Homework 3: Bayesian Estimation & Logistic Regression

UC Irvine CS275P: Graphical Models & Statistical Learning

Homework due at 11:59pm on April 29, 2025

Full Name: Viveka Agrawal

UCINetID (e.g., ucinetid@uci.edu): vivekaa@uci.edu

Question 0: (5 points)

Canvas includes detailed guidelines on how homework solutions should be formatted for submission to Gradescope. To receive full credit on this question, be sure that you read and follow these guidelines carefully! In particular, please take care that:

- There are two Gradescope assignments for Homework 3. For the first assignment, submit a PDF containing your answers for questions 1-3. For the second assignment, submit your Python code for questions 2-3, which will be checked by an autograder.
- For the PDF submission, you must fill in this PDF template with your answers. All pages must remain in their original order when uploading your assignment to Gradescope. *Do not rearrange, add, or remove any pages from the provided PDF template.*
- For multiple-choice questions, ensure that you *completely fill in the bubble next to your selected answer*, and provide an explanation or justification to support your answer.
- Enter your final answers (to a precision of 3 decimal places) in the answer boxes, and write your explanations and/or math justifications in the space provided below each question. *Always show the work needed to produce your answers, not just the final result.* Math may be handwritten, but if possible, please type any sentences.
- For questions 2-3, use the provided Python file. After adding your code to the template, submit that file separately to the second Gradescope assignment. In addition to uploading your Python code, be sure you provide answers and explanations in the designated areas of the PDF. *Do not include code for questions 2-3 in the PDF.*
- If you need to include code to show your work for any part of question 1, you may insert an image showing the relevant code in the PDF. If any question asks you to create a plot, insert an image showing the plot in the PDF.
- Check that your answers (especially scanned math) are readable within Gradescope, and that content has not been cut-off by the page margins.

Copyright © Erik B. Sudderth 2025. All rights reserved.
This content is protected and may not be shared, uploaded, or distributed.

Question 1: (25 points)

This question asks you to devise ML and Bayesian MAP estimators for a simple model of an uncalibrated sensor. Let the sensor output, X, be a random variable that ranges over the positive real numbers. We assume that, when tested over a range of environments, its outputs are uniformly distributed on some unknown interval $[0, \theta]$, so that

$$p(x \mid \theta) = \frac{1}{\theta}\mathbb{I}_{0,\theta}(x) = \begin{cases} 1/\theta & \text{if } 0 \le x \le \theta, \\ 0 & \text{otherwise.} \end{cases}$$

Here, $\mathbb{I}_{0,\theta}(x)$ is an *indicator function* that equals 1 when $0 \le x \le \theta$, and 0 otherwise. We denote this distribution by $\text{Unif}(0, \theta)$. To calibrate the sensor, we would like to infer θ.

a) Given N i.i.d. observations $x = (x_1, \ldots, x_N)$, $X_i \sim \text{Unif}(0, \theta)$, what is the likelihood function $p(x \mid \theta)$? What is the maximum likelihood (ML) estimator for θ? Give an explanation for why your estimator is in fact the ML estimator.

Likelihood: $p(x \mid \theta) = \prod_{i=1}^{N} \frac{1}{\theta} \mathbb{I}_{0,\theta}(x_i) = \frac{1}{\theta^N} \prod_{i=1}^{N} \mathbb{I}_{0,\theta}(x_i)$

ML estimate: $\hat{\theta} = \max(x_1, x_2, \ldots x_N)$

Brief explanation for how the ML estimate is derived:

