Homework 1 of CS 165A (Spring 2023)

University of California, Santa Barbara

To be discussed on Apr 19, 2023 (Wednesday)

Notes:

- The homework is optional. You do not need to submit your solutions anywhere and you will not be evaluated by these.
- To maximize your learning, you should try understanding the problems and try solving them as much as you can before the discussion class.
- Feel free to discuss with your peers / form small groups to solve these problems.
- Feel free to discuss any questions with the instructor and the TA in office hours or on Piazza.

---

1 Why should I do this homework?

This homework is about intelligent agent in general, classifier agents and machine learning. In Problem 1, you will practice organizing your thoughts, coming up with descriptions of agents. In Problem 2, you will zoom into classifier agents and develop an understanding of the idea of a “decision boundary”. Problem 3 helps you to understand how data splitting methods work mechanically. In Problem 4, you will do a simple theoretical exercise to see the gist of statistical learning theory, and to understand why data splitting works.

2 Homework problems

Problem 1. Intelligent agents, rationality and descriptions of the environment.

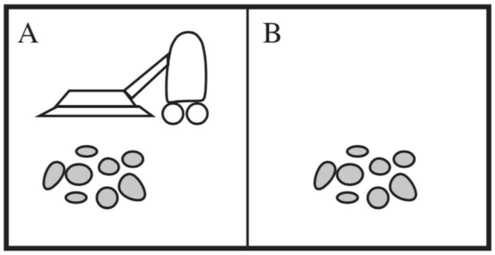
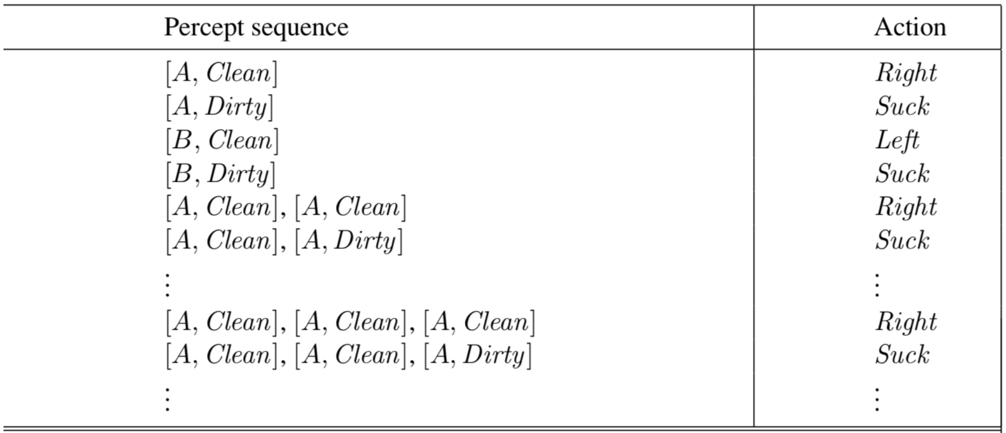


Figure 2.2 A vacuum-cleaner world with just two locations.



Percept sequence	Action
$[A, Clean]$	*Right*
$[A, Dirty]$	*Suck*
$[B, Clean]$	*Left*
$[B, Dirty]$	*Suck*
$[A, Clean], [A, Clean]$	*Right*
$[A, Clean], [A, Dirty]$	*Suck*
⋮	⋮
$[A, Clean], [A, Clean], [A, Clean]$	*Right*
$[A, Clean], [A, Clean], [A, Dirty]$	*Suck*
⋮	⋮

Figure 2.3 Partial tabulation of a simple agent function for the vacuum-cleaner world shown in Figure 2.2.

(a) Let us inspect the vacuum-cleaner example again from the textbook. Under the assumption listed on textbook page 38:

(i) Explain that why this vacuum-cleaner agent function described in Figure 2.3 is rational.

(ii) If each movement of the cleaner generate a unit cost, explain that now a rational agent needs to maintain an internal state.

(iii) Back to the original problem, now there is a naughty pet dog in the environment. At each time step, each clean square has a 50% chance of becoming dirty. Briefly explain how can you modified the rational agent in this case.

(b) For each of the following activities, give a **PEAS** description of the task environment and characterize it in terms of the properties listed in textbook Section 2.3.2.

