CPSC 340 Assignment 5 (due Friday March 23 at 9:00pm)

Instructions

Rubric: {mechanics:5}

The above points are allocated for following the general homework instructions. In addition to the usual instructions: if you're embedding your answers in a document that also contains the questions, your answers should be in a colour that clearly stands out, such as green or red. This should hopefully make it much easier for the grader to find your answers. To make something green, you can use the LaTeX macro `\gre{my text}`.

1 MAP Estimation

Rubric: {reasoning:10}

In class, we considered MAP estimation in a regression model where we assumed that:

- The likelihood $p(y_i|x_i, w)$ is a normal distribution with a mean of $w^T x_i$ and a variance of 1.
- The prior for each variable j, $p(w_j)$, is a normal distribution with a mean of zero and a variance of λ^{-1}.

Under these assumptions, we showed that this leads to the standard L2-regularized least squares objective function:

$$f(w) = \frac{1}{2}\|Xw - y\|^2 + \frac{\lambda}{2}\|w\|^2.$$

For each of the alternate assumptions below, show how the loss function would change (simplifying as much as possible):

1. We use a zero-mean Laplace prior for each variable with a scale parameter of λ^{-1}, so that

$$p(w_j) = \frac{\lambda}{2}\exp(-\lambda|w_j|).$$

2. We use a Laplace likelihood with a mean of $w^T x_i$ and a scale of 1, so that

$$p(y_i|x_i, w) = \frac{1}{2}\exp(-|w^T x_i - y_i|).$$

3. We use a Gaussian likelihood where each datapoint has variance σ^2 instead of 1,

$$p(y_i|x_i, w) = \frac{1}{\sqrt{2\sigma^2\pi}}\exp\left(-\frac{(w^T x_i - y_i)^2}{2\sigma^2}\right).$$

4. We use a Gaussian likelihood where each datapoint has its own variance σ_i^2,

$$p(y_i|x_i, w) = \frac{1}{\sqrt{2\sigma_i^2\pi}}\exp\left(-\frac{(w^T x_i - y_i)^2}{2\sigma_i^2}\right).$$

Answer:

1. This changes the regularizer to $\lambda\|w\|_1$.
2. This changes the data-fitting term to $\|Xw - y\|_1$.
3. This changes the data-fitting term to $\frac{1}{2\sigma^2}\|Xw - y\|^2$.
4. This changes the data-fitting term to $\frac{1}{2}(Xw - y)^T Z(Xw - y)$ where the diagonal matrix Z has $1/\sigma_i^2$ along its diagonals.

2 Principal Component Analysis

2.1 PCA by Hand

Rubric: {reasoning:3}

Consider the following dataset, containing 5 examples with 2 features each:

x_1	x_2
-2	-1
-1	0
0	1
1	2
2	3

Recall that with PCA we usually assume that the PCs are normalized ($\|w\| = 1$), we need to center the data before we apply PCA, and that the direction of the first PC is the one that minimizes the orthogonal distance to all data points.

1. What is the first principal component?
2. What is the (L2-norm) reconstruction error of the point (3,3)? (Show your work.)
3. What is the (L2-norm) reconstruction error of the point (3,4)? (Show your work.)

Answer: We do not need to center the first variable but we need to center the second one. The mean of the second variable is 1 so the centered data looks like this:

x_1	x_2
-2	-2
-1	-1
0	0
1	1
2	2

1. We see that all the centered variables lie along the $x_2 = x_1$. The direction of this line is $(1, 1)$, but since we normalize the principal components to have a norm of one the first PC is $w_1 = (1/\sqrt{2}, 1/\sqrt{2})$ so that $\sqrt{w_1} = \sqrt{(1/\sqrt{2})^2 + (1/\sqrt{2})^2} = \sqrt{1/2 + 1/2} = 1$.
2. To get the low-dimensional representation, we first subtract the means and then multiply by w_c,

$$z = (3 - 0)/\sqrt{2} + (3 - 1)/\sqrt{2} = 5/\sqrt{2}.$$

To go back to the original space, we multiply this by w_c and add back the means:

$$\hat{x} = \frac{5}{\sqrt{2}}(1/\sqrt{2}, 1/\sqrt{2}) + (0, 1) = (5/2, 7/2) = (2.5, 3.5),$$

so the reconstruction error is

$$\sqrt{(2.5-3)^2+(3-3.5)^2} = \sqrt{1/4+1/4} = 1/\sqrt{2}.$$

3.

$$z = (3-0)/\sqrt{2} + (4-1)/\sqrt{2} = 6/\sqrt{2}.$$

$$\hat{x} = \frac{6}{\sqrt{2}}(1/\sqrt{2}, 1/\sqrt{2}) + (0,1) = (6/2, 8/2) = (3,4),$$

which is the same as the original point so the reconstruction error is 0.

