Georgia Tech College of Computing

CS4641 Spring 2025

Naive Bayes Classifier:

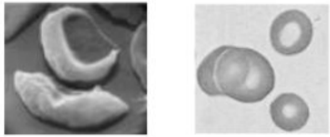
Discriminative vs. Generative Classifier

Bo Dai
School of CSE, Georgia Tech
bodai@cc.gatech.edu

Classification Tasks

	Feature, X	Label, Y	

Diagnosing sickle cell anemia



Anemic cell
Healthy cell

$$Y \in \{0, 1\}$$

Tax Fraud Detection

Refund	Marital Status	Taxable Income
No	Married	80K

Cheat
?

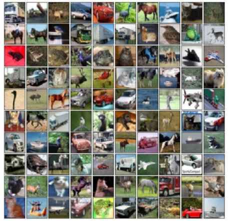
$$Y \in \{0, 1\}$$

Web Classification

Sports
Science
News

$$Y \in \{0, 1, 2, \ldots, k\}$$

Image Classification



Airplane
Automobile
Bird
...

$$Y \in \{0, 1, 2, \ldots, k\}$$

ML Algorithm Pipeline

General ML Algorithm Pipeline

1. Build probabilistic models
2. Derive loss function (by MLE or MAP….)
3. Select optimizer

Multiclass Logistic Regression Algorithms

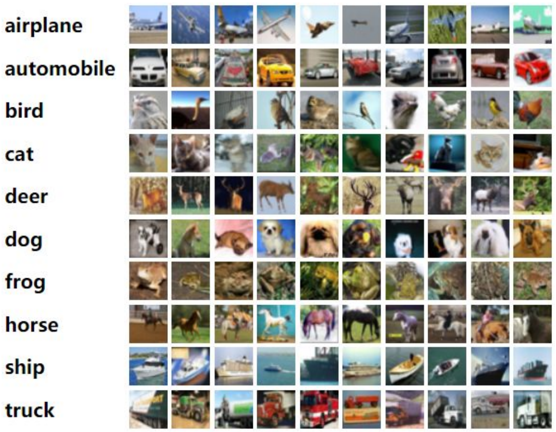
Training Data

$\{x^i, y^i\}_{i=1}^m$

Learning Algorithm

Classifier

$f : X \to Y$



Multiclass Classification $Y \in \{0, 1, \ldots, k\}$

Multiclass Logistic Regression Algorithms

Training Data

$\{x^i, y^i\}_{i=1}^m$

Learning Algorithm

Classifier

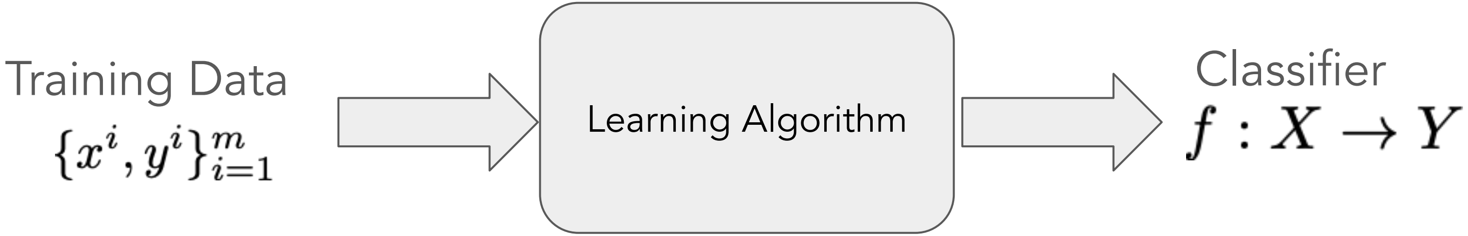
$f : X \to Y$

Multiclass Classification $Y \in \{0, 1, \ldots, k\}$

Multiclass Logistic Regression Pipeline

1. Build probabilistic models:
   Categorical Distribution + Linear Model
2. Derive loss function: MLE and MAP
3. Select optimizer: (Stochastic) Gradient Descent

