# Homework 4 Questions

## Instructions

- 6 questions.
- Write code where appropriate; feel free to include images or equations.
- Please make this document anonymous.
- This assignment is **fixed length**, and the pages have been assigned for you in Gradescope. As a result, **please do NOT add any new pages**. We will provide ample room for you to answer the questions. If you *really* wish for more space, please add a page *at the end of the document*.
- **We do NOT expect you to fill up each page with your answer.** Some answers will only be a few sentences long, and that is okay.

## Questions

### Q1:

(a) Define these common terms in machine learning:

(i) Bias

(ii) Variance

(b) Define these terms in the context of evaluating a classifier:

(i) Overfitting

(ii) Underfitting

(c) How do overfitting and underfitting relate to bias and variance?

*Please answer overleaf.*

**A1:** Your answer here.

(a) (i) Bias describes the error between model's prediction and the target value on our training dataset. (The error can be measured through metrics like RMSE)

(ii) Variance describes the difference between models' predictions when they are trained through different datasets.

(b) (i) When model has low bias and high variance. If classifier can predict very accurately on training set but performs poorly on test set, or varies significantly through training from different datasets, it is overfitting.

(ii) When model has high bias and low variance. IF classifier predicts very badly on across different datasets, it is underfitting.

(c) When model has low bias and high variance, it is overfitting. When model has high bias and low variance, it is underfitting.

**Q2:** Suppose we create a visual word dictionary using SIFT and k-means clustering for a scene recognition algorithm. Examining the SIFT features generated from our training database, we see that many are almost equidistant from two or more visual words.

(a) Why might this affect classification accuracy?

(b) Given the situation, describe *two* methods to improve classification accuracy, and explain why they would help.
*These can be for k-means, or otherwise.*

**A2:** Your answer here.

(a) Since many words have equal distance from a certain feature, that means two or more words are equally likely to be matched in this prediction. Therefore, the accuracy will be lower than $50\%$, which is bad. In this case, the model will output wrong results easily and randomly based on the training set.

(b) Firstly, we can increase the number of pixels covered by a feature, which means features are more likely to be different from each other. Another way is to use kernel trick increase the dimension of the features, which allow us to explore more potential information to identify each word's feature from SIFT.

Only Q2 and A2 answers should be on this page

**Q3:** The way that the bag of words representation handles the spatial layout of visual information can be both an advantage and a disadvantage.

(a) Describe an example scenario for each of these cases.

(b) Describe a modification or additional algorithm which might overcome the disadvantage.

(c) How might we determine whether bag of words is a good model?

**A3:** Your answer here.

(a) Advantage scenario: We can use bag of words to compare the relativity between two documents efficiently. Disadvantage Scenario: it is hard for bag of words to understand the sequence of a sentence and predict the next word of a sentence. (Reverse of a sentence does not post a difference in bag of words model, but it generates a significant difference in the semantic meaning).

(b) N-gram statistical model is a choice to model the sequence of words. It assumes that the Nth word has relationship with N-1 words before it. Therefore, it cuts documents into (N-word) vectors called N-gram. Since N words in N-gram together represent a feature in the feature space, the sequence of words is measured and semantic meaning is preserved.

(c) It depends on the task that we needs to complete. Bag of words model breaks links between words but preserves the spatial layout of the document. If the task does not require the information between words (sequence) but focuses on the general description (topic, content), then bag of words model is a good choice. Also, bag of words is fast and efficient.

**Q4:** Goal: Recognize a biased dataset and understand how you classify data affects the output and outcome of computer vision machine learning models.

Data bias affects machine learning, and recent national news has highlighted data bias in object detection. In one case, researchers discovered that current pedestrian detection models identified darker-skinned pedestrians with 5% less accuracy than lighter-skinned pedestrians. The researchers investigate multiple reasons for this inaccuracy, but one reason could be that the training dataset had 29% darker-skinned and 71% lighter-skinned labeled pedestrians. While measuring this difference in the data might be simple for pedestrians, other data biases can be harder to describe.

