The Mathematics of Machine Learning 2026

Lecture 4

Tim van Erven

University of Amsterdam

Outline

Plug-in Estimators for Classification

Linear and Quadratic Discriminant Analysis (Plug-in Estimators)

1. Gaussian In-class Distributions
2. Estimating the Covariance Matrix: LDA vs QDA
3. Shape of the Decision Boundary

Naive Bayes Classifier (Plug-in Estimator)

4. Naive Bayes
5. Discrete Features with Application to Spam Detection
6. Parameter Estimation
7. Discussion

Plug-in Estimators for Classification

Bayes-optimal predictor: $$f_{\mathrm{B}}(X) = \arg\max_k P^*(Y = k \mid X)$$

estimate

Plug-in estimators: $$\hat{f}(X) = \arg\max_k \hat{P}(Y = k \mid X)$$

Plug-in Estimators for Classification

Bayes-optimal predictor:

$$\begin{aligned} f_{\mathrm{B}}(X) &= \arg\max_k P^*(Y = k \mid X) \\ &= \arg\max_k \frac{P^*(X \mid Y = k) P^*(Y = k)}{P^*(X)} \\ &= \arg\max_k P^*(X \mid Y = k) P^*(Y = k) \end{aligned}$$

Plug-in estimators:

$$\hat{f}(X) = \arg\max_k \hat{P}(Y = k \mid X)$$

Plug-in Estimators for Classification

Bayes-optimal predictor:

$$\begin{aligned} f_{\mathrm{B}}(X) &= \arg\max_k P^*(Y = k \mid X) \\ &= \arg\max_k \frac{P^*(X \mid Y = k) P^*(Y = k)}{P^*(X)} \\ &= \arg\max_k P^*(X \mid Y = k) P^*(Y = k) \end{aligned}$$

Plug-in estimators:

$$\begin{aligned} \hat{f}(X) &= \arg\max_k \hat{P}(Y = k \mid X) \\ &= \arg\max_k \hat{P}(X \mid Y = k) \hat{P}(Y = k) \end{aligned}$$

NB Although we are using Bayes' rule, there is nothing inherently Bayesian about this approach. Can estimate $\hat{P}(X \mid Y = k)$ and $\hat{P}(Y = k)$ by Bayesian or non-Bayesian methods.

Plug-in Estimators for Classification

Bayes-optimal predictor: $f_{\mathrm{B}}(X) = \arg\max_k P^*(X \mid Y = k) P^*(Y = k)$

Plug-in estimators: $\hat{f}(X) = \arg\max_k \hat{P}(X \mid Y = k) \hat{P}(Y = k)$

Shorter Notation:

$$f_k(X) = P^*(X \mid Y = k) \qquad \pi_k = P^*(Y = k)$$
$$\hat{f}_k(X) = \hat{P}(X \mid Y = k) \qquad \hat{\pi}_k = \hat{P}(Y = k)$$

Plug-in Estimators for Classification

Bayes-optimal predictor: $$f_{\mathrm{B}}(X) = \arg\max_k \; f_k(X) \cdot \pi_k$$

Plug-in estimators: $$\hat{f}(X) = \arg\max_k \; \hat{f}_k(X) \cdot \hat{\pi}_k$$

Shorter Notation:

$$f_k(X) = P^*(X \mid Y = k) \qquad \pi_k = P^*(Y = k)$$

$$\hat{f}_k(X) = \hat{P}(X \mid Y = k) \qquad \hat{\pi}_k = \hat{P}(Y = k)$$

Outline

Plug-in Estimators for Classification

Linear and Quadratic Discriminant Analysis (Plug-in Estimators)

1. Gaussian In-class Distributions
2. Estimating the Covariance Matrix: LDA vs QDA
3. Shape of the Decision Boundary

Naive Bayes Classifier (Plug-in Estimator)

