

Introduction to Machine Learning

Instructor: Lara Dolecek

TA: Zehui (Alex) Chen, Ruiyi (John) Wu

Please upload your homework to Gradescope by April 23, 11:59 pm.

Please submit a single PDF directly on Gradescope. Make sure to make the page selection. You may type your homework or scan your handwritten version. Make sure all the work is discernible.

1. Consider the modified objective function of regularized logistic regression:

$$J(w) = - \sum_{n=1}^N [y_n \log h_w(x_n) + (1 - y_n) \log(1 - h_w(x_n))] + \frac{1}{2} \sum_i w_i^2 \quad (1)$$

where $h_w(x) = \sigma(w^T x)$ and the sigmoid function $\sigma(x) = \frac{1}{1+\exp(-x)}$. Find the partial derivative $\frac{\partial J}{\partial w_j}$ and derive the gradient descent update rules for the weights.

2. In class we have seen the probabilistic interpretation of the logistic regression objective, and the derivation of the gradient descent rule for maximizing the conditional likelihood. We have this maximum likelihood (ML) formulation for $w \in \mathbb{R}^m$:

$$w^* = \arg \max_w \prod_{i=1}^n P(y_i|x_i, w). \quad (2)$$

To prevent overfitting, we want the weights to be small. To achieve this, we consider some prior distribution of the weights $f(w)$ and use maximum a posteriori (MAP) estimation:

$$w^* = \arg \max_w \prod_{i=1}^n P(y_i|x_i, w)f(w). \quad (3)$$

Assuming a standard Gaussian prior $\mathcal{N}(0, I)$ for the weight vector, show that the MAP estimation is equivalent to minimizing the logistic regression objective with L2 regularization as shown in the previous question. Note: For $Z \sim \mathcal{N}(0, I_m)$,

$$f_Z(z) = \frac{1}{(2\pi)^{\frac{m}{2}}} \exp \left(-\sum_{i=1}^m \frac{z_i^2}{2} \right).$$

3. You are trying to choose between two restaurants (sample 9 and sample 10) to eat at. To do this, you will use a decision tree that will be trained based on your past experiences (sample 1-8). The features for each restaurants and your judgment on the goodness of sample 1-8 are summarized by the following chart.

Sample #	HasOutdoorSeating	HasBar	IsClean	HasGoodAtmosphere	IsGoodRestaurant
1	0	0	0	1	1
2	1	1	0	0	0
3	0	1	1	1	1
4	1	0	0	1	1
5	1	1	1	0	0
6	1	0	1	0	1
7	1	1	0	1	1
8	0	0	1	1	1
9	0	1	0	1	?
10	1	1	1	1	?

- (a) What is the entropy of IsGoodRestaurant, i.e., $H(\text{IsGoodRestaurant})$?
- (b) Calculate the conditional entropy of IsGoodRestaurant conditioning on HasOutdoorSeating. To do this, first compute $H(\text{IsGoodRestaurant}|\text{HasOutdoorSeating} = 0)$ and $H(\text{IsGoodRestaurant}|\text{HasOutdoorSeating} = 1)$, then weigh each term by the probabilities $P(\text{HasOutdoorSeating} = 0)$ and $P(\text{HasOutdoorSeating} = 1)$, respectively. Namely, calculate the following:

$$\begin{aligned} & H(\text{IsGoodRestaurant}|\text{HasOutdoorSeating}) \\ &= P(\text{HasOutdoorSeating} = 0)H(\text{IsGoodRestaurant}|\text{HasOutdoorSeating} = 0) \\ &+ P(\text{HasOutdoorSeating} = 1)H(\text{IsGoodRestaurant}|\text{HasOutdoorSeating} = 1). \end{aligned}$$

- (c) Similarly, calculate

$$H(\text{IsGoodRestaurant}|X), \text{ for } X \in \{\text{HasBar}, \text{IsClean}, \text{HasGoodAtmosphere}\},$$

i.e., the conditional entropy of IsGoodRestaurant conditioning on the other three features.

