EECS 545: Machine Learning
Lecture 5. Classification 2

Honglak Lee

1/27/2025

Last updated: 2/5/2025 12pm

Outline

- Probabilistic Discriminative models
  - Objective: maximize **conditional likelihood** over training data

$$\prod_i P(y^{(i)}|\mathbf{x}^{(i)}, \mathbf{w})$$

  - Logistic Regression (covered in previous lecture)
  - Softmax Regression: Multiclass extension of logistic regression
- Probabilistic Generative models
  - Objective: maximize **joint likelihood** over training data

$$\prod_i P(\mathbf{x}^{(i)}, y^{(i)}|\mathbf{w})$$

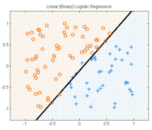
  - Gaussian Discriminant Analysis
  - Naive Bayes (part 1)

Softmax regression for multiclass classification

- For multiclass case, we can use softmax regression.
  - Softmax regression can be viewed as a generalization of logistic regression
- Recall that, logistic regression (binary classification) models class conditional probability as:

$$p(y=1 \mid \mathbf{x}; \mathbf{w}) = \frac{\exp\left(\mathbf{w}^\top \phi(\mathbf{x})\right)}{1+\exp\left(\mathbf{w}^\top \phi(\mathbf{x})\right)}$$

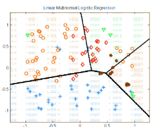
$$p(y=0 \mid \mathbf{x}; \mathbf{w}) = \frac{1}{1+\exp\left(\mathbf{w}^\top \phi(\mathbf{x})\right)}$$



  - Note that these probability sum to 1.
- For multiclass classification (with K classes), we use the following model

$$p(y=k \mid \mathbf{x}; \mathbf{w}) = \frac{\exp\left(\mathbf{w}_k^\top \phi(\mathbf{x})\right)}{1+\sum_{j=1}^{K-1} \exp\left(\mathbf{w}_j^\top \phi(\mathbf{x})\right)} \quad \text{for } k=\{1,\ldots,K-1\}$$

$$p(y=K \mid \mathbf{x}; \mathbf{w}) = \frac{1}{1+\sum_{j=1}^{K-1} \exp\left(\mathbf{w}_j^\top \phi(\mathbf{x})\right)} \quad \text{equivalent to setting } \mathbf{w}_K = 0$$



- Note that these probability sum to 1.

Softmax regression: Log-likelihood (objective function) and learning

- Defining $\mathbf{w}_K = 0$, we can write as:

$$p(y=k \mid \mathbf{x}; \mathbf{w}) = \frac{\exp\left(\mathbf{w}_k^\top \phi(\mathbf{x})\right)}{\sum_{j=1}^{K} \exp\left(\mathbf{w}_j^\top \phi(\mathbf{x})\right)}$$

or

$$p(y \mid \mathbf{x}; \mathbf{w}) = \prod_{k=1}^{K} \left[\frac{\exp\left(\mathbf{w}_k^\top \phi(\mathbf{x})\right)}{\sum_{j=1}^{K} \exp\left(\mathbf{w}_j^\top \phi(\mathbf{x})\right)}\right]^{\mathbb{I}(y=k)}$$

- Log-Likelihood

$$\log p(D|\mathbf{w}) = \sum_i \log p(y^{(i)}|\mathbf{x}^{(i)}, \mathbf{w})$$

$$= \sum_i \log \prod_{k=1}^{K} \left[\frac{\exp\left(\mathbf{w}_k^\top \phi(\mathbf{x}^{(i)})\right)}{\sum_{j=1}^{K} \exp\left(\mathbf{w}_j^\top \phi(\mathbf{x}^{(i)})\right)}\right]^{\mathbb{I}(y^{(i)}=k)}$$

- We can learn $\mathbf{w}$ by gradient ascent for maximizing the log-likelihood or iterative Newton's method (IRLS).