4. Naive Bayes
5. Discrete Features with Application to Spam Detection
6. Parameter Estimation
7. Discussion

Gaussian In-class Distributions

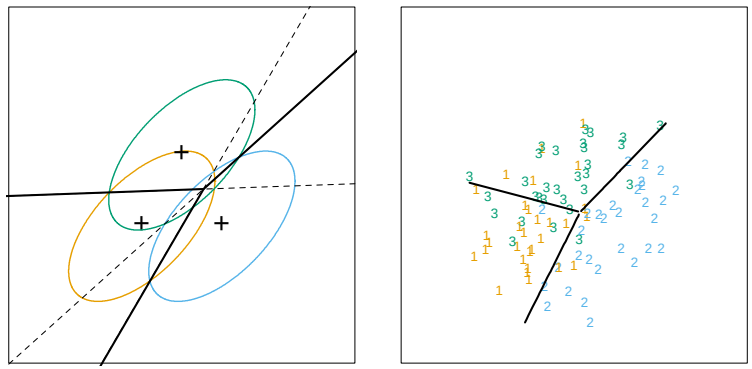
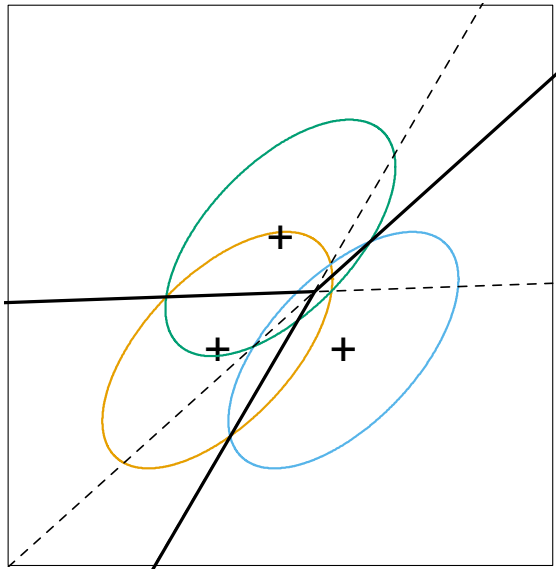
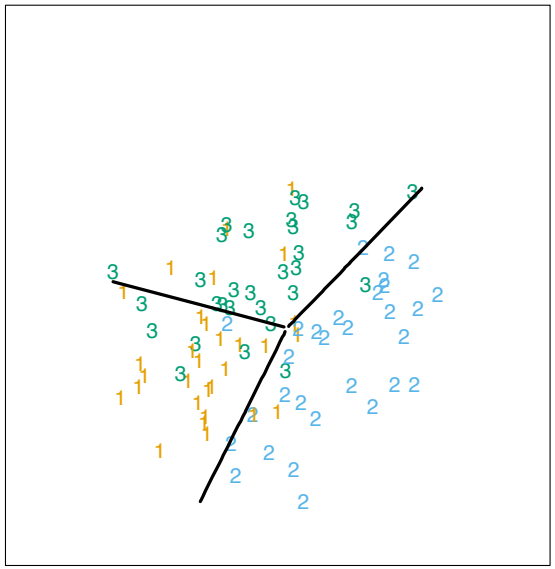


FIGURE 4.5. *The left panel shows three Gaussian distributions, with the same covariance and different means. Included are the contours of constant density enclosing* 95% *of the probability in each case. The Bayes decision boundaries between each pair of classes are shown (broken straight lines), and the Bayes decision boundaries separating all three classes are the thicker solid lines (a subset of the former). On the right we see a sample of* 30 *drawn from each Gaussian distribution, and the fitted LDA decision boundaries.*

Gaussian In-class Distributions

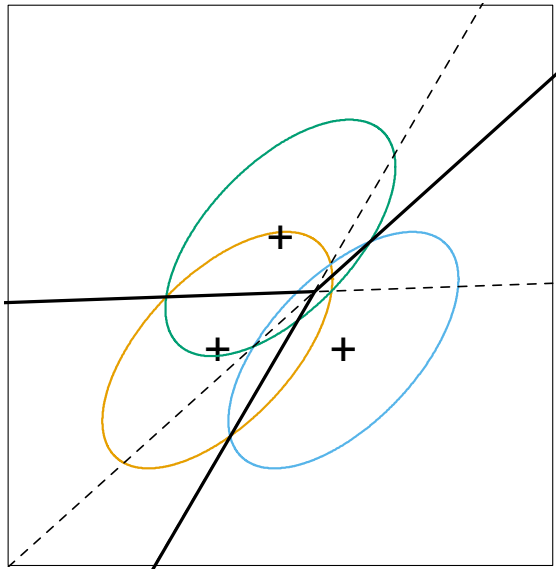


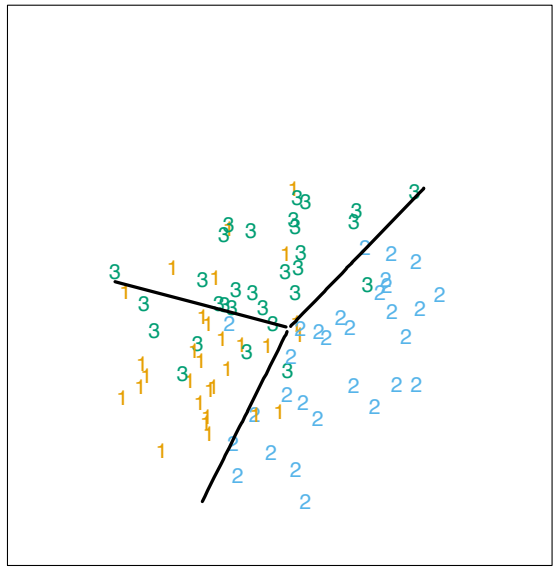


Plug-in estimator for continuous features:

f_k is **multivariate Gaussian** $\mathcal{N}(\mu_k, \Sigma_k)$ in p dimensions

Gaussian In-class Distributions





Plug-in estimator for continuous features:

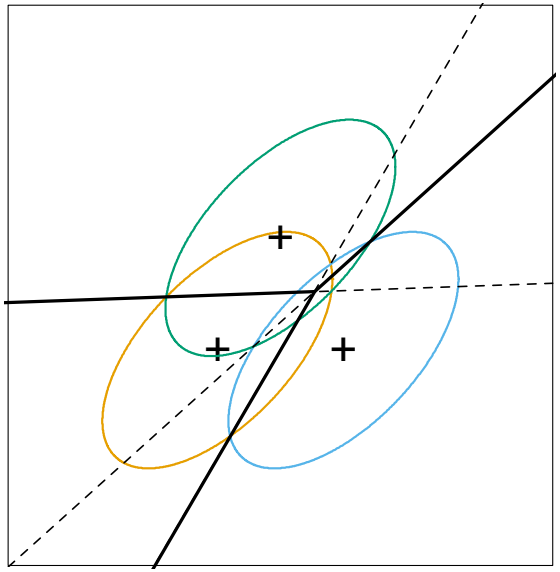
f_k is **multivariate Gaussian** $\mathcal{N}(\mu_k, \Sigma_k)$ in p dimensions

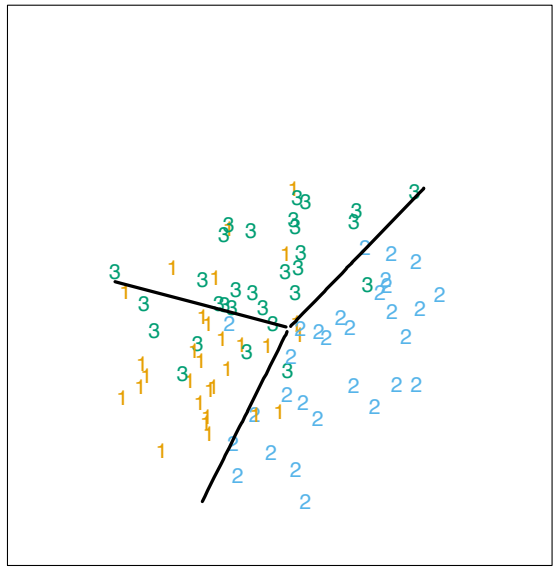
Parameter estimates from training data:

$$\hat{\mu}_k = \frac{1}{n_k} \sum_{i:Y_i=k} X_i \qquad \hat{\pi}_k = \frac{n_k}{N}$$

where $n_k = |\{i : Y_i = k\}|$ is the nr. of observations in class k.

Gaussian In-class Distributions





Plug-in estimator for continuous features:

f_k is **multivariate Gaussian** $\mathcal{N}(\mu_k, \Sigma_k)$ in p dimensions

Parameter estimates from training data:

p parameters per class

$$\hat{\mu}_k = \frac{1}{n_k} \sum_{i:Y_i=k} X_i$$

1 parameter per class

$$\hat{\pi}_k = \frac{n_k}{N}$$

where $n_k = |\{i : Y_i = k\}|$ is the nr. of observations in class k.

Estimating the Covariance Matrix: LDA vs QDA

Quadratic Discriminant Analysis (QDA):

$$\hat{\Sigma}_k = \frac{1}{n_k - 1} \sum_{i:Y_i=k} (X_i - \hat{\mu}_k)(X_i - \hat{\mu}_k)^\intercal \qquad \text{for } k = 1, \ldots, K$$

Estimating the Covariance Matrix: LDA vs QDA

Quadratic Discriminant Analysis (QDA):

$$\hat{\Sigma}_k = \frac{1}{n_k - 1} \sum_{i:Y_i=k} (X_i - \hat{\mu}_k)(X_i - \hat{\mu}_k)^\intercal \qquad \text{for } k = 1, \ldots, K$$

Linear Discriminant Analysis (LDA):

Take $\Sigma_k = \Sigma$ **the same for all classes** $k = 1, \ldots, K$, so we can pool data across classes to estimate it:

$$\hat{\Sigma} = \frac{1}{N - K} \sum_{k=1}^{K} \sum_{i:Y_i=k} (X_i - \hat{\mu}_k)(X_i - \hat{\mu}_k)^\intercal$$

Estimating the Covariance Matrix: LDA vs QDA

Quadratic Discriminant Analysis (QDA):

$$\hat{\Sigma}_k = \frac{1}{n_k - 1} \sum_{i:Y_i=k} (X_i - \hat{\mu}_k)(X_i - \hat{\mu}_k)^\intercal \qquad \text{for } k = 1, \ldots, K$$

Roughly $\frac{1}{2}p^2$ parameters per class $k \Rightarrow \frac{1}{2}Kp^2$ in total

Linear Discriminant Analysis (LDA):

Take $\Sigma_k = \Sigma$ **the same for all classes** $k = 1, \ldots, K$, so we can pool data across classes to estimate it:

$$\hat{\Sigma} = \frac{1}{N - K} \sum_{k=1}^{K} \sum_{i:Y_i=k} (X_i - \hat{\mu}_k)(X_i - \hat{\mu}_k)^\intercal$$

Roughly $\frac{1}{2}p^2$ parameters in total,
much less than QDA

Shape of the Decision Boundary

$$
\begin{aligned}
\arg\max_k \hat{f}_k(X) \cdot \hat{\pi}_k &= \arg\max_k \log \hat{f}_k(X) + \log \hat{\pi}_k \\
&= \arg\max_k \log \left(\frac{1}{(2\pi)^{p/2} |\hat{\Sigma}_k|^{1/2}} e^{-\frac{1}{2}(X-\hat{\mu}_k)^\intercal \hat{\Sigma}_k^{-1}(X-\hat{\mu}_k)} \right) + \log \hat{\pi}_k \\
&= \arg\max_k -\tfrac{1}{2} \log |\hat{\Sigma}_k| - \tfrac{1}{2}(X-\hat{\mu}_k)^\intercal \hat{\Sigma}_k^{-1}(X-\hat{\mu}_k) + \log \hat{\pi}_k
\end{aligned}
$$

