CSC 311: Introduction to Machine Learning

Tutorial 12 - Test 2 Review

University of Toronto

This tutorial

Cover example questions on several topics:

- Bias-Variance Decomposition
- Bagging / Boosting
- Probabilistic Models (Naïve Bayes, Gaussian Discriminant)
- Principal Component Analysis (Matrix factorization, Autoencoder)
- K-Means / EM

Useful mathematical concepts

- Working with logs / exponents
- MLE, MAP, Generative modeling
- Independence, conditional independence
- Bayes rule, law of total probability, marginalization.
- Properties of Covariance matrices (i.e., positive semidefinite) / spectral decomposition for PCA.
- Definition of expectation. Expectation/variance of a sum of variables

Bias-Variance Decomposition[1]

$$\mathbb{E}[(y-t)^2] = \underbrace{(y_\star - \mathbb{E}[y])^2}_{\text{bias}} + \underbrace{\mathrm{Var}(y)}_{\text{variance}} + \underbrace{\mathrm{Var}(t)}_{\text{Bayes error}}$$

- We just split the expected loss into three terms:
  - bias: how wrong the expected prediction is (corresponds to underfitting)
  - variance: the amount of variability in the predictions (corresponds to overfitting)
  - Bayes error: the inherent unpredictability of the targets
- Even though this analysis only applies to squared error, we often loosely use "bias" and "variance" as synonyms for "underfitting" and "overfitting".

[1] From Lecture 5, Slide 49

Ensembling Methods (Bagging/Boosting)

- **Bagging**: Train independent models on random subsets of the full training data
- **Boosting**: Train models sequentially, each time focusing on examples the previous model got wrong

	Bias	**Variance**	**Training**	**Ensemble Elements**
Bagging	$\approx$	$\downarrow$	Parallel	Minimize correlation
Boosting	$\downarrow$	$\uparrow$	Sequential	High dependency

Ensembling Methods (Bagging/Boosting)

Question: Suppose your classifier achieves poor accuracy on both the training and test sets. Which would be a better choice to try to improve the performance: bagging or boosting? Justify your answer.

Ensembling Methods (Bagging/Boosting)

Question: Suppose your classifier achieves poor accuracy on both the training and test sets. Which would be a better choice to try to improve the performance: bagging or boosting? Justify your answer.

Answer:

- The model is underfitting, has high bias
- Bagging reduces variance, whereas boosting reduces the bias
- Therefore, use **boosting**

Probabilistic Models: Naive Bayes

Question: True or False: Naive Bayes assumes that all features are independent.

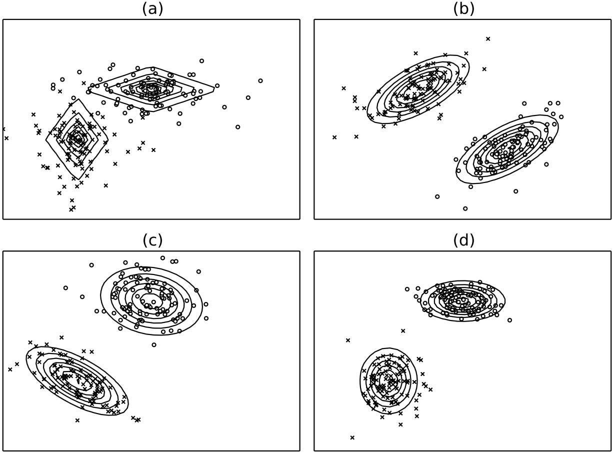
Probabilistic Models: Naive Bayes

Question: True or False: Naive Bayes assumes that all features are independent.

