Generative modeling of data

Instructor: Taylor Berg-Kirkpatrick
Slides: Sanjoy Dasgupta
Parametric versus nonparametric classifiers

Nearest neighbor classification:

size of classifier = size of data set

Nonparametric: Model complexity (\# of “parameters”) not fixed.

\( x \in \mathbb{R}^2 \)
\( y \in \{+, -, \emptyset\} \)
Parametric versus nonparametric classifiers

Nearest neighbor classification:

size of classifier = size of data set

Nonparametric: Model complexity (\# of “parameters”) not fixed.

- Good: Can fit any boundary, given enough data.
- Bad: Typically need a lot of data.
Parametric versus nonparametric classifiers

Nearest neighbor classification:

\[ \text{size of classifier} = \text{size of data set} \]

**Nonparametric:** Model complexity (\# of “parameters”) not fixed.

- Good: Can fit *any* boundary, given enough data.
- Bad: Typically need a lot of data.

**Parametric classifiers:** fixed number of parameters.
Parametric classifiers

Fixed number of parameters: e.g. lines.
Parametric classifiers

Fixed number of parameters: e.g. lines.

- Good: Can get a reasonable model with limited data
- Bad: Less flexibility in modeling
Parametric classifiers

Fixed number of parameters: e.g. lines.

- Good: Can get a reasonable model with limited data
- Bad: Less flexibility in modeling

Basic principle: pick a good approximation.
The generative approach to classification

Two ways to classify:

- **Generative**: model the individual classes.
- **Discriminative**: model the decision boundary between the classes.
Recall: Bayes’ rule

\[ \Pr(A|B) = \text{probability event } A \text{ occurs given that we have observed } B \text{ to occur} \]

\[ \Pr(A|B) = \frac{\Pr(A \text{ and } B)}{\Pr(B)} = \frac{\Pr(B|A)}{\Pr(B)} \times \Pr(A) \]
Recall: Bayes’ rule

\[ \Pr(A|B) = \frac{\Pr(A \text{ and } B)}{\Pr(B)} = \frac{\Pr(B|A)}{\Pr(B)} \times \Pr(A) \]

Example: Coin 1 has heads probability \( \frac{1}{6} \) and Coin 2 has heads probability \( \frac{1}{3} \). We close our eyes, pick a coin at random and flip it. It comes out heads. What is the probability it is Coin 1?

Parameters:
- \( p(A) \rightarrow \frac{1}{6} \) for Coin 1
- \( p(A) \rightarrow \frac{1}{3} \) for Coin 2
- \( A \in \{C1, C2\} \)
- \( B \in \{h, t\} \)

\[ \Pr(A = C1) = \frac{1}{2} \]
\[ \Pr(A = C2) = \frac{1}{2} \]

\[ \Pr(A = C1, B = h) = \frac{1}{12} \]
\[ \Pr(A = C2, B = h) = \frac{1}{6} \]

\[ \Pr(B = h) = \Pr(B = h|A = C1) \times \Pr(A = C1) + \Pr(B = h|A = C2) \times \Pr(A = C2) = \frac{1}{2} \]
Generative models

Example: data space $\mathcal{X} = \mathbb{R}$, classes/labels $\mathcal{Y} = \{1, 2, 3\}$

Caption: The diagram illustrates the joint distribution over $(x, y)$ via the probabilities of the classes, $\pi_j = \Pr(y = j)$, and the distribution of the individual classes, $P_1(x), P_2(x), P_3(x)$. The overall distribution is given by $\Pr(x, y) = \pi_y P_y(x)$. The notation $P(y) : \frac{\pi_1}{2} \frac{\pi_2}{3} \frac{\pi_3}{4}$ is used to depict the probability of classes 1, 2, and 3.
Generative models

Example: data space $\mathcal{X} = \mathbb{R}$, classes/labels $\mathcal{Y} = \{1, 2, 3\}$

Capture the joint distribution over $(x, y)$ via:

- the probabilities of the classes, $\pi_j = \Pr(y = j)$
- the distribution of the individual classes, $P_1(x), P_2(x), P_3(x)$
Generative models

Example: data space $\mathcal{X} = \mathbb{R}$, classes/labels $\mathcal{Y} = \{1, 2, 3\}$

Capture the joint distribution over $(x, y)$ via:

- the probabilities of the classes, $\pi_j = \Pr(y = j)$
- the distribution of the individual classes, $P_1(x), P_2(x), P_3(x)$

