Generative models for classification

1. Prediction functions

Fish on conveyor belt

Goal: fish-packing plant wants to automate the process of sorting incoming fish on conveyor belt according to species.

- Salmon or sea-bass?
- Side-information: measurements/observations that may inform your prediction. E.g., length of the fish.
- How can we model this problem statistically?

Prediction problem

Two Galton boards put side-by-side (with some overlap):

- A ball is dropped from one of the boards. Random variable $Y$: which side ball was dropped from (0 or 1).
- You observe the (horizontal) position of the ball. Random variable $X$: position of ball (real number).
- Our task: given observation, predict side the ball was dropped from. Given $X$, predict value of $Y$. 

1 / 37
Statistical model

- Which side ball is dropped from is a coin toss.
  \[ Y \sim \text{Bern}(p). \]
- Distribution of ball position depends on which side it was dropped from.
  \[ X | Y = y \sim N(\mu_y, \sigma^2_y) \quad \text{for each } y \in \{0, 1\}. \]

Note: This model ignores effect of right board when ball is dropped from left board, and vice versa.

- Parameters: \( p \in [0, 1], \mu_0, \mu_1 \in \mathbb{R}, \sigma^2_0, \sigma^2_1 > 0 \)
  Collect into a parameter vector \( \theta = (p, \mu_0, \mu_1, \sigma^2_0, \sigma^2_1) \).

This is an example of a generative model for classification.

Bayes’ rule

Generative model specifies dist. of \( Y \) and conditional dist. of \( X \) given \( Y \).

Bayes’ rule:

\[
\Pr(Y = y | X = x) = \frac{\Pr(Y = y \land X = x)}{\Pr(X = x)} = \frac{\Pr(Y = y) \cdot \Pr(X = x | Y = y)}{\Pr(X = x)}.
\]

- Observe that
  \[
  \arg \max_{y \in \{0, 1\}} \Pr(Y = y | X = x) = \arg \max_{y \in \{0, 1\}} \Pr(Y = y) \cdot \Pr(X = x | Y = y),
  \]
  since denominator \( \Pr(X = x) \) in Bayes’ rule does not involve \( y \).

- When \( X \) is continuous random variable, use its (conditional) density in place of \( \Pr(X = x | Y = y) \).

Can plug-in expressions for \( \Pr(Y = y) \) and \( \Pr(X = x | Y = y) \) using model parameters \( \theta \).

Prediction strategy

Suppose we know \( \theta = (p, \mu_0, \mu_1, \sigma^2_0, \sigma^2_1) \).
How should we predict given that we observe \( X = x \)?

- If \( \Pr(Y = 1 | X = x) > 1/2 \), then predict 1.
- If \( \Pr(Y = 1 | X = x) \leq 1/2 \), then predict 0.

This defines a prediction function (a.k.a. predictor) \( f^*: \mathbb{R} \to \{0, 1\} \):

\[
f^*(x) = 1 \{ \Pr(Y = 1 | X = x) > 1/2 \} \quad \text{for all } x \in \mathbb{R},
\]

which is the same as

\[
f^*(x) = \arg \max_{y \in \{0, 1\}} \Pr(Y = y | X = x) \quad \text{for all } x \in \mathbb{R}.
\]

Using this strategy, what is the probability that you predict incorrectly?

\[
\Pr(f^*(x) \neq Y | X = x) = \min_{y \in \{0, 1\}} 1 - \Pr(Y = y | X = x);
\]

\[
\Pr(f^*(x) \neq Y) = \mathbb{E}_{y \in \{0, 1\}} \min_{y \in \{0, 1\}} 1 - \Pr(Y = y | X).
\]

This is the best you can do!

How do we implement this prediction strategy using knowledge of \( \theta \)?