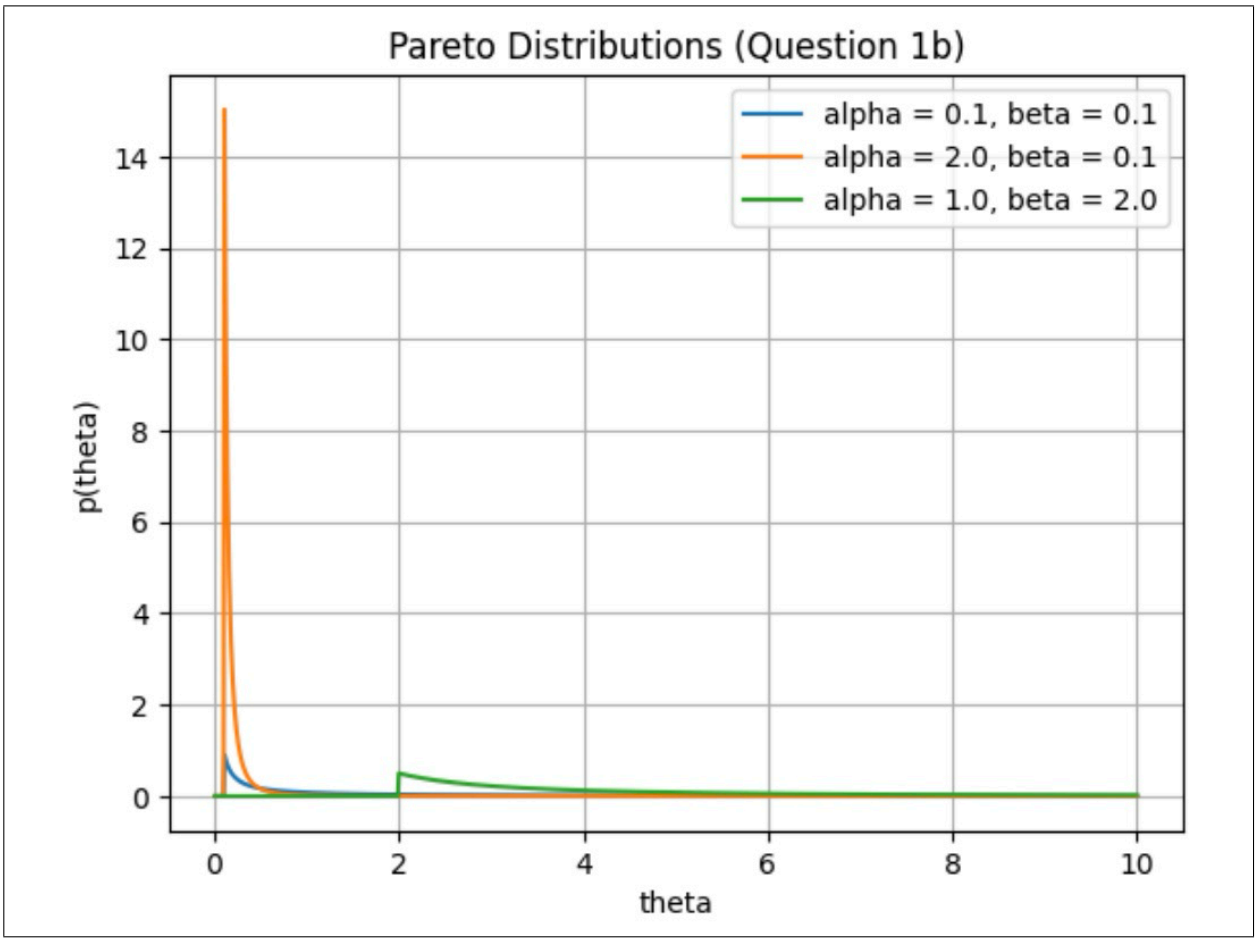
The likelihood function $\frac{1}{\theta^N}$ is strictly decreasing in θ, so to maximize the likelihood, the smallest possible value of θ that makes all observations possible ($\theta \geq \max(x_i)$) is needed.

Copyright © Erik B. Sudderth 2025. All rights reserved.
This content is protected and may not be shared, uploaded, or distributed.

b) Suppose that we place the following prior distribution on θ:

$$p(\theta) = \alpha\beta^{\alpha}\theta^{-\alpha-1}\mathbb{I}_{\beta,\infty}(\theta)\,.$$

This is known as a Pareto distribution. *We denote it by $\theta \sim \text{Pareto}(\alpha, \beta)$. Plot the three prior probability densities corresponding to the following three hyperparameter choices: $(\alpha, \beta) = (0.1, 0.1)$; $(\alpha, \beta) = (2.0, 0.1)$; $(\alpha, \beta) = (1.0, 2.0)$. Briefly describe the influence these parameters have on the properties of the Pareto distribution.*



Brief description of how the hyperparameters α and β impact Pareto distributions:

α controls how quickly the tail of the distribution decays. Larger α indicates faster decay. β controls the minimum value of θ. As β increases, the entire distribution shifts to the right.

Copyright © Erik B. Sudderth 2025. All rights reserved.
This content is protected and may not be shared, uploaded, or distributed.

c) If $\theta \sim \text{Pareto}(\alpha, \beta)$ and we observe N uniformly distributed observations $X_i \sim \text{Unif}(0, \theta)$, derive the posterior distribution $p(\theta \mid x)$. Is this a member of any standard family?

Posterior: $p(x \mid \theta) =$ $\text{Pareto}(\theta \mid N+\alpha, \max(\max(x_i), \beta))$

Yes, this is a member of a standard family

Derivation of posterior distribution, including alignment with standard family:

Bayes Rule: $p(\theta \mid x) = \frac{p(x \mid \theta) \cdot p(\theta)}{p(x)}$

Likelihood: $p(x \mid \theta) = \prod_{i=1}^{N} \frac{1}{\theta} I_{[0,\theta]}(x_i) = \frac{1}{\theta^N} \prod_{i=1}^{N} I_{[0,\theta]}(x_i)$

To be nonzero $(\theta \geq \max(x_1, \ldots, x_N))$: $p(x \mid \theta) = \frac{1}{\theta^N} I_{[\max(x_i), \infty)}(\theta)$

