y_i=0} \ln P(x_i \mid y_i; \theta_0) = \max_{\theta_0} \sum_{j=1}^F \sum_{i:y_i=0} \ln P(x_i(j) \mid y_i; \theta_0)$$

$$= \max_{\theta_0} \sum_{w=1}^W \sum_{j=1}^F \sum_{i:y_i=0} I(x_i(j) = w) \ln \theta_{0,w}$$

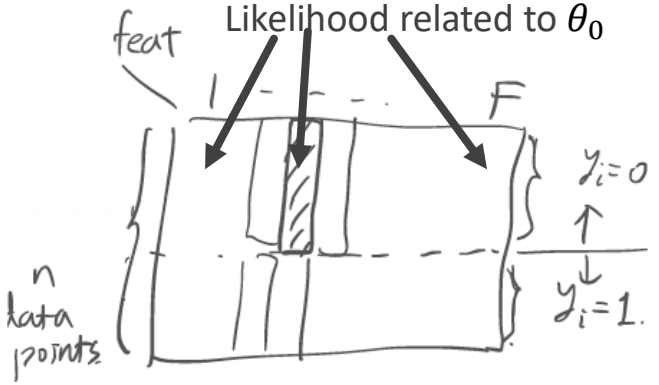
$$= \max_{\theta_0} \sum_{w=1}^W \ln \theta_{0,w} \, \#\{(i,j): y_i = 0, x_i(j) = w\}$$

  $\Rightarrow \hat{\theta}_{c,w} = \frac{\#\{(i,j): y_i=c, x_i(j)=w\}}{\#\{i: y_i=c\}}$

  Question: how to extend this to variable-length x_i's (e.g. for text classification)?
- [Test] Bayes optimal classification rule with $(\hat{\pi}, \hat{\theta}_0, \hat{\theta}_1)$ (exercise)



Likelihood related to θ_0



Probabilistic modeling: quick summary

- Now you see the flow:

  (1) Specify the generative story

  (2) design the maximum likelihood estimator

  => gives you a natural loss function for training

  (3) use the estimated model to make decisions

- Naïve Bayes: can mix different distributions for different features: $x(j) \mid y = c$
  - Bernoulli
  - Categorical
  - Continuous distributions (next)

Next lecture (10/17)

- Naïve Bayes with continuous feature distributions
- Midterm review
- Reading: CIML 9.4-9.7