Naive Bayes Classification



Multiclass Classification $\quad Y \in \{0, 1, \ldots, k\}$

Gaussian Naive Bayes Pipeline

1. Build probabilistic models
2. Derive loss function (by MLE or MAP)
3. Select optimizer

Bayes' Rule

Softmax in Multiclass Classification

$$p(y_i = 1|\theta_i^\top x) = \frac{\exp(\theta_i^\top x)}{\sum_{i=1}^k \exp(\theta_i^\top x)}$$

$$P(y|x) = \frac{P(x|y)P(y)}{P(x)} = \frac{P(x,y)}{\sum_z P(x,y)}$$

Bayes' Rule

$$P(y|x) = \frac{P(x|y)P(y)}{P(x)} = \frac{P(x,y)}{\sum_z P(x,y)}$$

Prior: $P(y) \quad \pi = (\pi_1, \pi_2, \ldots, \pi_k), \quad \sum_{i=1}^{k} \pi_i = 1, \pi_i \geq 0$

Likelihood (class conditional distribution : $p(x|y) = \mathcal{N}(x|\mu_y, \Sigma_y)$

Posterior: $P(y|x) = \dfrac{P(y)\mathcal{N}(x|\mu_y, \Sigma_y)}{\sum_y P(y)\mathcal{N}(x|\mu_y, \Sigma_y)}$

Decision with Bayes' Rule

- The posterior probability of a test point

$$q_i(x) := P(y = i|x) = \frac{P(x|y)P(y)}{P(x)}$$

- Bayes decision rule:
  - If $q_i(x) > q_j(x)$, then $y = i$, otherwise $y = j$

- Alternatively:
  - If ratic $l(x) = \frac{P(x|y=i)}{P(x|y=j)} > \frac{P(y=j)}{P(y=i)}$, then $y = i$, otherwise $y = j$
  - Or look at the log-likelihood ratio $h(x) = \ln \frac{q_i(x)}{q_j(x)}$

Naive Bayes Classifier

Training Data $\{x^i, y^i\}_{i=1}^{m}$ → Learning Algorithm → Classifier $f : X \rightarrow Y$

Multiclass Classification $Y \in \{0, 1, \ldots, k\}$

General ML Algorithm Pipeline

1. Build probabilistic models
2. Derive loss function (by MLE or MAP)
3. Select optimizer

MLE of Naive Bayes Classifier

$$\theta = [\mu, \Sigma, \pi], \quad Z = \sqrt{(2\pi)^D \det(\Sigma)}$$

$$p(x^i | y^i_j = 1, \theta) = \frac{1}{Z} \exp\left(-\frac{1}{2}(x^i - \mu_j)^\top \Sigma_j^{-1} (x^i - \mu_j) \right)$$

$$P(y) \qquad \pi = (\pi_1, \pi_2, \ldots, \pi_k), \quad \sum_{i=1}^{k} \pi_i = 1, \pi_i \geq 0$$

MLE of Naive Bayes Classifier

$$\theta = [\mu, \Sigma, \pi], \quad Z = \sqrt{(2\pi)^D \det(\Sigma)}$$

$$p(x^i | y^i_j = 1, \theta) = \frac{1}{Z} \exp\left(-\frac{1}{2}(x^i - \mu_j)^\top \Sigma_j^{-1} (x^i - \mu_j)\right)$$

$$P(y) \qquad \pi = (\pi_1, \pi_2, \ldots, \pi_k), \quad \sum_{i=1}^{k} \pi_i = 1, \pi_i \geq 0$$

$$\log L(\theta) = \log p(x, y | \theta)$$

MLE of Naive Bayes Classifier

$$\theta = [\mu, \Sigma, \pi], \quad Z = \sqrt{(2\pi)^D \det(\Sigma)}$$

$$p(x^i | y_j^i = 1, \theta) = \frac{1}{Z} \exp\left(-\frac{1}{2}(x^i - \mu_j)^\top \Sigma_j^{-1} (x^i - \mu_j)\right)$$

$$\log L(\theta) = \log p(x, y|\theta) = \log p(y|\theta) + \log p(x|y, \theta)$$

$$\log L(\theta) = \sum_{i=1}^{N} \sum_{j=1}^{k} y_j^i \log \pi_j - \sum_{i=1}^{N} \log Z - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{k} y_j^i (x^i - \mu_j)^\top \Sigma_j^{-1} (x^i - \mu_j)$$

