# CS 475 Machine Learning: Project 1
# Supervised Classifiers 1
Due: Thursday February 22, 2018, 11:59pm
50 Points Total Version 1.0

Mengqi Qin (mqin2)

## 1 Analytical (20 Points)

In addition to completing the analytical questions, your assignment for this homework is to learn Latex. All homework writeups must be PDFs compiled from Latex. Why learn latex?

1. It is incredibly useful for writing mathematical expressions.
2. It makes references simple.
3. Many academic papers are written in latex.

The list goes on. Additionally, it makes your assignments much easier to read than if you try to scan them in or complete them in Word.

We realize learning latex can be daunting. Fear not. There are many tutorials on the Web to help you learn. We recommend using pdflatex. It's available for nearly every operating system. Additionally, we have provided you with the tex source for this PDF, which means you can start your writeup by erasing much of the content of this writeup and filling in your answers. You can even copy and paste the few mathematical expressions in this assignment for your convenience. As the semester progresses, you'll no doubt become more familiar with latex, and even begin to appreciate using it.

Be sure to check out this cool latex tool for finding symbols. It uses machine learning! `http://detexify.kirelabs.org/classify.html`

**1) Supervised vs. Unsupervised Learning (3 points)** .

1. Give an example of a problem that could be solved with both supervised learning and unsupervised learning. Is data readily available for this problem? How would you measure your 'success' in solving this problem for each approach?
2. What are the pros and cons of each approach? Which approach do you think the problem better lends itself to?

**Answer:**

1. Lets say we want to classify German Shepherd Dog and wolfhound. The appearance of these two dogs are similar, but at most case, we can still tell whether an animal is a German Shepherd Dog or a wolfhound. Because most German Shepherd Dog are gentle, but the wolfhounds are always fierce.

Very few of the dogs will actually be recorded the type. And to get the label of the dog will be hard. So, we can get some labeled dogs which has been recorded online to do supervised learning on it. And for many other datasets (many other unlabeled dogs)which dont have the label, we can do unsupervised learning on it. Thus, we have the learning algorithm. And if we use other data sets which already have the label to predict the type(to see if an animal is German Shepherd Dog or wolf) using the algorithm we got from the learning procedure. If the result is the same with the label, then we succeed.

2. supervised learning

   Pros: Good performance

   Cons: Labeled data is difficult to find

   unsupervised learning

   Pros: Easy to find lots of data

   Cons: Finding patterns of interest

   In the example I listed above, I think unsupervised learning fits better. Because it's hard to get the label of a dog, very few of them has been recorded, but to get the dogs' feature like characteristic, appearance and some other feature is easier and use these data sets can help us cluster them using the patterns of interest and get the predicted label.

**2) Model Complexity (3 points)** Explain when you would want to use a simple model over a complex model and vice versa. Are there any approaches you could use to mitigate the disadvantages of using a complex model?

**Answer:**

When the requirement is not high, and there are small data sets, simple model can solve the problem, there is no need to use the complex one. It might even caught overfitting problems.But when the data sets are large. It will need the complex model.

approaches: enlarge training data sets; early stop to mitigate; combine different models; dropout

**3) Training and Generalization (3 points)** Suppose you're building a system to classify images of food into two categories: either the image contains a hot dog or it does not. You're given a dataset of 25,000 (image, label) pairs for training and a separate dataset of 5,000 (image, label) pairs.

1. Suppose you train an algorithm and obtain 96% accuracy on the larger training set. Do you expect the trained model to obtain similar performance if used on newly acquired data? Why or why not?

2. Suppose that, after training, someone gives you a new test set of 1,000 (image, label) pairs. Which do you expect to give greater accuracy on the test set: The model after trained on the dataset of 25,000 pairs or the model after trained on the dataset of 5,000 pairs? Explain your reasoning.

3. Suppose your models obtained greater than 90% accuracy on the test set. How might you proceed in hope of improving accuracy further?

**Answer:**

1. Yes. I expect a similar performance. Because the number of training data set is big to some extent. It forms a good training process and the accuracy is already very high to some extent. And use larger training set, the training process is almost the same. The accuracy might be a little different but will still be similar to 96%.

2. At most case, its the model after trained on the dataset of 25,000 pairs. Because at most case, the more data sets there are, the higher the accuracy will reach. With more data sets, the training procedure can be more precise, and it can learn better to label the new data. But to some extreme circumstance, if the new data sets are very similar to the 5000 pairs, and the 25000 pairs dont include useful information, it might be the 5000 pairs.

3. Enlarge data sets to improve the training process.

**4) Loss Function (3 points)** State whether each of the following is a valid loss function for binary classification. Wherever a loss function is not valid, state why. Here $y$ is the correct label and $\hat{y}$ is a decision confidence value, meaning that the predicted label is given by $\text{sign}(\hat{y})$ and the confidence on the classification increases with $|\hat{y}|$.