Overall distribution: $\Pr(x, y) = \sum_{y \in \mathcal{Y}} \pi_y P_y(x)$. 
Optimal predictions

For any data point \( x \in X \) and any candidate label \( j \),

\[
\Pr(y = j | x) = \frac{\Pr(y = j)}{\Pr(x)}
\]

\[
\Pr(x) = \sum_{k} \pi_j \Pr_j(x)
\]

Optimal prediction: the class \( j \) with largest \( \pi_j \Pr_j(x) \).
Optimal predictions

\[
\max_j f(j)
\]

\[
\arg\max_j f(j)
\]

\[
\text{setting of } j \text{ s.t. } f(j) \geq f(j') \quad \forall j'
\]

\[
\Pr(x)
\]

\[
P_1(x)
\]

\[
P_2(x)
\]

\[
P_3(x)
\]

\[
\pi_1 = 10\%
\]

\[
\pi_2 = 50\%
\]

\[
\pi_3 = 40\%
\]

\[
\Pr(y = j|x) = \frac{\Pr(y = j)\Pr(x|y = j)}{\Pr(x)} = \frac{\pi_j P_j(x)}{\sum_{i=1}^{k} \pi_i P_i(x)}
\]

\[
\text{Optimal prediction: the class } j \text{ with largest } \pi_j P_j(x)
\]

For any data point \( x \in \mathcal{X} \) and any candidate label \( j \),

\[
\hat{j} \quad \Pr(y = j|x) = \frac{\Pr(y = j)\Pr(x|y = j)}{\Pr(x)} = \frac{\pi_j P_j(x)}{\sum_{i=1}^{k} \pi_i P_i(x)}
\]
Optimal predictions

For any data point $x \in X$ and any candidate label $j$,

$$
Pr(y = j|x) = \frac{Pr(y = j)Pr(x|y = j)}{Pr(x)} = \frac{\pi_j P_j(x)}{\sum_{i=1}^k \pi_i P_i(x)}
$$

Optimal prediction: the class $j$ with largest $\pi_j P_j(x)$. 
A classification problem

You have a bottle of wine whose label is missing.

Which winery is it from, 1, 2, or 3?
A classification problem

You have a bottle of wine whose label is missing.

Which winery is it from, 1, 2, or 3?

Solve this problem using visual and chemical properties of the wine.
The data set

Training set obtained from 130 bottles

- Winery 1: 43 bottles
- Winery 2: 51 bottles
- Winery 3: 36 bottles
- For each bottle, 13 features:
The data set

Training set obtained from 130 bottles

- Winery 1: 43 bottles
- Winery 2: 51 bottles
- Winery 3: 36 bottles
- For each bottle, 13 features:
  - 'Alcohol', 'Malic acid', 'Ash', 'Alcalinity of ash', 'Magnesium',
  - 'Total phenols', 'Flavanoids', 'Nonflavanoid phenols',
  - 'Proanthocyanins',
  - 'Color intensity', 'Hue', 'OD280/OD315 of diluted wines',
  - 'Proline'

Also, a separate test set of 48 labeled points.
Recall: the generative approach

For any data point $x \in \mathcal{X}$ and any candidate label $j$,

$$
Pr(y = j|x) = \frac{Pr(y = j)Pr(x|y = j)}{Pr(x)} = \frac{\pi_j P_j(x)}{\sum_{i=1}^{k} \pi_i P_i(x)}
$$

Optimal prediction: the class $j$ with largest $\pi_j P_j(x)$. 
Fitting a generative model

Training set of 130 bottles:

- Winery 1: 43 bottles, winery 2: 51 bottles, winery 3: 36 bottles
- For each bottle, 13 features: 'Alcohol', 'Malic acid', 'Ash', 'Alcalinity of ash', 'Magnesium', 'Total phenols', 'Flavanoids', 'Nonflavanoid phenols', 'Proanthocyanins', 'Color intensity', 'Hue', 'OD280/OD315 of diluted wines', 'Proline'

\[
\mathcal{D} = \{ y^{(1)}, y^{(2)}, \ldots, y^{(130)} \} \\
\Theta = (\pi_1, \pi_2, \pi_3) \\
P_\Theta(\mathcal{D}) = \prod_{i=1}^{130} P_\Theta(y^{(i)}) = \prod_{i=1}^{130} \pi_{y^{(i)}}
\]
Fitting a generative model

Training set of 130 bottles:

