CSCI567 HW4

Ritu Pravakar and Charlene Yuen

November 2022

Theory-based Questions

Problem 1: Decision Trees (12pts)

Consider a binary dataset with 400 examples, where half of them belong to class A and the rest belong to class B. Next, consider two decision stumps (i.e. trees with depth 1) $\mathcal{T}_1$ and $\mathcal{T}_2$, each with two children. For $\mathcal{T}_1$, the left child has 150 examples in class A and 50 examples in class B. For $\mathcal{T}_2$, the left child has 0 examples in class A and 100 examples in class B. (You can infer the number of examples in the right child using the total number of examples.)

1.1 (6 pts) In class, we discussed entropy and Gini impurity as measures of uncertainty at a leaf. Another possible metric is the classification error at the leaf, assuming that the prediction at the leaf is the majority class among all examples that belong to that leaf. For each leaf of $\mathcal{T}_1$ and $\mathcal{T}_2$, compute the entropy (base e), Gini impurity and classification error. You can either exactly express the final numbers in terms of fractions and logarithms, or round them to two decimal places.

Entropy (base e):
If root tests on $\mathcal{T}_1$:
Left child: $-(\frac{150}{200}\ln\frac{150}{200}+\frac{50}{200}\ln\frac{50}{200})\approx 0.56$
Right child: $-(\frac{50}{200}\ln\frac{50}{200}+\frac{150}{200}\ln\frac{150}{200})\approx 0.56$

If root tests on $\mathcal{T}_2$:
Left child: $-(\frac{0}{100}\ln\frac{0}{100}+\frac{100}{100}\ln\frac{100}{100})=0$
Right child: $-(\frac{200}{300}\ln\frac{200}{300}+\frac{100}{300}\ln\frac{100}{300})\approx 0.64$

Gini impurity:
If root tests on $\mathcal{T}_1$:
Left child: $\frac{150}{200}(1-\frac{150}{200})+\frac{50}{200}(1-\frac{50}{200})=\frac{3}{8}$
Right child: $\frac{50}{200}(1-\frac{50}{200})+\frac{150}{200}(1-\frac{150}{200})=\frac{3}{8}$

If root tests on $\mathcal{T}_2$:
Left child: $\frac{0}{100}(1-\frac{0}{100})+\frac{100}{100}(1-\frac{100}{100})=0$
Right child: $\frac{200}{300}(1-\frac{200}{300})+\frac{100}{300}(1-\frac{100}{300})=\frac{4}{9}$

Classification error:
If root tests on $\mathcal{T}_1$:
Left child (classified A): $1-\frac{150}{200}=\frac{1}{4}$
Right child (classified B): $1-\frac{150}{200}=\frac{1}{4}$

If root tests on $\mathcal{T}_2$:
Left child (classified B): $1-\frac{100}{100}=0$
Right child (classified A): $1-\frac{200}{300}=\frac{1}{3}$

1.2 (6 pts) Compare the quality of $\mathcal{T}_1$ and $\mathcal{T}_2$ (that is, the two different splits of the root) based on conditional entropy (base e), weighted Gini impurity and total classification error. Intuitively, which of $\mathcal{T}_1$ or $\mathcal{T}_2$ appears to be a better split to you (there may not necessarily be one correct answer to this)? Based on your conditional entropy, Gini impurity and classification error calculations, which of the metrics appear to be more suitable choices to decide which variable to split on?

Conditional Entropy (base e):
If root tests on $\mathcal{T}_1$:
$\frac{200}{400} * 0.56 + \frac{200}{400} * 0.56 \approx 0.56$
If root tests on $\mathcal{T}_2$:
$\frac{100}{400} * 0 + \frac{300}{400} * 0.64 \approx 0.48$

Weighted Gini impurity:
If root tests on $\mathcal{T}_1$:
$\frac{200}{400} * \frac{3}{8} + \frac{200}{400} * \frac{3}{8} = \frac{3}{8} = 0.375$
If root tests on $\mathcal{T}_2$:
$\frac{100}{400} * 0 + \frac{300}{400} * \frac{4}{9} = \frac{1}{3} = 0.33$

Total classification error:
If root tests on $\mathcal{T}_1$:
$\frac{200}{400} * \frac{1}{4} + \frac{200}{400} * \frac{1}{4} = \frac{1}{4}$
If root tests on $\mathcal{T}_2$:
$\frac{100}{400} * 0 + \frac{300}{400} * \frac{1}{3} = \frac{1}{3}$

To me intuitively, $\mathcal{T}_2$ is the better split, since from a glance there seems to be less entropy, especially in the left child, which is unambiguously class B. Even for the right child, class A dominates over class B by 200/300. This intuition is confirmed by the calculations, as the entropy from all 3 calculations are less on $\mathcal{T}_2$. Based on the calculations, the metrics of conditional entropy seems to be the better choice since it seems to be more sensitive to subtle changes in probability than the total classification error and better than weighted Gini impurity in the case of imbalanced datasets similar to the given dataset.