Example

“Two Galton boards” model:

\[
Y \sim \text{Bern}(p),
\]
\[
X | Y = y \sim N(\mu_y, \sigma^2_y) \quad \text{for each } y \in \{0, 1\}.
\]

\[
\Pr(Y = y) \cdot \Pr(X = x | Y = y) = p^y(1-p)^{1-y} \cdot \frac{1}{\sqrt{2\pi\sigma^2_y}} \exp \left( -\frac{(x - \mu_y)^2}{2\sigma^2_y} \right).
\]

Optimal predictor:

\[
f^*(x) = \arg \max_{y \in \{0,1\}} p^y(1-p)^{1-y} \cdot \frac{1}{\sqrt{2\pi\sigma^2_y}} \exp \left( -\frac{(x - \mu_y)^2}{2\sigma^2_y} \right).
\]

Example:

\[
p = 0.7, \quad \mu_0 = 1, \quad \sigma^2_0 = 4, \quad \mu_1 = 0, \quad \sigma^2_1 = 1.
\]
General setting for classification problems

$(X, Y)$ is pair of random variables. Goal is to predict $Y$ after observing $X$.

- $X$ takes values in $\mathcal{X}$ (feature space).
  
  E.g., $\mathcal{X} = \mathbb{R}$.
  
  This is side-information that is supposed to help us predict $Y$.

- $Y$ takes values in $\mathcal{Y}$ (label space or output space).
  
  In this lecture, $\mathcal{Y} = \{1, \ldots, K\}$ or $\mathcal{Y} = \{0, 1\}$ (classification problems).

Use predictor $f : \mathcal{X} \to \mathcal{Y}$ to form prediction $\hat{Y} = f(X)$.

Risk of predictor $f$:

$$R(f) := P(f(X) \neq Y).$$

Optimal predictor $f^* : \mathcal{X} \to \mathcal{Y}$ with smallest risk is

$$f^*(x) = \arg \max_{y \in \mathcal{Y}} P(Y = y | X = x) \quad \text{for all } x \in \mathcal{X}.$$ 

Also called the Bayes predictor.

Note: optimal predictor depends on dist. of $(X, Y)$, which is typically unknown!

IID model

IID model: training data $(X_1, Y_1), \ldots, (X_n, Y_n)$ and test example $(X, Y)$ are $n+1$ iid pairs from probability distribution $P_0$ with parameter vector $\theta$.

1. Estimate unknowns $\theta$ using training data $(X_1, Y_1), \ldots, (X_n, Y_n)$.

2. Plug estimate $\hat{\theta}$ into formula for optimal predictor.

E.g., for “Two Galton boards” model: with $\hat{\theta} = (\hat{p}, \hat{\mu}_0, \hat{\sigma}_0^2, \hat{\sigma}_1^2)$, form predictor $\hat{f}$ given by

$$\hat{f}(x) := \arg \max_{y \in \{0, 1\}} \hat{p}^y (1 - \hat{p})^{1-y} \frac{1}{\sqrt{2\pi \hat{\sigma}_0^2}} \exp \left( -\frac{(x - \hat{\mu}_0)^2}{2\hat{\sigma}_0^2} \right).$$

We call $\hat{f}$ a plug-in predictor.

3. Prediction of $Y$ given $X$:

$$\hat{Y} := \hat{f}(X).$$

Maximum likelihood estimation

Parametric statistical model:

$\mathcal{P} = \{P_0 : \theta \in \Theta\}$, a collection of probability distributions for observed data.

- $\Theta$: parameter space.

- $\theta \in \Theta$: a particular parameter vector.

- $P_0$: a particular probability distribution for observed data.

Likelihood of $\theta \in \Theta$ given observed data $z$:

$$\mathcal{L}(\theta) := P_0(z).$$

Maximum likelihood estimator (MLE):

Let $\hat{\theta}$ be the $\theta \in \Theta$ of highest likelihood given observed data.
MLE for “Two Galton boards” model

\[ \mathcal{P} = \text{distributions on } (X_1, Y_1), \ldots, (X_n, Y_n) \text{ treated as iid and} \]
\[ Y_i \sim \text{Bern}(p); \quad X_i \mid Y_i = y \sim N(\mu_y, \sigma_y^2) \quad \text{for each } y \in \{0, 1\}. \]

1. \( \Theta = \{ \theta = (p, \mu_0, \mu_1, \sigma_0^2, \sigma_1^2) : 0 \leq p \leq 1; \mu_0, \mu_1 \in \mathbb{R}; \sigma_0^2, \sigma_1^2 > 0 \}. \)