- (d) Calculate the information gain:

$$I(\text{IsGoodRestaurant}; X) = H(\text{IsGoodRestaurant}) - H(\text{IsGoodRestaurant}|X),$$

for

$$X \in \{\text{HasOutdoorSeating}, \text{HasBar}, \text{IsClean}, \text{HasGoodAtmosphere}\}.$$

- (e) Based on the information gain, determine the first attribute to split on.

- (f) Make the full decision tree. After each split, treat the sets of samples with $X = 0$ and $X = 1$ as two separate sets and redo (b), (c), (d) and (e) on each of them. X is the feature for previous split and is thus excluded from the available features which can be split on next. Terminate splitting if after the previous split, the entropy of *IsGoodRestaurant* in the current set is 0. For example, if we choose *HasGoodAtmosphere* as our first feature to split, we get $H(\text{IsGoodRestaurant}|\text{HasGoodAtmosphere} = 1) = 0$. We thus stop splitting the tree in this branch. Draw the tree and indicate the split at each node.
- (g) Now, determine if restaurants 9 and 10 are good or not.

4. When training decision trees for classification, we generally use entropy or gini index to help determine good splits. These functions are examples of impurity measures. We can formally define impurity measures as follows.

Assume we are performing binary classification. Let V be a set of data points and let $\{y_i \in \{+1, -1\}, \forall i \in V\}$ be the set of labels. Let $V_1 \subseteq V$. We define p and q as follows:

$$p(V_1, V) = \frac{|V_1|}{|V|} \quad q(V_1) = \frac{|\{i : i \in V_1, y_i = 1\}|}{|V_1|}$$

where $|\cdot|$ is the cardinality of a set.

Let $i(q(V))$ measure the impurity of a set V . Two desired properties of $i(q(V))$ are:

- $i(q(V)) = 0$ when the set has only one class
- $i(q(V))$ reaches its maximum value for sets where the two classes are perfectly balanced.

When a split is performed of a set V , the result is two sets $V_1 \cup V_2 = V$ such that $V_1 \cap V_2 = \emptyset$. The information gain of this split is defined as

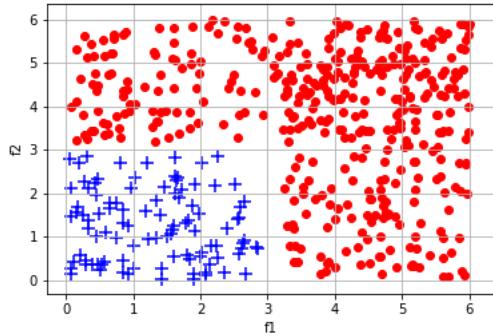
$$I(V_1, V_2, V) = i(q(V)) - (p(V_1, V)i(q(V_1)) + p(V_2, V)i(q(V_2))).$$

Using the previous notation, we define the binary gini Index and binary entropy as:

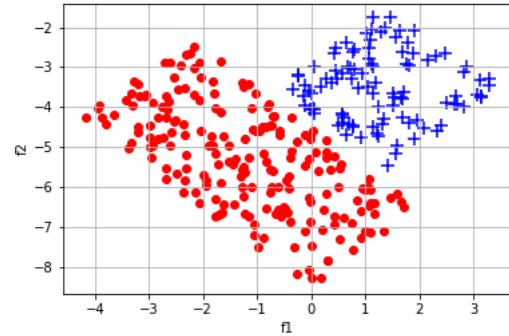
$$\begin{aligned} gini(q(V)) &= 2q(V)(1 - q(V)) \\ H(q(V)) &= -(q(V)\log(q(V)) + (1 - q(V))\log(1 - q(V))) \end{aligned}$$

- (a) Like binary entropy, gini index is an alternative impurity measure that is commonly used. Plot both the Gini index and the binary entropy function for $q(V)$ from 0 to 1. Comment on their similarities.
- (b) Show that if $i(q(V))$ is concave in $q(V)$ then $I(V_1, V_2, V) \geq 0 \forall V_1 \cup V_2 = V, V_1 \cap V_2 = \emptyset$. This means that every split does not lose any information. Recall that a function is concave if $\forall \lambda \in [0, 1]$ and $\forall q_1, q_2$ then $i(\lambda q_1 + (1 - \lambda)q_2) \geq \lambda i(q_1) + (1 - \lambda)i(q_2)$.
- (c) Show that binary entropy is concave. Hint: Show that the 2nd derivative is always non-positive.
- (d) Show that the binary Gini index is concave.