In Homework 4, we will train a scene recognition model using data from Lazebnik et al. 2006. Please review its data to check for biases: observe image samples in the data/train and data/test directories and consider their class labels.

(a) Does this dataset contain potentially harmful biases? If so, describe them and why they might be harmful. (3–4 sentences)

In many cases these data sets are collected by graduate students and meant to be used for academic research.

(b) What implications does this have if other organizations use these datasets for their models? Describe an application that uses this scene data in an unanticipated way. How might this reveal other overlooked biases? (3–4 sentences)

Please read the following two short articles: Article 1 and Article 2.

(c) Do these articles change your answers? Why or why not? (2–3 sentences)

**A4:** Your answer here.

(a) In the dataset, we have mostly gray-scaled pictures and a small amount of colored pictures (most in ”Industrial” and ”Store”) which may cause bias in recognizing colorful inputs and scenes in those cases. Also, we have lots of cars appearing in ”highway” pictures, which may cause data bias when classifying ”insdeCity” since there are also many cars inside the city. Also, the people appearing on the Mountain may also cause data bias for the same reason. Also, the ”Kitchen” mostly contain western styles tools and layouts, which cause data bias in recognizing Eastern kitchens.

(b) Since the graduate students have limited resources for collecting data, and they may only collect data that matches their knowledge for a certain class. If an application is built on this dataset, the bias will exist because of different understanding of the same class between different areas of the world. For example, the ”InsideCity” class may have different features in America and in Mexico. Also, the ”Office” class may also have different representations in America and in China. Therefore, this application will have low accuracy when used in different countries.

(c) The Article1 provides additional data bias to mine, which is the household items in living room for different families. The Ariticle2 matches my findings in (b).

**Q5:** Given a linear classifier such as SVM which separates two classes (binary decision), how might we use multiple linear classifiers to create a new classifier which separates $k$ classes?

Below, we provide pseudocode for a linear classifier. It trains a model on a training set, and then classifies a new test example into one of two classes. Please edit the pseudo-code to convert this into a multi-class classifier.

*Hints:* See slides in supervised learning crash course deck, plus your own research. You can take either the one vs. all (or one vs. others) approach or the one vs. one approach in the slides; please declare which approach you take.

*More hints:* Be aware that 1) the input labels in the multi-class case are different, and you will need to match the expected label input for the `train_linear_classifier` function, 2) you need to make a new decision on how to aggregate or decide on the most confident prediction.

*Note:* A more efficient software application would separate the classifier training and testing into two different functions so that the model could be reused without retraining. Feel free to ignore this for now.

*Please answer overleaf.*

**A5:** Your answer here.

I use the one vs. others approach.

```
# Inputs
#   train_feats: N x d matrix of N features each d descriptor long
#   train_labels: N x 1 array containing values of either -1 (class 0)
                                    or 1 (class 1)
#   test_feat: 1 x d image for which we wish to predict a label
#
# Outputs
#   -1 (class 0) or 1 (class 1)
#
# Please turn this into a multi-class classifier for k classes.
# Inputs:
#    As before, except
#    train_labels: N x 1 array of class label integers from 0 to k-1
# Outputs:
#    A class label integer from 0 to k-1
#
def classify(train_feats, train_labels, test_feat):
    # Train classification hyperplane use one vs all method
    classifier = {}
    for i in range(k):
        train_k_label = [1 if i == k else 0 for i in train_labels]
        classifier[i] = train_linear_classifier(train_feats,
                                           train_k_label)
    # Compute distance from hyperplane
    scores = {}
    for i in classifier.keys():
        weights, bias - classifier[i]
        test_score = weights * test_feats + bias
        scores[i] = test_score
    highest = [k for k,_ in sorted(scores.items(), lambda x: x[1])][0]

    return highest
```

**Q6:** In Homework 4, we will use a feature descriptor called HOG—'Histogram of Oriented Gradients'. As its name implies, it works similarly to SIFT. In classification, we might extract HOG features across the entire image (not just at corners) to create visual words.