Pareto Prior: $p(\theta) = \alpha \beta^\alpha \theta^{-\alpha-1} I_{[\beta, \infty)}(\theta)$

Multiply likelihood and prior: $p(\theta \mid x) \propto p(x \mid \theta) \cdot p(\theta)$

$p(\theta \mid x) \propto \frac{1}{\theta^N} I_{[\max(x_i), \infty)}(\theta) \cdot \alpha \beta^\alpha \theta^{-\alpha-1} I_{[\beta, \infty)}(\theta)$

Simplify Indicator functions $(\theta \geq \max(\max(x_i), \beta)$

$p(\theta \mid x) \propto \frac{1}{\theta^N} \cdot \alpha \beta^\alpha \theta^{-\alpha-1} \cdot I_{[\max(\max(x_i)\ \beta), \infty)}(\theta)$

$p(\theta \mid x) \propto \alpha \beta^\alpha \theta^{-N} \theta^{-\alpha-1} \cdot I_{[\max(\max(x_i), \beta), \infty)}(\theta)$

$p(\theta \mid x) \propto \alpha \beta^\alpha \theta^{-(N+\alpha+1)} I_{[\max(\max(x_i), \beta), \infty)}(\theta)$

$p(\theta \mid x) \propto \theta^{-(N+\alpha+1)} I_{[\max(\max(x_i), \beta), \infty)}(\theta)$

$p(\theta \mid x) = (N+\alpha) \cdot \frac{[\max(\max(x_i), \beta)]^{N+\alpha}}{\theta^{N+\alpha+1}} \cdot I_{[\max(\max(x_i), \beta), \infty)}(\theta)$

Copyright © Erik B. Sudderth 2025. All rights reserved.
This content is protected and may not be shared, uploaded, or distributed.

d) For the posterior derived in part (c), what is the corresponding MAP estimator of θ? How does this compare to the ML estimator?

MAP estimate: $\hat{\theta} =$ $\max(\max(x_1, \ldots, x_N), \beta)$

Brief discussion of the relationship between the MAP and ML parameter estimates:

The MAP estimate is the same as the ML estimate if the largest observed value is greater than or equal to the beta (β) value. If all data falls below β, however, the prior takes over, and the MAP estimate becomes β. This shows that the prior has more influence when the data is smaller or does not support higher values.

Copyright © Erik B. Sudderth 2025. All rights reserved.
This content is protected and may not be shared, uploaded, or distributed.

e) Recall that the quadratic loss is defined as $L(\theta, \hat{\theta}) = (\theta - \hat{\theta})^2$. For the posterior derived in part (c), what estimator of θ minimizes the posterior expected quadratic loss? Simplify your answer as much as possible.

MMSE estimate: $\hat{\theta} =$ $\boxed{\dfrac{(N+\alpha)\max(\max(x_i), \beta)}{N+\alpha-1}}$

Derivation of the minimum mean squared error (MMSE) parameter estimate:

MMSE estimator minimizes posterior expected quadratic loss:

$$\hat{\theta}_{MMSE} = E[\theta|x] = \int \theta p(\theta|x) d\theta$$

Mean exists when $\alpha' > 1$ and equals:

$E[\theta] = \dfrac{\alpha'\beta'}{\alpha'-1}$ (this is for Pareto(α', β') distribution)