Want $\arg\max_{\theta} \log L(\theta)$ subject to $\sum_{j=1}^{k} \pi_j = 1$

MAP of Naive Bayes Classifier

$$p(\theta) \propto \exp(-\lambda\|\theta\|_2^2)$$

$$\log L(\theta) = \log p(x, y|\theta) = \log p(y|\theta) + \log p(x|y, \theta)$$

$$\max_{\theta} \log p(\theta|\{x^i, y^i\}_{i=1}^m) = \log L(\theta) + \log p(\theta)$$

Naive Bayes Classifier

Training Data $\{x^i, y^i\}_{i=1}^m$ $\Rightarrow$ Learning Algorithm $\Rightarrow$ Classifier $f : X \to Y$

Multiclass Classification $Y \in \{0, 1, \ldots, k\}$

General ML Algorithm Pipeline

1. Build probabilistic models
2. Derive loss function (by MLE or MAP)
3. Select optimizer

Select Optimizer

$$\sum_{i=1}^{N}\sum_{j=1}^{k} y_j^i \log \pi_j - \log Z - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{k} y_j^i (x^i - \mu_j)^\top \Sigma_j^{-1} (x^i - \mu_j)$$

- Necessary Condition
- (Stochastic) Gradient Descent

Gradient Calculation of MLE

$$Z = \sqrt{(2\pi)^D \det(\Sigma)}$$

Take derivative w.r.t μ_k

$$\sum_{i=1}^{N}\sum_{j=1}^{k} y_j^i \log \pi_j - \log Z - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{k} y_j^i (x^i - \mu_j)^\top \Sigma_j^{-1}(x^i - \mu_j)$$

$$\frac{\partial \log L}{\partial \mu_k} = \sum_{i=1}^{N} y_k^i \Sigma_k^{-1}(x^i - \mu_k) = 0$$

$$\frac{\partial}{\partial \mathbf{s}}(\mathbf{x}-\mathbf{s})^T\mathbf{W}(\mathbf{x}-\mathbf{s}) = -2\mathbf{W}(\mathbf{x}-\mathbf{s})$$

Gradient Calculation of MLE

$$Z = \sqrt{(2\pi)^D \det(\Sigma)}$$

Take derivative w.r.t μ_k

$$\sum_{i=1}^{N}\sum_{j=1}^{k} y_j^i \log \pi_j - \log Z - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{k} y_j^i (x^i - \mu_j)^\top \Sigma_j^{-1}(x^i - \mu_j)$$

$$\frac{\partial \log L}{\partial \mu_k} = \sum_{i=1}^{N} y_k^i \Sigma_k^{-1}(x^i - \mu_k) = 0$$

$$\boxed{\mu_k = \frac{\sum_{i=1}^{N} y_k^i x^i}{\sum_{i=1}^{N} y_k^i}}$$

Necessary Condition for MLE

Take derivative w.r.t Σ_k^{-1} (not Σ_k)

$$Z = \sqrt{(2\pi)^D \det(\Sigma)}$$

Note:

$$\frac{\partial \log L}{\partial \Sigma_k^{-1}} = -\sum_{i=1}^{N} y_k^i \left[-\frac{\partial \log Z_k}{\partial \Sigma_k^{-1}} - \frac{1}{2}(x^i - \mu_k)(x^i - \mu_k)^\top \right] = 0$$