2. Likelihood of \( \theta = (p, \mu_0, \mu_1, \sigma_0^2, \sigma_1^2) \) given data \((X_1, Y_1), \ldots, (X_n, Y_n) = ((x_1, y_1), \ldots, (x_n, y_n)):\)
\[
\mathcal{L} = \prod_{i=1}^{n} \left\{ \left( \frac{1}{2p(1-p)} \right)^{x_i} \left( \frac{1}{\sqrt{2\pi \sigma_0^2}} \right)^{y_i} \exp \left( -\frac{(x_i - \mu_0)^2}{2\sigma_0^2} \right) \right\}^{1-y_i} \times \left\{ \left( 1-p \right)^{x_i} \left( \frac{1}{\sqrt{2\pi \sigma_1^2}} \right)^{y_i} \exp \left( -\frac{(x_i - \mu_1)^2}{2\sigma_1^2} \right) \right\}^{y_i}.
\]

3. Using calculus, we find that the maximizing value of \( \theta \) is given by
\[
\hat{p} := \frac{|S_1|}{n}, \quad \hat{\mu}_0 := \text{Sample Mean}(S_0), \quad \hat{\mu}_1 := \text{Sample Mean}(S_1),
\]
\[
\hat{\sigma}_0^2 := \text{Sample Variance}(S_0), \quad \hat{\sigma}_1^2 := \text{Sample Variance}(S_1),
\]
where \( S_0 := \{ x_i : y_i = 0 \} \) and \( S_1 := \{ x_i : y_i = 1 \}. \)

**Example**

Plug-in predictor:
\[
\hat{f}(x) = \begin{cases} 
1 & \text{if } x \in [0.38, 2.29]; \\
0 & \text{otherwise}.
\end{cases}
\]

Dotted lines = decision boundary.

(Here, \( \hat{\pi}_0 = 1 - \hat{p} \) and \( \hat{\pi}_1 = \hat{p} \).)

### Generative models for classification

A **generative model for classification** has the following components:

- **Distribution of \( Y \): class prior**
  - E.g., categorical distribution specified by \( \pi_y = \mathbb{P}(Y = y) \) for each \( y \in \mathcal{Y} \).

- **Conditional distributions of \( X \) given \( Y \): class conditional distributions**
  - E.g., \( X \mid Y = y \sim N(\mu_y, \sigma_y^2) \) for each \( y \in \mathcal{Y} \).

**Parameter estimation:**

When class prior & class conditional distributions have disjoint parameters — e.g., \( \theta = (\pi_1, \pi_K, \theta_1, \ldots, \theta_K) \) — then MLE \( \hat{\theta} = (\hat{\pi}_1, \ldots, \hat{\pi}_K, \hat{\theta}_1, \ldots, \hat{\theta}_K) \) given data \( ((x_1, y_1), \ldots, (x_n, y_n)) \) decomposes as follows:

- **(\( \hat{\pi}_1, \ldots, \hat{\pi}_K \)):** MLE for \( (\pi_1, \ldots, \pi_K) \) given \( (y_1, \ldots, y_n) \)
  - (i.e., \( \hat{\pi}_1, \ldots, \hat{\pi}_K \) only depends on labels).

- **\( \hat{\theta}_y \):** MLE for \( \theta_y \) given \( (x_i : y_i = y) \), for each \( y \in \mathcal{Y} \)
  - (i.e., \( \hat{\theta}_y \) only depends on \( x_i \)'s with label \( y \)).

Proof on next slide.