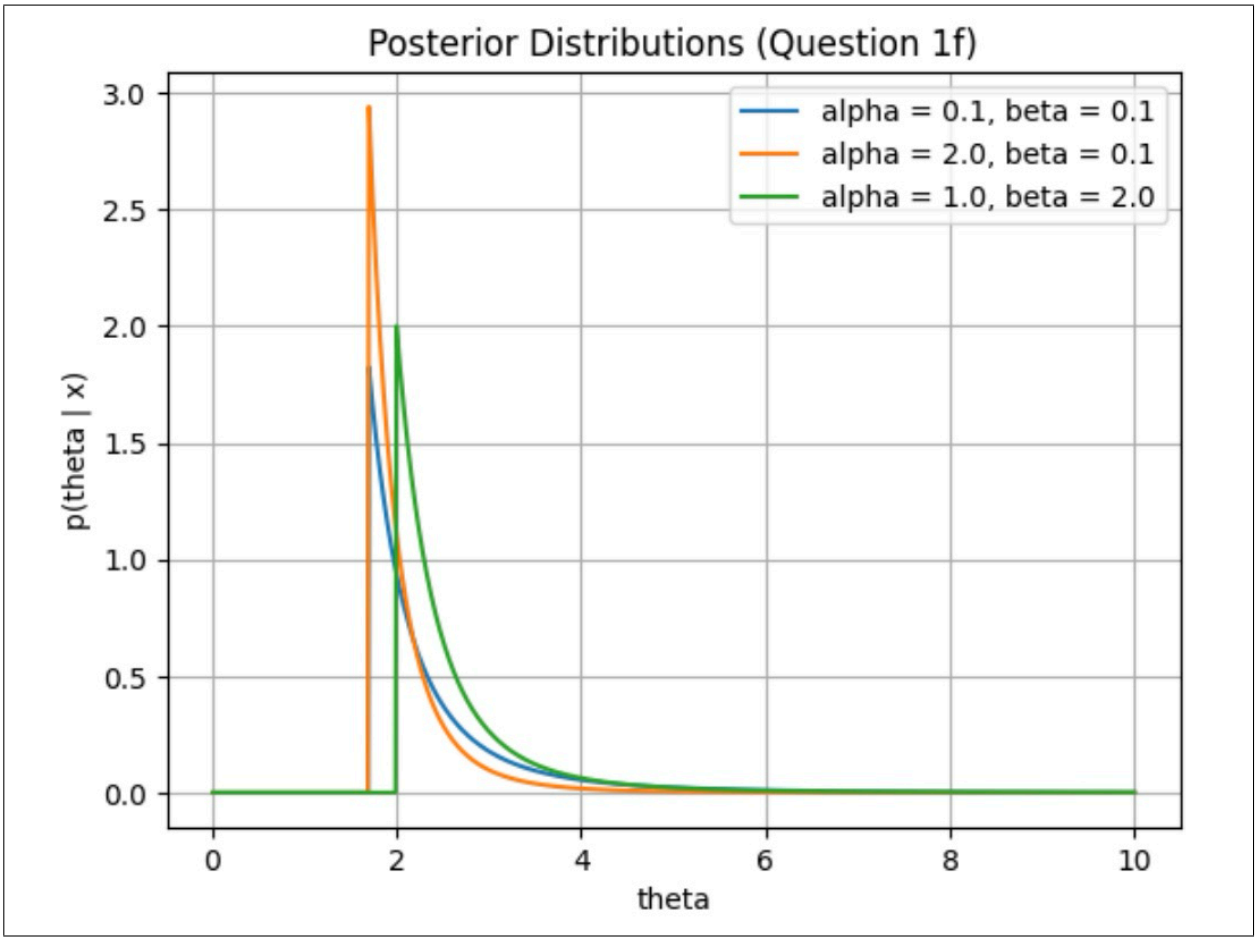
Substitute posterior parameters:

$$\hat{\theta}_{MMSE} = \frac{(N+\alpha)\max(\max(x_i), \beta)}{N+\alpha-1}$$

which is valid when $N+\alpha > 1$

Copyright © Erik B. Sudderth 2025. All rights reserved.
This content is protected and may not be shared, uploaded, or distributed.

f) Suppose that we observe three observations $x = (0.7, 1.3, 1.7)$. Determine the posterior distribution of θ for each of the priors in part (b), and plot the corresponding posterior densities. What is the MAP estimate for each hyperparameter choice? What estimate minimizes the quadratic loss for each hyperparameter choice?



Parameter estimates minimizing expected loss for each choice of prior distribution:

	MAP Estimate	**MMSE Estimate**
$(\alpha, \beta) = (0.1, 0.1) : \hat{\theta} =$	1.7	2.509
$(\alpha, \beta) = (2.0, 0.1) : \hat{\theta} =$	1.7	2.125
$(\alpha, \beta) = (1.0, 2.0) : \hat{\theta} =$	2.0	2.667

Copyright © Erik B. Sudderth 2025. All rights reserved.
This content is protected and may not be shared, uploaded, or distributed.

Question 2-3:

In the next two questions, we use multinomial logistic regression models to predict one of K discrete classes. Let $t_{nk} = 1$ if observation n is an instance of class k, and $t_{nk} = 0$ otherwise. Given input variable x_n, let $\phi(x_n) \in \mathbb{R}^M$ be a fixed feature function. The multinomial logistic regression model then defines the conditional probability of class k as follows:

$$p(t_{nk} = 1 \mid x_n, w) = \frac{\exp(w_k^T \phi(x_n))}{\sum_{\ell=1}^{K} \exp(w_\ell^T \phi(x_n))}.$$

Here $w_k \in \mathbb{R}^M$ are the feature weights associated with class k, and $w \in \mathbb{R}^{KM}$ is a vector concatenating the weights for all classes.

