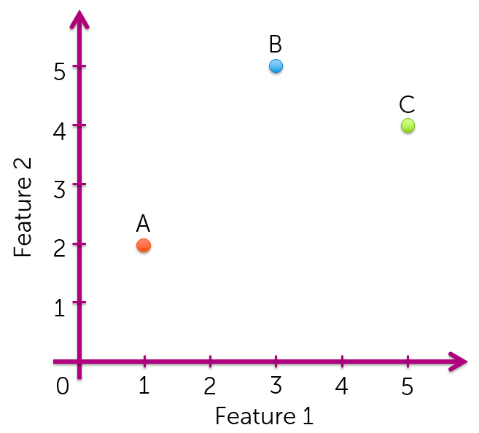
## Week 2a – Representations and Metrics

1. Which two points are closest to each other with respect to Euclidean distance?



* 1. B and C.

1. Which two points are closest to each other with respect to cosine similarity (smallest cosine distance)?
   1. A and B.
2. Consider the following sentences:
   1. Sentence 1: The quick brown fox jumps over the lazy dog.
   2. Sentence 2: A quick brown dog outpaces a quick fox.

Compute the Euclidean distance using word counts. To compute word counts, turn all words into lower case and strip all punctuation, so that “The” and “the” are counted as the same token. That is, document 1 would be represented as:

where # word is the count of that word in that document.

1. Recall that . Compute the cosine distance between the sentences using word counts.
2. For positive features, cosine similarity is always between 0 and 1.
   1. True.
3. Which of the following describes the word count document representation?
   1. Ignores the order of words.
   2. Assigns a high score to a frequently occurring word.

## Week 2b – KD-trees

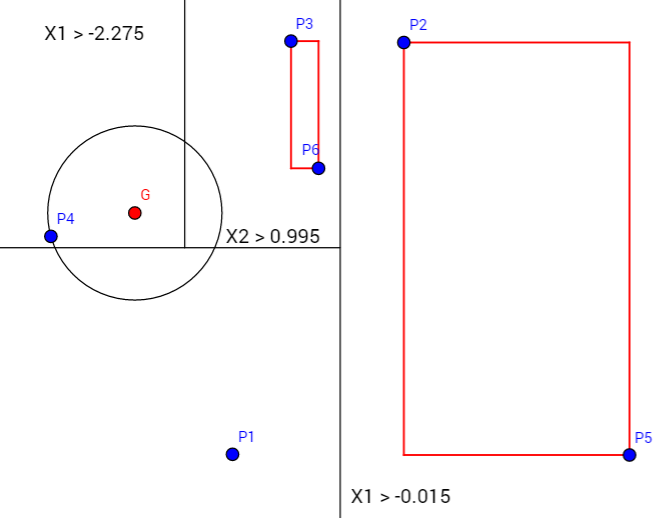
1. Which of the following is true about KD-trees?
   1. It divides the feature space into nested axis-aligned boxes.
   2. It prunes parts of the feature space away from consideration by inspecting smallest possible distances that can be achieved.
   3. The query time scales sublinearly with the number of data points and exponentially with the number of dimensions.
   4. It works best in low to medium settings.

|  |  |  |
| --- | --- | --- |
|  | X1 | X2 |
| Data point 1 | -1.58 | -2.01 |
| Data point 2 | 0.91 | 3.98 |
| Data point 3 | -0.73 | 4.00 |
| Data point 4 | -4.22 | 1.16 |
| Data point 5 | 4.19 | -2.02 |
| Data point 6 | -0.33 | 2.15 |

Train a KD-tree by hand as follows:

* First split using X1 and then using X2. Alternate between X1 and X2.
* Use “middle-of-the-range” heuristic for each split. Take the maximum and minimum of the coordinates of the member points.
* Keep sub-dividing until each leaf node contains two or fewer data points.

1. What is the split value used for the first split?
   1. -0.015
2. What is the split value used for the second split?
   1. -0.995
3. Given a query point (-3, 1.5), which of the data points belong to the same leaf node as the query point?
   1. Data point 4.
4. Perform back-tracking with the query point to perform exact nearest neighbour search. Which of the data points would be pruned from the search? Assume that each node in the KD-tree remembers the **tight bound** on the coordinates of its member points as follows:



* 1. Data points 2, 5, 1, 3 and 6.

## Week 2c – Locality Sensitive Hashing (LSH)

1. Like KD-trees, LSH lets us compute exact nearest neighbours while inspecting only a fraction of the data points in the training set.
   1. False.
2. Given two data points with high cosine similarity, the probability that a randomly drawn line would separate the two points is small.
   1. True.
3. The true nearest neighbour of the query is guaranteed to fall into the same bin as the query.
   1. False.
4. LSH is more efficient than KD-trees in high dimensional settings.
   1. True.
5. Suppose you trained a LSH model and performed a lookup using the bin index of the query. You notice that the list of candidates returned aren’t at all similar to the query item. Which of the following changes would produce a more relevant list of candidates?
   1. Use multiple tables.
   2. Inspect more neighbouring bins to the bin containing the query.
   3. Decrease the number of random lines/hyperplanes.