Answer: False. Naive Bayes assumes that the input features x_i are **conditionally independent** give the class c:

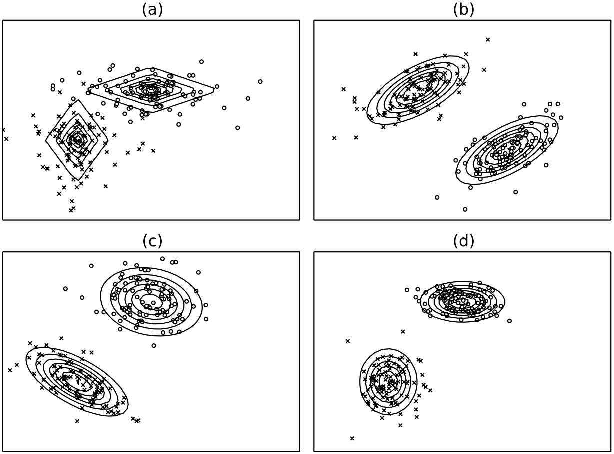
$$p(c, x_1, \ldots, {}_D) = p(c)p(x_1|c) \cdots p(x_D|c)$$

Question: Which of the following diagrams could be a visualization of a Naive Bayes classifier? Select all that applies.



Probabilistic Models: Naive Bayes

Question: Which of the following diagrams could be a visualization of a Naive Bayes classifier? Select all that applies.



Answer: A, D

Probabilistic Models: Naïve Bayes

Question:

- Consider the following problem, in which we have two classes: $\{Tainted, Clean\}$, and each data x has 3 attributes: (a_1, a_2, a_3).
- These attributes are also binary variables: $a_1 \in \{on, off\}$, $a_2 \in \{blue, red\}$, $a_3 \in \{light, heavy\}$.
- We are given a training set as follows:
  1. $Tainted$: $(on, blue, light)$ $(off, red, light)$ $(on, red, heavy)$
  2. $Clean$: $(off, red, heavy)$ $(off, blue, light)$ $(on, blue, heavy)$

(A) Manually construct Naïve Bayes Classifier based on the above training data. Compute the following probability tables: a) the class prior probability, b) the class conditional probabilities of each attribute.

Probabilistic Models: Naïve Bayes

(a) Class prior probability:

- $p(c = Tainted) = 3/6 = 1/2,$
- $p(c = Clean) = 1/2$

(a) Class prior probability:

- $p(c = Tainted) = 3/6 = 1/2$,
- $p(c = Clean) = 1/2$

(b) The class conditional distributions:

- $p(a_1 = on|c = Tainted) = 2/3$, $p(a_1 = off|c = Tainted) = 1/3$

Probabilistic Models: Naïve Bayes

(a) Class prior probability:

- $p(c = Tainted) = 3/6 = 1/2$,
- $p(c = Clean) = 1/2$

(b) The class conditional distributions:

- $p(a_1 = on|c = Tainted) = 2/3$, $p(a_1 = off|c = Tainted) = 1/3$
- $p(a_2 = blue|c = Tainted) = 1/3$, $p(a_2 = red|c = Tainted) = 2/3$
- $p(a_3 = light|c = Tainted) = 2/3$,
  $p(a_3 = heavy|c = Tainted) = 1/3$
- $p(a_1 = on|c = Clean) = 1/3$, $p(a_1 = off|c = Clean) = 2/3$
- $p(a_2 = blue|c = Clean) = 2/3$, $p(a_2 = red|c = Clean) = 1/3$
- $p(a_3 = light|c = Clean) = 1/3$, $p(a_3 = heavy|c = Clean) = 2/3$

(B) Classify a new example $(on, red, light)$ using the classifier you built above. You need to compute the posterior probability (up to a constant) of class given this example.

(B) Classify a new example $(on, red, light)$ using the classifier you built above. You need to compute the posterior probability (up to a constant) of class given this example.

Answer: To classify $\mathbf{x} = (on, red, light)$, we have:

$$p(c|\mathbf{x}) = \frac{p(c)p(x|c)}{p(c = Tainted)p(x|c = Tainted) + p(c = Clean)p(x|c = Clean)}$$

Computing each term:

$$\begin{aligned} p(c = T)p(x|c = T) &= \big(p(c = T)p(a_1 = on|c = T)p(a_2 = red|c = T) \\ &\quad p(a_3 = light|c = T)\big) \\ &= \frac{1}{2} \times \frac{2}{3} \times \frac{2}{3} \times \frac{2}{3} \\ &= \frac{8}{54} \end{aligned}$$

(B) Classify a new example $(on, red, light)$ using the classi
er you built above. You need to compute the posterior probability (up to a constant) of class given this example.

