Introduction to Machine Learning

Lecture 11 Classification - Introduction (Linear Classification)

Hongteng Xu

中国人民大學 RENMIN UNIVERSITY OF CHINA

高瓴人工智能学院 Gaoling School of Artificial Intelligence

Outline

Review

- Bayesian inference of Gaussian mixture model and MCMC
- Nonparametric clustering and kernel density estimation
- Mean shift algorithm

Outline

Review

- Bayesian inference of Gaussian mixture model and MCMC
- Nonparametric clustering and kernel density estimation
- Mean shift algorithm

Today

- Classification, definition, evaluation
- Linear classifiers (Naïve Bayes, Linear discriminant analysis, Logistic regression)

Classification Problems

- Object recognition and detection
- Face recognition
- ....

A Viewpoint of Statistical ML

- Given a set of labeled data $\{\boldsymbol{x}_n \in \mathcal{X}, y_n \in \mathcal{C}\}_{n=1}^{N}$
- Train a model (a.k.a. a classifier) $f : \mathcal{X} \mapsto \mathcal{C}$

Classification Problems

- Object recognition and detection
- Face recognition
- ....

A Viewpoint of Statistical ML

- Given a set of labeled data $\{\boldsymbol{x}_n \in \mathcal{X}, y_n \in \mathcal{C}\}_{n=1}^N$
- Train a model (a.k.a. a classifier) $f : \mathcal{X} \mapsto \mathcal{C}$
- **Key modeling target:** $p(y|\boldsymbol{x})$ or $p(y, \boldsymbol{x})$ (or equivalently, $p(\boldsymbol{x}|y)$)

Classification Problems

- Object recognition and detection
- Face recognition
- ....

A Viewpoint of Statistical ML

- Given a set of labeled data $\{\boldsymbol{x}_n \in \mathcal{X}, y_n \in \mathcal{C}\}_{n=1}^{N}$
- Train a model (a.k.a. a classifier) $f : \mathcal{X} \mapsto \mathcal{C}$
- **Key modeling target:** $p(y|\boldsymbol{x})$ or $p(y, \boldsymbol{x})$ (or equivalently, $p(\boldsymbol{x}|y)$)
- Recall generative and discriminative modeling,

Classification Problems

- Object recognition and detection
- Face recognition
- ....

A Viewpoint of Statistical ML

- Given a set of labeled data $\{\boldsymbol{x}_n \in \mathcal{X}, y_n \in \mathcal{C}\}_{n=1}^N$
- Train a model (a.k.a. a classifier) $f : \mathcal{X} \mapsto \mathcal{C}$
- **Key modeling target:** $p(y|\boldsymbol{x})$ or $p(y, \boldsymbol{x})$ (or equivalently, $p(\boldsymbol{x}|y)$)
- Recall generative and discriminative modeling, and their pros and cons.

Classification Problems

- Object recognition and detection
- Face recognition
- ....

A Viewpoint of Statistical ML

- Given a set of labeled data $\{\boldsymbol{x}_n \in \mathcal{X}, y_n \in \mathcal{C}\}_{n=1}^N$
- Train a model (a.k.a. a classifier) $f : \mathcal{X} \mapsto \mathcal{C}$
- **Key modeling target:** $p(y|\boldsymbol{x})$ or $p(y, \boldsymbol{x})$ (or equivalently, $p(\boldsymbol{x}|y)$)
- Recall generative and discriminative modeling, and their pros and cons.

Special Cases of Classification Problems

Outlier Detection (One-class Recognition):

- Classify one class and detect outliers.

Special Cases of Classification Problems

Outlier Detection (One-class Recognition):

- Classify one class and detect outliers.

Constrastive Learning:

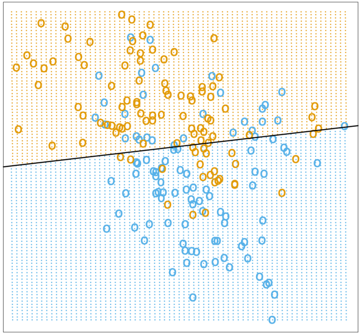
- Given a pair of samples, judge whether they are in the same class or not.

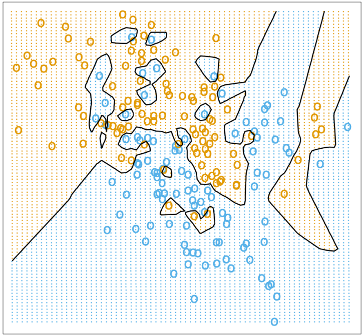
Key Challenges of Classification Compared to Regression

- Discrete label/output space $\Rightarrow$ essentially f is non-differentiable.

Key Challenges of Classification Compared to Regression

- Discrete label/output space $\Rightarrow$ essentially f is non-differentiable.
- Ambiguity on boundaries, and the trade-off between precision and robustness





K-Nearest Neighbor Classifier: Maybe The Simplest Classifier

- Given a set of labeled data $\{\boldsymbol{x}_n, y_n\}$, where $y_n \in \{0, 1\}$
- 1-NN: For a sample $\boldsymbol{x}$

$$y_{\boldsymbol{x}} := \arg\min_{y_n} d(\boldsymbol{x}_n, \boldsymbol{x})$$

K-Nearest Neighbor Classifier: Maybe The Simplest Classifier

- Given a set of labeled data $\{\boldsymbol{x}_n, y_n\}$, where $y_n \in \{0, 1\}$
- 1-NN: For a sample $\boldsymbol{x}$

$$y_{\boldsymbol{x}} := \arg\min_{y_n} d(\boldsymbol{x}_n, \boldsymbol{x}) \tag{1}$$

- Obviously, it is a nonparametric model.