- Winery 1: 43 bottles, winery 2: 51 bottles, winery 3: 36 bottles

Class weights:

\[ \pi_1 = \frac{43}{130} = 0.33, \quad \pi_2 = \frac{51}{130} = 0.39, \quad \pi_3 = \frac{36}{130} = 0.28 \]
Training set of 130 bottles:

- Winery 1: 43 bottles, winery 2: 51 bottles, winery 3: 36 bottles
- For each bottle, 13 features: 'Alcohol', 'Malic acid', 'Ash', 'Alcalinity of ash', 'Magnesium', 'Total phenols', 'Flavanoids', 'Nonflavanoid phenols', 'Proanthocyanins', 'Color intensity', 'Hue', 'OD280/OD315 of diluted wines', 'Proline'

Class weights:

\[ \pi_1 = \frac{43}{130} = 0.33, \quad \pi_2 = \frac{51}{130} = 0.39, \quad \pi_3 = \frac{36}{130} = 0.28 \]

Need distributions \( P_1, P_2, P_3 \), one per class.
Base these on a single feature: 'Alcohol'.
The univariate Gaussian

The Gaussian $N(\mu, \sigma^2)$ has mean $\mu$, variance $\sigma^2$, and density function

$$p(x) = \frac{1}{(2\pi\sigma^2)^{1/2}} \exp\left(-\frac{(x - \mu)^2}{2\sigma^2}\right).$$
The distribution for winery 1

Single feature: 'Alcohol'
The distribution for winery 1

Single feature: 'Alcohol'

Mean $\mu = 13.72$, Standard deviation $\sigma = 0.44$ (variance 0.20)

$\hat{\mu}, \hat{\sigma}^2 = \arg\max_{\mu, \sigma^2} P(D|\mu, \sigma^2)$
All three wineries

- $\pi_1 = 0.33$, $P_1 = N(13.7, 0.20)$
- $\pi_2 = 0.39$, $P_2 = N(12.3, 0.28)$
- $\pi_3 = 0.28$, $P_3 = N(13.2, 0.27)$
All three wineries

π₁ = 0.33, \( P_1 = N(13.7, 0.20) \)

π₂ = 0.39, \( P_2 = N(12.3, 0.28) \)

π₃ = 0.28, \( P_3 = N(13.2, 0.27) \)

To classify \( x \): Pick the \( j \) with highest \( \pi_j P_j(x) \)
All three wineries

To classify $x$: Pick the $j$ with highest $\pi_j P_j(x)$

Test error: $14/48 = 29\%$
All three wineries

- $\pi_1 = 0.33$, $P_1 = N(13.7, 0.20)$
- $\pi_2 = 0.39$, $P_2 = N(12.3, 0.28)$
- $\pi_3 = 0.28$, $P_3 = N(13.2, 0.27)$

To classify $x$: Pick the $j$ with highest $\pi_j P_j(x)$

Test error: $14/48 = 29\%$

What if we use two features?
The data set, again

Training set obtained from 130 bottles

- Winery 1: 43 bottles
- Winery 2: 51 bottles
- Winery 3: 36 bottles
- For each bottle, 13 features:
  - 'Alcohol', 'Malic acid', 'Ash', 'Alcalinity of ash', 'Magnesium', 'Total phenols', 'Flavanoids', 'Nonflavanoid phenols', 'Proanthocyanins', 'Color intensity', 'Hue', 'OD280/OD315 of diluted wines', 'Proline'

Also, a separate test set of 48 labeled points.
The data set, again

Training set obtained from 130 bottles

- Winery 1: 43 bottles
- Winery 2: 51 bottles
- Winery 3: 36 bottles

Also, a separate test set of 48 labeled points.

This time: ‘Alcohol’ and ‘Flavanoids’.
Why it helps to add features

Better separation between the classes!
Why it helps to add features

Better separation between the classes!

Error rate drops from 29% to 8%.
Why it helps to add features

Better separation between the classes!

Error rate drops from 29% to 8%.
Why it helps to add features

Better separation between the classes!