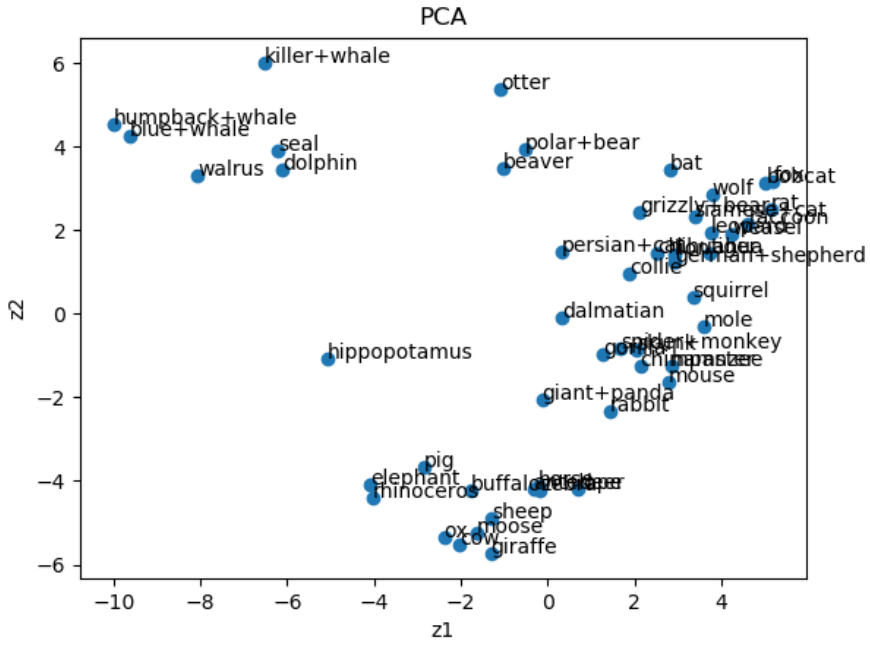
2.2 Data Visualization

Rubric: {reasoning:2}

If you run `python main.py -q 2`, it will load the animals dataset and create a scatterplot based on two randomly selected features. We label some random points, but because of the binary features the scatterplot shows us almost nothing about the data.

The class *pca.PCA* applies the classic PCA method (orthogonal bases via SVD) for a given k. Use this class so that the scatterplot uses the latent features z_i from the PCA model. Make a scatterplot of the two columns in Z, and label a bunch of the points in the scatterplot. Hand in your code and the scatterplot.

Answer: The scatterplot should look roughly like this:



The points that are labeled could vary. (Roughly it looks like as we move from the upper-left to the lower-right we go from terrestrial to aquatic animals, and as we move from the lower-left to the upper-right the animals are getting bigger. But they are definitely exceptions to these rules that the higher-order principal components might capture, and the 'crowding' effect places some very dissimilar animals next to each other.)

2.3 Data Compression

Rubric: {reasoning:2}

1. How much of the variance is explained by our 2-dimensional representation from the previous question?
2. How many PCs are required to explain 50% of the variance in the data?

Answer: Computing the amount of variance explained by the PCs can be done using the last line of the code in the previous answer.

1. With $k = 2$ only about 30.19% of the variance is explained, so our 2D visualization is missing a lot of information.
2. We need $k \geq 5$ to explain 50% of the variance.

3 PCA Generalizations

3.1 Robust PCA

Rubric: {code:10}

If you run `python main -q 3.1` the code will load a dataset X where each row contains the pixels from a single frame of a video of a highway. The demo applies PCA to this dataset and then uses this to reconstruct the original image. It then shows the following 3 images for each frame (pausing and waiting for input between each frame):

1. The original frame.
2. The reconstruction based on PCA.
3. A binary image showing locations where the reconstruction error is non-trivial.

Recently, latent-factor models have been proposed as a strategy for "background subtraction": trying to separate objects from their background. In this case, the background is the highway and the objects are the cars on the highway. In this demo, we see that PCA does an ok job of identifying the cars on the highway in that it does tend to identify the locations of cars. However, the results aren't great as it identifies quite a few irrelevant parts of the image as objects.

Robust PCA is a variation on PCA where we replace the L2-norm with the L1-norm,

$$f(Z, W) = \sum_{i=1}^{n} \sum_{j=1}^{d} |w^{jT} z_i - x_{ij}|,$$

and it has recently been proposed as a more effective model for background subtraction. Complete the class *pca.RobustPCA*, that uses a smooth approximation to the absolute value to implement robust PCA. Comment on the quality of the results.

Hint: most of the work has been done for you in the class *pca.AlternativePCA*. This work implements an alternating minimization approach to minimizing the (L2) PCA objective (without enforcing orthogonality). This gradient-based approach to PCA can be modified to use a smooth approximation of the L1-norm. Note that the log-sum-exp approximation to the absolute value may be hard to get working due to numerical issues, and a numerically-nicer approach is to use the "multi-quadric" approximation:

$$|\alpha| \approx \sqrt{\alpha^2 + \epsilon},$$

where ϵ controls the accuracy of the approximation (a typical value of ϵ is 0.0001).