Probabilistic Generative Models

Learning the Classifier

- For classification, we want to compute $p(C_k \mid \mathbf{x})$
  - (a) **Discriminative** models: Directly model $p(C_k \mid \mathbf{x})$ and learn parameters from the training set.
    - Logistic regression
    - Softmax regression
  - (b) **Generative** models: Learn joint densities $p(\mathbf{x}, C_k)$ by learning $p(\mathbf{x} \mid C_k)$ and $p(C_k)$, and then use Bayes rule for predicting the class C_k given $\mathbf{x}$:
    - Gaussian Discriminant Analysis
    - Naive Bayes

Probabilistic Generative Models

- Bayes' theorem reduces the classification problem $p(C_k \mid \mathbf{x})$ to estimating the distribution of the data:

$$p(C_k|\mathbf{x}) = \frac{p(\mathbf{x}|C_k)p(C_k)}{p(\mathbf{x})} = \frac{p(\mathbf{x}|C_k)p(C_k)}{\sum_{k'} p(\mathbf{x}|C_{k'})p(C_{k'})}$$

- Density estimation can be decomposed into learning distributions from training data.
  - $p(C_k)$
  - $p(\mathbf{x} \mid C_k)$
- Maximum likelihood estimation for $p(\mathbf{x}, C_k)$

Probabilistic Generative Models

- For two classes, Bayes' theorem says:

$$p(C_1|\mathbf{x}) = \frac{p(\mathbf{x}|C_1)p(C_1)}{p(\mathbf{x}|C_1)p(C_1) + p(\mathbf{x}|C_2)p(C_2)}$$

- Use *log odds* (i.e., logit "score"):

$$a = \log \frac{p(C_1|\mathbf{x})}{p(C_2|\mathbf{x})} = \log \frac{p(\mathbf{x}|C_1)p(C_1)}{p(\mathbf{x}|C_2)p(C_2)}$$

- Then we can define the posterior via the *sigmoid*:

$$p(C_1|\mathbf{x}) = \frac{1}{1+\exp(-a)} = \sigma(a)$$

Gaussian Discriminant Analysis

Gaussian Discriminant Analysis

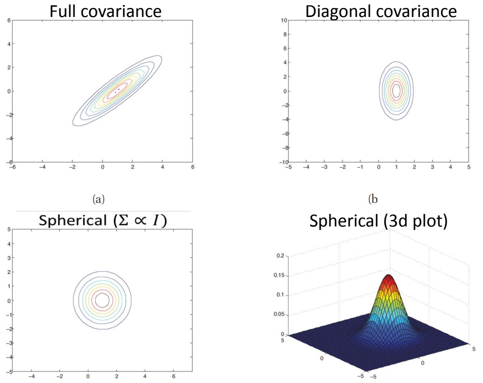
- Probability of class label
  - $p(C_k)$: Constant (e.g., Bernoulli)
- Conditional probability of data given a class
  - $p(\mathbf{x} \mid C_k)$: Gaussian distribution

$$p(\mathbf{x} \mid C_k) = \frac{1}{(2\pi)^{D/2}} \frac{1}{|\mathbf{\Sigma}|^{1/2}} \exp\left\{ -\frac{1}{2}(\mathbf{x}-\mu_k)^\top \mathbf{\Sigma}^{-1}(\mathbf{x}-\mu_k) \right\}$$

- Classification: use Bayes rule (previous slide)

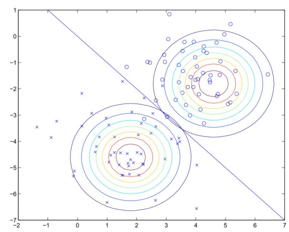
Examples of Gaussian Distributions

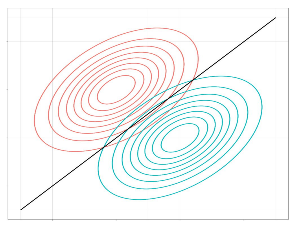
- Probability density p(x) for 2 dimensional case



Gaussian Discriminant Analysis

- Basic GDA assumes the same covariance for all classes
  - The figure below shows class-specific density and decision boundary. Note the linear decision boundary for any types of covariance matrices!