K-Nearest Neighbor Classifier: Maybe The Simplest Classifier

- Given a set of labeled data $\{\boldsymbol{x}_n, y_n\}$, where $y_n \in \{0, 1\}$
- 1-NN: For a sample $\boldsymbol{x}$

$$y_{\boldsymbol{x}} := \arg\min_{y_n} d(\boldsymbol{x}_n, \boldsymbol{x}) \tag{1}$$

- Obviously, it is a nonparametric model.
- How to extend to multi-class cases?

K-Nearest Neighbor Classifier: Maybe The Simplest Classifier

- Given a set of labeled data $\{\boldsymbol{x}_n, y_n\}$, where $y_n \in \{0, 1\}$
- 1-NN: For a sample $\boldsymbol{x}$

$$y_{\boldsymbol{x}} := \arg\min_{y_n} d(\boldsymbol{x}_n, \boldsymbol{x}) \tag{1}$$

- Obviously, it is a nonparametric model.
- How to extend to multi-class cases? Represent y_n as one-hot vector.

K-Nearest Neighbor Classifier: Maybe The Simplest Classifier

- Given a set of labeled data $\{\boldsymbol{x}_n, y_n\}$, where $y_n \in \{0, 1\}$
- 1-NN: For a sample $\boldsymbol{x}$

$$y_{\boldsymbol{x}} := \arg\min_{y_n} d(\boldsymbol{x}_n, \boldsymbol{x}) \tag{1}$$

- Obviously, it is a nonparametric model.
- How to extend to multi-class cases? Represent y_n as one-hot vector.
- How to estimate $p(y|\boldsymbol{x})$?

K-Nearest Neighbor Classifier: Maybe The Simplest Classifier

- Given a set of labeled data $\{\boldsymbol{x}_n, y_n\}$, where $y_n \in \{0, 1\}$
- 1-NN: For a sample $\boldsymbol{x}$

$$y_{\boldsymbol{x}} := \arg\min_{y_n} d(\boldsymbol{x}_n, \boldsymbol{x}) \tag{1}$$

- Obviously, it is a nonparametric model.
- How to extend to multi-class cases? Represent y_n as one-hot vector.
- How to estimate $p(y|\boldsymbol{x})$? K-NN

$$\hat{p}(y|\boldsymbol{x}) = \frac{1}{|\mathcal{N}_h(\boldsymbol{x})|} \sum_{n \in \mathcal{N}_h(\boldsymbol{x})} \{d(\boldsymbol{x}_n, \boldsymbol{x}) \leq h\} y_n$$

K-Nearest Neighbor Classifier: Maybe The Simplest Classifier

- Given a set of labeled data $\{\boldsymbol{x}_n, y_n\}$, where $y_n \in \{0, 1\}$
- 1-NN: For a sample $\boldsymbol{x}$

$$y_{\boldsymbol{x}} := \arg\min_{y_n} d(\boldsymbol{x}_n, \boldsymbol{x}) \tag{1}$$

- Obviously, it is a nonparametric model.
- How to extend to multi-class cases? Represent y_n as one-hot vector.
- How to estimate $p(y|\boldsymbol{x})$? K-NN

$$\hat{p}(y|\boldsymbol{x}) = \frac{1}{|\mathcal{N}_h(\boldsymbol{x})|} \sum_{n \in \mathcal{N}_h(\boldsymbol{x})} \{d(\boldsymbol{x}_n, \boldsymbol{x}) \leq h\} y_n \tag{2}$$

- Is it a special case of Nadaraya-Watson estimator?

Naïve Bayes Classifier: A Typical Classifier

Motivations:

- Modeling $p(y, \boldsymbol{x})$ and assume the features are independent.

Naïve Bayes Classifier: A Typical Classifier

Motivations:

- Modeling $p(y, \boldsymbol{x})$ and assume the features are independent.

Principle:

- Independent features $\Rightarrow$ The joint distribution can be factorized:

$$p(y, \boldsymbol{x}) = p(\boldsymbol{x}|y)p(y)$$

Naïve Bayes Classifier: A Typical Classifier

Motivations:

- Modeling $p(y, \boldsymbol{x})$ and assume the features are independent.

Principle:

- Independent features $\Rightarrow$ The joint distribution can be factorized:

$$p(y, \boldsymbol{x}) = p(\boldsymbol{x}|y)p(y) = p(y) \prod_{d=1}^{D} p(x_d|y)$$

Naïve Bayes Classifier: A Typical Classifier

Motivations:

- Modeling $p(y, \boldsymbol{x})$ and assume the features are independent.

Principle:

- Independent features $\Rightarrow$ The joint distribution can be factorized:

$$p(y, \boldsymbol{x}) = p(\boldsymbol{x}|y)p(y) = p(y) \prod_{d=1}^{D} p(x_d|y) \tag{3}$$

- **Gaussian Naïve Bayes:** For $\boldsymbol{x} \in \mathbb{R}^D$, each $p(x_d|y)$ is assumed to be Gaussian

$$p(x_d|y = k) = \frac{1}{\sqrt{2\pi}\sigma_{k,d}} \exp\left(-\frac{(x_d - \mu_{k,d})^2}{2\sigma_{k,d}^2}\right)$$

Naïve Bayes Classifier: A Typical Classifier

Motivations:

- Modeling $p(y, \boldsymbol{x})$ and assume the features are independent.

Principle:

- Independent features $\Rightarrow$ The joint distribution can be factorized:

$$p(y, \boldsymbol{x}) = p(\boldsymbol{x}|y)p(y) = p(y) \prod_{d=1}^{D} p(x_d|y) \tag{3}$$

- **Gaussian Naïve Bayes:** For $\boldsymbol{x} \in \mathbb{R}^D$, each $p(x_d|y)$ is assumed to be Gaussian

$$p(x_d|y = k) = \frac{1}{\sqrt{2\pi}\sigma_{k,d}} \exp\left(-\frac{(x_d - \mu_{k,d})^2}{2\sigma_{k,d}^2}\right) \tag{4}$$

- How to estimate $\{\mu_{k,d}, \sigma_{k,d}\}_{k,d}$?

