BerkeleyX: CS190.1x Scalable Machine Learning

ONE-HOT-ENCODING (1/1 point)

Using one-hot-encoding, a categorical feature with four distinct values would be represented by how many features?

- 1 feature
- 2 features
- 3 features
- 4 features ✔

EXPLANATION

OHE, as performed in the lab, adds one new feature for every distinct feature/value combination in the dataset.

CHECK HIDE ANSWER

RARE EVENTS (1/1 point)

For rare events it is often a good idea to predict probabilities instead of classes.

- True ✔
- False

EXPLANATION

True. When dealing with rare events (like click events), we often want to use probabilities and log-loss rather than class predictions and 0 / 1 loss. For example, if the probability of a click event is relatively high, say 10%, we would still classify that observation as not a click event given the typical threshold of 50%, so using probabilities provides us with much more granular information.

CHECK HIDE ANSWER

FEATURE REPRESENTATION (1/1 point)

The OHE features in the lab are stored in a:

- [ ] Dense representation
- [x] Sparse representation ✔

✔

Note: Make sure you select all of the correct options—there may be more than one!

EXPLANATION

The lab uses MLlib's SparseVector to store OHE features.

CHECK HIDE ANSWER

PREDICTION TARGET (1/1 point)

In the lab, we're trying to predict

- [ ] Revenue from click events
- [x] Probability of a click event ✔
- [ ] Click-through vs not click event

✔

Note: Make sure you select all of the correct options—there may be more than one!

EXPLANATION

We're trying to predict the probability of a click event so that we can minimize the log-loss.

CHECK HIDE ANSWER

OHE FEATURES (1/1 point)

In the lab, using the OHE method on the training data in Part (3c) creates a dictionary with:

- ( ) 23,328 features
- ( ) 36,177 features
- (x) 233,286 features ✔
- ( ) 361,772 features

EXPLANATION

The OHE method generates over 200,000 different features, and we are only working with a small sample of the full dataset. Note that the model you created in Part (4a) also has over 200,000 weights, which are used to convert these features into predictions.

CHECK HIDE ANSWER

FEATURE HASHING (1/1 point)

The feature hashing performed in the lab:

- [ ] Discards rare features
- [ ] Increases the number of features
- [ ] Requires calculating the OHE dictionary
- [x] Causes feature collisions for certain observations ✔

✔

Note: Make sure you select all of the correct options—there may be more than one!

EXPLANATION

No features are discarded. Instead, several features map to the same buckets. The total number of features is decreased substantially. Feature hashing does not require computing an OHE dictionary. Several of the training observations (almost 5,000) have hash collisions, which can seen by running: hashTrainData.filter(lambda lp: np.any(lp.features.values > 1)).count().

CHECK HIDE ANSWER

SPARSE VECTORS (3/3 points)

In Part (1b) we use a sparse vector representation to efficiently store a one-hot-encoded (OHE) feature vector. Imagine that we have 1000 OHE features, and that for a particular data point, we have *s* non-zero OHE features.

If *s* = 10, how much smaller is the storage footprint of the sparse vector representation versus the dense representation (assume that all indices and values are stored as floats)?

- ○ 100x
- ◉ 50x ✔
- ○ 10x
- ○ they are the same size

If *s* = 500, how much smaller is the storage footprint of the sparse vector representation versus the dense representation (assume that all indices and values are stored as floats)?

- ○ 100x
- ○ 50x
- ○ 10x
- ◉ they are the same size ✔

Suppose we would like to compute a dot product between this feature vector and a dense vector, and assume *s* = 10. How many fewer scalar multiplications must we perform if we use a sparse vector representation versus a dense representation of the feature vector (assume we have random access to the entries of the dense vector)?

- (●) 100x ✔
- ( ) 50x
- ( ) 10x
- ( ) they are the same size

EXPLANATION

The dense representation requires 1000 floats, while the sparse representation requires 2*s* floats, since for each non-zero we must store the index and the value. When computing a dot product, we only need to consider the non-zero entries of the feature vector, so we only need to perform *s* scalar multiples when using a sparse representation, versus 1000 scalar multiplies when using a dense representation.

CHECK HIDE ANSWER

HASHING (3/3 points)

In Part (5a) of the coding assignment we hashed the three sample points using numBuckets=4 and numBuckets=100. Complete the three statements below about these hashed features summarized in the following table using each answer once.

Name	Raw Features	4 Buckets	100 Buckets
sampleOne	[(0, 'mouse'), (1, 'black')]	{2: 1.0, 3: 1.0}	{14: 1.0, 31: 1.0}
sampleTwo	[(0, 'cat'), (1, 'tabby'), (2, 'mouse')]	{0: 2.0, 2: 1.0}	{40: 1.0, 16: 1.0, 62: 1.0}
sampleThree	[(0, 'bear'), (1, 'black'), (2, 'salmon')	{0: 1.0, 1: 1.0, 2: 1.0}	{72: 1.0, 5: 1.0, 14: 1.0}

With 100 buckets, sampleOne and sampleThree both contain index 14 due to _________.

underlying properties of the data ▾ underlying properties of the data ✔

It is likely that sampleTwo has two indices with 4 buckets, but three indices with 100 buckets due to _________.

the fact we go from 4 to 100 buckets ▼ the fact we go from 4 to 100 buckets ✔

With 4 buckets, sampleTwo and sampleThree both contain index 0 due to _________.

a hash collision ▼ a hash collision ✔

EXPLANATION

sampleOne and sampleThree both contain the feature (1, 'black'), and this property of the data explains why they both contain index 14 when hashing with 100 buckets.

As we increase the number of buckets, we reduce the likelihood of hash collisions. Given that we have 7 unique features to hash, collisions must occur when we hash with 4 buckets, yet are unlikely to occur when we hash with 100 buckets.

sampleTwo and sampleThree do not share any features in common, but as mentioned above, when using only 4 hash buckets with 7 unique features, hash collisions will occur.

CHECK HIDE ANSWER

SURVEY: LAB4 COMPLETION TIME (1/1 point)

How long did Lab FOUR take you to complete (in hours - decimals are OK)?

4 ✔

4

Please click "Check" to save your answers.

CHECK SHOW ANSWER