Prediction: Class-Conditional Densities

- Suppose we model $p(\mathrm{x} \mid C_k)$ as Gaussians with the <u>same covariance</u> matrix.

$$p(\mathbf{x} \mid C_k) = \frac{1}{(2\pi)^{D/2}} \frac{1}{|\mathbf{\Sigma}|^{1/2}} \exp\left\{ -\frac{1}{2}(\mathbf{x}-\mu_k)^\top \mathbf{\Sigma}^{-1}(\mathbf{x}-\mu_k) \right\}$$

- This gives us $p(C_1 \mid \mathbf{x}) = \sigma(\mathbf{w}^\top \mathbf{x} + w_0)$
  - where $\mathbf{w} = \mathbf{\Sigma}^{-1}(\mu_1 - \mu_2)$

  and $w_0 = -\frac{1}{2}\mu_1^\top \mathbf{\Sigma}^{-1}\mu_1 + \frac{1}{2}\mu_2^\top \mathbf{\Sigma}^{-1}\mu_2 + \log\frac{p(C_1)}{p(C_2)}$

Derivation

$$\begin{aligned} P(x, C_1) &= P(x \mid C_1)\, P(C_1) \\ &= \frac{1}{(2\pi)^{D/2}} \frac{1}{|\Sigma|^{1/2}} \exp\left\{ -\frac{1}{2}(x-\mu_1)^\top \Sigma^{-1}(x-\mu_1) \right\} P(C_1) \\ P(x, C_2) &= P(x \mid C_2)\, P(C_2) \\ &= \frac{1}{(2\pi)^{D/2}} \frac{1}{|\Sigma|^{1/2}} \exp\left\{ -\frac{1}{2}(x-\mu_2)^\top \Sigma^{-1}(x-\mu_2) \right\} P(C_2) \end{aligned}$$

Derivation

$$\begin{aligned} P(x, C_1) &= P(x \mid C_1)\, P(C_1) \\ &= \frac{1}{(2\pi)^{D/2}} \frac{1}{|\Sigma|^{1/2}} \exp\left\{ -\frac{1}{2}(x-\mu_1)^\top \Sigma^{-1}(x-\mu_1) \right\} P(C_1) \\ P(x, C_2) &= P(x \mid C_2)\, P(C_2) \\ &= \frac{1}{(2\pi)^{D/2}} \frac{1}{|\Sigma|^{1/2}} \exp\left\{ -\frac{1}{2}(x-\mu_2)^\top \Sigma^{-1}(x-\mu_2) \right\} P(C_2) \end{aligned}$$

$$\log\frac{P(C_1 \mid \mathbf{x})}{P(C_2 \mid \mathbf{x})} = \log\frac{P(C_1 \mid \mathbf{x})}{1 - P(C_1 \mid \mathbf{x})}$$ "Log-odds"

Derivation

$$\begin{aligned} P(x, C_1) &= P(x \mid C_1)\, P(C_1) \\ &= \frac{1}{(2\pi)^{D/2}} \frac{1}{|\Sigma|^{1/2}} \exp\left\{ -\frac{1}{2}(x-\mu_1)^\top \Sigma^{-1}(x-\mu_1) \right\} P(C_1) \\ P(x, C_2) &= P(x \mid C_2)\, P(C_2) \\ &= \frac{1}{(2\pi)^{D/2}} \frac{1}{|\Sigma|^{1/2}} \exp\left\{ -\frac{1}{2}(x-\mu_2)^\top \Sigma^{-1}(x-\mu_2) \right\} P(C_2) \end{aligned}$$

$$\log\frac{P(C_1 \mid \mathbf{x})}{P(C_2 \mid \mathbf{x})} = \log\frac{P(C_1 \mid \mathbf{x})}{1 - P(C_1 \mid \mathbf{x})}$$ "Log-odds"

$$= \log\frac{\exp\left\{-\frac{1}{2}(\mathbf{x}-\mu_1)^\top \Sigma^{-1}(\mathbf{x}-\mu_1)\right\}}{\exp\left\{-\frac{1}{2}(\mathbf{x}-\mu_2)^\top \Sigma^{-1}(\mathbf{x}-\mu_2)\right\}} + \log\frac{P(C_1)}{P(C_2)}$$

