1. A common preprocessing step of many learning algorithms is to normalize each feature to be zero mean and unit variance. Give the formula for the normalized feature $\tilde{x}_j$ as a function of the original feature x_j and the mean μ_j and standard deviation σ_j of that feature.

2. We showed that each step of K-means reduces a particular cost function. What is that cost function?

3. Suppose your classifier achieves poor accuracy on both the training and test sets. Which would be a better choice to try to improve the performance: bagging or boosting? Justify your answer.

4. We showed that AdaBoost can be viewed as minimizing the exponential loss.

   (a) Give the definition of exponential loss.

   $$\mathcal{L}_{\mathrm{E}}(y, t) =$$

   (b) TRUE or FALSE: there is some value ε such that if the sum of the exponential loss on all the training examples is less than ε, then all the training examples are classified correctly. Justify your answer.

5. Recall two linear classification methods we considered:

   Model 1:

   $$\begin{aligned} y &= \mathbf{w}^\top \mathbf{x} + b \\ \mathcal{L}_{\mathrm{SE}}(y, t) &= \tfrac{1}{2}(y - t)^2 \end{aligned}$$

   Model 2:

   $$\begin{aligned} z &= \mathbf{w}^\top \mathbf{x} + b \\ y &= \sigma(z) \\ \mathcal{L}_{\mathrm{SE}}(y, t) &= \tfrac{1}{2}(y - t)^2 \end{aligned}$$

   Here, σ denotes the logistic function, and the targets t take values in $\{0, 1\}$. Briefly explain our reason for preferring Model 2 to Model 1.

6. Consider a discounted Markov decision process (MDP) with discount parameter γ. It has a transition distribution $\mathcal{P}(\cdot \mid s, a)$ and deterministic reward function $r(s, a)$. The agent's policy is a deterministic function $\pi : \mathcal{S} \to \mathcal{A}$.

(a) Give the definition of the state-action value function Q^π for a policy π. It should be given in terms of γ and the immediate rewards $R_t = r(S_t, A_t)$ for $t = 0, \ldots, \infty$. You don't need to justify your answer.

$$Q^\pi(s, a) =$$

(b) Give the Bellman recurrence for Q^π, i.e. the formula expressing $Q^\pi(s, a)$ in terms of an expectation over successor states. You don't need to justify your answer.

$$Q^\pi(s, a) =$$

7. Consider the following NumPy code for computing cross-entropy loss.

```
def cross_entropy_loss(z, t):
    y = 1 / (1 + np.exp(-z))
    return -t * np.log(y) - (1-t) * np.log(1-y)
```

The formulas for y and $\mathcal{L}$ are correct, but there's something wrong with this code.

(a) What is wrong with the code? *Hint: what happens when z is large?*

(b) Provide NumPy code implementing `cross_entropy_loss` which doesn't have this problem. You may want to use the function `np.logaddexp`, which takes two arguments a and b and returns $\log(e^a + e^b)$.

8. We showed that the Support Vector Machine (SVM) can be viewed as minimizing hinge loss:

$$\min_{\mathbf{w},b} \sum_{i=1}^{N} \mathcal{L}_{\text{H}}(y, t) + \frac{1}{2\gamma}\|\mathbf{w}\|_2^2$$

where hinge loss is defined as:

$$\mathcal{L}_{\text{H}}(y, t) = \max(0, 1 - ty)$$

(a) TRUE or FALSE: if the total hinge loss is zero, then every training example must be classified correctly. Justify your answer.

(b) Suppose we replace the hinge loss with the following:

$$\mathcal{L}(y, t) = \max(0, -ty)$$

and otherwise keep the soft-margin SVM objective the same. What would go wrong?

9. The Laplace distribution, parameterized by μ and β, is defined as follows:

$$\text{Laplace}(w; \mu, \beta) = \frac{1}{2\beta} \exp\left(-\frac{|w - \mu|}{\beta}\right).$$

Consider a variant of Bayesian linear regression where we assume the prior over the weights $\mathbf{w}$ consists of an independent zero-centered Laplace distribution for each dimension, with shared parameter β:

$$w_j \sim \text{Laplace}(0, \beta)$$
$$t \mid \mathbf{w} \sim \mathcal{N}(t; \mathbf{w}^\top \boldsymbol{\psi}(\mathbf{x}), \sigma)$$

For reference, the Gaussian PDF is:

$$\mathcal{N}(x; \mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x - \mu)^2}{2\sigma^2}\right).$$

(a) Suppose you have a labeled training set $\{(\mathbf{x}^{(i)}, t^{(i)})\}_{i=1}^N$. Give the cost function you would minimize to find the MAP estimate of $\mathbf{w}$. (It should be expressed in terms of mathematical operations.)

