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**

\[
\mathbb{E}[(y - t)^2] = (y^* - \mathbb{E}[y])^2 + \text{Var}(y) + \text{Var}(t)
\]

- 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

<table>
<thead>
<tr>
<th></th>
<th>Bias</th>
<th>Variance</th>
<th>Training</th>
<th>Ensemble Elements</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bagging</td>
<td>≈</td>
<td>↓</td>
<td>Parallel</td>
<td>Minimize correlation</td>
</tr>
<tr>
<td>Boosting</td>
<td>↓</td>
<td>↑</td>
<td>Sequential</td>
<td>High dependency</td>
</tr>
</tbody>
</table>
**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**
Question: True or False: Naïve Bayes assumes that all features are independent.
Probabilistic Models: Naïve Bayes

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

**Answer:** False. Naïve Bayes assumes that the input features $x_i$ are **conditionally independent** given the class $c$:

$$p(c, x_1, \ldots, x_D) = p(c)p(x_1|c) \cdots p(x_D|c)$$
**Question:** Which of the following diagrams could be a visualization of a Naïve Bayes classifier? Select all that applies.
Question: Which of the following diagrams could be a visualization of a Naïve Bayes classifier? Select all that applies.

Answer: A, D
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:
- \( p(c = \text{Tainted}) = \frac{3}{6} = \frac{1}{2} \),
- \( p(c = \text{Clean}) = \frac{1}{2} \)

(b) The class conditional distributions:
- \( p(a_1 = \text{on} | c = \text{Tainted}) = \frac{2}{3} \), \( p(a_1 = \text{off} | c = \text{Tainted}) = \frac{1}{3} \)
(a) Class prior probability:

- $p(c = \text{Tainted}) = \frac{3}{6} = \frac{1}{2}$,
- $p(c = \text{Clean}) = \frac{1}{2}$

(b) The class conditional distributions:

- $p(a_1 = \text{on} | c = \text{Tainted}) = \frac{2}{3}$, $p(a_1 = \text{off} | c = \text{Tainted}) = \frac{1}{3}$
- $p(a_2 = \text{blue} | c = \text{Tainted}) = \frac{1}{3}$, $p(a_2 = \text{red} | c = \text{Tainted}) = \frac{2}{3}$
- $p(a_3 = \text{light} | c = \text{Tainted}) = \frac{2}{3}$, $p(a_3 = \text{heavy} | c = \text{Tainted}) = \frac{1}{3}$
- $p(a_1 = \text{on} | c = \text{Clean}) = \frac{1}{3}$, $p(a_1 = \text{off} | c = \text{Clean}) = \frac{2}{3}$
- $p(a_2 = \text{blue} | c = \text{Clean}) = \frac{2}{3}$, $p(a_2 = \text{red} | c = \text{Clean}) = \frac{1}{3}$
- $p(a_3 = \text{light} | c = \text{Clean}) = \frac{1}{3}$, $p(a_3 = \text{heavy} | c = \text{Clean}) = \frac{2}{3}$
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.

\[
p(c|x) = p(c)p(x|c) = p(Tainted)p(a_1=on|x=Tainted)p(a_2=red|x=Tainted)p(a_3=light|x=Tainted) + p(Clean)p(a_1=on|x=Clean)p(a_2=red|x=Clean)p(a_3=light|x=Clean)
\]

Computing each term:

\[
p(Tainted)p(a_1=on|x=Tainted)p(a_2=red|x=Tainted)p(a_3=light|x=Tainted) = \frac{1}{2} \times \frac{2}{3} \times \frac{2}{3} \times \frac{2}{3} = \frac{8}{54}
\]
(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 \(x = (on, red, light)\), we have:

\[
p(c|x) = \frac{p(c)p(x|c)}{p(c = Tainted)p(x|c = Tainted) + p(c = Clean)p(x|c = Clean)}
\]

Computing each term:

\[
p(c = T)p(x|c = T) = (p(c = T)p(a_1 = on|c = T)p(a_2 = red|c = T)\]
\[
\quad \cdot p(a_3 = light|c = T))
\]
\[
= \frac{1}{2} \times \frac{2}{3} \times \frac{2}{3} \times \frac{2}{3}
\]
\[
= \frac{8}{54}
\]
(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:** Similarly,

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

Therefore, \( p(c = \text{Tainted}|x) = 8/9 \) and \( p(c = \text{Clean}|x) = 1/9 \), according to Naïve Bayes classifier this example should be classified as Tainted.
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.
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
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?
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
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 the **M-step** in EM maximizes generative likelihood.
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.