### Decomposability of MLE for generative models

Log-likelihood of \( \pi \) and \( (\theta_y : y \in \mathcal{Y}) \):
\[
\log \prod_{i=1}^{n} \prod_{y \in \mathcal{Y}} \left[ \pi_y \cdot \mathbb{P}_{\theta_y}(X = x_i) \right]^{\mathbb{1}_{\{y_i = y\}}}
\]
\[
= \sum_{i=1}^{n} \sum_{y \in \mathcal{Y}} \left[ \mathbb{1}_{\{y_i = y\}} \log \pi_y + \mathbb{1}_{\{y_i = y\}} \log \mathbb{P}_{\theta_y}(X = x_i) \right]
\]
\[
= \sum_{i=1}^{n} \sum_{y \in \mathcal{Y}} \mathbb{1}_{\{y_i = y\}} \log \pi_y + \sum_{y \in \mathcal{Y}} \sum_{y_i = y} \mathbb{P}_{\theta_y}(X = x_i)
\]

- **\( \pi \) only involved in term \( T_{\pi} \), which is log-likelihood given \( (y_1, \ldots, y_n) \).**
- **\( \theta_y \) only involved in term \( T_y \), which is log-likelihood given \( (x_i : y_i = y) \).**

These terms \( (T_{\pi} \text{ and } T_y \text{ for each } y \in \mathcal{Y}) \) can be maximized separately to maximize the overall log-likelihood objective.
3. **Naïve Bayes models**

**Naïve Bayes**: generative model where class conditional distributions treat features as independent.

$$P(X = x | Y = y) = \prod_{j=1}^{d} P(X_j = x_j | Y = y).$$

**Special case**: Naïve Bayes with binary features ($X = \{0, 1\}^d$):

$$X_j | Y = y \sim \text{Bern}(\mu_{y,j}).$$

E.g., $x_1 = 1$ (length > 1 meter), $x_2 = 1$ (tastes fishy), ...

Model parameters:

1. Class priors: $\pi = (\pi_1, \pi_2, \ldots, \pi_K)$.
2. Class conditionals: $\mu_y = (\mu_{y,1}, \mu_{y,2}, \ldots, \mu_{y,d})$ for each $y \in \{1, \ldots, K\}$.

---

**Structure of Naïve Bayes classifiers**

What is the form of Bayes classifier for a Naïve Bayes distribution?

Let $\pi, \mu_1, \ldots, \mu_K$ be the parameters of the distribution.

$$f^*(x) = \arg \max_{y \in \{1, \ldots, K\}} \log \left( \frac{P(Y = y) \cdot P(X = x | Y = y)}{n} \right).$$

$$= \arg \max_{y \in \{1, \ldots, K\}} \log \left( \frac{\pi_y \cdot \prod_{j=1}^{d} \mu_{y,j}^{x_j}(1 - \mu_{y,j})^{1-x_j}}{|S_y|} \right).$$

$$= \arg \max_{y \in \{1, \ldots, K\}} \log \left[ \frac{\pi_y \cdot \prod_{j=1}^{d} (1 - \mu_{y,j})}{|S_y|} \right] + \sum_{j=1}^{d} \log \left[ \frac{\mu_{y,j}}{1 - \mu_{y,j}} \right] \cdot x_j.$$

"Score" for class $y$ is an affine function of $x$.

Can pre-compute coefficients to speed-up classifier evaluation.

---

**Parameter estimation for Naïve Bayes with binary features**

Let $(x^{(1)}, y^{(1)}), \ldots, (x^{(n)}, y^{(n)})$ be the training data.

MLE for $\theta = (\pi, \mu_1, \ldots, \mu_K)$:

$$\hat{\pi}_y := \frac{|S_y|}{n},$$

$$\hat{\mu}_{y,j} := \frac{\sum_{x^{(i)} \in S_y} x^{(i)}_j}{|S_y|} \quad \text{for each } j = 1, \ldots, d.$$

where $S_y := (x^{(i)} : y^{(i)} = y)$ for each $y \in \{1, \ldots, K\}$.

Caveat: MLE is not a good estimator when $\hat{\mu}_{y,j}$ turns out to be 0 or 1.

Alternative to MLE: Laplace smoothing estimate

$$\hat{\mu}_{y,j} := \frac{1 + \sum_{x^{(i)} \in S_y} x^{(i)}_j}{2 + |S_y|} \in (0, 1).$$
**Example: 20 Newsgroups**

**Data set:** “20 Newsgroups”

- \( \approx 11 \times 10^3 \) messages from 20 message boards.
  - (“alt.atheism”, “comp.graphics”, “comp.os.ms-windows.misc”, \ldots)
- Extract vocabulary of \( d = 61188 \) unique words.
  - (“archive”, “name”, “atheism”, “resources”, \ldots)
- Represent each message as a binary vector \( \mathbf{x} \in \{0, 1\}^d \):
  \[ x_i = \mathbb{1}\{ \text{message contains the } i\text{-th vocabulary word} \} \]
  - E.g., \( x_1 = \mathbb{1}\{ \text{message contains “archive”} \} \).