Derivation

$$
\begin{aligned}
P(x, C_1) &= P(x \mid C_1)\, P(C_1) \\
&= \frac{1}{(2\pi)^{D/2}} \frac{1}{|\Sigma|^{1/2}} \exp\left\{-\frac{1}{2}(x-\mu_1)^\top \Sigma^{-1}(x-\mu_1)\right\} P(C_1) \\
P(x, C_2) &= P(x \mid C_2)\, P(C_2) \\
&= \frac{1}{(2\pi)^{D/2}} \frac{1}{|\Sigma|^{1/2}} \exp\left\{-\frac{1}{2}(x-\mu_2)^\top \Sigma^{-1}(x-\mu_2)\right\} P(C_2)
\end{aligned}
$$

$$\log \frac{P(C_1 \mid \mathbf{x})}{P(C_2 \mid \mathbf{x})} = \log \frac{P(C_1 \mid \mathbf{x})}{1 - P(C_1 \mid \mathbf{x})} \qquad \text{“Log-odds”}$$

$$
\begin{aligned}
&= \log \frac{\exp\left\{-\frac{1}{2}(\mathbf{x}-\mu_1)^\top \Sigma^{-1}(\mathbf{x}-\mu_1)\right\}}{\exp\left\{-\frac{1}{2}(\mathbf{x}-\mu_2)^\top \Sigma^{-1}(\mathbf{x}-\mu_2)\right\}} + \log \frac{P(C_1)}{P(C_2)} \\
&= \left\{-\frac{1}{2}(\mathbf{x}-\mu_1)^\top \Sigma^{-1}(\mathbf{x}-\mu_1)\right\} - \left\{-\frac{1}{2}(\mathbf{x}-\mu_2)^\top \Sigma^{-1}(\mathbf{x}-\mu_2)\right\} + \log \frac{P(C_1)}{P(C_2)}
\end{aligned}
$$

Derivation

$$
\begin{aligned}
P(x, C_1) &= P(x \mid C_1)\, P(C_1) \\
&= \frac{1}{(2\pi)^{D/2}} \frac{1}{|\Sigma|^{1/2}} \exp\left\{-\frac{1}{2}(x-\mu_1)^\top \Sigma^{-1}(x-\mu_1)\right\} P(C_1) \\
P(x, C_2) &= P(x \mid C_2)\, P(C_2) \\
&= \frac{1}{(2\pi)^{D/2}} \frac{1}{|\Sigma|^{1/2}} \exp\left\{-\frac{1}{2}(x-\mu_2)^\top \Sigma^{-1}(x-\mu_2)\right\} P(C_2)
\end{aligned}
$$

$$\log \frac{P(C_1 \mid \mathbf{x})}{P(C_2 \mid \mathbf{x})} = \log \frac{P(C_1 \mid \mathbf{x})}{1 - P(C_1 \mid \mathbf{x})} \qquad \text{“Log-odds”}$$

$$
\begin{aligned}
&= \log \frac{\exp\left\{-\frac{1}{2}(\mathbf{x}-\mu_1)^\top \Sigma^{-1}(\mathbf{x}-\mu_1)\right\}}{\exp\left\{-\frac{1}{2}(\mathbf{x}-\mu_2)^\top \Sigma^{-1}(\mathbf{x}-\mu_2)\right\}} + \log \frac{P(C_1)}{P(C_2)} \\
&= \left\{-\frac{1}{2}(\mathbf{x}-\mu_1)^\top \Sigma^{-1}(\mathbf{x}-\mu_1)\right\} - \left\{-\frac{1}{2}(\mathbf{x}-\mu_2)^\top \Sigma^{-1}(\mathbf{x}-\mu_2)\right\} + \log \frac{P(C_1)}{P(C_2)} \\
&= (\mu_1-\mu_2)^\top \Sigma^{-1}\mathbf{x} - \frac{1}{2}\mu_1^\top \Sigma^{-1}\mu_1 + \frac{1}{2}\mu_2^\top \Sigma^{-1}\mu_2 + \log \frac{P(C_1)}{P(C_2)}
\end{aligned}
$$

