

**Problem 2** (Maximum likelihood in classification, 15pts)

Consider now a generative  $K$ -class model. We adopt class prior  $p(\mathbf{y} = C_k; \boldsymbol{\pi}) = \pi_k$  for all  $k \in \{1, \dots, K\}$  (where  $\pi_k$  is a parameter of the prior). Let  $p(\mathbf{x}|\mathbf{y} = C_k)$  denote the class-conditional density of features  $\mathbf{x}$  (in this case for class  $C_k$ ). Consider the data set  $D = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^n$  where as above  $\mathbf{y}_i \in \{C_k\}_{k=1}^K$  is encoded as a one-hot target vector and the data are independent.

1. Write out the log-likelihood of the data set,  $\ln p(D; \boldsymbol{\pi})$ .
2. Since the prior forms a distribution, it has the constraint that  $\sum_k \pi_k - 1 = 0$ . Using the hint on Lagrange multipliers below, give the expression for the maximum-likelihood estimator for the prior class-membership probabilities, i.e.  $\hat{\pi}_k$ . Make sure to write out the intermediary equation you need to solve to obtain this estimator. Briefly state why your final answer is intuitive.

For the remaining questions, let the class-conditional probabilities be Gaussian distributions with the same covariance matrix

$$p(\mathbf{x}|\mathbf{y} = C_k) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}), \text{ for } k \in \{1, \dots, K\}$$

and different means  $\boldsymbol{\mu}_k$  for each class.

3. Derive the gradient of the log-likelihood with respect to vector  $\boldsymbol{\mu}_k$ . Write the expression in matrix form as a function of the variables defined throughout this exercise. Simplify as much as possible for full credit.
4. Derive the maximum-likelihood estimator  $\hat{\boldsymbol{\mu}}_k$  for vector  $\boldsymbol{\mu}_k$ . Briefly state why your final answer is intuitive.
5. Derive the gradient for the log-likelihood with respect to the covariance matrix  $\boldsymbol{\Sigma}$  (i.e., looking to find an MLE for the covariance). Since you are differentiating with respect to a *matrix*, the resulting expression should be a matrix!
6. Derive the maximum likelihood estimator  $\hat{\boldsymbol{\Sigma}}$  of the covariance matrix.

**Hint: Lagrange Multipliers.** Lagrange Multipliers are a method for optimizing a function  $f$  with respect to an equality constraint, i.e.

$$\min_{\mathbf{x}} f(\mathbf{x}) \text{ s.t. } g(\mathbf{x}) = 0.$$

This can be turned into an unconstrained problem by introducing a Lagrange multiplier  $\lambda$  and constructing the Lagrangian function,

$$L(\mathbf{x}, \lambda) = f(\mathbf{x}) + \lambda g(\mathbf{x}).$$

It can be shown that it is a necessary condition that the optimum is a critical point of this new function. We can find this point by solving two equations:

$$\frac{\partial L(\mathbf{x}, \lambda)}{\partial \mathbf{x}} = 0 \quad \text{and} \quad \frac{\partial L(\mathbf{x}, \lambda)}{\partial \lambda} = 0$$

**Cookbook formulas.** Here are some formulas you might want to consider using to compute difficult gradients. You can use them in the homework without proof. If you are looking to hone your matrix calculus skills, try to find different ways to prove these formulas yourself (will not be part of the evaluation of this homework). In general, you can use any formula from the matrix cookbook, as long as you cite it. We opt for the following common notation:  $\mathbf{X}^{-\top} := (\mathbf{X}^{\top})^{-1}$

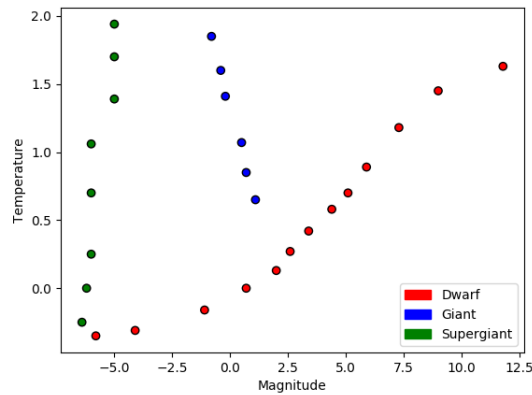
$$\frac{\partial \mathbf{a}^{\top} \mathbf{X}^{-1} \mathbf{b}}{\partial \mathbf{X}} = -\mathbf{X}^{-\top} \mathbf{a} \mathbf{b}^{\top} \mathbf{X}^{-\top}$$

$$\frac{\partial \ln |\det(\mathbf{X})|}{\partial \mathbf{X}} = \mathbf{X}^{-\top}$$



### Problem 3 (Classifying Stars)

In this problem, you will code up three different classifiers to classify different types of stars. The file `data/hr.csv` contains data on magnitude and temperature. The data can be plotted on these two axes:



Please implement the following classifiers in the `SoftmaxRegression` and `KNNClassifier` classes:

- A generative classifier with Gaussian class-conditional densities with a *shared covariance matrix*** across all classes. Feel free to re-use your Problem 2 results.
- Another generative classifier with Gaussian class-conditional densities, but now with a *separate covariance matrix*** learned for each class. (Note: The staff implementation can switch between the two Gaussian generative classifiers with just a few lines of code.)
- A multi-class logistic regression classifier** using the softmax activation function. In your implementation of gradient descent, **make sure to include a bias term and use L2 regularization** with regularization parameter  $\lambda = 0.001$ . Limit the number of iterations of gradient descent to 200,000, and set the learning rate to be  $\eta = 0.001$ .
- Another multi-class logistic regression classifier** with feature map  $\phi(\mathbf{x}) = [\ln(x_1 + 10), x_2^\top]^\top$ , where  $x_1$  and  $x_2$  represent the values for magnitude and temperature, respectively.
- A kNN classifier** in which you classify based on the  $k = 1$  and  $k = 5$  nearest neighbors and the following distance function:

$$\text{dist}(\text{star}_1, \text{star}_2) = (\text{mag}_1 - \text{mag}_2)^2/9 + (\text{temp}_1 - \text{temp}_2)^2$$

where nearest neighbors are those with the smallest distances from a given point.

Note 1: When there are more than two labels, no label may have the majority of neighbors. Use the label that has the most votes among the neighbors as the choice of label.

Note 2: The grid of points for which you are making predictions should be interpreted as our test space. Thus, it is not necessary to make a test point that happens to be on top of a training point ignore itself when selecting neighbors.

After implementing the above classifiers, complete the following exercises:

- Plot the decision boundaries generated by each classifier for the dataset. Include them in your PDF. Identify the similarities and differences among the classifiers. What explains the differences—in particular, which aspects or properties of each model dictate the shape of its decision boundary?
- Consider a star with Magnitude 3 and Temperature -2. To which class does each classifier assign this star? Report the classification probabilities of this star for each model.

Interpret how each model makes its classification decision. What are the pros and cons of each interpretation? What else should we, the modelers, be aware of when making predictions on a test point “far” from our training data? **Your response should no be longer than 5 sentences.**

**Name**

**Collaborators and Resources**

Whom did you work with, and did you use any resources beyond cs181-textbook and your notes?