Problem 2: Gaussian Mixture Model and EM

2.1

$$\underset{\pi_j,\mu_j,\Sigma_j}{\operatorname{argmax}} \sum_i \sum_j \gamma_{ij} \ln \pi_j + \sum_i \sum_j \gamma_{ij} \ln N(\mathbf{x}_i \mid \mu_j, \Sigma_j)$$

$$\text{s.t. } \pi_j \geq 0 \text{ and } \sum_{j=1}^{k} \pi_j = 1$$

$$\text{Let } P = \sum_i \sum_j \gamma_{ij} \ln \pi_j + \sum_i \sum_j \gamma_{ij} \ln N(\mathbf{x}_i \mid \mu_j, \Sigma_j)$$

Using the given fact with $a_j = \sum_i \gamma_{ij}$ and with $q_j = \pi_j$ and $q_j^* = \frac{a_j}{\sum_{k'} a_{k'}}$, we can see that $\pi_j^* = \frac{\sum_i \gamma_{ij}}{\sum_i \sum_{k'} \gamma_{ik'}} = \frac{\sum_i \gamma_{ij}}{n}$

To find μ_j^* we can rewrite P as:

$$P = \sum_i \sum_j \gamma_{ij} \ln \pi_j + \sum_i \sum_j \gamma_{ij} (\ln \frac{1}{(\sqrt{2\pi})^d |\Sigma_j|^{\frac{1}{2}}} + \ln \exp\left(-\frac{1}{2}(\mathbf{x}_i - \mu_j)^T \Sigma_j^{-1} (\mathbf{x}_i - \mu_j)\right))$$

$$\frac{\partial P}{\partial \mu_j} = 0 + \sum_i \gamma_{ij} \frac{1}{N(\mathbf{x}_i \mid \mu_j, \Sigma_j)} \frac{\partial}{\partial \mu_j} N(\mathbf{x}_i \mid \mu_j, \Sigma_j)$$

$$= \sum_i \gamma_{ij} \frac{\partial}{\partial \mu_j} (\ln \frac{1}{(\sqrt{2\pi})^d |\Sigma_j|^{\frac{1}{2}}} - \frac{1}{2}(\mathbf{x}_i - \mu_j)^T \Sigma_j^{-1} (\mathbf{x}_i - \mu_j))$$

$$= \sum_i \gamma_{ij} \frac{\partial}{\partial \mu_j} (-\frac{1}{2}(\mathbf{x}_i - \mu_j)^T \Sigma_j^{-1} (\mathbf{x}_i - \mu_j))$$

Since Σ_j is symmetric,

$$= \sum_i \gamma_{ij} \ (-\frac{1}{2}(-2)\Sigma_j^{-1}(\mathbf{x}_i - \mu_j)) = \sum_i \gamma_{ij} \ \Sigma_j^{-1}(\mathbf{x}_i - \mu_j)$$

$$\frac{\partial P}{\partial \mu_j} = \sum_i \gamma_{ij} \ \Sigma_j^{-1}(\mathbf{x}_i - \mu_j) = 0$$

$$\sum_i \gamma_{ij} \ \Sigma_j \Sigma_j^{-1}(\mathbf{x}_i - \mu_j) = \Sigma_j 0$$

$$\sum_i \gamma_{ij} \ (\mathbf{x}_i - \mu_j) = 0$$

$$\sum_i \gamma_{ij} \ \mathbf{x}_i = \sum_i \gamma_{ij} \ \mu_j$$

$$\mu_j^* = \frac{\sum_i \gamma_{ij} \ \mathbf{x}_i}{\sum_i \gamma_{ij}}$$

To find Σ_j^* :

$$\frac{\partial P}{\partial \Sigma_j} = 0 + \sum_i \gamma_{ij} \frac{1}{N(\mathbf{x}_i \mid \mu_j, \Sigma_j)} \frac{\partial}{\partial \Sigma_j} N(\mathbf{x}_i \mid \mu_j, \Sigma_j)$$

$$= \sum_i \gamma_{ij} \frac{\partial}{\partial \Sigma_j} (\ln(\sqrt{2\pi})^{-d} |\Sigma_j|^{-\frac{1}{2}} - \frac{1}{2}(\mathbf{x}_i - \mu_j)^T \Sigma_j^{-1} (\mathbf{x}_i - \mu_j))$$

$$= \sum_i \gamma_{ij} \frac{\partial}{\partial \Sigma_j} (-\frac{1}{2} \ln |\Sigma_j| - \frac{1}{2}(\mathbf{x}_i - \mu_j)^T \Sigma_j^{-1} (\mathbf{x}_i - \mu_j))$$