Derivation

$$
\begin{aligned}
P(x, C_1) &= P(x \mid C_1)\, P(C_1) \\
&= \frac{1}{(2\pi)^{D/2}} \frac{1}{|\Sigma|^{1/2}} \exp\left\{-\frac{1}{2}(x-\mu_1)^\top \Sigma^{-1}(x-\mu_1)\right\} P(C_1) \\
P(x, C_2) &= P(x \mid C_2)\, P(C_2) \\
&= \frac{1}{(2\pi)^{D/2}} \frac{1}{|\Sigma|^{1/2}} \exp\left\{-\frac{1}{2}(x-\mu_2)^\top \Sigma^{-1}(x-\mu_2)\right\} P(C_2)
\end{aligned}
$$

$$\log \frac{P(C_1 \mid \mathbf{x})}{P(C_2 \mid \mathbf{x})} = \log \frac{P(C_1 \mid \mathbf{x})}{1 - P(C_1 \mid \mathbf{x})} \qquad \text{“Log-odds”}$$

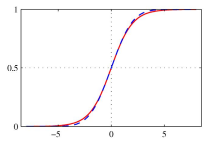
$$
\begin{aligned}
&= \log \frac{\exp\left\{-\frac{1}{2}(\mathbf{x}-\mu_1)^\top \Sigma^{-1}(\mathbf{x}-\mu_1)\right\}}{\exp\left\{-\frac{1}{2}(\mathbf{x}-\mu_2)^\top \Sigma^{-1}(\mathbf{x}-\mu_2)\right\}} + \log \frac{P(C_1)}{P(C_2)} \\
&= \left\{-\frac{1}{2}(\mathbf{x}-\mu_1)^\top \Sigma^{-1}(\mathbf{x}-\mu_1)\right\} - \left\{-\frac{1}{2}(\mathbf{x}-\mu_2)^\top \Sigma^{-1}(\mathbf{x}-\mu_2)\right\} + \log \frac{P(C_1)}{P(C_2)} \\
&= (\mu_1-\mu_2)^\top \Sigma^{-1}\mathbf{x} - \frac{1}{2}\mu_1^\top \Sigma^{-1}\mu_1 + \frac{1}{2}\mu_2^\top \Sigma^{-1}\mu_2 + \log \frac{P(C_1)}{P(C_2)} \\
&= \left(\Sigma^{-1}(\mu_1-\mu_2)\right)^\top \mathbf{x} + w_0
\end{aligned}
$$

$$\text{where } w_0 = -\frac{1}{2}\mu_1^\top \mathbf{\Sigma}^{-1}\mu_1 + \frac{1}{2}\mu_2^\top \mathbf{\Sigma}^{-1}\mu_2 + \log \frac{p(C_1)}{p(C_2)}$$

Prediction: Class-Conditional Densities for shared covariances

- $p(C_k \mid \mathbf{x})$ is a sigmoid function:

$$\sigma(a) = \frac{1}{1+\exp(-a)}$$



  - with log-odds (*logit* function):

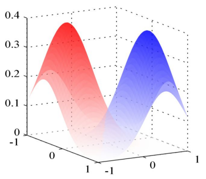
$$a = \log\left(\frac{\sigma}{1-\sigma}\right) = \left(\mathbf{\Sigma}^{-1}(\mu_1-\mu_2)\right)^\top \mathbf{x} + w_0$$

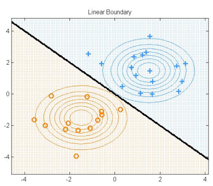
$$\text{where } w_0 = -\frac{1}{2}\mu_1^\top \mathbf{\Sigma}^{-1}\mu_1 + \frac{1}{2}\mu_2^\top \mathbf{\Sigma}^{-1}\mu_2 + \log \frac{p(C_1)}{p(C_2)}$$

- Generalizes to *normalized exponential*, or *softmax* : $p_i = \dfrac{\exp(q_i)}{\sum_j \exp(q_j)}$

Prediction: Linear Decision Boundaries

- At decision boundary, we have $p(C_1 \mid \mathrm{x}) = p(C_2 \mid \mathrm{x})$
- With the same covariance matrices, the boundary $p(C_1 \mid \mathrm{x}) = p(C_2 \mid \mathrm{x})$ is linear.
  - Different class $p(C_1)$, $p(C_2)$ just shift it around.