Naïve Bayes Classifier: A Typical Classifier

Motivations:

- Modeling $p(y, \boldsymbol{x})$ and assume the features are independent.

Principle:

- Independent features $\Rightarrow$ The joint distribution can be factorized:

$$p(y, \boldsymbol{x}) = p(\boldsymbol{x}|y)p(y) = p(y) \prod_{d=1}^{D} p(x_d|y) \tag{3}$$

- **Gaussian Naïve Bayes:** For $\boldsymbol{x} \in \mathbb{R}^D$, each $p(x_d|y)$ is assumed to be Gaussian

$$p(x_d|y = k) = \frac{1}{\sqrt{2\pi}\sigma_{k,d}} \exp\left(-\frac{(x_d - \mu_{k,d})^2}{2\sigma_{k,d}^2}\right) \tag{4}$$

- How to estimate $\{\mu_{k,d}, \sigma_{k,d}\}_{k,d}$? MLE

Naïve Bayes Classifier: A Typical Classifier

Motivations:

- Modeling $p(y, \boldsymbol{x})$ and assume the features are independent.

Principle:

- Independent features $\Rightarrow$ The joint distribution can be factorized:

$$p(y, \boldsymbol{x}) = p(\boldsymbol{x}|y)p(y) = p(y) \prod_{d=1}^{D} p(x_d|y) \tag{3}$$

- **Gaussian Naïve Bayes:** For $\boldsymbol{x} \in \mathbb{R}^D$, each $p(x_d|y)$ is assumed to be Gaussian

$$p(x_d|y = k) = \frac{1}{\sqrt{2\pi}\sigma_{k,d}} \exp\left(-\frac{(x_d - \mu_{k,d})^2}{2\sigma_{k,d}^2}\right) \tag{4}$$

- How to estimate $\{\mu_{k,d}, \sigma_{k,d}\}_{k,d}$? MLE
- How to deal with non-Gaussian distribution?

Naïve Bayes Classifier: A Typical Classifier

Motivations:

- Modeling $p(y, \boldsymbol{x})$ and assume the features are independent.

Principle:

- Independent features $\Rightarrow$ The joint distribution can be factorized:

$$p(y, \boldsymbol{x}) = p(\boldsymbol{x}|y)p(y) = p(y)\prod_{d=1}^{D} p(x_d|y) \tag{3}$$

- **Gaussian Naïve Bayes:** For $\boldsymbol{x} \in \mathbb{R}^D$, each $p(x_d|y)$ is assumed to be Gaussian

$$p(x_d|y=k) = \frac{1}{\sqrt{2\pi}\sigma_{k,d}} \exp\left(-\frac{(x_d - \mu_{k,d})^2}{2\sigma_{k,d}^2}\right) \tag{4}$$

- How to estimate $\{\mu_{k,d}, \sigma_{k,d}\}_{k,d}$? MLE
- How to deal with non-Gaussian distribution? KDE

Naïve Bayes Classifier: A Typical Classifier

Motivations:

- Modeling $p(y, \boldsymbol{x})$ and assume the features are independent.

Principle:

- Independent features $\Rightarrow$ The joint distribution can be factorized:

$$p(y, \boldsymbol{x}) = p(\boldsymbol{x}|y)p(y) = p(y) \prod_{d=1}^{D} p(x_d|y) \tag{3}$$

- **Gaussian Naïve Bayes:** For $\boldsymbol{x} \in \mathbb{R}^D$, each $p(x_d|y)$ is assumed to be Gaussian

$$p(x_d|y = k) = \frac{1}{\sqrt{2\pi}\sigma_{k,d}} \exp\left(-\frac{(x_d - \mu_{k,d})^2}{2\sigma_{k,d}^2}\right) \tag{4}$$

- How to estimate $\{\mu_{k,d}, \sigma_{k,d}\}_{k,d}$? MLE
- How to deal with non-Gaussian distribution? KDE

$y_{\boldsymbol{x}} = k \;\Leftarrow\; k = \arg\max_k p(y = k)p(\boldsymbol{x}|y = k)$

Naïve Bayes Classifier: A Typical Classifier

Motivations:

- Modeling $p(y, \boldsymbol{x})$ and assume the features are independent.

Principle:

- Independent features $\Rightarrow$ The joint distribution can be factorized:

$$p(y, \boldsymbol{x}) = p(\boldsymbol{x}|y)p(y) = p(y) \prod_{d=1}^{D} p(x_d|y) \tag{3}$$

- **Gaussian Naïve Bayes:** For $\boldsymbol{x} \in \mathbb{R}^D$, each $p(x_d|y)$ is assumed to be Gaussian

$$p(x_d|y = k) = \frac{1}{\sqrt{2\pi}\sigma_{k,d}} \exp\left(-\frac{(x_d - \mu_{k,d})^2}{2\sigma_{k,d}^2}\right) \tag{4}$$

- How to estimate $\{\mu_{k,d}, \sigma_{k,d}\}_{k,d}$? MLE
- How to deal with non-Gaussian distribution? KDE

$$y_{\boldsymbol{x}} = k \ \Leftarrow \ k = \arg\max_k p(y = k)p(\boldsymbol{x}|y = k) = \arg\max_k p(y = k) \prod_{d=1}^{D} p(x_d|y = k) \tag{5}$$

Naïve Bayes Classifier: More Extensions

Multinomial Naïve Bayes: For $\boldsymbol{x} \in \mathbb{N}^D$:

$$p(\boldsymbol{x}|y = k) = \frac{(\sum_{d=1}^{D} x_d)!}{\prod_{d=1}^{D} x_d!} \prod_{d=1}^{D} p_{k,d}^{x_d}$$