(b) Based on your answer to part (a), how might the MAP solution for a Laplace prior differ from the MAP solution if you use a Gaussian prior?

10. Consider one layer of a multilayer perceptron (MLP), whose computations are defined as follows:

$$z_i = \sum_j w_{ij} h_j + b_i$$
$$y_i = \phi(z_i),$$

where ϕ is a nonlinear activation function, h_j denotes the input to this layer (i.e. the previous layer's hidden units), and y_i denotes the output of this layer.

Give the backprop rules for $\overline{z_i}$, $\overline{h_j}$ and $\overline{w_{ij}}$ in terms of the error signal $\overline{y_i}$. You can use ϕ' to denote the derivative of ϕ.

$$\overline{z_i} =$$

$$\overline{h_j} =$$

$$\overline{w_{ij}} =$$

11. Recall that the beta distribution is defined by

$$\text{Beta}(\theta; a, b) = \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} \theta^{a-1}(1-\theta)^{b-1},$$

where Γ is the gamma function. Give values of a and b such that the distribution is highly concentrated around $\theta = 0.75$.

Hint: If you've forgotten the shape of the distribution, you can find the mode as a function of a and b by differentiating the log density.

$$a =$$

$$b =$$

12. Recall that the optimal PCA subspace can be determined from the eigendecomposition of the empirical covariance matrix $\boldsymbol{\Sigma} = \text{Cov}(\mathbf{x})$. Also recall that the eigendecomposition can be expressed in terms of the following spectral decomposition of $\boldsymbol{\Sigma}$:

$$\boldsymbol{\Sigma} = \mathbf{Q}\boldsymbol{\Lambda}\mathbf{Q}^\top,$$

where $\mathbf{Q}$ is an orthogonal matrix and $\boldsymbol{\Lambda}$ is a digonal matrix. Assume the eigenalues are sorted from largest to smallest. You may assume all of the eigenvalues are distinct.

(a) If you've already computed the eigendecomposition (i.e. $\mathbf{Q}$ and $\boldsymbol{\Lambda}$), how do you obtain the orthogonal basis $\mathbf{U}$ for the optimal PCA subspace?

(b) The PCA code vector for a data point $\mathbf{x}$ is given by $\mathbf{z} = \mathbf{U}^\top(\mathbf{x} - \boldsymbol{\mu})$. Show that the dimensions of $\mathbf{z}$ are uncorrelated. (Hint: start by finding a formula for $\text{Cov}(\mathbf{z})$.)

13. In this question, you will derive the E-M update rules for a univariate Gaussian mixture model (GMM) with two mixture components. Unlike the GMMs we covered in the course, the mean μ will be shared between the two mixture components, but each component will have its own standard deviation σ_k. The mixture component is indicated by a latent variable $z \in \{0, 1\}$. The model is defined as follows:

$$\begin{aligned} z &\sim \text{Bernoulli}(\theta) \\ x \,|\, z = k &\sim \mathcal{N}(\mu, \sigma_k) \qquad \text{for } k \in \{0, 1\} \end{aligned}$$

The parameters of the model are θ, μ, σ_0, and σ_1. Suppose we observe a dataset $\{x^{(i)}\}_{i=1}^N$.

For reference, the PDF of the Gaussian distribution is as follows:

$$\mathcal{N}(x; \mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

(a) Write the complete data log-likelihood for this model.

(b) In the E-step, for each data point $x^{(i)}$, we need to compute the posterior probability $r^{(i)} = \Pr(z^{(i)} = 1 \,|\, x^{(i)})$. Give the formula for $r^{(i)}$. In your formula, you may use $\mathcal{N}(x^{(i)}; \mu, \sigma)$ to denote the Gaussian PDF, rather than writing it out explicitly.

$$r^{(i)} =$$

(c) Write out the expected complete data loglikelihood i.e. the objective that is to be maximized in the M-step. It should be expressed in terms of the $r^{(i)}$ and the Gaussian PDF $\mathcal{N}(x^{(i)}; \mu, \sigma)$.

(d) Derive the M-step update rule for μ by maximizing this objective with respect to μ. (In this step, the σ_k are fixed to their previous values.)

(e) Derive the M-step update rule for σ_1 by maximizing the objective with respect to σ_1. (In this step, assume μ is fixed to its previous value.)