Likelihood function of generative models

- The likelihood of Data $\{(\mathbf{x}^{(n)}, y^{(n)})\}$

$$P(D|\mathbf{w}) = \prod_{i=1}^{N} P(\mathbf{x}^{(i)}, y^{(i)}|\mathbf{w}) \longrightarrow P(\mathbf{X}, \mathbf{y}|\mathbf{w})$$

Compact notation:
This is called joint likelihood.

Decomposition of the joint probability

$$= \prod_{i=1}^{N} P(\mathbf{x}^{(i)}|y^{(i)}, \mathbf{w}) P(y^{(i)}|\mathbf{w})$$

Learning parameters via maximum likelihood

- Given training data $\{(\mathbf{x}^{(1)}, y^{(1)}), \cdots, (\mathbf{x}^{(N)}, y^{(N)})\}$ and a generative model (“shared covariance”)

$$p(y) = \phi^y (1-\phi)^{1-y}$$

$$p(\mathbf{x}|y=0) = \frac{1}{(2\pi)^{D/2}|\Sigma|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(\mathbf{x}-\mu_0)^\top \Sigma^{-1}(\mathbf{x}-\mu_0)\right)$$

$$p(\mathbf{x}|y=1) = \frac{1}{(2\pi)^{D/2}|\Sigma|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(\mathbf{x}-\mu_1)^\top \Sigma^{-1}(\mathbf{x}-\mu_1)\right)$$

Learning via maximum likelihood

- Maximum likelihood estimation (HW2):

$$\phi = \frac{1}{N}\sum_{i=1}^{N} \mathbb{I}\{y^{(i)} = 1\}$$

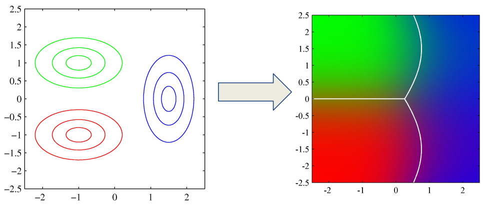
$$\mu_0 = \frac{\sum_{i=1}^{N} \mathbb{I}\{y^{(i)} = 0\}\mathbf{x}^{(i)}}{\sum_{i=1}^{N} \mathbb{I}\{y^{(i)} = 0\}}$$

$$\mu_1 = \frac{\sum_{i=1}^{N} \mathbb{I}\{y^{(i)} = 1\}\mathbf{x}^{(i)}}{\sum_{i=1}^{N} \mathbb{I}\{y^{(i)} = 1\}}$$

$$\Sigma = \frac{1}{N}\sum_{i=1}^{N} (\mathbf{x}^{(i)} - \mu_{y^{(i)}})(\mathbf{x}^{(i)} - \mu_{y^{(i)}})^\top$$

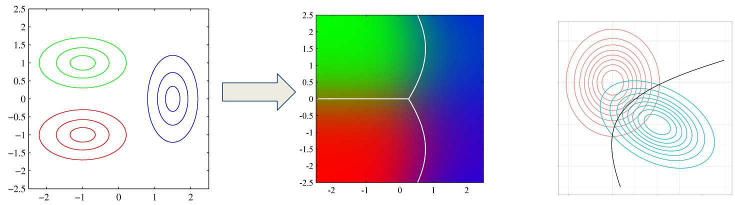
Different Covariance

- Decision boundaries between some classes can be quadratic when they have **different** covariances.



Different Covariance

- Decision boundaries between some classes can be quadratic when they have **different** covariances.



Comparison between GDA and Logistic regression (or softmax regression)

- Logistic regression:
  - For an M-dimensional feature space, this model has M parameters to fit.
- Gaussian Discriminative Analysis
  - $2M$ parameters for the means of $p(\mathbf{x} \mid C_1)$ and $p(\mathbf{x} \mid C_2)$
  - $M(M+1)/2$ parameters for the shared covariance matrix
- Logistic regression has less parameters and is more flexible about data distribution.
- GDA has a stronger modeling assumption, and works well when the distribution follows the assumption.