Shape of the Decision Boundary

$$
\begin{aligned}
\arg\max_k \hat{f}_k(X) \cdot \hat{\pi}_k &= \arg\max_k \log \hat{f}_k(X) + \log \hat{\pi}_k \\
&= \arg\max_k \log \left(\frac{1}{(2\pi)^{p/2} |\hat{\Sigma}_k|^{1/2}} e^{-\frac{1}{2}(X-\hat{\mu}_k)^\intercal \hat{\Sigma}_k^{-1}(X-\hat{\mu}_k)} \right) + \log \hat{\pi}_k \\
&= \arg\max_k \underbrace{-\tfrac{1}{2} \log |\hat{\Sigma}_k| - \tfrac{1}{2}(X-\hat{\mu}_k)^\intercal \hat{\Sigma}_k^{-1}(X-\hat{\mu}_k) + \log \hat{\pi}_k}_{\delta_k(X)}
\end{aligned}
$$

Shape of the Decision Boundary

$$\arg\max_k \hat{f}_k(X) \cdot \hat{\pi}_k$$

$$= \arg\max_k \underbrace{-\tfrac{1}{2}\log|\hat{\Sigma}_k| - \tfrac{1}{2}(X-\hat{\mu}_k)^\intercal \hat{\Sigma}_k^{-1}(X-\hat{\mu}_k) + \log\hat{\pi}_k}_{\delta_k(X)}$$

Shape of the Decision Boundary

$$\arg\max_k \hat{f}_k(X) \cdot \hat{\pi}_k$$

$$= \arg\max_k \underbrace{-\tfrac{1}{2}\log|\hat{\Sigma}_k| - \tfrac{1}{2}(X-\hat{\mu}_k)^\intercal \hat{\Sigma}_k^{-1}(X-\hat{\mu}_k) + \log\hat{\pi}_k}_{\delta_k(X)}$$

Decision boundary between any classes k_1 and k_2: x such that

$$\delta_{k_1}(x) = \delta_{k_2}(x) \quad \Leftrightarrow \quad \delta_{k_1}(x) - \delta_{k_2}(x) = 0$$

- Quadratic decision boundary $\Rightarrow$ "Quadratic Discriminant Analysis"

Shape of the Decision Boundary

$$\arg\max_k \hat{f}_k(X) \cdot \hat{\pi}_k$$

$$= \arg\max_k \underbrace{-\tfrac{1}{2}\log|\hat{\Sigma}_k| - \tfrac{1}{2}(X-\hat{\mu}_k)^\intercal \hat{\Sigma}_k^{-1}(X-\hat{\mu}_k) + \log\hat{\pi}_k}_{\delta_k(X)}$$

Decision boundary between any classes k_1 and k_2: x such that

$$\delta_{k_1}(x) = \delta_{k_2}(x) \quad \Leftrightarrow \quad \delta_{k_1}(x) - \delta_{k_2}(x) = 0$$

- Quadratic decision boundary $\Rightarrow$ "Quadratic Discriminant Analysis"
- If $\Sigma_k = \Sigma$ for all k, then

$$(x-\hat{\mu}_k)^\intercal \hat{\Sigma}^{-1}(x-\hat{\mu}_k) = x^\intercal \hat{\Sigma}^{-1} x - 2x\hat{\Sigma}^{-1}\hat{\mu}_k + \hat{\mu}_k^\intercal \hat{\Sigma}^{-1}\hat{\mu}_k$$

Shape of the Decision Boundary

$$\arg\max_k \hat{f}_k(X) \cdot \hat{\pi}_k$$

$$= \arg\max_k \underbrace{-\tfrac{1}{2}\log|\hat{\Sigma}_k| - \tfrac{1}{2}(X-\hat{\mu}_k)^\intercal \hat{\Sigma}_k^{-1}(X-\hat{\mu}_k) + \log\hat{\pi}_k}_{\delta_k(X)}$$

Decision boundary between any classes k_1 and k_2: x such that

$$\delta_{k_1}(x) = \delta_{k_2}(x) \quad \Leftrightarrow \quad \delta_{k_1}(x) - \delta_{k_2}(x) = 0$$

- Quadratic decision boundary $\Rightarrow$ "Quadratic Discriminant Analysis"
- If $\Sigma_k = \Sigma$ for all k, then

$$(x-\hat{\mu}_k)^\intercal \hat{\Sigma}^{-1}(x-\hat{\mu}_k) = \underbrace{x^\intercal \hat{\Sigma}^{-1} x} - 2x\hat{\Sigma}^{-1}\hat{\mu}_k + \hat{\mu}_k^\intercal \hat{\Sigma}^{-1}\hat{\mu}_k$$

quadratic part does not depend on k, so cancels in $\delta_{k_1}(x) - \delta_{k_2}(x)$