## Week 3a – K-Means

1. K-means always converges to a local optimum.
   1. True.
2. The clustering objective is non-increasing throughout a run of k-means.
   1. True.
3. Running k-means with a larger value of k always enables a lower possible final objective value than running k-means with a smaller value.
   1. True.
4. Any initialisation of the centroids in k-means is just as good as any other.
   1. False.
5. Initialising centroids using k-means++ guarantees convergence to a global optimum.
   1. False.
6. Initialising centroids using k-means++ costs more than random initialisation in the beginning, but can pay off eventually by speeding up convergence.
   1. True.
7. Using k-means++ can only influence the number of iterations to convergence, not the quality of the final assignments (i.e. objective value at convergence).
   1. False.
8. Consider the following dataset:

|  |  |  |
| --- | --- | --- |
|  | X1 | X2 |
| Data point 1 | -1.88 | 2.05 |
| Data point 2 | -0.71 | 0.42 |
| Data point 3 | 2.41 | -0.67 |
| Data point 4 | 1.85 | -3.8 |
| Data point 5 | -3.69 | -1.33 |

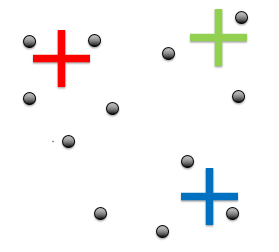
Perform k-means with k = 2 until the cluster assignment doesn’t change between successive iterations. Use the following initialisation for the centroids.

|  |  |  |
| --- | --- | --- |
|  | X1 | X2 |
| Cluster 1 | 2 | 2 |
| Cluster 2 | -2 | -2 |

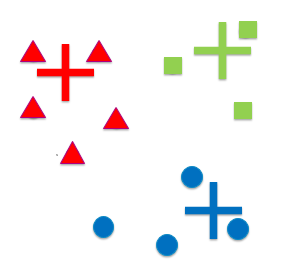
Which of the 5 data points changed its cluster assignment most often?

* 1. Data point 2

1. Suppose we initialise k-means with the following centroids.



What best describes the cluster assignment in the first iteration of k-means?

* 1. 

## Week 3b – MapReduce for K-Means

1. Suppose we are operating on a 1D vector. Which operations are data parallel over the vector elements?
   1. Compute the average of the elements.
2. A single mapper can emit multiple (key, value) pairs.
   1. True.
3. More than one reducer can emit (key, value) pairs with the same key simultaneously.
   1. False.
4. Suppose we are running k-means using MapReduce. Some mappers may be launched for a new k-means iteration even if some reducers from the previous iteration are still running.
   1. False.
5. Which binary operations can be used for the reduce step of MapReduce?

The reduce step requires a binary operator that satisfies **both** of the following operations:

* Commutative:
* Associative:

## Week 4 – EM for Gaussian Mixtures

1. While the EM algorithm maintains uncertainty about the cluster assignment for each observation via soft assignments, the model assumes that every observation comes from only one cluster.
   1. True.
2. In high dimensions, the EM algorithm runs the risk of setting cluster variances to 0.
   1. True.
3. In the EM algorithm, what do the E and M step represent respectively?
   1. Estimate cluster responsibilities, Maximise likelihood over parameters.
4. Suppose we have data that comes from a mixture of 6 Gaussians i.e. that is the true data structure. Which model would we expect to have the highest log-likelihood after fitting via the EM algorithm?
   1. A mixture of Gaussians with 10 component clusters.
5. Which of the following correctly describes the differences between EM for mixtures of Gaussians and k-means?
   1. EM is better at capturing clusters of different sizes and orientations.
   2. EM is better at capturing clusters with overlaps.
   3. K-means is equivalent to running EM with infinitesimally small diagonal covariances.
6. Suppose we are running the EM algorithm. After an E-step, we obtain the following responsibility matrix:

|  |  |  |  |
| --- | --- | --- | --- |
| Cluster responsibilities | Cluster A | Cluster B | Cluster C |
| Data point 1 | 0.20 | 0.40 | 0.40 |
| Data point 2 | 0.50 | 0.10 | 0.40 |
| Data point 3 | 0.70 | 0.20 | 0.10 |

Which is the **most probable** cluster for data point 3?

* 1. Cluster A.