Naïve Bayes Classifier: More Extensions

Multinomial Naïve Bayes: For $\boldsymbol{x} \in \mathbb{N}^D$:

$$p(\boldsymbol{x}|y=k) = \frac{(\sum_{d=1}^{D} x_d)!}{\prod_{d=1}^{D} x_d!} \prod_{d=1}^{D} p_{k,d}^{x_d}$$

$$\Rightarrow \log p(y=k|\boldsymbol{x}) \propto \log\left(p(y=k) \prod_{d=1}^{D} p_{k,d}^{x_d}\right)$$

Naïve Bayes Classifier: More Extensions

Multinomial Naïve Bayes: For $\boldsymbol{x} \in \mathbb{N}^D$:

$$p(\boldsymbol{x}|y=k) = \frac{(\sum_{d=1}^{D} x_d)!}{\prod_{d=1}^{D} x_d!} \prod_{d=1}^{D} p_{k,d}^{x_d}$$

$$\Rightarrow \log p(y=k|\boldsymbol{x}) \propto \log \left(p(y=k) \prod_{d=1}^{D} p_{k,d}^{x_d} \right)$$

$$= \log p(y=k) + \sum_{d=1}^{D} x_d \log p_{k,d} = b + \boldsymbol{w}_k^T \boldsymbol{x}.$$

Naïve Bayes Classifier: More Extensions

Multinomial Naïve Bayes: For $\boldsymbol{x} \in \mathbb{N}^D$:

$$\begin{aligned} p(\boldsymbol{x}|y=k) &= \frac{(\sum_{d=1}^{D} x_d)!}{\prod_{d=1}^{D} x_d!} \prod_{d=1}^{D} p_{k,d}^{x_d} \\ \Rightarrow \log p(y=k|\boldsymbol{x}) &\propto \log \left(p(y=k) \prod_{d=1}^{D} p_{k,d}^{x_d} \right) \\ &= \log p(y=k) + \sum_{d=1}^{D} x_d \log p_{k,d} = b + \boldsymbol{w}_k^T \boldsymbol{x}. \end{aligned} \tag{6}$$

Bernoulli Naïve Bayes: For $\boldsymbol{x} \in \{0,1\}^D$:

$$p(\boldsymbol{x}|y=k) = \prod_{d=1}^{D} p_{k,d}^{x_d} (1 - p_{k,d})^{(1-x_d)} \tag{7}$$

Linear Discriminant Analysis (LDA)

Motivations:

- The data belonging to different classes have clustering structures.

Linear Discriminant Analysis (LDA)

Motivations:

- The data belonging to different classes have clustering structures.
- Relax the assumption of feature independence

Principle: (Take two-class case as an example)

- Assume $p(\boldsymbol{x}|y=0)$ and $p(\boldsymbol{x}|y=1)$ are normal distributions:

$$\boldsymbol{x}|y=\boldsymbol{i} \sim \mathcal{N}(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i), \quad \boldsymbol{i}=0,\ 1.$$

Linear Discriminant Analysis (LDA)

Motivations:

- The data belonging to different classes have clustering structures.
- Relax the assumption of feature independence

Principle: (Take two-class case as an example)

- Assume $p(\boldsymbol{x}|y=0)$ and $p(\boldsymbol{x}|y=1)$ are normal distributions:

$$\boldsymbol{x}|y=i \sim \mathcal{N}(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i), \quad i=0,\ 1. \tag{8}$$

- **Logarithm of likelihood ratio:**

$$\log \frac{p(\boldsymbol{x}|y=1)}{p(\boldsymbol{x}|y=0)} = \log p(\boldsymbol{x}|y=1) - \log p(\boldsymbol{x}|y=0)$$

Linear Discriminant Analysis (LDA)

Motivations:

- The data belonging to different classes have clustering structures.
- Relax the assumption of feature independence

Principle: (Take two-class case as an example)

- Assume $p(\boldsymbol{x}|y=0)$ and $p(\boldsymbol{x}|y=1)$ are normal distributions:

$$\boldsymbol{x}|y=i \sim \mathcal{N}(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i), \quad i=0,\ 1. \tag{8}$$

- **Logarithm of likelihood ratio:**

$$\log \frac{p(\boldsymbol{x}|y=1)}{p(\boldsymbol{x}|y=0)} = \log p(\boldsymbol{x}|y=1) - \log p(\boldsymbol{x}|y=0) \tag{9}$$

- **Bayes optimal solution:**

$$y_{\boldsymbol{x}} = 1 \quad \Leftrightarrow \quad \exists T,\ \log \frac{p(\boldsymbol{x}|y=1)}{p(\boldsymbol{x}|y=0)} > T \tag{10}$$

Typically, we set $T=0$.

Linear Discriminant Analysis (LDA)

Given labeled data $\{\boldsymbol{x}_n, y_n\}_{n=1}^{N}$, estimate the conditional distributions:

- Recall the unbiased estimation of $\{\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i\}_{i=0,1}$.

Linear Discriminant Analysis (LDA)

Given labeled data $\{\boldsymbol{x}_n, y_n\}_{n=1}^N$, estimate the conditional distributions:

- Recall the unbiased estimation of $\{\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i\}_{i=0,1}$.
- LDA further assumes $\boldsymbol{\Sigma}_0 = \boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}$

Linear Discriminant Analysis (LDA)

Given labeled data $\{\boldsymbol{x}_n, y_n\}_{n=1}^{N}$, estimate the conditional distributions:

- Recall the unbiased estimation of $\{\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i\}_{i=0,1}$.
- LDA further assumes $\boldsymbol{\Sigma}_0 = \boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}$ (How to estimate $\boldsymbol{\Sigma}$?)

