MATH 462 ASSIGNMENT 3
DRAFT VERSION October 24, 2022

ADAM M. OBERMAN

Instructions. Refer to class notes and typed notes posted on `https://adam-oberman.github.io/Math462/`, as well as textbook references.

Submit your solutions on MyCourses course page.

Math Solutions should be handwritten. You can get help from other students, but you should do the write up yourself. Code solutions can be a PDF of a workbook, you can use any language.

3.1. Projections and PCA (Theory).

Exercise 3.1. Let S^m be a collection of m vectors in $\mathbb{R}^n$. Form the $m \times d$ matrix whose ith row is $x_i^\top$.

$$X = [x_1, \dots, x_m]^\top \in \mathbb{R}^{m \times d}$$

Show that

$$X^\top X = \sum_{i=1}^{m} x_i x_i^\top$$

Hint: write out an expression for $(X^\top X)_{ij}$ as the dot product of the ith row and jth column of the corresponding matrices.

Exercise 3.2. Let U be a linear subspace of $\mathbb{R}^n$.

(a) Show that $\| \operatorname{Proj}_U x\| \leq \|x\|$.
(b) Show that $\operatorname{Proj}_U x = x$ if and only if $x \in U$.
(c) Show that if $\operatorname{Proj}_U x = x$ for all $x \in \mathbb{R}^n$, then $U = \mathbb{R}^n$.

Exercise 3.3. Let O be an orthogonal matrix. Prove that $\|Ox\| = \|x\|$.

Exercise 3.4. Let A be a symmetric positive definite matrix, and consider the optimization problem

$$\max_{\|x\|=1} x^T A x \tag{1}$$

Prove that every critical point of the constrained optimization problem (3.4) satisfies $Ax = \lambda x$ for some scalar λ. Assume that x^* is an optimizer of (3.4). Prove that is corresponds to the eigenvector of A with the largest eigenvalue. Hint: use the technique of Lagrange multipliers for constrained optimization.

3.2. Decision Trees (Example and Theory).

Exercise 3.5. Consider the following data set comprised of three binary input attributes A_1, A_2, and A_3 and one binary output:

Example	A_1	A_2	A_3	Output y
x_1	1	0	0	0
x_2	1	0	1	0
x_3	0	1	0	0
x_4	1	1	1	1
x_5	1	1	0	1

Use the algorithm in Figure 1 to learn a decision tree for these data. Show the computations made to determine the attribute to split at each node.

Exercise 3.6 (leaf-classification-exercise). In the recursive construction of decision trees, it sometimes happens that a mixed set of positive and negative examples remains at a leaf node, even after all the attributes have been used. Suppose that we have p positive examples and n negative examples.

(1) ~~Show that the solution used by the Decision-Tree-Learning algorithm , which picks the majority classification, minimizes the absolute error over the set of examples at the leaf.~~
(2) Show that the class probability $p/(p+n)$ minimizes the sum of squared errors.

Exercise 3.7 (nonnegative-gain-exercise). Suppose that an attribute splits the set of examples E into subsets E_k and that each subset has p_k positive examples and n_k negative examples. Show that the attribute has strictly positive information gain unless the ratio $p_k/(p_k + n_k)$ is the same for all k.

function DECISION-TREE-LEARNING(*examples*, *attributes*, *parent_examples*) **returns** a tree

  if *examples* is empty **then return** PLURALITY-VALUE(*parent_examples*)
  else if all *examples* have the same classification **then return** the classification
  else if *attributes* is empty **then return** PLURALITY-VALUE(*examples*)
  else
    $A \leftarrow \text{argmax}_{a \in attributes}$ IMPORTANCE(a, *examples*)
    tree ← a new decision tree with root test A
    for each value v_k of A **do**
      $exs \leftarrow \{e : e \in examples$ **and** $e.A = v_k\}$
      subtree ← DECISION-TREE-LEARNING(*exs*, *attributes* − A, *examples*)
      add a branch to *tree* with label $(A = v_k)$ and subtree *subtree*
    return *tree*

Figure 18.5 The decision-tree learning algorithm. The function IMPORTANCE is described in Section 18.3.4. The function PLURALITY-VALUE selects the most common output value among a set of examples, breaking ties randomly.

FIGURE 1. Decision Tree Algorithm

3.3. **PCA coding.** Use the code `https://colab.research.google.com/drive/1MjaWPqB9-sQSI9r_Egj9Cu1AhBX276eU?usp=sharing`

Exercise 3.8.
(1) Use the code provided to compute the covariance matrix for the data Y produced by the code sample. Print out the empirical covariance matrix $\widehat{C}$ and the true covariance matrix C. Plot the eigenvectors of the covariance matrix along with the data.
(2) Now, choose the principal eigenvector, v_1 (the one with the larger eigenvalue) and plot the the projections of the data onto U, the linear subspace spanned by v_1. Hint: the formula for the projection is $P_U(x) = (x^T v)v$.