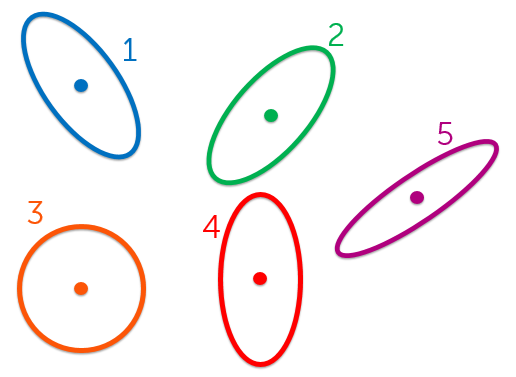
1. Also suppose that the data points are as follows:

|  |  |  |  |
| --- | --- | --- | --- |
| Dataset | X | Y | Z |
| Data point 1 | 3 | 1 | 2 |
| Data point 2 | 0 | 0 | 3 |
| Data point 3 | 1 | 3 | 7 |

Let us compute the new mean for cluster A. What is the **Z coordinate** of the new mean?

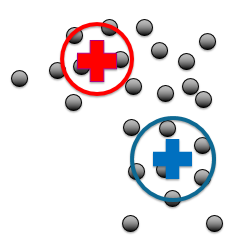
* 1. 4.857

1. Which contour plots describe a Gaussian distribution with diagonal covariance?

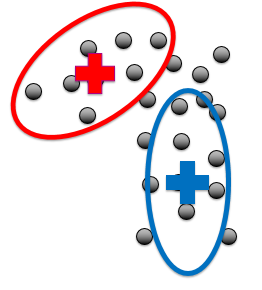


* 1. 3 and 4.

1. Suppose we initialise EM for mixtures of Gaussians (using full covariance matrices) with the following clusters:



What best describes the updated clusters after the first iteration of EM?

1. 

## Week 5a – Latent Dirichlet Allocation

1. According to the assumption of LDA, each document in the corpus contains words about a single topic.
   1. False.
2. Using LDA to analyse a set of documents is an example of a supervised learning task.
   1. False.
3. When training an LDA model, changing the ordering of words in a document doesn’t affect the overall joint probability.
   1. True.
4. Suppose in a trained LDA model two documents have no topics in common (i.e. one document has 0 weight on any topic with non-zero weight in the other document). As a result, a single word in the vocabulary cannot have high joint probability of occurring in both documents.
   1. False.
5. Topic models are guaranteed to produce weights on words that are coherent and easily interpretable by humans.
   1. False.

## Week 5b – Training LDA Models with Gibbs Sampling

1. Each iteration of Gibbs sampling for Bayesian inference in topic modelling is guaranteed to yield a higher joint probability than the previous sample.
   1. False.
2. Bayesian methods such as Gibbs sampling can be advantageous because they:
   1. Account for uncertainty over parameters when making predictions.
   2. Regularise parameter estimates to avoid extreme values.
3. For the standard LDA model, how many parameters are required to represent the distributions defining the topics?
   1. [# unique words] \* [# topics]
4. Suppose we have a collection of documents, and we are focusing our analysis to the use of the following 10 words. We ran several iterations of Gibbs sampling for a LDA model with k=2 topics and alpha=10 and gamma=0.1. The corpus-wide assignments at our most recent collapsed Gibbs iteration are summarised in the following table of counts:

|  |  |  |
| --- | --- | --- |
| Word | Count in topic 1 | Count in topic 2 |
| Baseball | 52 | 0 |
| Homerun | 15 | 0 |
| Ticket | 9 | 2 |
| Price | 9 | 25 |
| Manager | 20 | 37 |
| Owner | 17 | 32 |
| Company | 1 | 23 |
| Stock | 0 | 75 |
| Bankrupt | 0 | 19 |
| Taxes | 0 | 29 |

We also have a single document *i* with the following topic assignments for each word:

|  |  |
| --- | --- |
| Word | Topic |
| Baseball | 1 |
| Manager | 2 |
| Ticket | 1 |
| Price | 2 |
| Owner | 1 |

Suppose we want to re-compute the topic assignments for the word “manager”. To sample a new topic, we need to compute several terms to determine how much the document likes each topic, and how much each topic likes the word “manager”.

What is the value of i.e. the number of times the word “manager” has been assigned to topic 1?

* 1. 20

1. What is the value of , where the sum is taken over all the words in the vocabulary?
   1. 123.
2. What is the value of for this document i.e. the number of words in document assigned to topic 1?
   1. 3.
3. “manager” was assigned to topic 2. When we remove that assignment prior to sampling, we need to decrement the associated counts. After decrementing, what is the value of ?
   1. 1.
4. After decrementing, what is the value of ?
   1. 36.
5. After decrementing, what is the value of ?
   1. 241.
6. The unnormalised probability of assigning to topic 1 is: where V is the total size of the vocabulary. Similarly, the unnormalised probability of assigning to topic 2 is: . Using these equations, what is the probability of assigning the word “manager” to topic 1. (Remember to normalise across the two topic options so that the probabilities of all possible assignments – topic 1 and topic 2 – sum to 1.)