**Goal:** Given a message (with message headers removed), predict which of the 20 message boards it was posted to.

We’ll fit the Naïve Bayes model (with MLE+Laplace smoothing) to this data.

Naïve Bayes predictor:

\[
\hat{f}(\mathbf{x}) = \arg \max_{y \in \{1, \ldots, 20\}} \log \left( \hat{\pi}_y \prod_{j=1}^{d} \left( 1 - \hat{\mu}_{y,j} \right) \right) + \sum_{j=1}^{d} \log \left( \frac{\hat{\mu}_{y,j}}{1 - \hat{\mu}_{y,j}} \right) \cdot x_j
\]

- The 29-th word in the vocabulary is “the”. What do you think \( \hat{\mu}_{y,29} \) is? (Probably should’ve removed stop words before fitting model. Oh well!)
- Class 1 is “alt.atheism”; class 17 is “talk.politics.guns”.
  - 38733-th word in vocabulary is “firearms”.
  \[
  \hat{\mu}_{1,38733} \approx 0.0021, \quad \hat{\mu}_{17,38733} \approx 0.1901
  \]
  so
  \[
  \log \left( \frac{\hat{\mu}_{17,38733}}{1 - \hat{\mu}_{17,38733}} \right) - \log \left( \frac{\hat{\mu}_{1,38733}}{1 - \hat{\mu}_{1,38733}} \right) \approx 4.7267.
  \]
- On separate collection of \( 7.5 \times 10^3 \) messages, get test error rate of 37.6%.

**Problems with Naïve Bayes**

Features typically not independent (even conditional on class label).

- E.g., \( x_1 = \text{height}, \ x_2 = \text{weight} \).

Alternative: use statistical models that model dependencies between features.

E.g., **multivariate Gaussian distributions**.

**4. Multivariate Gaussian distributions**
Standard Gaussian distributions on $\mathbb{R}^d$

**Standard normal (Gaussian) distribution on $\mathbb{R}^1$**

$X \sim N(0, 1)$, density

$$\varphi_{0,1}(x) = \frac{1}{\sqrt{2\pi}} \exp \left( -\frac{x^2}{2} \right) \quad \text{for all } x \in \mathbb{R}.$$ 

**Standard normal (Gaussian) distribution on $\mathbb{R}^d$**

$X = (X_1, X_2, \ldots, X_d) \sim N(0, I)$, density

$$\varphi_{0,I}(x) = \prod_{i=1}^{d} \frac{1}{\sqrt{2\pi}} \exp \left( -\frac{x_i^2}{2} \right) \quad \text{for all } x = (x_1, \ldots, x_d) \in \mathbb{R}^d.$$ 

Usually written as

$$\varphi_{0,I}(x) = \frac{1}{(2\pi)^{d/2}} \exp \left( -\frac{\|x\|^2}{2} \right).$$

(General) Gaussian distributions on $\mathbb{R}^d$ come from applying two operations to another (e.g., the standard) Gaussian distribution:

$X \mapsto A X + \mu$

for some vector $\mu \in \mathbb{R}^d$ and invertible linear map $A \in \mathbb{R}^{d \times d}$.

**Fact**: Let $\mu \in \mathbb{R}^d$ be any vector, and $A \in \mathbb{R}^{d \times d}$ be any invertible matrix. For any random vector $X$ in $\mathbb{R}^d$ with $E(X) = 0$ and $\text{cov}(X) = I$, the random vector $Y = AX + \mu$ satisfies

$$E(Y) = \mu, \quad \text{cov}(Y) = AA^T.$$ 

Furthermore, if $X \sim N(0, I)$, then $Y \sim N(\mu, AA^T)$.