Shape of the Decision Boundary

$$\arg\max_k \hat{f}_k(X) \cdot \hat{\pi}_k$$

$$= \arg\max_k \underbrace{-\tfrac{1}{2}\log|\hat{\Sigma}_k| - \tfrac{1}{2}(X-\hat{\mu}_k)^\intercal \hat{\Sigma}_k^{-1}(X-\hat{\mu}_k) + \log\hat{\pi}_k}_{\delta_k(X)}$$

Decision boundary between any classes k_1 and k_2: x such that

$$\delta_{k_1}(x) = \delta_{k_2}(x) \quad \Leftrightarrow \quad \delta_{k_1}(x) - \delta_{k_2}(x) = 0$$

- Quadratic decision boundary $\Rightarrow$ "Quadratic Discriminant Analysis"
- If $\Sigma_k = \Sigma$ for all k, then

$$(x-\hat{\mu}_k)^\intercal \hat{\Sigma}^{-1}(x-\hat{\mu}_k) = \underbrace{x^\intercal \hat{\Sigma}^{-1} x} - 2x\hat{\Sigma}^{-1}\hat{\mu}_k + \hat{\mu}_k^\intercal \hat{\Sigma}^{-1}\hat{\mu}_k$$

quadratic part does not depend on k, so cancels in $\delta_{k_1}(x) - \delta_{k_2}(x)$
$\Rightarrow$ "Linear Discriminant Analysis"

Outline

Plug-in Estimators for Classification

Linear and Quadratic Discriminant Analysis (Plug-in Estimators)

1. Gaussian In-class Distributions
2. Estimating the Covariance Matrix: LDA vs QDA
3. Shape of the Decision Boundary

Naive Bayes Classifier (Plug-in Estimator)

4. Naive Bayes
5. Discrete Features with Application to Spam Detection
6. Parameter Estimation
7. Discussion

Naive Bayes

Model: features are **independent** given their class k

For each class k and feature vector $X_i \in \mathbb{R}^p$:

$$f_k(X_i) = \prod_{j=1}^{p} f_{kj}(X_{ij})$$

Naive Bayes

Model: features are **independent** given their class k

For each class k and feature vector $X_i \in \mathbb{R}^p$:

$$f_k(X_i) = \prod_{j=1}^{p} f_{kj}(X_{ij})$$

- Independent features very useful simplification if p is very large, because then we have no hope of learning the true joint distribution exactly anyway

Naive Bayes

Model: features are **independent** given their class k

For each class k and feature vector $X_i \in \mathbb{R}^p$:

$$f_k(X_i) = \prod_{j=1}^{p} f_{kj}(X_{ij})$$

For continuous features $X_{ij} \in \mathbb{R}$:
For each feature j and every class k:
Estimate f_{kj} by fitting a univariate Gaussian $\mathcal{N}(\mu_{kj}, \sigma_{kj}^2)$

Naive Bayes

Model: features are **independent** given their class k

For each class k and feature vector $X_i \in \mathbb{R}^p$:

$$f_k(X_i) = \prod_{j=1}^{p} f_{kj}(X_{ij})$$

For continuous features $X_{ij} \in \mathbb{R}$:
For each feature j and every class k:
Estimate f_{kj} by fitting a univariate Gaussian $\mathcal{N}(\mu_{kj}, \sigma_{kj}^2)$
$\Rightarrow$ f_k multivariate Gaussian $\mathcal{N}\left(\mu_k, \Sigma_k\right)$ with

$$\mu_k = \begin{pmatrix} \mu_{k1} \\ \vdots \\ \mu_{kp} \end{pmatrix}, \qquad \Sigma_k = \begin{pmatrix} \sigma_{k1}^2 & & \\ & \ddots & \\ & & \sigma_{kp}^2 \end{pmatrix}$$

Naive Bayes

Model: features are **independent** given their class k

For each class k and feature vector $X_i \in \mathbb{R}^p$:

$$f_k(X_i) = \prod_{j=1}^{p} f_{kj}(X_{ij})$$

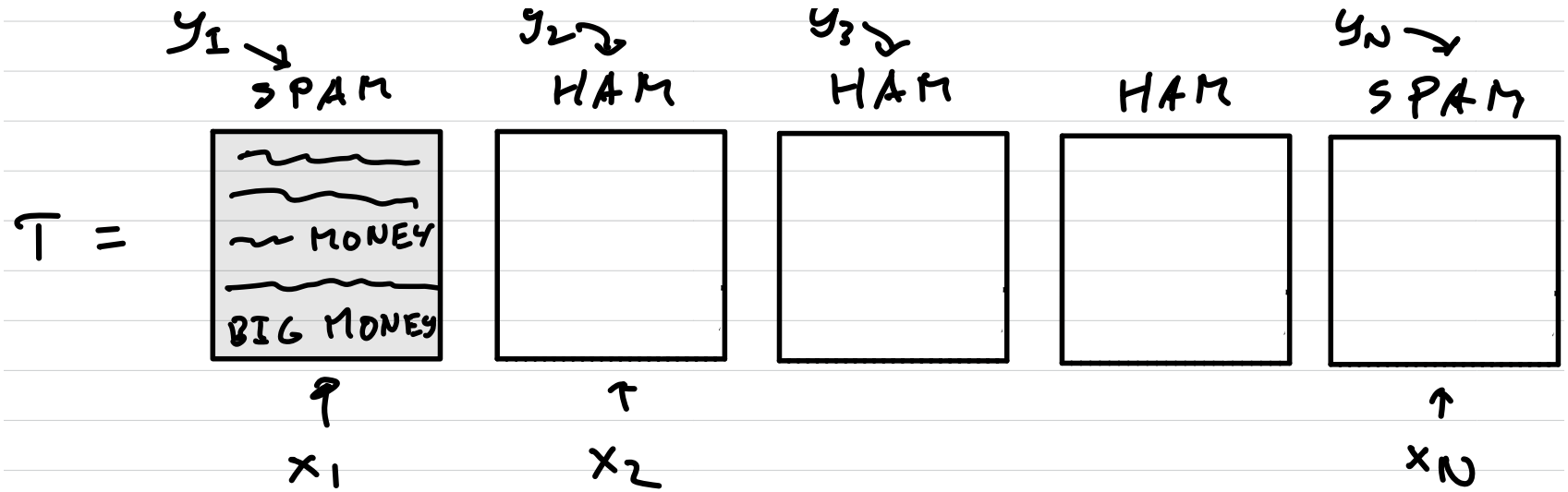
For continuous features $X_{ij} \in \mathbb{R}$:
For each feature j and every class k:
Estimate f_{kj} by fitting a univariate Gaussian $\mathcal{N}(\mu_{kj}, \sigma_{kj}^2)$
$\Rightarrow$ f_k multivariate Gaussian $\mathcal{N}\left(\mu_k, \Sigma_k\right)$ with

$$\mu_k = \begin{pmatrix} \mu_{k1} \\ \vdots \\ \mu_{kp} \end{pmatrix}, \qquad \Sigma_k = \begin{pmatrix} \sigma_{k1}^2 & & \\ & \ddots & \\ & & \sigma_{kp}^2 \end{pmatrix}$$

- Very similar to QDA, but restricts to diagonal covariance matrix **($2p$ parameters per class k)**

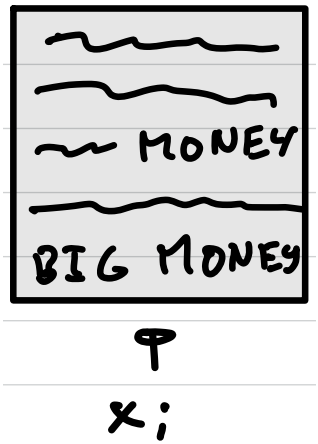
Discrete Features

Consider SPAM classification as running example:



How to **represent** e-mails (or other text documents) for classification?

Discrete Features



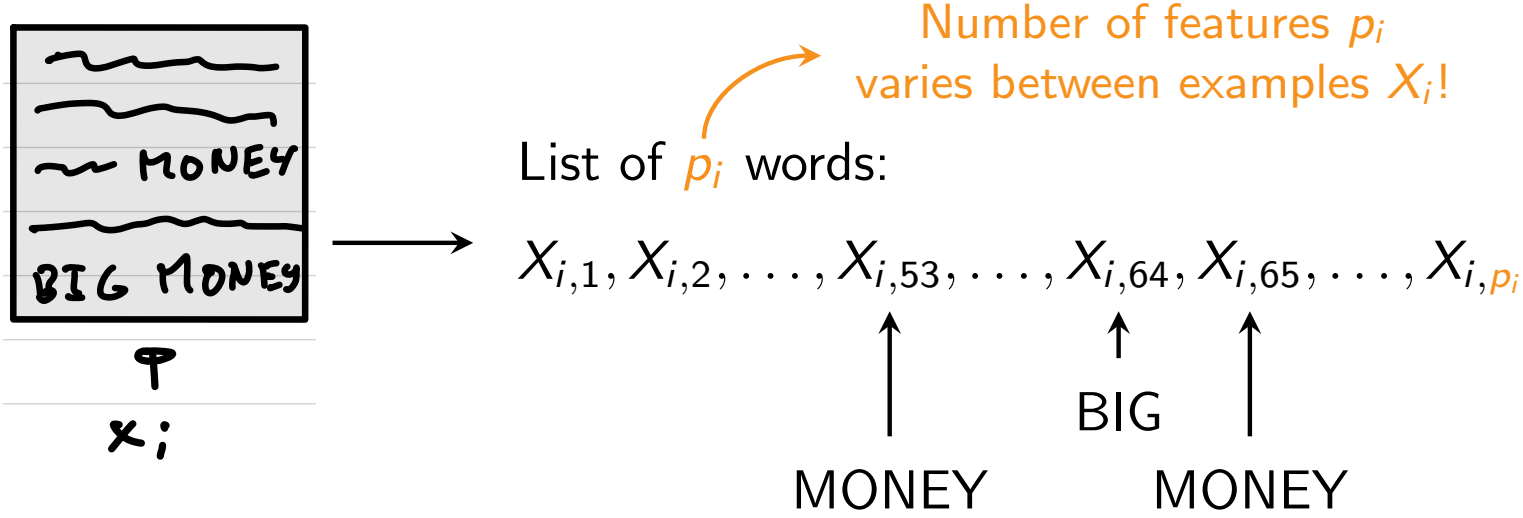
Number of features p_i varies between examples X_i!