Linear Discriminant Analysis (LDA)

Given labeled data $\{\boldsymbol{x}_n, y_n\}_{n=1}^N$, estimate the conditional distributions:

- Recall the unbiased estimation of $\{\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i\}_{i=0,1}$.
- LDA further assumes $\boldsymbol{\Sigma}_0 = \boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}$ (How to estimate $\boldsymbol{\Sigma}$?)
- Write (9) as

$$\log \frac{p(\boldsymbol{x}|y=1)}{p(\boldsymbol{x}|y=0)} = 2(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)^T \boldsymbol{\Sigma}^{-1} \boldsymbol{x} - (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{\mu}_1 + \boldsymbol{\mu}_0) > 0 (= T) \tag{11}$$

Derive it.

Linear Discriminant Analysis (LDA)

Given labeled data $\{\boldsymbol{x}_n, y_n\}_{n=1}^N$, estimate the conditional distributions:

- Recall the unbiased estimation of $\{\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i\}_{i=0,1}$.
- LDA further assumes $\boldsymbol{\Sigma}_0 = \boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}$ (How to estimate $\boldsymbol{\Sigma}$?)
- Write (9) as

$$\log \frac{p(\boldsymbol{x}|y=1)}{p(\boldsymbol{x}|y=0)} = 2(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)^T \boldsymbol{\Sigma}^{-1} \boldsymbol{x} - (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{\mu}_1 + \boldsymbol{\mu}_0) > 0 (= T) \tag{11}$$

  Derive it.

- **Decision criterion:**

$$y_{\boldsymbol{x}} = 1 \quad \Leftrightarrow \quad \langle \boldsymbol{w}, \boldsymbol{x} \rangle > c, \ \text{where}$$
$$\boldsymbol{w} = \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0), \quad c = \langle \boldsymbol{w}, \frac{1}{2}(\boldsymbol{\mu}_1 + \boldsymbol{\mu}_0) \rangle.$$

Linear Discriminant Analysis (LDA)

Given labeled data $\{\boldsymbol{x}_n, y_n\}_{n=1}^N$, estimate the conditional distributions:

- Recall the unbiased estimation of $\{\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i\}_{i=0,1}$.
- LDA further assumes $\boldsymbol{\Sigma}_0 = \boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}$ (How to estimate $\boldsymbol{\Sigma}$?)
- Write (9) as

$$\log \frac{p(\boldsymbol{x}|y=1)}{p(\boldsymbol{x}|y=0)} = 2(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)^T \boldsymbol{\Sigma}^{-1} \boldsymbol{x} - (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{\mu}_1 + \boldsymbol{\mu}_0) > 0 (= T) \tag{11}$$

  Derive it.

- **Decision criterion:**

$$\begin{aligned} y_{\boldsymbol{x}} = 1 \quad &\Leftrightarrow \quad \langle \boldsymbol{w}, \boldsymbol{x} \rangle > c, \text{ where} \\ \boldsymbol{w} = \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0), &\quad c = \langle \boldsymbol{w}, \frac{1}{2}(\boldsymbol{\mu}_1 + \boldsymbol{\mu}_0) \rangle. \end{aligned} \tag{12}$$

- Let's think about the geometrical interpretation of LDA.

Generalized LDA/QDA: Fisher's Linear Discriminant

Motivations:

- Relax the assumption of shared covariance matrix
- Relax the assumption of Gaussian conditional distribution

Principle:

- For the two classes, estimate their means and covariances $\{\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i\}_{i=0,1}$.
- Project the data to 1D along a direction $\boldsymbol{w}$ (Derive the projected means and covariances)

Generalized LDA/QDA: Fisher's Linear Discriminant

Motivations:

- Relax the assumption of shared covariance matrix
- Relax the assumption of Gaussian conditional distribution

Principle:

- For the two classes, estimate their means and covariances $\{\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i\}_{i=0,1}$.
- Project the data to 1D along a direction $\boldsymbol{w}$ (Derive the projected means and covariances)
- **Fisher's separation** between two distributions: the ratio of the variance between the projected classes to the variance within the projected classes

$$S = \frac{\sigma^2_{\text{between}}}{\sigma^2_{\text{within}}} = \frac{(\boldsymbol{w}^T(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0))^2}{\boldsymbol{w}^T(\boldsymbol{\Sigma}_0 + \boldsymbol{\Sigma}_1)\boldsymbol{w}} \tag{13}$$

Generalized LDA/QDA: Fisher's Linear Discriminant

Principle:

- Fisher's linear discriminant finds the optimal projection $\boldsymbol{w}$ to maximize the separation:

$$\max_{\boldsymbol{w}} S \quad \Rightarrow \quad \boldsymbol{w} \propto (\boldsymbol{\Sigma}_0 + \boldsymbol{\Sigma}_1)^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)$$

Generalized LDA/QDA: Fisher's Linear Discriminant

Principle:

- Fisher's linear discriminant finds the optimal projection $\boldsymbol{w}$ to maximize the separation:

$$\max_{\boldsymbol{w}} S \quad \Rightarrow \quad \boldsymbol{w} \propto (\boldsymbol{\Sigma}_0 + \boldsymbol{\Sigma}_1)^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0) \tag{14}$$

(Derive it. Hint: Consider $\frac{\partial \log S}{\partial \boldsymbol{w}} = \mathbf{0}$)

Generalized LDA/QDA: Fisher's Linear Discriminant

Principle:

- Fisher's linear discriminant finds the optimal projection $\boldsymbol{w}$ to maximize the separation:

$$\max_{\boldsymbol{w}} S \quad \Rightarrow \quad \boldsymbol{w} \propto (\boldsymbol{\Sigma}_0 + \boldsymbol{\Sigma}_1)^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0) \tag{14}$$

(Derive it. Hint: Consider $\frac{\partial \log S}{\partial \boldsymbol{w}} = \mathbf{0}$)

- Recall the decision criterion is

$$\langle \boldsymbol{w}, \boldsymbol{x} \rangle > c, \quad \text{where} \quad c = \langle \boldsymbol{w}, \frac{1}{2}(\boldsymbol{\mu}_0 + \boldsymbol{\mu}_1) \rangle = \frac{1}{2}\boldsymbol{\mu}_1 \boldsymbol{\Sigma}_1^{-1} \boldsymbol{\mu}_1 - \frac{1}{2}\boldsymbol{\mu}_0 \boldsymbol{\Sigma}_0^{-1} \boldsymbol{\mu}_0. \tag{15}$$

Multi-class LDA

- From find a hyperplane to find a subspace which appears to contain all of the class variability.