This model is not identifiable: matching translations of the weight vectors w_k for different classes can produce identical probabilities. To avoid this ambiguity and improve robustness, we place a zero-mean, diagonal-covariance Gaussian prior on the weight vector:

$$p(w) = \mathcal{N}(w \mid 0, \alpha^{-1} I_{KM}) \propto \exp\left(-\frac{\alpha}{2} w^T w\right).$$

Here, α is a tunable inverse-variance parameter that controls the degree of regularization. The negative log-posterior distribution, or equivalently the objective whose minimum equals the MAP estimate of the weight vector w, is then

$$\begin{aligned} f(w) &= C - \log p(w) - \sum_{n=1}^{N} \log p(t_n \mid x_n, w) \\ &= \frac{\alpha}{2} w^T w - \sum_{n=1}^{N} \left[\sum_{k=1}^{K} t_{nk} w_k^T \phi(x_n) - \log\left(\sum_{\ell=1}^{K} \exp(w_\ell^T \phi(x_n)) \right) \right], \end{aligned}$$

ignoring constants C which do not depend on w. Taking derivatives with respect to some weight w_{km} and simplifying, we can then show that

$$\frac{\partial f(w)}{\partial w_{km}} = \alpha w_{km} - \sum_{n=1}^{N} (t_{nk} - \mu_{nk}) \phi_m(x_n), \qquad \mu_{nk} = \frac{\exp(w_k^T \phi(x_n))}{\sum_{\ell=1}^{K} \exp(w_\ell^T \phi(x_n))}.$$

These derivatives define the gradients needed for MAP estimation via gradient descent.

Copyright © Erik B. Sudderth 2025. All rights reserved.
This content is protected and may not be shared, uploaded, or distributed.

Question 2: (25 points for PDF answers, 15 points for Python code submission)

We reexamine the gamma ray data set from Homework 1, but instead apply a logistic regression model for the binary classification of star showers. We use the same split of the data into 15,216 training examples and 3,804 test examples. We compare the performance of three different feature mappings of the $D = 10$ raw inputs. In all cases, the first feature should be a constant bias or offset term, $\phi_0(x_n) = 1$. The three feature sets are then:

1. $D + 1$ linear features, the bias feature plus the raw inputs $\phi_m(x_n) = x_{nm}, 1 \leq m \leq D$.
2. $2D + 1$ diagonal quadratic features, including the $D + 1$ linear features from set 1, as well as D quadratic features $\phi_{D+m}(x_n) = x_{nm}^2, 1 \leq m \leq D$.
3. $(D + 1)D/2 + D + 1$ general quadratic features, including the $2D + 1$ features from set 2, as well as products of all pairs of input dimensions $x_{nm}x_{nm'}, m \neq m'$.

Given this data and the objective from part (a), we will use the `minimize` function from the `scipy.optimize` package to find the weight vector w that minimizes the negative log-posterior distribution $f(w)$. You should provide the optimizer with the gradient of $f(w)$, and write your *own* function to compute the objective $f(w)$ and its gradient $\nabla f(w)$. (You may *not* simply use numerical approximations to the gradient.) To get started with `minimize`, see the sample code released with the homework, which includes suggestions for appropriate convergence tolerance parameters.

For all of the following questions, set the regularization constant to $\alpha = 10^{-6}$, and test each of the three feature sets defined above.

Copyright © Erik B. Sudderth 2025. All rights reserved.
This content is protected and may not be shared, uploaded, or distributed.

a) For each of the three feature sets, train the logistic regression model by running gradient-based optimization to convergence. Report the accuracy and negative log-posterior probability of the classifier when training and evaluating on `train`*.*

	Accuracy	**Negative Log-Post**
Linear Features:	0.790	6992.988
Diagonal Quadratic Features:	0.845	5901.021
General Quadratic Features:	0.861	5122.974

b) For each of the three feature sets, and the MAP weight estimates from part (a), evaluate and report the `test` *accuracy and* `test` *negative log-posterior probability.*

	Accuracy	**Negative Log-Post**
Linear Features:	0.797	1706.716
Diagonal Quadratic Features:	0.853	1435.365
General Quadratic Features:	0.871	1275.312

Copyright © Erik B. Sudderth 2025. All rights reserved.
This content is protected and may not be shared, uploaded, or distributed.

c) *Compare the test accuracies of the logistic regression models to the Gaussian naive Bayes classifier from Homework 1. Which method is more accurate? Using concepts from lecture, briefly discuss possible reasons for the observed performance differences.*

The Gaussian naive Bayes classifier had an accuracy of 0.749 for the area under the ROC curve. This indicates that its separation between positive and negative classes is weaker than the logistic regression models (which all had higher classification accuracy and lower negative log-post). What can be learned from this is that logistic regression gives a more reliable model for the dataset compared to the conditional independence assumptions made by the Gaussian naive Bayes classifier.

Copyright © Erik B. Sudderth 2025. All rights reserved.
This content is protected and may not be shared, uploaded, or distributed.

Question 3: (25 points for PDF answers, 5 points for Python code submission)

This question uses synthetic data to compare the properties of logistic regression and linear regression for classification. Each "toy" data item has two continuous features $x \in \mathbb{R}^2$ and is labeled as one of either $K = 2$ or $K = 3$ classes.