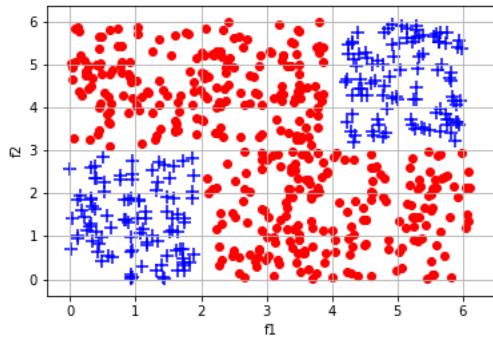
5. Consider the following plots:



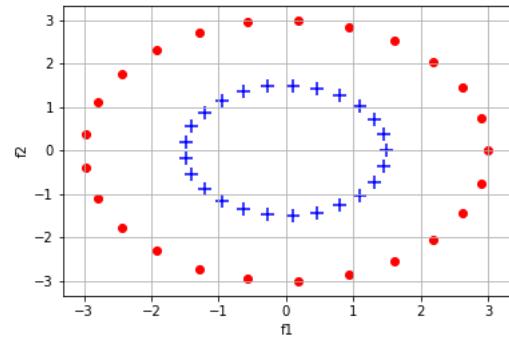
(1) Decision Tree Example 1



(2) Decision Tree Example 2



(3) Decision Tree Example 3



(4) Decision Tree Example 4

Assume that you are using a binary split decision tree to classify the points. At each node, you may ask questions like: “Is $f_1 \geq 3$?” or “Is $f_2 \leq 1.8$?” . Here, f_1 and f_2 are the variables on the horizontal axis and vertical axis of the examples, respectively.

- (a) Which examples can be fully separated using a depth 2 decision tree?
- (b) Which example would have the most complex decision tree in terms of the number of splits? Explain why.
- (c) If you used a depth 4 decision tree, is one more example now separable? If so, show how you can separate it using a depth 4 decision tree.

6. On April 15, 1912, the largest passenger liner ever made collided with an iceberg during her maiden voyage. When the Titanic sank it killed 1502 out of 2224 passengers and crew. This sensational tragedy shocked the international community and led to better safety regulations for ships. One of the reasons that the shipwreck resulted in such loss of life was that there were not enough lifeboats for the passengers and crew. Although there was some element of luck involved in surviving the sinking, some groups of people were more likely to survive than others.

In this section, you will use decision trees to predict whether a person survives or not. You will be using the data set provided in *dataTesting_X.csv*, *dataTesting_Y.csv*, *dataTraining_X.csv* and *dataTraining_Y.csv* which contains feature values and labels for both training and testing data. The data is normalized so that all feature values are between 0 and 1. If you are interested in the actual meaning of each feature, we provide the original data in *titanic.csv* where we use the mapping (Female:0, Male:1). (For python, feel free to use the following libraries: math, collections, numpy, matplotlib and sklearn.)

- (a) Before starting, it is useful to have a baseline accuracy to compare against. A simple baseline to consider is majority voting where the classifier always outputs the class with the majority in the training set. Provide the accuracy of this classifier on both the training and the testing set. To find the accuracy, calculate the fraction in both the training and the testing set that has the same class as the majority class in the training set.
- (b) Now, use **fitctree** (**sklearn.tree.DecisionTreeClassifier** for python user) to train a decision tree classifier on the training data. Make sure to set the splitting criteria to “deviance” (“entropy” for python user) to make it use cross entropy. What is the training and testing accuracy of this model?