Density for $X \sim N(\mu, AA^T)$ for $\mu \in \mathbb{R}^d$ and symmetric pos. def. matrix $AA^T$:

$$\phi_{\mu, AA^T}(x) = \frac{1}{(2\pi)^{d/2} |\det(\text{AA}^T)|^{1/2}} \exp \left( -\frac{1}{2} \|A^{-1}(x - \mu)\|^2 \right).$$
Examples of linear maps

Write $x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$.

1. If $A = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}$, then $Ax = \begin{bmatrix} x_1 \\ 2x_2 \end{bmatrix}$.
   (Scale coordinates $x_1$ and $x_2$ by, respectively, 1 and 2.)

2. If $A = \begin{bmatrix} \frac{\sqrt{2}}{\sqrt{3}} & \frac{\sqrt{3}}{\sqrt{2}} \\ \frac{\sqrt{3}}{\sqrt{2}} & \frac{\sqrt{2}}{\sqrt{3}} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}$, then $Ax = x_1 \begin{bmatrix} \frac{\sqrt{2}}{\sqrt{3}} \\ \frac{1}{\sqrt{3}} \end{bmatrix} + 2x_2 \begin{bmatrix} \frac{\sqrt{3}}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix}$.
   (Coordinate scaling as above, followed by rotation.)

MLE for Gaussian parameters

$\mathcal{P}$ = statistical model that treats $X_1, \ldots, X_n$ as iid $N(\mu, \Sigma)$ random vectors.

- MLE for $\mu$ given $(X_1, \ldots, X_n) = (x_1, \ldots, x_n)$:
  \[ \hat{\mu} = \frac{1}{n} \sum_{i=1}^{n} x_i. \]
  \[ 1 \text{ sample mean} \]

- MLE for $\Sigma$ given $(X_1, \ldots, X_n) = (x_1, \ldots, x_n)$:
  \[ \hat{\Sigma} = \frac{1}{n} \sum_{i=1}^{n} (x_i - \hat{\mu})(x_i - \hat{\mu})^T \]
  \[ 1 \text{ sample covariance} \]

where $\hat{\mu}$ is the sample mean.

(This assumes $\Sigma$ is invertible; if not, then MLE does not exist!)

General Gaussian distributions on $\mathbb{R}^d$

$X \sim N(\mu, AA^T)$

$\begin{bmatrix} \mu \\ A \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}$

Multivariate Gaussian class conditionals

Example: $\mathcal{X} = \mathbb{R}^d$, $\mathcal{Y} = \{0, 1\}$, and using multivariate Gaussian class conditional densities.

Bayes classifier corresponding to distribution with parameters $\pi_0, \pi_1, \mu_0, \Sigma_0, \mu_1, \Sigma_1$:

$\Sigma_0 = \Sigma_1$

Bayes classifier: linear decision boundary

$\Sigma_0 \neq \Sigma_1$

Bayes classifier: quadratic decision boundary
Example: quadratic decision boundary

Suppose

- \( \pi_0 = \pi_1 = 1/2 \);
- \( \mu_0 \neq \mu_1 \);
- \( \Sigma_0 = I \) and \( \Sigma_1 = 0.01I \).

What is the shape of the decision boundary?

Example: Classifying irises by sepal measurements

\( X = \mathbb{R}^1, \ Y = \{1, 2, 3\} \)

- \( x_1 = \text{ratio of sepal length to width} \)

Training data: 120 examples
Test data: 30 examples

Test error rate: 30%

Example: Classifying irises by petal measurements

\( X = \mathbb{R}^1, \ Y = \{1, 2, 3\} \)

- \( x_2 = \text{ratio of petal length to width} \)

Training data: 120 examples
Test data: 30 examples

Test error rate: 23.33%

Example: Classifying irises with both features

\( X = \mathbb{R}^2, \ Y = \{1, 2, 3\} \)

- \( x_1 = \text{ratio of sepal length to width} \)
- \( x_2 = \text{ratio of petal length to width} \)

Training data: 120 examples
Test data: 30 examples

Test error rate: 16.67%
5. Beyond Gaussians

Beyond Gaussians: exponential families

Gaussians capture pairwise correlations between features: more powerful than Na"ive Bayes, but still limited.