Linear regression code (as in Homework 2) can be adapted for classification as follows. For K classes, each response is encoded as a row vector $t_n = [t_{n1}, \ldots, t_{nK}]$, where $t_{nk} = 1$ for an example of class k, and zero otherwise. For N data samples we define the $N \times K$ matrix T as a matrix of 0's and 1's, with each row having a single 1. We fit a linear regression model to each of the columns of T as follows:

$$\hat{T} = \Phi(\Phi^T\Phi)^{-1}\Phi^T T.$$

Here, Φ is the $N \times 3$ model matrix of corresponding to the feature function $\phi(x_n) = [1, x_{n1}, x_{n2}]^T$, i.e. the raw 2D input data augmented by a constant bias feature. The weights corresponding to the least squares prediction above equal

$$\hat{W} = (\Phi^T\Phi)^{-1}\Phi^T T.$$

Here, $\hat{W}$ is a $3 \times K$ matrix where the k^{th} column $\hat{w}_k$ represents the linear regression fit for class k. Finally, we can use this linear regression model to classify a new observation as

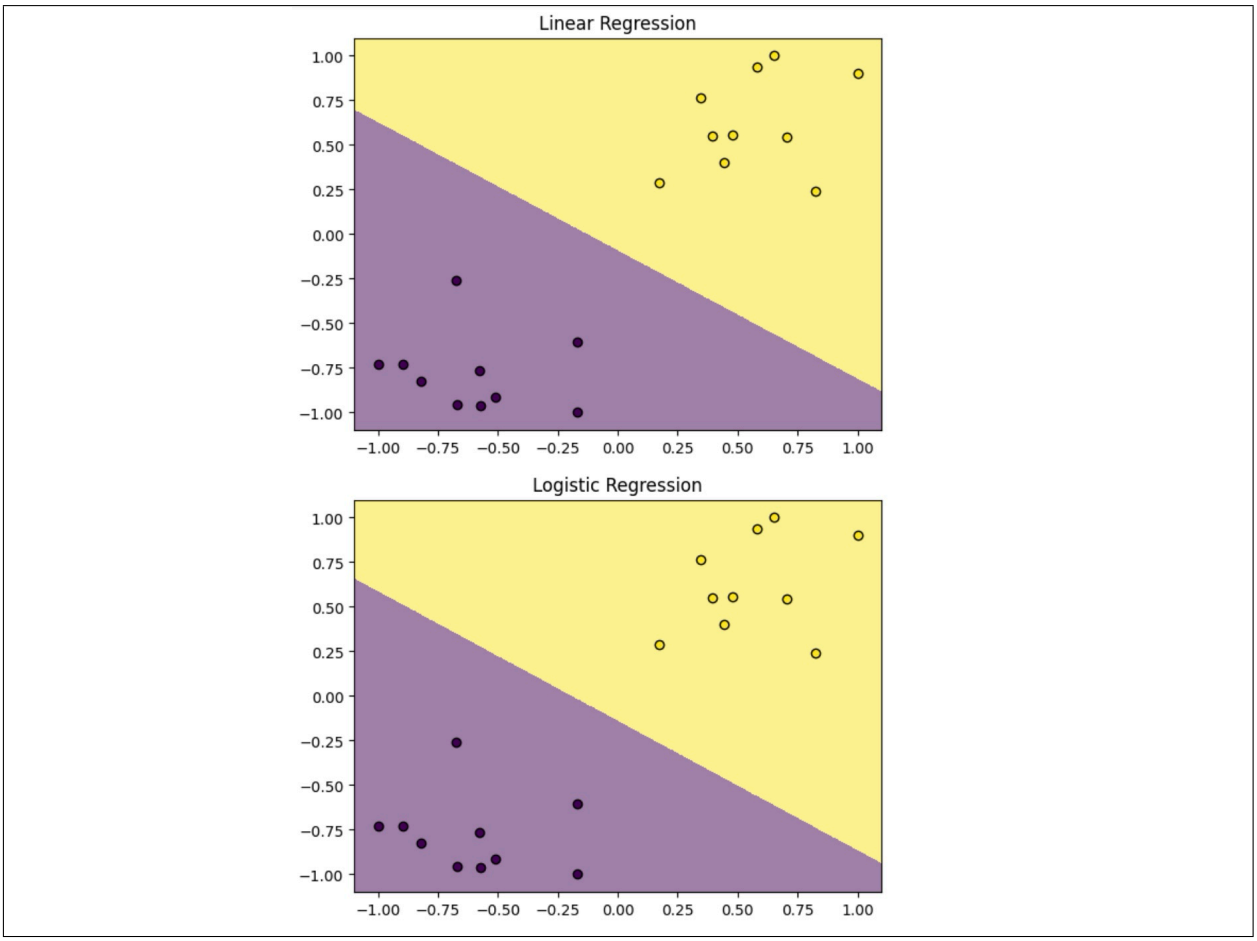
$$y(x) = \arg\max_k \ \phi(x)^T \hat{w}_k.$$

The supplied function `plotter_classifier` can be used to visualize decision boundaries.