Multi-class LDA

- From find a hyperplane to find a subspace which appears to contain all of the class variability.

Practical Implementation:

- Suppose that we obtain the means for C classes $\{\boldsymbol{\mu}_i\}_{i=1}^{C}$ and the classes share the same covariance $\boldsymbol{\Sigma}$.

Multi-class LDA

- From find a hyperplane to find a subspace which appears to contain all of the class variability.

Practical Implementation:

- Suppose that we obtain the means for C classes $\{\boldsymbol{\mu}_i\}_{i=1}^{C}$ and the classes share the same covariance $\boldsymbol{\Sigma}$.
- **The covariance of all the classes**:

$$\boldsymbol{\Sigma}_{\text{between}} = \frac{1}{C}\sum_{i=1}^{C}(\boldsymbol{\mu}_i - \boldsymbol{\mu})(\boldsymbol{\mu}_i - \boldsymbol{\mu})^T, \quad \text{where} \quad \boldsymbol{\mu} = \frac{1}{C}\sum_{i=1}^{C}\boldsymbol{\mu}_i$$

Multi-class LDA

- From find a hyperplane to find a subspace which appears to contain all of the class variability.

Practical Implementation:

- Suppose that we obtain the means for C classes $\{\boldsymbol{\mu}_i\}_{i=1}^{C}$ and the classes share the same covariance $\boldsymbol{\Sigma}$.
- **The covariance of all the classes**:

$$\boldsymbol{\Sigma}_{\text{between}} = \frac{1}{C}\sum\nolimits_{i=1}^{C}(\boldsymbol{\mu}_i - \boldsymbol{\mu})(\boldsymbol{\mu}_i - \boldsymbol{\mu})^T, \quad \text{where} \quad \boldsymbol{\mu} = \frac{1}{C}\sum\nolimits_{i=1}^{C}\boldsymbol{\mu}_i \tag{16}$$

- Fisher's separation becomes

$$S = \frac{\boldsymbol{w}^T\boldsymbol{\Sigma}_{\text{between}}\boldsymbol{w}}{\boldsymbol{w}^T\boldsymbol{\Sigma}\boldsymbol{w}}.$$

Multi-class LDA

- From find a hyperplane to find a subspace which appears to contain all of the class variability.

Practical Implementation:

- Suppose that we obtain the means for C classes $\{\boldsymbol{\mu}_i\}_{i=1}^{C}$ and the classes share the same covariance $\boldsymbol{\Sigma}$.
- **The covariance of all the classes**:

$$\boldsymbol{\Sigma}_{\text{between}} = \frac{1}{C}\sum_{i=1}^{C}(\boldsymbol{\mu}_i - \boldsymbol{\mu})(\boldsymbol{\mu}_i - \boldsymbol{\mu})^T, \quad \text{where} \quad \boldsymbol{\mu} = \frac{1}{C}\sum_{i=1}^{C}\boldsymbol{\mu}_i \tag{16}$$

- Fisher's separation becomes

$$S = \frac{\boldsymbol{w}^T\boldsymbol{\Sigma}_{\text{between}}\boldsymbol{w}}{\boldsymbol{w}^T\boldsymbol{\Sigma}\boldsymbol{w}}. \tag{17}$$

- When $\boldsymbol{w}$ is an eigenvector of $\boldsymbol{\Sigma}^{-1}\boldsymbol{\Sigma}_{\text{between}}$, the separation is equal to the corresponding eigenvalue.

Multi-class LDA

One against the rest

- Train C LDA classifiers, each of which maximizes the separation of one class and the remaining classes.

Multi-class LDA

One against the rest

- Train C LDA classifiers, each of which maximizes the separation of one class and the remaining classes.

Pairwise classification

- Train $C(C-1)/2$ LDA classifiers, each of which maximize the separation of arbitrary two classes.

Logistic Regression (LR)

Motivation:

- Instead of modeling $p(\boldsymbol{x}|y)$, modeling $p(y|\boldsymbol{x})$ is often easier. (Recall the introduction of generative and discriminative models)

Logistic Regression (LR)

Motivation:

- Instead of modeling $p(\boldsymbol{x}|y)$, modeling $p(y|\boldsymbol{x})$ is often easier. (Recall the introduction of generative and discriminative models)

Principle:

- Essentially, logistic regression is a special case of generalized linear model (GLM, recall lecture 4)

Logistic Regression (LR)

Motivation:

- Instead of modeling $p(\boldsymbol{x}|y)$, modeling $p(y|\boldsymbol{x})$ is often easier. (Recall the introduction of generative and discriminative models)

Principle:

- Essentially, logistic regression is a special case of generalized linear model (GLM, recall lecture 4)
  1. An **exponential family** of probability distributions to generate the output.
  2. A **linear predictor** $\eta = X\beta$
  3. A **link function** g: $\mathbb{E}[Y|X] = \mu = g^{-1}(\eta)$.