$$\frac{\partial \det(A)}{\partial A} = \det(A) A^{-\top}$$

$$\frac{\partial \log Z_k}{\partial \Sigma_k^{-1}} = \frac{1}{Z_k}\frac{\partial Z_k}{\partial \Sigma_k^{-1}} = (2\pi)^{-D/2} \det(\Sigma_k)^{-1/2} (2\pi)^{D/2} \frac{\partial \det(\Sigma_k^{-1})^{-1/2}}{\partial \Sigma_k^{-1}}$$

$$= \det(\Sigma_k^{-1})^{1/2} \left(-\frac{1}{2}\right) \det(\Sigma_k^{-1})^{-3/2} \det(\Sigma_k^{-1}) \Sigma_k^\top = -\frac{1}{2}\Sigma_k$$

$$\det(A^{-1}) = \det(A)^{-1}$$

$$\frac{\partial x^\top A x}{\partial A} = x x^\top$$

$$\Sigma^\top = \Sigma$$

Necessary Condition for MLE

Take derivative w.r.t Σ_k^{-1} (not Σ_k)

Note:

$$\frac{\partial \log L}{\partial \Sigma_k^{-1}} = -\sum_{i=1}^{N} y_k^i \left[-\frac{\partial \log Z_k}{\partial \Sigma_k^{-1}} - \frac{1}{2}(x^i - \mu_k)(x^i - \mu_k)^\top \right] = 0$$

$$\boxed{\Sigma_k = \frac{\sum_{i=1}^{N} y_k^i (x^i - \mu_k)(x^i - \mu_k)^\top}{\sum_{i=1}^{N} y_k^i}}$$

Necessary Condition for MLE

Take derivative w.r.t Σ_k^{-1} (not Σ_k)

Note:

$$\frac{\partial \det(A)}{\partial A} = \det(A) A^{-\top}$$

$$\det(A^{-1}) = \det(A)^{-1}$$

$$\frac{\partial x^\top A x}{\partial A} = x x^\top$$

$$\Sigma^\top = \Sigma$$

$$\frac{\partial \log L}{\partial \Sigma_k^{-1}} = -\sum_{i=1}^{N} y_k^i \left[-\frac{\partial \log Z_k}{\partial \Sigma_k^{-1}} - \frac{1}{2}(x^i - \mu_k)(x^i - \mu_k)^\top \right] = 0$$

Necessary Condition for MLE

$$Z_k = \sqrt{(2\pi)^D \det(\Sigma_k)}$$

$$\frac{\partial \log Z_k}{\partial \Sigma_k^{-1}} = \frac{1}{Z_k}\frac{\partial Z_k}{\partial \Sigma_k^{-1}} = (2\pi)^{-D/2}\det(\Sigma_k)^{-1/2}(2\pi)^{D/2}\frac{\partial \det(\Sigma_k^{-1})^{-1/2}}{\partial \Sigma_k^{-1}}$$

$$= \det(\Sigma_k^{-1})^{1/2}\left(-\frac{1}{2}\right)\det(\Sigma_k^{-1})^{-3/2}\det(\Sigma_k^{-1})\Sigma_k^{\top} = -\frac{1}{2}\Sigma_k$$

$$\frac{\partial \log L}{\partial \Sigma_k^{-1}} = -\sum_{i=1}^{N} y_k^i \left[\frac{1}{2}\Sigma_k - \frac{1}{2}(x^i - \mu_k)(x^i - \mu_k)^{\top}\right] = 0$$

$$\boxed{\Sigma_k = \frac{\sum_{i=1}^N y_k^i (x^i - \mu_k)(x^i - \mu_k)^{\top}}{\sum_{i=1}^N y_k^i}}$$