Error rate drops from 29% to 8%.
The bivariate Gaussian

\[ \text{mean } \mu = (13.7, 3.0) \]

\[ \text{covariance matrix } \Sigma = \begin{pmatrix} 0.20 & 0.06 \\ 0.06 & 0.12 \end{pmatrix} \]
The bivariate Gaussian

Model class 1 by a bivariate Gaussian, parametrized by:

\[
\text{mean } \mu = \begin{pmatrix} 13.7 \\ 3.0 \end{pmatrix} \quad \text{and covariance matrix } \Sigma = \begin{pmatrix} 0.20 & 0.06 \\ 0.06 & 0.12 \end{pmatrix}
\]
Suppose $X_1$ has mean $\mu_1$ and $X_2$ has mean $\mu_2$.

\[ \text{Cov}(X_1, X_2) = \text{Cov}(X_1, X_1) \]

Can measure dependence between them by their covariance:

- $\text{Cov}(X_1, X_2) = \mathbb{E}[(X_1 - \mu_1)(X_2 - \mu_2)] = \mathbb{E}[X_1X_2] - \mu_1\mu_2$
- Maximized when $X_1 = X_2$, in which case it is $\text{var}(X_1)$.
- It is at most $\text{std}(X_1)\text{std}(X_2)$. 
The bivariate (2-d) Gaussian

A distribution over \((x_1, x_2) \in \mathbb{R}^2\), parametrized by:

- **Mean** \((\mu_1, \mu_2) \in \mathbb{R}^2\), where \(\mu_1 = \mathbb{E}(X_1)\) and \(\mu_2 = \mathbb{E}(X_2)\)

- **Covariance matrix** \(\Sigma = \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix}\) where

\[
\begin{align*}
\Sigma_{11} &= \text{var}(X_1) \\
\Sigma_{22} &= \text{var}(X_2) \\
\Sigma_{12} &= \Sigma_{21} = \text{cov}(X_1, X_2)
\end{align*}
\]
The bivariate (2-d) Gaussian

A distribution over \((x_1, x_2) \in \mathbb{R}^2\), parametrized by:

- **Mean** \((\mu_1, \mu_2) \in \mathbb{R}^2\), where \(\mu_1 = \mathbb{E}(X_1)\) and \(\mu_2 = \mathbb{E}(X_2)\)

- **Covariance matrix** \(\Sigma = \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix}\) where

  \[
  \begin{cases} 
  \Sigma_{11} = \text{var}(X_1) \\
  \Sigma_{22} = \text{var}(X_2) \\
  \Sigma_{12} = \Sigma_{21} = \text{cov}(X_1, X_2)
  \end{cases}
  \]

Density is highest at the mean, falls off in ellipsoidal contours.
Density of the bivariate Gaussian

- **Mean** \((\mu_1, \mu_2) \in \mathbb{R}^2\), where \(\mu_1 = \mathbb{E}(X_1)\) and \(\mu_2 = \mathbb{E}(X_2)\)

- **Covariance matrix** \(\Sigma = \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix} \)

\[
p(x_1, x_2) = \frac{1}{2\pi|\Sigma|^{1/2}} \exp \left( -\frac{1}{2} \begin{bmatrix} x_1 - \mu_1 \\ x_2 - \mu_2 \end{bmatrix}^T \Sigma^{-1} \begin{bmatrix} x_1 - \mu_1 \\ x_2 - \mu_2 \end{bmatrix} \right)
\]

\[
\exp \left( -\frac{1}{2} \frac{(x - u)^T}{\sigma^2} \right)
\]
Bivariate Gaussian: examples

In either case, the mean is $(1, 1)$.

\[
\Sigma = \begin{bmatrix} 4 & 0 \\ 0 & 1 \end{bmatrix}
\]

\[
\Sigma = \begin{bmatrix} 4 & 1.5 \\ 1.5 & 1 \end{bmatrix}
\]
The decision boundary

Go from 1 to 2 features: error rate goes from 29% to 8%.

\[ y \in \{1, 2\}, \quad x \in \mathbb{R}^2 \]

\[ y = \begin{cases} 
    1 & \text{if } \sum p(y=1|x) > \sum p(y=2|x) \\
    2 & \text{o.w.}
\end{cases} \]

\[ s.t. \quad p(y=1|x) = \frac{p(x|y=1)p(y=1)}{p(x)} \]

\[ s.t. \quad \frac{1}{c_1} \exp \left(-\frac{1}{2} (x-\mu_1)^T \Sigma_1^{-1} (x-\mu_1)\right) \Pi_1 = \frac{1}{c_2} \exp \left(-\frac{1}{2} (x-\mu_2)^T \Sigma_2^{-1} (x-\mu_2)\right) \Pi_2 \]
The decision boundary

Go from 1 to 2 features: error rate goes from 29% to 8%.
The decision boundary

Go from 1 to 2 features: error rate goes from 29% to 8%.

What kind of function is this? And, can we use more features?