Logistic Regression (LR)

Motivation:

- Instead of modeling $p(\boldsymbol{x}|y)$, modeling $p(y|\boldsymbol{x})$ is often easier. (Recall the introduction of generative and discriminative models)

Principle:

- Essentially, logistic regression is a special case of generalized linear model (GLM, recall lecture 4)
  1. An **exponential family** of probability distributions to generate the output.
  2. A **linear predictor** $\eta = X\beta$
  3. A **link function** g: $\mathbb{E}[Y|X] = \mu = g^{-1}(\eta)$.
- For classification:
  1. The distribution of output is **Bernoulli**

Logistic Regression (LR)

Motivation:

- Instead of modeling $p(\boldsymbol{x}|y)$, modeling $p(y|\boldsymbol{x})$ is often easier. (Recall the introduction of generative and discriminative models)

Principle:

- Essentially, logistic regression is a special case of generalized linear model (GLM, recall lecture 4)
  1. An **exponential family** of probability distributions to generate the output.
  2. A **linear predictor** $\eta = X\beta$
  3. A **link function** g: $\mathbb{E}[Y|X] = \mu = g^{-1}(\eta)$.
- For classification:
  1. The distribution of output is **Bernoulli**
  2. The link function is **Logit**:

$$X\beta = \eta = g(\mu) = \ln\left(\frac{\mu}{1-\mu}\right)$$

Logistic Regression (LR)

Motivation:

- Instead of modeling $p(\boldsymbol{x}|y)$, modeling $p(y|\boldsymbol{x})$ is often easier. (Recall the introduction of generative and discriminative models)

Principle:

- Essentially, logistic regression is a special case of generalized linear model (GLM, recall lecture 4)
  1. An **exponential family** of probability distributions to generate the output.
  2. A **linear predictor** $\eta = X\beta$
  3. A **link function** g: $\mathbb{E}[Y|X] = \mu = g^{-1}(\eta)$.
- For classification:
  1. The distribution of output is **Bernoulli**
  2. The link function is **Logit**:

$$X\beta = \eta = g(\mu) = \ln\left(\frac{\mu}{1-\mu}\right) \Rightarrow \quad \mu = \frac{\exp(X\beta)}{1+\exp(X\beta)} = \underbrace{\frac{1}{1+\exp(-X\beta)}}_{\text{Sigmoid}} \tag{18}$$

Learning Logistic Regression (LR) via The MLE of The GLM

- Given a set of labeled data $\{\boldsymbol{x}_n, y_n\}_{n=1}^N$, the MLE of LR is

$$\max_{\boldsymbol{\beta}} \sum_{n=1}^{N} \log p(y_n|\boldsymbol{x}_n; \boldsymbol{\beta})$$

Learning Logistic Regression (LR) via The MLE of The GLM

- Given a set of labeled data $\{\boldsymbol{x}_n, y_n\}_{n=1}^{N}$, the MLE of LR is

$$\max_{\boldsymbol{\beta}} \sum_{n=1}^{N} \log p(y_n | \boldsymbol{x}_n; \boldsymbol{\beta}) \tag{19}$$

- Derive the MLE of Bernoulli distribution

Learning Logistic Regression (LR) via The MLE of The GLM

- Given a set of labeled data $\{\boldsymbol{x}_n, y_n\}_{n=1}^N$, the MLE of LR is

$$\max_{\boldsymbol{\beta}} \sum_{n=1}^{N} \log p(y_n|\boldsymbol{x}_n; \boldsymbol{\beta}) \tag{19}$$

- Derive the MLE of Bernoulli distribution
- As a result, (19) becomes the (binary) cross entropy loss:

$$\max_{\boldsymbol{\beta}} \sum_{y_n=1} \log p(1|\boldsymbol{x}_n; \boldsymbol{\beta}) + \sum_{y_n=0} \log \underbrace{p(0|\boldsymbol{x}_n; \boldsymbol{\beta})}_{1-p(1|\boldsymbol{x}_n;\boldsymbol{\beta})}$$

Learning Logistic Regression (LR) via The MLE of The GLM

- Given a set of labeled data $\{\boldsymbol{x}_n, y_n\}_{n=1}^N$, the MLE of LR is

$$\max_{\boldsymbol{\beta}} \sum_{n=1}^N \log p(y_n|\boldsymbol{x}_n; \boldsymbol{\beta}) \tag{19}$$

- Derive the MLE of Bernoulli distribution
- As a result, (19) becomes the (binary) cross entropy loss:

$$\max_{\boldsymbol{\beta}} \sum_{y_n=1} \log p(1|\boldsymbol{x}_n; \boldsymbol{\beta}) + \sum_{y_n=0} \log \underbrace{p(0|\boldsymbol{x}_n; \boldsymbol{\beta})}_{1-p(1|\boldsymbol{x}_n;\boldsymbol{\beta})}$$

$$= \max_{\boldsymbol{\beta}} \sum_{n=1}^N (y_n \log p_n + (1 - y_n) \log(1 - p_n))$$

Learning Logistic Regression (LR) via The MLE of The GLM

- Given a set of labeled data $\{\boldsymbol{x}_n, y_n\}_{n=1}^N$, the MLE of LR is

$$\max_{\boldsymbol{\beta}} \sum_{n=1}^{N} \log p(y_n|\boldsymbol{x}_n; \boldsymbol{\beta}) \tag{19}$$

- Derive the MLE of Bernoulli distribution
- As a result, (19) becomes the (binary) cross entropy loss:

$$\begin{aligned} &\max_{\boldsymbol{\beta}} \sum_{y_n=1} \log p(1|\boldsymbol{x}_n; \boldsymbol{\beta}) + \sum_{y_n=0} \log \underbrace{p(0|\boldsymbol{x}_n; \boldsymbol{\beta})}_{1-p(1|\boldsymbol{x}_n;\boldsymbol{\beta})} \\ &= \max_{\boldsymbol{\beta}} \sum_{n=1}^{N} (y_n \log p_n + (1-y_n)\log(1-p_n)) \end{aligned} \tag{20}$$

- No closed-form solution, GD or SGD is required.