Necessary Condition for MLE

Use Lagrange multiplier to derive π_k $$\sum_{i=1}^{N}\sum_{j=1}^{k} y_j^i \log \pi_j - \log Z - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{k} y_j^i (x^i - \mu_j)^\top \Sigma_j^{-1} (x^i - \mu_j)$$

$$\frac{\partial L(\theta)}{\partial \pi_k} + \lambda \frac{\partial \sum_k \pi_k}{\partial \pi_k} = 0 \Rightarrow \lambda = -\sum_{i=1}^{N} y_k^i \frac{1}{\pi_k}$$

$$\pi_k = -\frac{\sum_{i=1}^{N} y_k^i}{\lambda}$$

Apply constraint: $\sum_k \pi_k = 1 \Rightarrow \lambda = -N$

$$\boxed{\pi_k = \frac{\sum_{i=1}^{N} y_k^i}{N}}$$

Naive Bayes Classifier

Training Data $\{x^i, y^i\}_{i=1}^m$ → Learning Algorithm → Classifier $f : X \to Y$

Multiclass Classification $Y \in \{0, 1, \ldots, k\}$

Gaussian Naive Bayes Pipeline

1. Build probabilistic models: Gaussian Likelihood
2. Derive loss function: MLE or MAP
3. Select optimizer: Necessary Condition

Discriminative vs. Generative Classifier

$$P(y|x) = \frac{P(x|y)P(y)}{P(x)} = \frac{P(x,y)}{\sum_y P(x,y)}$$

Discriminative

- Directly estimate decision boundary $h(x) = -\ln \frac{q_i(x)}{q_j(x)}$ or posterior distribution $p(y|x)$
- $h(x)$ or $f(x) := p(y=1|x)$ is a function of x, and
  - Does not have probabilistic meaning
  - Hence can not be used to sample data points

Generative

- Estimate the probabilistic generative mechanism $P(x|y)P(y)$
- Derive decision boundary through Bayes' rule

Discriminative vs. Generative Classifier

Binary
Logistic Regression

$$p(y=1|x,\theta) = \frac{1}{1+\exp(-\theta^\top x)}$$

$$p(y=0|x,\theta) = 1 - \frac{1}{1+\exp(-\theta^\top x)} = \frac{\exp(-\theta^\top x)}{1+\exp(-\theta^\top x)}$$

Gaussian
Naive Bayes Classifier

$$P(y|x) = \frac{P(x|y)P(y)}{P(x)} = \frac{P(x,y)}{\sum_y P(x,y)}$$

$$= \frac{P(y)\mathcal{N}(x|\mu_y,\Sigma_y)}{\sum_y P(y)\mathcal{N}(x|\mu_y,\Sigma_y)}$$

Posterior of Gaussian Naive Bayes Classifier

$$\frac{p(x, y=1)}{p(x, y=0) + p(x, y=1)} = \frac{\pi_1 \mathcal{N}(x|\mu_1, \Sigma)}{\pi_0 \mathcal{N}(x|\mu_0, \Sigma) + \pi_1 \mathcal{N}(x|\mu_1, \Sigma)}$$

Posterior of Gaussian Naive Bayes Classifier

$$\frac{p(x, y=1)}{p(x, y=0) + p(x, y=1)} = \frac{\pi_1 \mathcal{N}(x|\mu_1, \Sigma)}{\pi_0 \mathcal{N}(x|\mu_0, \Sigma) + \pi_1 \mathcal{N}(x|\mu_1, \Sigma)}$$

$$= \left\{1 + \frac{\pi_0}{\pi_1} \exp\left[-\frac{1}{2}(x-\mu_0)^\top \Sigma^{-1}(x-\mu_0) + \frac{1}{2}(x-\mu_1)^\top \Sigma^{-1}(x-\mu_1)\right]\right\}^{-1}$$

$$= \left\{1 + \exp\left[\log \frac{\pi_0}{\pi_1} + (\mu_0 - \mu_1)^\top \Sigma^{-1} x + \frac{1}{2}(\mu_0^\top \Sigma^{-1} \mu_0 - \mu_1^\top \Sigma^{-1} \mu_1)\right]\right\}^{-1}$$

$$= \frac{1}{1 + \exp(-\theta^\top x - b)}$$

Posterior of Gaussian Naive Bayes Classifier

$$\frac{p(x, y=1)}{p(x, y=0) + p(x, y=1)} = \frac{\pi_1 \mathcal{N}(x|\mu_1, \Sigma)}{\pi_0 \mathcal{N}(x|\mu_0, \Sigma) + \pi_1 \mathcal{N}(x|\mu_1, \Sigma)}$$