- To find distributions that capture other dependencies among features, consider exponential families:
  \[ p_\theta(x) \propto \exp \left( \theta^T \phi(x) \right) \nu(x) \]
  where \( \phi(x) \in \mathbb{R}^p \) is the vector of \( p \) “sufficient statistics” of data point \( x \), and \( \nu \) is some “base” probability distribution.

- Multivariate normal is special case where \( \phi(x) = (x, \text{vec}(xx^T)) \) so \( p = d + d^2 \).

- Can consider other “sufficient statistics” that include higher-order interactions among features (e.g., three-way interaction \( x_1x_2x_3 \)).

- Very closely related to graphical models (see Dave Blei’s class).

Beyond Gaussians: non-parametric methods

For most flexibility, use non-parametric methods to model density of \( X \).

Example: kernel density estimator

\[ \hat{p}(x) = \frac{1}{nh} \sum_{i=1}^{n} K \left( \frac{x - X_i}{h} \right), \]

where \( K \) is some radially-symmetric function on \( \mathbb{R}^d \), e.g., \( K(\delta) = \exp(-\|\delta\|_2) \), and \( h > 0 \) is the bandwidth parameter.

Caveat: Non-parametric methods may not work well when \( d \) is large!

Caveat about caveat: good classification performance does not necessarily require very accurate density estimation!

Example: \( k \)-NN density estimator

\[ \hat{p}(x) = \frac{k}{n v_d \cdot r_k(x)^d} \]

where
- \( r_k(x) \) is distance from \( x \) to \( k \)-th nearest neighbor among \( X_1, \ldots, X_n \);
- \( v_d \) is the volume of the unit ball in \( \mathbb{R}^d \).

Main idea:

If probability density \( p \) of \( X \) is “smooth”, then probability mass of \( B(x,r) \) (i.e., ball of radius \( r \) around \( x \)) is

\[ \int_{B(x,r)} p(z) \, dz \approx p(x) \cdot \int_{B(x,r)} \, dz = p(x) \cdot \text{vol}(B(x,r)) = p(x) \cdot v_d \cdot r^d. \]

For \( r \approx r_k(x) \), LHS is about \( k/n \), so

\[ \frac{k}{n} \approx p(x) \cdot v_d \cdot r_k(x)^d. \]
Example: \( k \)-NN density estimates in generative models

Estimation of class priors and class conditional distributions:

- Distribution of \( Y \): estimate using MLE
  \[
  \hat{\pi}_0 := \frac{1}{n} \sum_{i=1}^{n} (1 - Y_i), \quad \hat{\pi}_1 := \frac{1}{n} \sum_{i=1}^{n} Y_i
  \]
  (for \( K = 2 \) classes, \( \mathcal{Y} = \{0, 1\} \)).

- Conditional distribution of \( X \) given \( Y = y \): estimate using \( k \)-NN density estimator
  \[
  \hat{p}_y(x) = \frac{k/|S_y|}{v_d \cdot r_{k,y}(x)}
  \]
  where \( |S_y| \) are the \( X_i \)'s with label \( Y_i = y \), and \( r_{k,y}(x) \) is distance to \( k \)-th nearest neighbor among \( S_y \).

Plug-in classifier:
\[
\hat{f}(x) := \arg \max_{y \in \mathcal{Y}} \hat{\pi}_y \cdot \hat{p}_y(x).
\]

Final remarks

Some redeeming qualities of classifiers based on generative models:

- Simple recipe, many variations.
- Can also get predictions of conditional probabilities:
  \[
  \hat{P}(Y = y \mid X = x) = \frac{\hat{P}(Y = y) \cdot \hat{P}(X = x \mid Y = y)}{\sum_{y' \in \mathcal{Y}} \hat{P}(Y = y') \cdot \hat{P}(X = x \mid Y = y')},
  \]
  which is a special kind of real-valued prediction.
  (Important: denominator does matter for this!)
- Multi-class is easy to handle (see above).

Key takeaways

1. Generative structure of Bayes classifier.
2. Basic properties of multivariate Gaussians.
3. Basic recipe for learning a classifier based on a generative model.