1. $\ell(y, \hat{y}) = \frac{3}{4}(y - \hat{y})^2$
2. $\ell(y, \hat{y}) = |(y - \hat{y})|/\hat{y}$
3. $\ell(y, \hat{y}) = \max(0, 1 - y \cdot \hat{y})$

**Answer:**

1. It's not valid. The minimum point is when y equals $\hat{y}$. But $|y - \hat{y}|$ have same result for 2 values and will got the same result. E.g. when y = 1, $\hat{y} = 2$ or 0 has the same loss function which is not correct.

2. Not valid. Loss function should not be negative. This function obviously doesn't meet the requirement.

3. Valid. If it predicts correctly, the y should have the same sign with $\hat{y}$. When the $\hat{y}$ predicts correctly, the loss function decreases monotonously, until the loss function reaches 0, that's the best prediction. When the $\hat{y}$ predicts wrong, the loss function increases monotonously.

**5) Linear Regression (4 points)** Suppose you observe $n$ data points $(x_1, y_1), \ldots, (x_n, y_n)$, where all $x_i$ and all $y_i$ are scalars.

1. Suppose you choose the model $\hat{y} = wx$ and aim to minimize the sum of squares error $\sum_i (y_i - \hat{y}_i)^2$. Derive the closed-form solution for $w$ from scratch, where 'from scratch' means without using the least-squares solution presented in class.

2. Suppose you instead choose the model $\hat{y} = w \sin(x)$ and aim to minimize the sum of squares error $\sum_i (y_i - \hat{y}_i)^2$. Is there a closed-form solution for $w$? If so, what is it?

**Answer:**

1. I use the method of completing square to get the minimum sum of squares error.

$$\begin{aligned}\sum_i (y_i - \hat{y}_i)^2 &= \sum_i (y_i - wx_i)^2 \\ &= \sum_i (y_i^2 - 2x_i y_i w + w^2 x_i^2) \\ &= (\sum_i x_i^2) w^2 - 2(\sum_i x_i y_i) w + \sum_i y_i^2 \\ &= (\sum_i x_i^2)[w^2 - 2\frac{\sum_i x_i y_i}{\sum_i x_i^2} w] + \sum_i y_i^2 \\ &= (\sum_i x_i^2)(w - \frac{\sum_i x_i y_i}{\sum_i x_i^2})^2 - \frac{(\sum_i x_i y_i)^2}{\sum_i x_i^2} + \sum_i y_i^2\end{aligned}$$

   Based on the above equation, I can get that when
   $w = \frac{\sum_i x_i y_i}{\sum_i x_i^2}$, the sum of squares error is minimized and equals to $\sum_i y_i^2 - \frac{(\sum_i x_i y_i)^2}{\sum_i x_i^2}$

2. Yes. I still use the method of completing square to get the minimum sum of squares error.

$$\begin{aligned}\sum_i (y_i - \hat{y}_i)^2 &= \sum_i (y_i - w\sin(x_i))^2 \\ &= \sum_i (y_i^2 - 2\sin(x_i) y_i w + w^2 \sin^2(x_i)) \\ &= (\sum_i \sin^2(x_i)) w^2 - 2(\sum_i sin(x_i) y_i) w + \sum_i y_i^2 \\ &= (\sum_i \sin^2(x_i))[w^2 - 2\frac{\sum_i \sin(x_i) y_i}{\sum_i \sin^2(x_i)} w] + \sum_i y_i^2 \\ &= (\sum_i \sin^2(x_i))(w - \frac{\sum_i \sin(x_i) y_i}{\sum_i \sin^2(x_i)})^2 - \frac{(\sum_i \sin(x_i) y_i)^2}{\sum_i \sin^2(x_i)} + \sum_i y_i^2\end{aligned}$$

   Based on the above equation, I can get that when

$$w = \frac{\sum_i sin(x_i) y_i}{\sum_i \sin^2(x_i)}$$

   and the sum of squares error is minimized and equals to

$$\sum_i y_i^2 - \frac{(\sum_i \sin(x_i) y_i)^2}{\sum_i \sin^2(x_i)}$$

**6) Logistic Regression (4 points)** Explain whether each statement is true or false. If false, explain why.

1. In the case of binary classification, optimizing the logistic loss is equivalent to minimizing the sum-of-squares error between our predicted probabilities for class 1, $\hat{\mathbf{y}}$, and the observed probabilities for class 1, $\mathbf{y}$.

2. One possible advantage of stochastic gradient descent is that it can sometimes escape local minima. However, in the case of logistic regression, the global minimum is the only minimum, and stochastic gradient descent is therefore never useful.

**Answer:**

1. **False**. The logistic function is $l(\mathbf{w}, \mathbf{x}, y) = \ln(1 + e^{-y[\mathbf{w} \cdot \mathbf{x}]})$
and the sum-of-square graph got a minimum value but not have to be the same with the weight we got from the logistic function.

2. **False.** The stochastic gradient descent is useful when you can pick one data from the datasets at a time to train unlike logistic regression, you have to iterate through all datasets to do the training process. So when there is a large datasets, the stochastic gradient descent is fast.