Since $\frac{\partial}{\partial \Sigma_j} \ln |\Sigma_j| = (\Sigma_j^T)^{-1} = \Sigma_j^{-1}$ and $\frac{\partial}{\partial \Sigma_j} (\mathbf{x}_i - \mu_j)^T \Sigma_j^{-1} (\mathbf{x}_i - \mu_j) = -\Sigma_j^{-1} (\mathbf{x}_i - \mu_j)(\mathbf{x}_i - \mu_j)^T \Sigma_j^{-1}$:

$$\frac{\partial P}{\partial \Sigma_j} = \sum_i \gamma_{ij} (-\frac{1}{2}\Sigma_j^{-1} + \frac{1}{2}\Sigma_j^{-1}(\mathbf{x}_i - \mu_j)(\mathbf{x}_i - \mu_j)^T \Sigma_j^{-1}) = 0$$

$$-\frac{1}{2}\sum_i \gamma_{ij}(\Sigma_j^{-1}) + \frac{1}{2}\sum_i \gamma_{ij}\Sigma_j^{-1}(\mathbf{x}_i - \mu_j)(\mathbf{x}_i - \mu_j)^T\Sigma_j^{-1} = 0$$

$$\sum_i \gamma_{ij}(\Sigma_j^{-1})\Sigma_j = \sum_i \Sigma_j^{-1}\gamma_{ij}(\mathbf{x}_i - \mu_j)(\mathbf{x}_i - \mu_j)^T\Sigma_j^{-1}\Sigma_j$$

$$\Sigma_j \sum_i \gamma_{ij} = \sum_i \Sigma_j\Sigma_j^{-1}\gamma_{ij}(\mathbf{x}_i - \mu_j)(\mathbf{x}_i - \mu_j)^T$$

$$\Sigma_j^* = \frac{1}{\sum_i \gamma_{ij}}\sum_i \gamma_{ij}(\mathbf{x}_i - \mu_j)(\mathbf{x}_i - \mu_j)^T$$

Programming-based Questions

Problem 3: Exploring Decision Trees and Random Forests (12pts)

3.1 (2 pts)

It seems like as `max_depth` increases, the training accuracy increases until `max_depth`=10 before slowing down its increase until `max_depth`=15, and stagnating until `max_depth`=20 as the training accuracy approaches 1 for both decision trees and random forests. The test accuracy also increases until `max_depth`=10 before stagnating, with decision trees being around an accuracy of around 0.85 and random forests being around 0.9. The generalization gap seems to increase with an increase in `max_depth` in both cases, which could happen from overfitting. The generalization gap also seems to be larger for the decision tree, since it tends to have poor prediction performance due to not generalizing well and being unstable high variance models that can overfit. However, random trees randomly select k of d input variables for splitting, which would differentiate and decorrelate trees further, which reduces overfitting and increases accuracy, as we see in the results of the plots.

3.2 (2 pts)

As shown by the plot, the accuracy increases with greater `n_estimators`, though it starts stagnating after around `n_estimators`=25. The generalization gap increases after the initial increase in `n_estimators` and remains the same throughout the rest of the graph. The perferred range of values would prefer are likely smaller values around `n_estimators`=25 and largest around `n_estimators`=50. Having overly large values of `n_estimators` has no increase on accuracy, and its redundancy may simply decrease efficiency, which means that smaller values are preferred.

3.3 (2 pts)

It seems that low values of `min_samples_leaf` will have higher training and test accuracy, while high values will have lower training and test accuracy overall for both samples. Since the smaller amount of 1000 samples may be more prone to stochasticity in random forests, as seen by the greater test accuracy than training accuracy at `min_samples_leaf`=40. From this trend, it would likely be better to go with a middle value of `min_samples_leaf` that isn't too large for low accuracy or too high in case of overfitting and having too low a requirement to have meaningful classification. The reason for this behavior may be a small amount of `min_samples_leaf` may overfit and allow just any sample or a small amount of samples to be a leaf node, and as `min_samples_leaf` increases, the required number becomes more reasonable, and there is less overfitting and a smaller generalization gap.