We compare the performance of this linear least squares classifier to a multinomial logistic regression classifier, both using the same features. To fit multinomial logistic regression models, use your implementation from Question 2, with a small but positive regularization constant $\alpha = 10^{-6}$ to ensure identifiability.

Copyright © Erik B. Sudderth 2025. All rights reserved.
This content is protected and may not be shared, uploaded, or distributed.

a) *The first dataset contains two classes which lie in well-separated clouds. Implement the least squares classifier described above. Estimate weights for both linear regression and logistic regression classifiers from training data, and plot the learned decision boundaries together with the training points. If implemented correctly, your test accuracy should be 100% for both classifiers. Is this the case?*



Linear regression test accuracy: 1.000

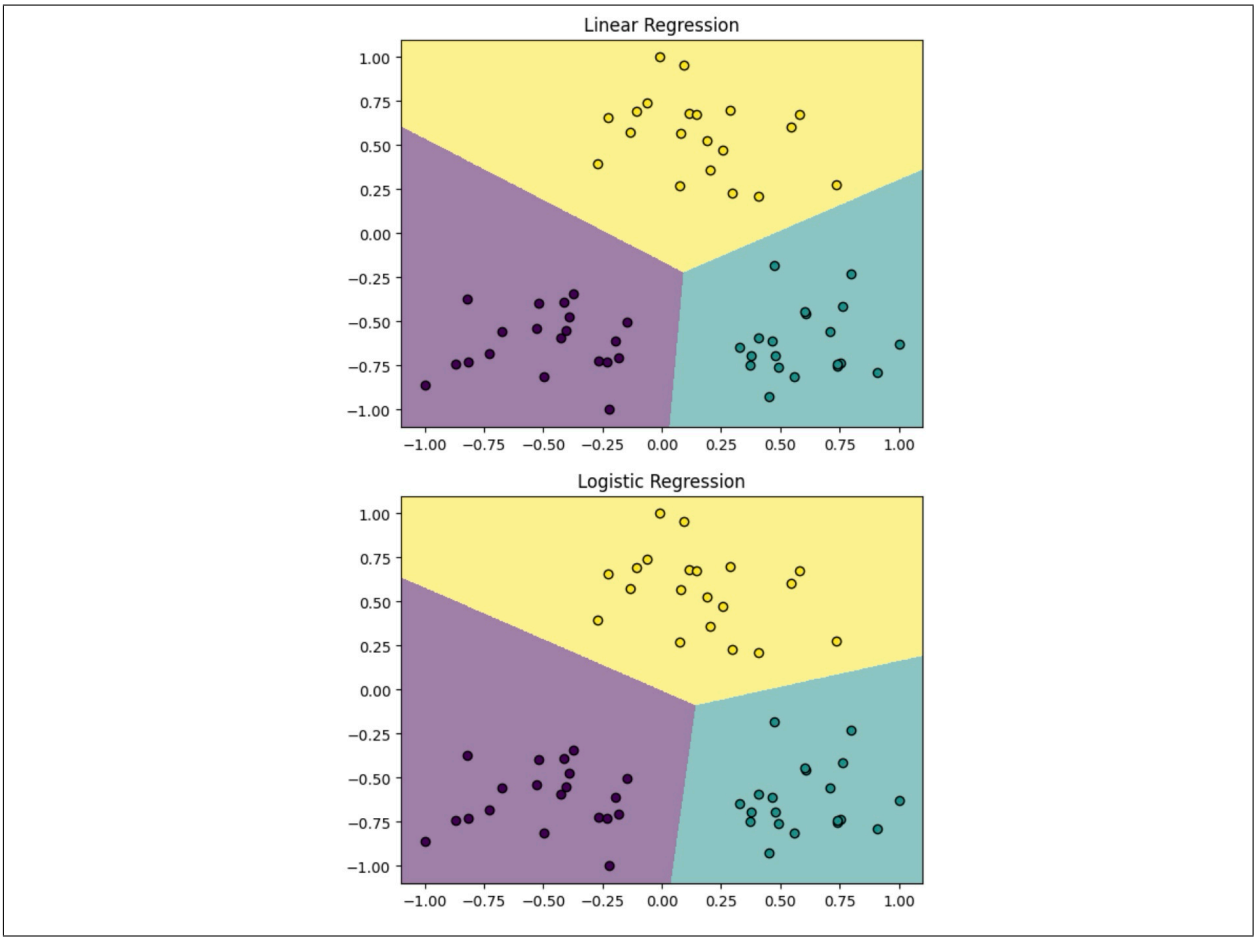
Logistic regression test accuracy: 1.000

Brief discussion of any performance differences:

Both test accuracies are 100%, so it is implemented correctly.