HOG creates a feature descriptor per image 'block'. Each block is split into 'cells' covering pixels. HOG outputs a 9-bin histogram of oriented gradients per cell. We append these together to obtain the feature descriptor for each block. As a result, if we have $(z, z)$ cells per block, the feature descriptor for each block will be of size $z \times z \times 9$.

*Blocks can overlap as displayed in the diagram below.*

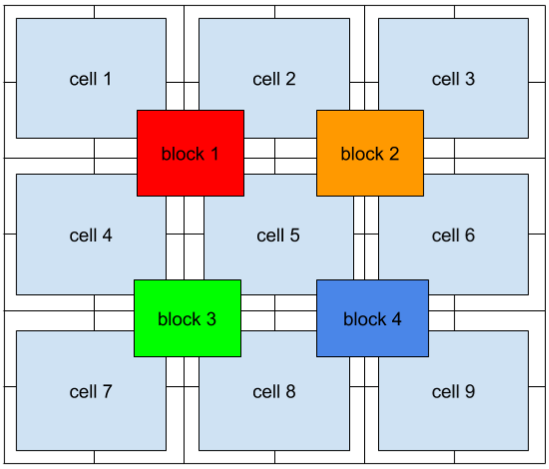

image dimensions = 6x6

pixels per cell = 2x2
cells per block = 2x2

There are 9 cells and
4 blocks according to
these specifications

When using HOG, the parameters such as pixels per cell and cells per block impact the resulting feature descriptor and so our performance on a classification task.

(a) Given a $72 \times 72$ image, calculate the number of cells, blocks, and feature vector size that will occur when we extract HOG features with the following parameters.

Scenario 1: Pixels per cell = $4 \times 4$, cells per block = $4 \times 4$
*Calculate:*
Number of cells:
Number of blocks:
Dimensions of resulting feature descriptor:

Scenario 2: Pixels per cell = $8 \times 8$, cells per block = $2 \times 2$.
*Calculate:*
Number of cells:
Number of blocks:
Dimensions of resulting feature descriptor:

(b) What are the pros and cons of the two parameter combinations? Which might you expect to have better performance?

*Note: You may find it useful to look at the thesis of Navneet Dalal (co-inventor of HOG) for more on this topic. [Link to thesis] (pages 39, 41 in Section 4.3).*

**A6:** Your answer here.

a) Scenario 1:
Number of cells: 324
Number of blocks: 225
Dimensions of resulting feature descriptor: $4 \times 4 \times 9$

Scenario 2:
Number of cells: 81
Number of blocks: 64
Dimensions of resulting feature descriptor: $2 \times 2 \times 9$

What are the pros and cons of the two parameter combinations? Which might you expect to have better performance?
The pros is that it can improve the performance. The cons is that it greatly increases the size of the descriptor. I though combination of blocks with different scales might have better performance.

**Secret something to think about:** Given a linear classifier like SVM, how might we handle data that are not linearly separable? How does the *kernel trick* help in these cases?

*Hint: See slides in supervised learning crash course deck, plus your own research.*

**A:** Your answer here. The non-linear kernel increase the dimension of the features of data, which may become separable in higher dimensions. As in maths, if the original vector space contain limited dimension, which means the feature number is limited, then there must exists a higher dimension feature that can sperate the sample vector space. Since some data can be separated non-linearly, such as a circle or an eclipse, we need non-linear SVM to solve this problem. However, it s hard to solve a non-linear problem in Maths, but we can transform the original vector space into a new vector space (higher dimension) in order that the circle or eclipse becomes a line. Hence, the problem becomes linear separable, which we can solve easily. The kernel here performs the transformation and also avoids the crash brought by significantly high dimension input space. (by separating kernel into inner product of two mapping functions)

## Feedback? (Optional)

Please help us make the course better. If you have any feedback for this assignment, we'd love to hear it!

Only feedback should be on this page