(i) Playing soccer as a robot.

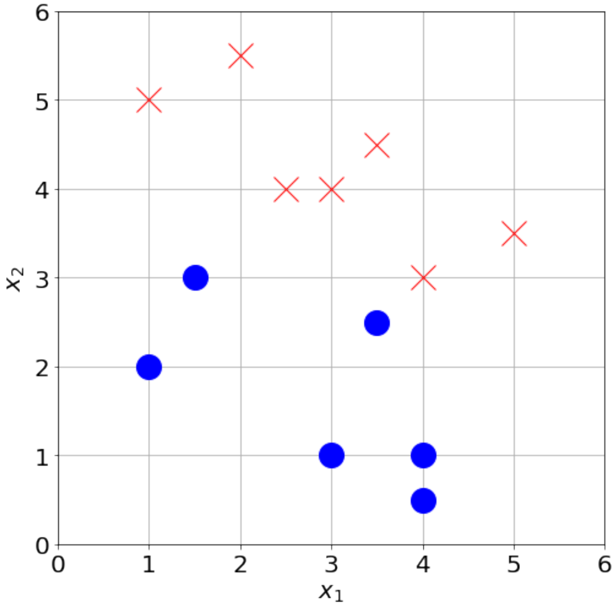
(ii) Internet laptop-shopping agent. (an intelligent agent that helps a customer to shop for a laptop on a website.)

Problem 2. Classifiers and Machine Learning.

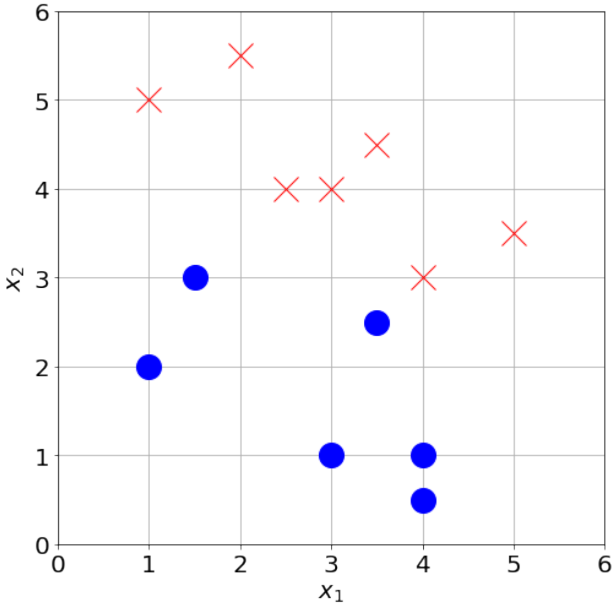
(a) What are the P,E,A,S description of a classifier agent?

(b) What are the “modelling”, “Learning”, “inference” components of a classifier agent design?

(c) Let the feature space be $\mathbb{R}^2$, i.e., two continuous features. Consider a decision tree classifier with depth = 2 (branching on one variable by thresholding on each level, the variables to branch on can repeat). What are the possible boundaries representable by this classifier on a 2D feature plane? Is there a depth 2 decision tree that gives perfect classification in the example above?



(d) Draw the decision boundary of a 1-nearest neighbor classifier in the example above.



Problem 3. Holdout and K-Fold Cross Validation.

Consider a dataset D with n instances that you want to use to train a classifier. Your task is to explore the concepts of holdout and k-fold cross validation methods and understand why we cannot use the training dataset to evaluate the performance of a classifier.

(a) **Holdout Method**: Divide the dataset D into two disjoint sets: a training set D_{train} and a validation set D_{val}. Explain the purpose of these two sets and their roles in training and validating a classifier.

(b) **K-Fold Cross Validation**: Explain the concept of k-fold cross validation, including the process of creating k partitions of the dataset, and how it helps in evaluating the performance of a classifier.

(c) **Comparison**: Compare the holdout method and k-fold cross validation in terms of their advantages and disadvantages. Discuss the situations in which you would prefer one method over the other.

(d) **Understanding**: Explain why it is not appropriate to use the training dataset to evaluate the performance of a classifier. Discuss the concept of overfitting and its relationship to evaluating classifiers.