Copyright © Erik B. Sudderth 2025. All rights reserved.
This content is protected and may not be shared, uploaded, or distributed.

b) *The second dataset contains three classes with means arranged in a triangular pattern. Train a least squares classifier as above, as well as a multinomial logistic regression classifier using the same features. Plot the training decision boundaries for both classifiers, and report test accuracy for each. Explain any performance differences.*



Linear regression test accuracy: 1.000

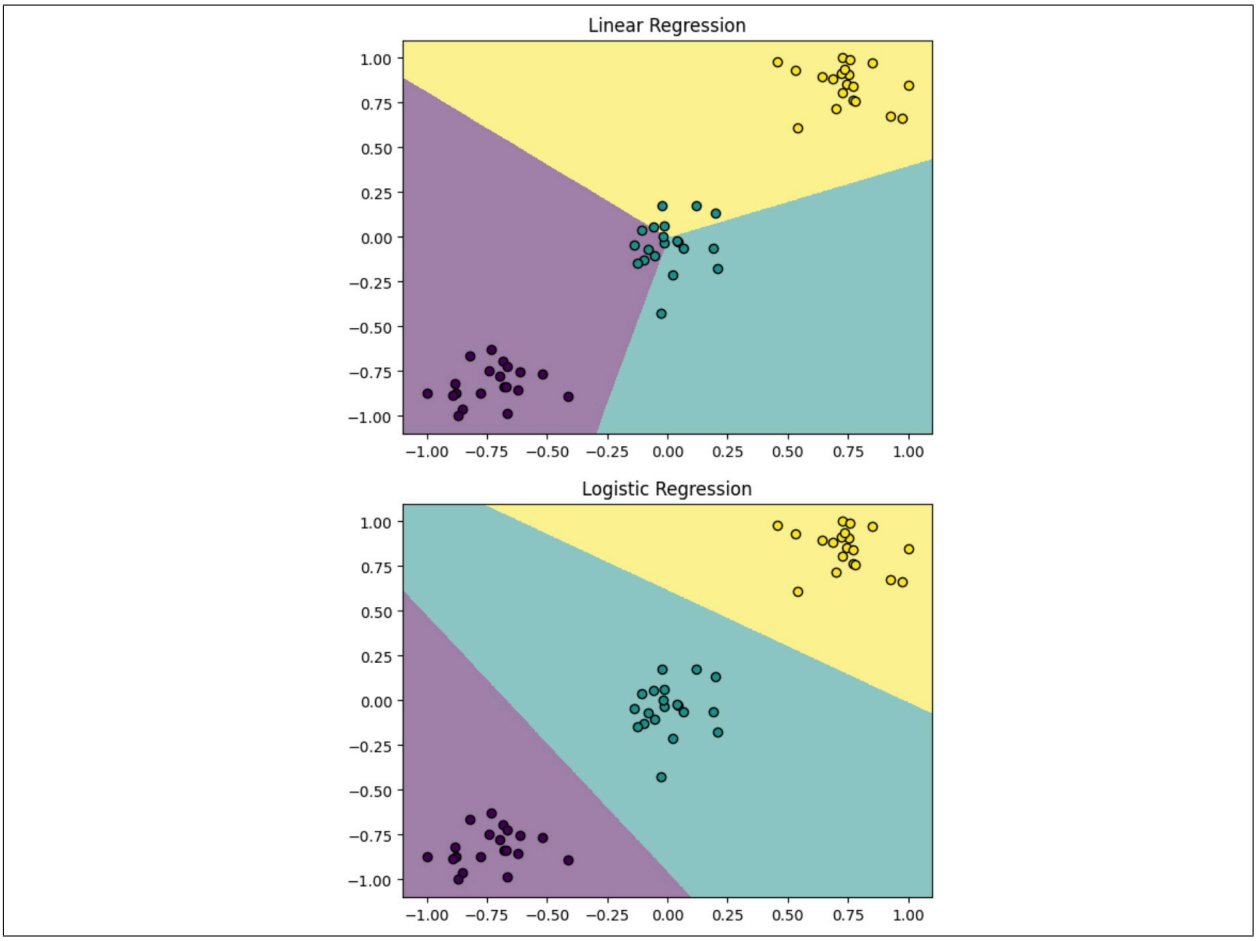
Logistic regression test accuracy: 1.000

Brief discussion of any performance differences:

Both test accuracies are 100%, so it is implemented correctly.

Copyright © Erik B. Sudderth 2025. All rights reserved.
This content is protected and may not be shared, uploaded, or distributed.

c) *The third dataset contains three classes with means arranged in a straight line. Train a least squares classifier as above, as well as a multinomial logistic regression classifier using the same features. Plot the training decision boundaries for both classifiers, and report test accuracy for each. Explain any performance differences.*



Linear regression test accuracy: 0.733

Logistic regression test accuracy: 1.000

Brief discussion of any performance differences:

The logistic regression model has higher accuracy than the linear regression model as it is designed specifically for classification and is able to model class boundaries more effectively.

Copyright © Erik B. Sudderth 2025. All rights reserved.
This content is protected and may not be shared, uploaded, or distributed.