3.4 (2 pts)

It seems that low values of `max_samples` result in low generalization gaps but lower training and testing accuracy, while high values result in higher training data and test accuracy but greater generalization gaps. Thus, we would prefer intermediate values for the parameter in both cases. The larger values of `max_samples` you train on will result in each decision tree being more similar as more samples can overlap, making each one too correlated and eventually having any larger number of samples will not add to the accuracy, but having too small values will be enough for the tree to be accurate enough. With a larger sample size of 4000, the generalization gap is smaller than that of the smaller sample size of 1000, since it is less prone to overfitting than a smaller sample size.

3.5 (2 pts)

The accuracies for low `max_features` seem to be lower than higher values, but stagnates quickly after `max_features`=100, resulting in an increasing generalization gap. The best range of values in both cases seems to be around `max_features`=50 to 100. The values for both training set sizes are similar, as after this range, the accuracies begin to decrease, so it would be optimal to maximize the accuracy and consider a smaller subset of features. The larger training set may have less prone to overfitting, as shown by the smaller generalization gap as `max_features` increases, than the smaller training set.

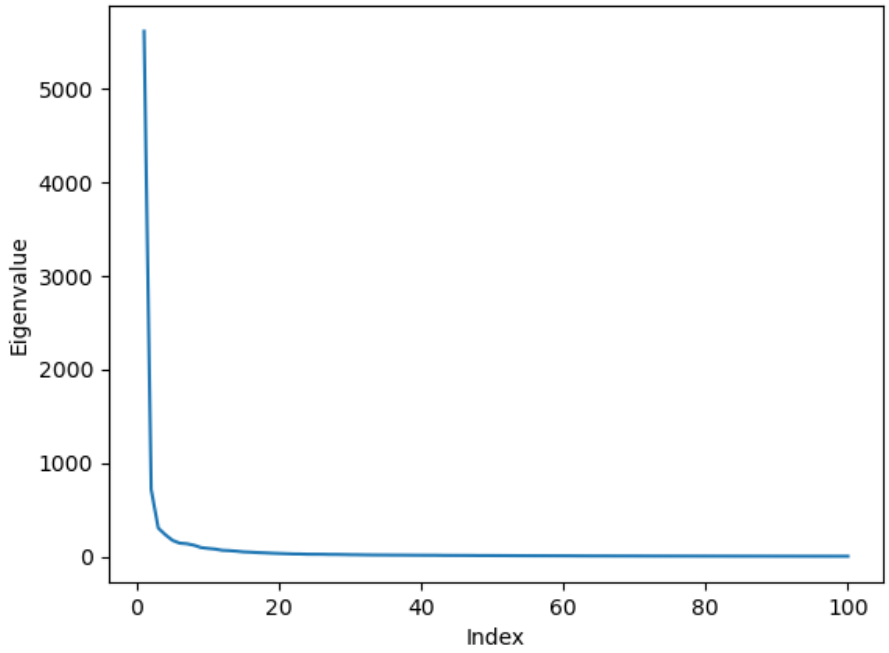
3.6 (2 pts)

The image seems to be focusing more on pixels that overlap when overlaying all the digits, which is where it may be more ambiguous compared to other numbers when classifying. The pixels given higher importance will allow the model to hone in on improving upon mistakes on similar looking digits, rather than wasting resources on classifying more easily differentiable numbers such as 0 or 1 compared to digits larger than or equal to 5. Thus, this gives reasonable estimates as to which feature to split on in the decision tree.

Problem 4: PCA for Learning Word Embeddings

4.1

The eigenvalues decay after the first few values, as the first value has the largest value of around 5618.77; the second drops to around 719, and then up until the 8th eigenvalue, the value slowly decreases and remains in the hundreds, and then after drops gradually to the two digit values and later one digit values. It seems that the first 100 eigenvalues capture around 75.67% out of the 10,000 eigenvalues in total.



4.2

The closest words seem relevant to the given word when tested, such as "university" being linked to "college" (school), "learning" with "teaching" (classroom), and "California" with "Florida" (states). Overall, it seems that finding the similarities between words using their embeddings does decently at finding a correlated word within some general topic.