Decision Boundary Gaussian Naive Bayes Classifier

$$p(x, y = 0) = p(x, y = 1)$$

$$\log \pi_1 - \frac{1}{2}(x - \mu_1)^\top \Sigma_1^{-1}(x - \mu_1) = \log \pi_0 - \frac{1}{2}(x - \mu_0)^\top \Sigma_0^{-1}(x - \mu_0)$$

$$x^\top (\Sigma_1^{-1} - \Sigma_0^{-1})x - 2\left(\mu_1^\top \Sigma_1^{-1} - \mu_0^\top \Sigma_0^{-1}\right)x + \left(\mu_0^\top \Sigma_0^{-1} \mu_0 - \mu_1^\top \Sigma_1^{-1} \mu_1\right) = C$$

$$\Rightarrow x^\top Q x - 2b^\top x + c = 0$$

The decision boundary is a quadratic function. In 2-d case, it looks like an ellipse, or a parabola, or a hyperbola.

Decision with Bayes' Rule

- The posterior probability of a test point

$$q_i(x) := P(y = i|x) = \frac{P(x|y)P(y)}{P(x)}$$

- Bayes decision rule:
  - If $q_i(x) > q_j(x)$, then $y = i$, otherwise $y = j$

- Alternatively:
  - If ratic $l(x) = \frac{P(x|y=i)}{P(x|y=j)} > \frac{P(y=j)}{P(y=i)}$, then $y = i$, otherwise $y = j$
  - Or look at the log-likelihood ratio $h(x) = \ln \frac{q_i(x)}{q_j(x)}$

Decision Boundary Gaussian Naive Bayes Classifier

$$p(x, y = 0) = p(x, y = 1)$$

Decision Boundary Gaussian Naive Bayes Classifier

$$p(x, y=0) = p(x, y=1)$$

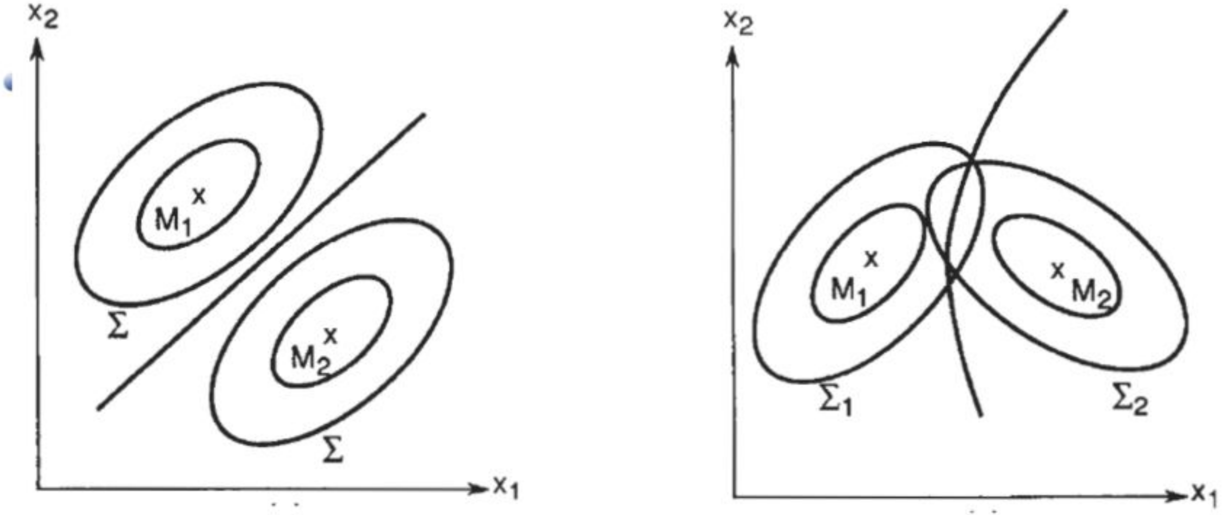
$$\log \pi_1 - \frac{1}{2}(x-\mu_1)^\top \Sigma_1^{-1}(x-\mu_1) = \log \pi_0 - \frac{1}{2}(x-\mu_0)^\top \Sigma_0^{-1}(x-\mu_0)$$