List of p_i words:

$$X_{i,1}, X_{i,2}, \ldots, X_{i,53}, \ldots, X_{i,64}, X_{i,65}, \ldots, X_{i,p_i}$$

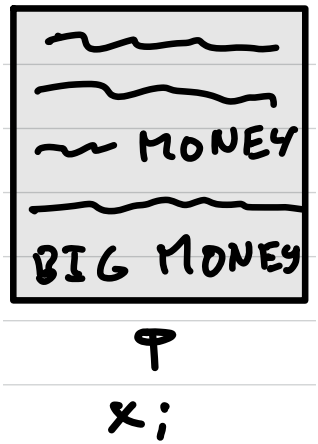
MONEY BIG MONEY

Discrete Features



1. Only keep words from a known **dictionary** $z_1, \ldots, z_d$ of d known words. E.g. $d = 10\,000$.

Discrete Features



Number of features p_i varies between examples X_i!

List of p_i words:

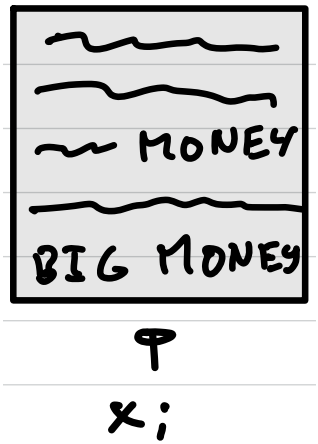
$$X_{i,1}, X_{i,2}, \ldots, X_{i,53}, \ldots, X_{i,64}, X_{i,65}, \ldots, X_{i,p_i}$$

MONEY BIG MONEY

1. Only keep words from a known **dictionary** $z_1, \ldots, z_d$ of d known words. E.g. $d = 10\,000$.
2. Model **words independently** per class k (Naive Bayes assumption):

$$f_k(X_i) = \prod_{j=1}^{p_i} f_{kj}(X_{ij})$$

Discrete Features



Number of features p_i varies between examples X_i!

List of p_i words:

$$X_{i,1}, X_{i,2}, \ldots, X_{i,53}, \ldots, X_{i,64}, X_{i,65}, \ldots, X_{i,p_i}$$

MONEY BIG MONEY

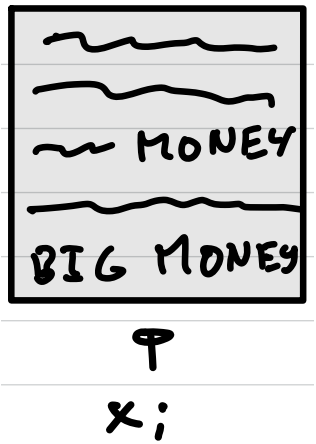
1. Only keep words from a known **dictionary** $z_1, \ldots, z_d$ of d known words. E.g. $d = 10\,000$.
2. Model **words independently** per class k (Naive Bayes assumption):

$$f_k(X_i) = \prod_{j=1}^{p_i} f_{kj}(X_{ij})$$

3. Further assume f_{kj} is the **same for all j** (allows variable p_i):

$$f_{kj}(X_{ij} = z_a) = \theta_{k,a} \qquad \left(\theta_{k,a} \geq 0,\ \sum_{a=1}^{d} \theta_{k,a} = 1\right)$$

Discrete Features



Number of features p_i varies between examples X_i!

List of p_i words:

$$X_{i,1}, X_{i,2}, \ldots, X_{i,53}, \ldots, X_{i,64}, X_{i,65}, \ldots, X_{i,p_i}$$

MONEY BIG MONEY

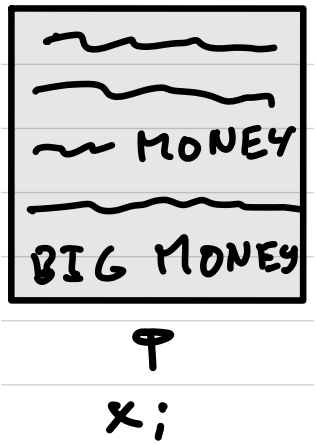
1. Only keep words from a known **dictionary** $z_1, \ldots, z_d$ of d known words. E.g. $d = 10\,000$.
2. Model **words independently** per class k (Naive Bayes assumption):

$$f_k(X_i) = \prod_{j=1}^{p_i} f_{kj}(X_{ij}) = \prod_{j=1}^{p_i} \theta_{k,X_{ij}}$$

3. Further assume f_{kj} is the **same for all j** (allows variable p_i):

$$f_{kj}(X_{ij} = z_a) = \theta_{k,a} \qquad \left(\theta_{k,a} \geq 0,\ \sum_{a=1}^{d} \theta_{k,a} = 1\right)$$

Discrete Features



Number of features p_i varies between examples X_i!