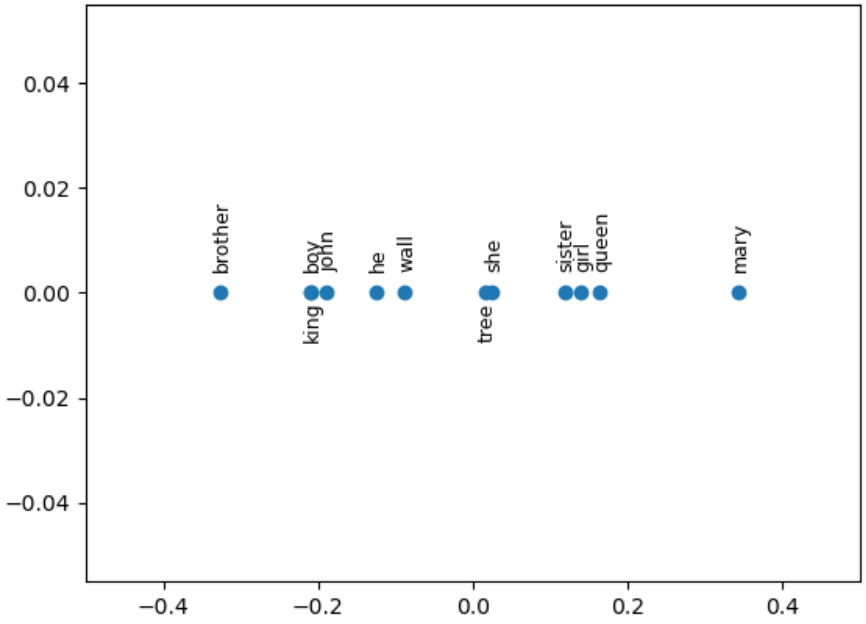
4.3

Interesting eigenvectors:
1) The first PC gives nouns closely related to the topic of university activities such as sports divisions and music, which can help form a general sentence about school.
2) The second PC gives a list of names, which are all proper nouns that can be used in a sentence.
3) The sixth PC gives a list of verbs, which can all be used in conjunction with nouns from other PCs and help form a grammatically correct sentence.
4) The ninth PC gives a list of words that can form a general sentence about someone winning a championship, cup, award with both nouns and verbs.
5) The tenth PC gives a list of words seemingly correlated to World War 2, as Germany, the Soviet Union, France, Britain, and Poland were countries mentioned, as well as aircraft, naval ships, squadrons, navy, and the military.

We can't do it for all 100 eigenvectors, since as PCs with smaller eigenvalues begins to give words less correlated and make less sense overall, making it hard to detect any semantic structure.

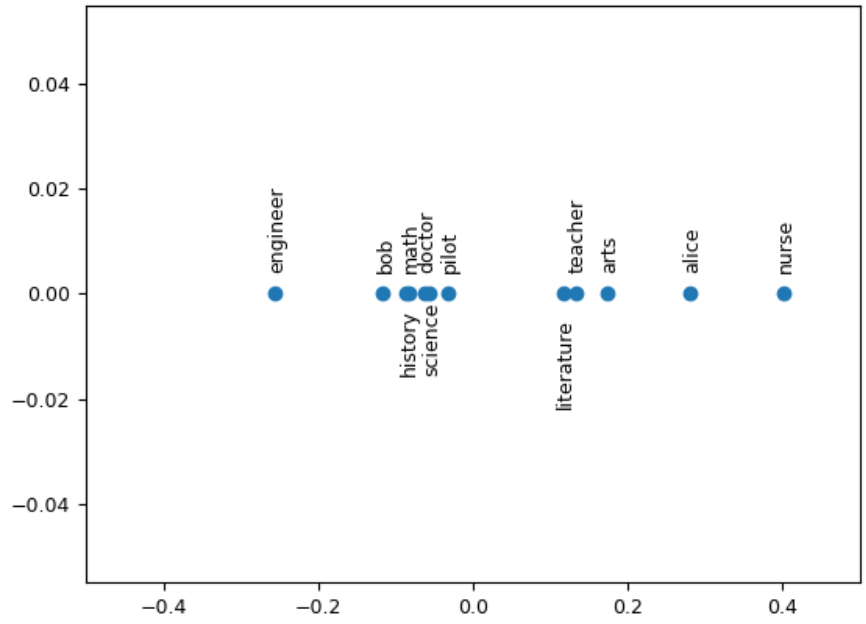
4.4

4.4.1



From the plot, we observe that the words boy, brother, king, john, and he are closer together on the left, while she, sister, girl, queen, and mary are closer together on the right, while tree and wall are in the center, as they are gender neutral objects.

4.4.2



From this plot, we observe that engineer, bob, math, history, doctor, science, and pilot are closer together on the left, while literature, teacher, arts, alice, and nurse are closer together on the right. This may be the case that engineer-

ing, math, doctors, science, and piloting are known as male-dominated fields, which may cause gender bias in the calculated correlations. An issue with this would be societal bias leaking into a supposedly unbiased result from a machine. If LinkedIn used word embeddings, this may result in non-males from being excluded from the job search and a possibly suitable candidate, which would create inequality.