$$x^\top(\Sigma_1^{-1} - \Sigma_0^{-1})x - 2\left(\mu_1^\top \Sigma_1^{-1} - \mu_0^\top \Sigma_0^{-1}\right)x + \left(\mu_0^\top \Sigma_0^{-1}\mu_0 - \mu_1^\top \Sigma_1^{-1}\mu_1\right) = C$$

$$\Rightarrow x^\top Q x - 2b^\top x + c = 0$$

The decision boundary is a quadratic function. In 2-d case, it looks like an ellipse, or a parabola, or a hyperbola.

- Depending on the Gaussian distributions, the decision boundary can be very different



- Decision boundary: $h(\mathbf{x}) = -\ln \frac{q_i(x)}{q_j(x)} = 0$

Gaussian Naive Bayes vs. Logistic Regression

Set of Gaussian
Naive Bayes parameters
(feature variance independent
of class label)

Set of Logistic
Regression parameters

Number of parameters

- Naive Bayes: $4D + 1$
  - When all random variables are binary
  - $4D + 1$ for Gaussians: $2D$ mean, $2D$ variance, and 1 for prior
- Logistic Regression: D
  - $\theta_1, \theta_2, ..., \theta_D$
- where D represents the number of features in the input data.

Gaussian Naive Bayes vs. Logistic Regression

Set of Gaussian Naive Bayes parameters (feature variance independent of class label) $\Leftrightarrow$ Set of Logistic Regression parameters

- Estimation method:
  - Naive Bayes parameter estimates are decoupled (easy)
  - Logistic regression parameter estimates are coupled (less easy)

Gaussian Naive Bayes vs. Logistic Regression

Set of Gaussian
Naive Bayes parameters
(feature variance independent
of class label)

Set of Logistic
Regression parameters

- Representation equivalence (both yield linear decision boundaries)
    - But only in special case!!! (GNB with class-independent variances)
    - LR makes no assumptions about $P(\mathbf{X}|Y)$ in learning!!!
    - Optimize different functions! Obtain different solutions

Gaussian Naive Bayes vs. Logistic Regression

- Asymptotic comparison (# training examples → infinity)

- When model assumptions correct
  - Naive Bayes, logistic regression produce identical classifiers
  - Naive Bayes converges faster

- When model assumptions incorrect
  - logistic regression is less biased - does not assume conditional independence
  - logistic regression has fewer parameters
  - therefore expected to outperform Naive Bayes

Gaussian Naive Bayes vs. Logistic Regression

Exploration Unlabeled Data

$$P(y|x) = \frac{P(x|y)P(y)}{P(x)}$$

$$P(x) = \sum_y P(x|y)P(y)$$

Gaussian Naive Bayes vs. Logistic Regression

Exploration Unlabeled Data

$$P(y|x) = \frac{P(x|y)P(y)}{P(x)}$$

$$P(x) = \sum_y P(x|y)P(y)$$

MLE

$$\max_\theta \log P_\theta(x) = \log \sum_y P(x|y)P(y)$$

Gaussian Naive Bayes vs. Logistic Regression

Exploration Unlabeled Data

$$P(y|x) = \frac{P(x|y)P(y)}{P(x)}$$

$$P(x) = \sum_y P(x|y)P(y) \;=\; \sum\nolimits_y P(y)\mathcal{N}(x|\mu_y, \Sigma_y)$$

MLE

$$\max_\theta \log P_\theta(x) = \log \sum_y P(x|y)P(y)$$

Gaussian Mixture Model!

ML Algorithm Pipeline

General ML Algorithm Pipeline

1. Build probabilistic models
2. Derive loss function (by MLE or MAP….)
3. Select optimizer

Q&A