List of p_i words:

$$X_{i,1}, X_{i,2}, \ldots, X_{i,53}, \ldots, X_{i,64}, X_{i,65}, \ldots, X_{i,p_i}$$

MONEY BIG MONEY

1. Only keep words from a known **dictionary** $z_1, \ldots, z_d$ of d known words. E.g. $d = 10\,000$.
2. Model **words independently** per class k (Naive Bayes assumption):

$$f_k(X_i) = \prod_{j=1}^{p_i} f_{kj}(X_{ij}) = \prod_{j=1}^{p_i} \theta_{k,X_{ij}} = \prod_{a=1}^{d} \theta_{k,a}^{n_{i,a}} \qquad n_{i,a} = |\{j \mid X_{ij} = z_a\}|$$

3. Further assume f_{kj} is the **same for all j** (allows variable p_i):

$$f_{kj}(X_{ij} = z_a) = \theta_{k,a} \qquad \left(\theta_{k,a} \geq 0,\ \sum_{a=1}^{d} \theta_{k,a} = 1\right)$$

Bag of Words Representation

In-class probability of X_i

$$f_k(X_i) = \prod_{a=1}^{d} \theta_{k,a}^{n_{i,a}}$$

only depends on how often each dictionary word z_a occurs in X_i:

$$n_{i,a} = |\{j \mid X_{ij} = z_a\}|$$

So we have completely forgotten about the ordering of words and are treating X_i as a **bag of words**.

Bag of Words Representation

In-class probability of X_i

$$f_k(X_i) = \prod_{a=1}^{d} \theta_{k,a}^{n_{i,a}}$$

only depends on how often each dictionary word z_a occurs in X_i:

$$n_{i,a} = |\{j \mid X_{ij} = z_a\}|$$

So we have completely forgotten about the ordering of words and are treating X_i as a **bag of words**.

It is therefore sufficient to represent X_i instead by only the word counts:

$$\tilde{X}_i = \begin{pmatrix} n_{i,1} \\ \vdots \\ n_{i,d} \end{pmatrix}$$

Parameter Estimation

$$\hat{\pi}_k = \frac{\text{nr. of e-mails in class } k}{\text{total nr. of e-mails}} = \frac{|\{i : Y_i = k\}|}{N}$$

Parameter Estimation

$$\hat{\pi}_k = \frac{\text{nr. of e-mails in class } k}{\text{total nr. of e-mails}} = \frac{|\{i : Y_i = k\}|}{N}$$

$$\hat{\theta}_{k,a} = \frac{\text{nr. of occurrences of word } z_a \text{ in e-mails of class } k + \alpha}{\text{total nr. of words in e-mails of class } k + \alpha d}$$

$$= \frac{\sum_{i:Y_i=k} n_{i,a} \; + \; \alpha}{\sum_{i:Y_i=k} p_i \; + \; \alpha d} = \frac{\sum_{i:Y_i=k} n_{i,a} \; + \; \alpha}{\sum_{a'=1}^{d} \left(\sum_{i:Y_i=k} n_{i,a'} + \alpha \right)}$$

$\alpha > 0$ is a small number.

Parameter Estimation

$$\hat{\pi}_k = \frac{\text{nr. of e-mails in class } k}{\text{total nr. of e-mails}} = \frac{|\{i : Y_i = k\}|}{N}$$

$$\hat{\theta}_{k,a} = \frac{\text{nr. of occurrences of word } z_a \text{ in e-mails of class } k + \alpha}{\text{total nr. of words in e-mails of class } k + \alpha d}$$
$$= \frac{\sum_{i:Y_i=k} n_{i,a} \; + \; \alpha}{\sum_{i:Y_i=k} p_i \; + \; \alpha d} = \frac{\sum_{i:Y_i=k} n_{i,a} \; + \; \alpha}{\sum_{a'=1}^{d} \left(\sum_{i:Y_i=k} n_{i,a'} + \alpha \right)}$$

$\alpha > 0$ is a small number. Interpretation: add **fake data containing each word α times** $\Rightarrow$ prevents $\hat{\theta}_{k,a}$ from being zero.

Without α:

- If SPAM e-mail contains one word we have never seen in SPAM e-mails in the training data, then $\hat{f}_{\text{SPAM}}(X) = 0$ and e-mail will be classified as HAM!
- So single word could throw off classification.

Discussion

Too simple/naive?

- Only need $\hat{P}(Y = k_1|X) > \hat{P}(Y = k_2|X)$ whenever $P^*(Y = k_1|X) > P^*(Y = k_2|X)$
- Possible even if $\hat{P}$ is poor estimate of P^*!

Thus good classification $\not\Longrightarrow$ good probability estimates

Discussion

Too simple/naive?

- Only need $\hat{P}(Y = k_1|X) > \hat{P}(Y = k_2|X)$ whenever $P^*(Y = k_1|X) > P^*(Y = k_2|X)$
- Possible even if $\hat{P}$ is poor estimate of P^*!

Thus good classification $\not\Longrightarrow$ good probability estimates

Known Issue:

- Longer e-mails $\rightarrow$ more words $\rightarrow$ large influence on $\hat{\theta}_{k,a}$ estimates
- Somewhat undesirable.