Remark. Data splitting methods (including holdout and cross validation) are powerful for evaluating the performance of an agent because they make no assumptions about the the distribution $\mathcal{D}$, type of classifier to use, size of the data and so on.

Problem 4. (More on training error, test error, Hold-out data and cross validation) The next step is to build some tool for evaluating the performance of a classifier learned by a machine learning algorithm.

Let h be a classifier. More formally, $h : \mathcal{X} \to \mathcal{Y}$, where $\mathcal{X}$ is the feature space and $\mathcal{Y}$ is the label space. And we denote the space of all classifiers to be $\mathcal{H}$. Let $D = (x_1, y_1), ..., (x_n, y_n)$ be a dataset

The most natural performance metric is the classification error, which measures the expected error rate.

$$\text{Err}(h) := \mathbb{E}_{(x,y)\sim\mathcal{D}}[\mathbf{1}(h(x) \neq y)]$$

where $\mathbf{1}(\cdot)$ is the indicator function that outputs 1 is the condition is true and 0 otherwise.

We can also define the error that we calculate on a dataset. We denote the empirical error on this data set as

$$\hat{\text{Err}}(h, D) := \frac{1}{\text{\# of data points in } D} \sum_{(x,y)\in D} \mathbf{1}(h(x) \neq y).$$

Furthermore, we define the generalization error to be

$$\text{GenErr} := \max_{h\in\mathcal{H}} |Err(h) - \hat{Err}(h)|$$

— the difference between the training error and the *expected* error on a **new data point** from the same distribution $\mathcal{D}$ on **all** classifier $h \in \mathcal{H}$.

(a) Let $\hat{h}$ be the learned classifier and h^* be the optimal classifier., i.e.

$$\hat{h} = \arg\min_h \hat{\text{Err}}(h, D), \quad h^* = \arg\min_h \text{Err}(h)$$

One can prove that

$$\text{Err}(\hat{h}) \leq \text{Err}(h^*) + 2\text{GenErr}.$$

The basic question is that: what does the equation say?

The advanced question is for you to come up with a proof of this fact. (Hint: if $a^* = \arg\max_a f(a)$ for some f, then for any b, we know $f(a) \geq f(b)$. Similarly for any $b \in S$, $f(b) \leq \max_{a \in S}(f(a))$. Also, you may need to use the triangle inequality of absolute value: $|a + b| \leq |a| + |b|$.)

(b) Assume that the data points in dataset D are drawn i.i.d. (independently and identically distributed) from some unknown distributions $\mathcal{D}$ defined on $\mathcal{X} \times \mathcal{Y}$. In practice, we can evaluate a classifier with data splitting. We randomly partition the data into a training data set D_{train} and a holdout validation dataset D_{val}.

Now let $\tilde{h}$ be any classifier we obtained by training on D_{train} (note that this doesn't have to be the classifier that minimizes the error, it can be one that minimizes the logistic loss, or one that maximizes the logistic loss or anything else we obtain using D_{train}).

One can show that

$$\mathbb{E}[\hat{\text{Err}}(\tilde{h}, D_{\text{Val}})|\tilde{h}] = \text{Err}(\tilde{h}).$$

Notice that both $\tilde{h}, D_{\text{Val}}$ are random ($\tilde{h}$ is random because its training data is random). In the above equation, we are conditioning on $\tilde{h}$ thus the expectation on the LHS is only averaging over that of the data points in the validation dataset.

The basic question is: What does the equation say?

The advanced question is for you to come up with a proof of this fact. (Hint: apply the definition of expectation and the fact that all data points in the validation dataset are independently drawn from $\mathcal{D}$.)

(c) (For advanced students) Let $E_1, ..., E_K$ be the error estimates you get from each of the K-fold in your cross validation setup. Are $E_1, ..., E_K$ independent to each other?

Assume dataset D is iid and n is an integer multiple of K. Further assume that the procedure for training in each fold is the same, one can prove that

$$\mathbb{E}[\frac{1}{K}\sum_{k=1}^{K} E_k] = \mathbb{E}[\text{Err}(\tilde{h}_1)],$$

where $\tilde{h}_1$ is the classifier trained on the first fold (using $(k-1)/k$ fraction of the dataset D) and the expectation is now taken over all randomness.

What does this equation say? For very advanced students, can you prove it?