Answer: Similarly,

$$p(c = Clean)p(x|c = Clean) = \frac{1}{2} \times \frac{1}{3} \times \frac{1}{3} \times \frac{1}{3} = \frac{1}{54}$$

Therefore, $p(c = Tainted|\mathbf{x}) = 8/9$ and $p(c = Clean|\mathbf{x}) = 1/9$, according to Naïve Bayes classifier this example should be classified as **Tainted**.

Principal Component Analysis (PCA)

1. The principal components of a dataset can be found by either minimizing an objective or, equivalently, maximizing a different objective. In words, describe the objective in each case using a single sentence.

Principal Component Analysis (PCA)

1. The principal components of a dataset can be found by either minimizing an objective or, equivalently, maximizing a different objective. In words, describe the objective in each case using a single sentence.

Answer:

- **Minimizing**: Reconstruction error i.e. the distance between the original point and its projection onto the principal component subspace

Principal Component Analysis (PCA)

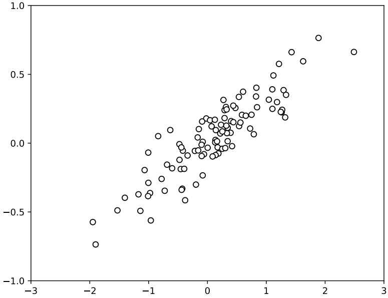
1. The principal components of a dataset can be found by either minimizing an objective or, equivalently, maximizing a different objective. In words, describe the objective in each case using a single sentence.

Answer:

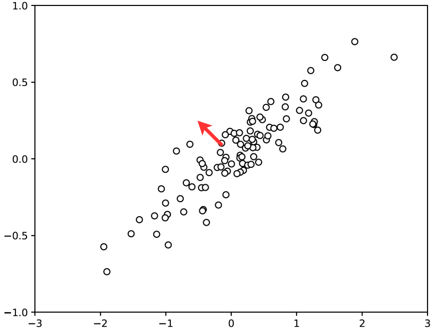
- **Minimizing**: Reconstruction error i.e. the distance between the original point and its projection onto the principal component subspace
- **Maximizing**: Variance between the code vectors i.e. the variance between the coordinate representations of the data in the principal component subspace

Principal Component Analysis (PCA)

2. The figure below shows a two-dimensional dataset. Draw the vector corresponding to the **second** principal component.



2. The figure below shows a two-dimensional dataset. Draw the vector corresponding to the **second** principal component.



1. What is the difference between K-Means and Soft K-Means algorithm?

K-Means / EM

1. What is the difference between K-Means and Soft K-Means algorithm?

Answer:

- Hard K-Means assigns a point to 1 particular cluster, whereas Soft K-Means assigns responsibilities (summing to 1) across clusters

2. K-means algorithm can be seen as a special case of the EM algorithm. Describe the steps in K-means that correspond to the E and M steps, respectively.

2. K-means algorithm can be seen as a special case of the EM algorithm. Describe the steps in K-means that correspond to the E and M steps, respectively.

Answer:

- **Assignment** step in K-Means is similar to the **E-step** in EM, computing responsibilities assessment

K-Means / EM

2. K-means algorithm can be seen as a special case of the EM algorithm. Describe the steps in K-means that correspond to the E and M steps, respectively.

Answer:

- **Assignment** step in K-Means is similar to the **E-step** in EM, computing responsibilities assessment
- **Refitting** step in K-Means minimizes the cluster distance while **M-step** in EM maximizes generative likelihood

K-Means / EM

2. K-means algorithm can be seen as a special case of the EM algorithm. Describe the steps in K-means that correspond to the E and M steps, respectively.

Answer:

- **Assignment** step in K-Means is similar to the **E-step** in EM, computing responsibilities assessment
- **Refitting** step in K-Means minimizes the cluster distance while **M-step** in EM maximizes generative likelihood
- Soft K-Means is equivalent to having spherical covariance (shared diagonal) while EM can have arbitrary covariance.