Learning Logistic Regression (LR) via The MLE of The GLM

- Given a set of labeled data $\{\boldsymbol{x}_n, y_n\}_{n=1}^{N}$, the MLE of LR is

$$\max_{\boldsymbol{\beta}} \sum_{n=1}^{N} \log p(y_n|\boldsymbol{x}_n;\boldsymbol{\beta}) \tag{19}$$

- Derive the MLE of Bernoulli distribution
- As a result, (19) becomes the (binary) cross entropy loss:

$$\begin{aligned}
&\max_{\boldsymbol{\beta}} \sum_{y_n=1} \log p(1|\boldsymbol{x}_n;\boldsymbol{\beta}) + \sum_{y_n=0} \log \underbrace{p(0|\boldsymbol{x}_n;\boldsymbol{\beta})}_{1-p(1|\boldsymbol{x}_n;\boldsymbol{\beta})} \\
&= \max_{\boldsymbol{\beta}} \sum_{n=1}^{N} (y_n \log p_n + (1-y_n)\log(1-p_n))
\end{aligned} \tag{20}$$

- No closed-form solution, GD or SGD is required.
- Derive $\frac{\mathrm{dSigmoid}(s)}{\mathrm{d}s}$.

From Logistic Regression to Softmax Regression

Given labeled data for C classes:

- From Bernoulli distribution to Categorical Distribution: $p(\boldsymbol{y}) = \prod_{i=1}^{C} p_i^{y_i}$, where $\boldsymbol{y}$ is one-hot label vector.

From Logistic Regression to Softmax Regression

Given labeled data for C classes:

- From Bernoulli distribution to Categorical Distribution: $p(\boldsymbol{y}) = \prod_{i=1}^{C} p_i^{y_i}$, where $\boldsymbol{y}$ is one-hot label vector.
- From Sigmoid to Softmax:

$$p_i = p(y_i = 1|\boldsymbol{x}) = \frac{\exp(\boldsymbol{x}^T\boldsymbol{\beta}_i)}{\sum_{c=1}^{C}\exp(\boldsymbol{x}^T\boldsymbol{\beta}_c)}$$

From Logistic Regression to Softmax Regression

Given labeled data for C classes:

- From Bernoulli distribution to Categorical Distribution: $p(\boldsymbol{y}) = \prod_{i=1}^{C} p_i^{y_i}$, where $\boldsymbol{y}$ is one-hot label vector.
- From Sigmoid to Softmax:

$$p_i = p(y_i = 1|\boldsymbol{x}) = \frac{\exp(\boldsymbol{x}^T \boldsymbol{\beta}_i)}{\sum_{c=1}^{C} \exp(\boldsymbol{x}^T \boldsymbol{\beta}_c)} \tag{21}$$

(Question: Is softmax shift-invariant?)

From Logistic Regression to Softmax Regression

Given labeled data for C classes:

- From Bernoulli distribution to Categorical Distribution: $p(\boldsymbol{y}) = \prod_{i=1}^{C} p_i^{y_i}$, where $\boldsymbol{y}$ is one-hot label vector.
- From Sigmoid to Softmax:

$$p_i = p(y_i = 1|\boldsymbol{x}) = \frac{\exp(\boldsymbol{x}^T \boldsymbol{\beta}_i)}{\sum_{c=1}^{C} \exp(\boldsymbol{x}^T \boldsymbol{\beta}_c)} \tag{21}$$

  (Question: Is softmax shift-invariant?)

- MLE:

$$\max_{\{\boldsymbol{\beta}_i\}_{i=1}^{C}} \sum_{n=1}^{N} \sum_{c=1}^{C} y_{nc} \log p(y_{nc} = 1|\boldsymbol{x}_n; \boldsymbol{\beta}_c) \tag{22}$$

The Differences between LDA and LR

LDA:

- Generative modeling: first obtain $p(\boldsymbol{x}|y)$ explicitly, then determine $p(y|\boldsymbol{x})$ in the projected space.

The Differences between LDA and LR

LDA:

- Generative modeling: first obtain $p(\boldsymbol{x}|y)$ explicitly, then determine $p(y|\boldsymbol{x})$ in the projected space.
- Gaussian assumption on $p(\boldsymbol{x}|y)$

The Differences between LDA and LR

LDA:

- Generative modeling: first obtain $p(\boldsymbol{x}|y)$ explicitly, then determine $p(y|\boldsymbol{x})$ in the projected space.
- Gaussian assumption on $p(\boldsymbol{x}|y)$
- The logarithm of likelihood ratio indicates the confidence of classification implicitly.

LR

- Discriminative modeling: modeling $p(y|\boldsymbol{x})$ directly.

The Differences between LDA and LR

LDA:

- Generative modeling: first obtain $p(\boldsymbol{x}|y)$ explicitly, then determine $p(y|\boldsymbol{x})$ in the projected space.
- Gaussian assumption on $p(\boldsymbol{x}|y)$
- The logarithm of likelihood ratio indicates the confidence of classification implicitly.

LR

- Discriminative modeling: modeling $p(y|\boldsymbol{x})$ directly.
- A generalized linear model assumption.

The Differences between LDA and LR

LDA:

- Generative modeling: first obtain $p(\boldsymbol{x}|y)$ explicitly, then determine $p(y|\boldsymbol{x})$ in the projected space.
- Gaussian assumption on $p(\boldsymbol{x}|y)$
- The logarithm of likelihood ratio indicates the confidence of classification implicitly.

LR

- Discriminative modeling: modeling $p(y|\boldsymbol{x})$ directly.
- A generalized linear model assumption.
- $p(y|\boldsymbol{x})$ indicates the confidence of classification directly.

In many tasks, their performance is similar, especially for simple 2-classification problems for low-dimensional data.

In Summary

- Classification problem and its challenges
- Linear discriminant analysis (LDA)
- Logistic regression

Next...

- Support-Vector Machine (SVM)