Naive Bayes Classifier

(Brief Intro: to be continued in the next lecture)

Naive Bayes classifier

- Probability of class label:
  - $p(C_k)$: Constant (e.g., Bernoulli)
- Conditional probability of data given the class
  - Naive Bayes assumption: $p(\mathbf{x} \mid C_k)$ is factorized
    (Each coordinate of $\mathbf{x}$ is conditionally independent of other coordinates given the class label)

$$P(x_1, ..., x_M|C_k) = P(x_1|C_k) \cdots P(x_M|C_k) = \prod_{j=1}^{M} P(x_j|C_k)$$

- Classification: use Bayes rule

(binary) $$P(C_1|\mathbf{x}) = \frac{P(C_1, \mathbf{x})}{P(\mathbf{x})} = \frac{P(C_1, \mathbf{x})}{P(C_1, \mathbf{x}) + P(C_2, \mathbf{x})}$$

Naive Bayes classifier

- When classifying, we can simply find the class C_k that maximizes $P(C_k|\mathbf{x})$ using the Bayes rule:

$$\arg\max_k P(C_k|\mathbf{x}) = \arg\max_k P(C_k, \mathbf{x})$$

Naive Bayes classifier

- When classifying, we can simply find the class C_k that maximizes $P(C_k|\mathbf{x})$ using the Bayes rule:

$$\begin{aligned}\arg\max_k P(C_k|\mathbf{x}) &= \arg\max_k P(C_k, \mathbf{x}) \\ &= \arg\max_k P(C_k) P(\mathbf{x}|C_k)\end{aligned}$$

Naive Bayes classifier

- When classifying, we can simply find the class C_k that maximizes $P(C_k|\mathbf{x})$ using the Bayes rule:

$$\begin{aligned}\arg\max_k P(C_k|\mathbf{x}) &= \arg\max_k P(C_k, \mathbf{x}) \\ &= \arg\max_k P(C_k) P(\mathbf{x}|C_k) \\ &= \arg\max_k P(C_k) \prod_{j=1}^{M} P(x_j|C_k)\end{aligned}$$

Naive Bayes assumption

Example: Naive Bayes for real-valued inputs

- Probability of class label:
  - $p(C_k)$: Constant (e.g., Bernoulli)
- Conditional probability of data given the class
  - Naive Bayes assumption: $P(\mathbf{x}|C_k)$ is factorized (e.g., 1D Gaussian)

$$\begin{aligned} P(x_1, \ldots, x_M|C_k) &= P(x_1|C_k) \cdots P(x_M|C_k) \\ &= \prod_{j=1}^{M} P(x_j|C_k) \\ &= \prod_{j=1}^{M} \frac{1}{\sqrt{2\pi\sigma_j^2}} \exp\left(-\frac{(x_j-\mu_j)^2}{2\sigma_j^2}\right) \end{aligned}$$

  - Note: this is equivalent to GDA with diagonal covariance!!

Comparison: Discriminative vs. Generative

- The *generative* approach is typically model-based, and it can generate synthetic data from $p(\mathbf{x}|\ C_k)$.
  - By comparing the synthetic data and real data, we get a sense of how good the generative model is.
- The *discriminative* approach will typically have fewer parameters to estimate and have less assumptions about data distribution.
  - Linear (e.g. logistic regression) v/s quadratic (e.g., Gaussian discriminant analysis) in the dimension of the input.
  - Less generative assumptions about the data (however, constructing the features may require domain knowledge)

Any feedback (about lecture, slide, homework, project, etc.)?
(via **anonymous** google form: https://forms.gle/fpYmiBtG9Me5qbP37)

Change Log of lecture slides:
https://docs.google.com/document/d/e/2PACX-1vSSIHJjklypK7rKFSR1-5GYXyBCEW8UPtpSfCR9AR6M1l7K9ZQEmxfFwaWaW7kLDxusthsF8WlCyZJ-/pub