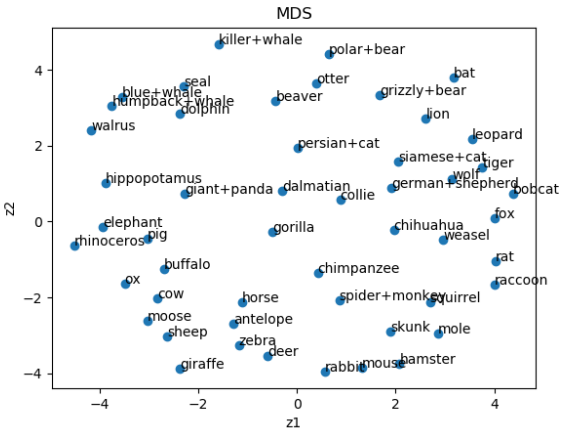
Answer: The code just needs to change all cases where the function is computed and where we compute the derivative with respect to the residual. See *pca.RobustPCA*.

4 Multi-Dimensional Scaling

If you run `python main.py -q 4`, the code will load the animals dataset and then apply gradient descent to minimize the following multi-dimensional scaling (MDS) objective (starting from the PCA solution):

$$f(Z) = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=i+1}^{n}(\|z_i - z_j\| - \|x_i - x_j\|)^2. \tag{1}$$

The result of applying MDS is shown below.



Although this visualization isn't perfect (with "gorilla" being placed close to the dogs and "otter" being placed close to two types of bears), this visualization does organize the animals in a mostly-logical way.

4.1 ISOMAP

Rubric: {code:10}

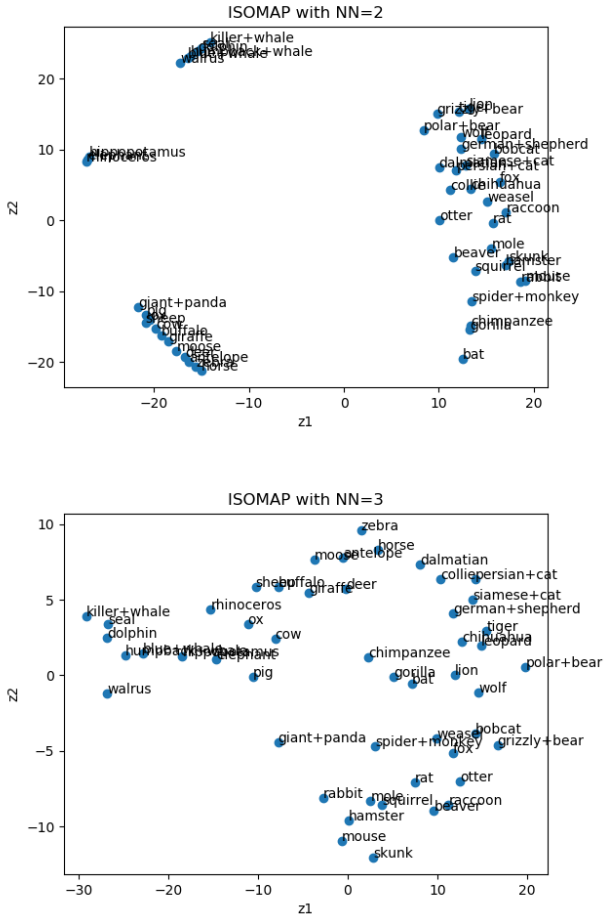
Euclidean distances between very different animals are unlikely to be particularly meaningful. However, since related animals tend to share similar traits we might expect the animals to live on a low-dimensional manifold. This suggests that ISOMAP may give a better visualization. Fill in the class *ISOMAP* so that it computes the approximate geodesic distance (shortest path through a graph where the edges are only between nodes that are k-nearest neighbours) between each pair of points, and then fits a standard MDS model (1) using gradient descent. Plot the results using 2 and using 3-nearest neighbours.

Note: when we say 2 nearest neighbours, we mean the two closest neigbours excluding the point itself. This is the opposite convention from what we used in KNN at the start of the course.

The function *utils.dijskstra* can be used to compute the shortest (weighted) distance between two points in a weighted graph. This function requires an $n \times n$ matrix giving the weights on each edge (use 0 as the weight for absent edges). Note that ISOMAP uses an undirected graph, while the k-nearest neighbour graph might be asymmetric. One of the usual heuristics to turn this into a undirected graph is to include an edge i to j if

i is a KNN of j or if j is a KNN of i. (Another possibility is to include an edge only if i and j are mutually KNNs.)

Answer: We can implement ISOMAP by changing the distance function. See *manifold.ISOMAP*. This assumes that we add an edge if i is a KNN of j or vice versa, but other distance functions in the spirit of constructing a KNN graph are also acceptable. The result of using this weighting, for 2 and 3 neighbours respectively, is:



4.2 Reflection

Rubric: {reasoning:2}

Briefly comment on PCA vs. MDS vs. ISOMAP for dimensionality reduction on this particular data set. In your opinion, which method did the best job and why?

Answer: This is a matter of opinion and the goal of the question is to make sure you at least looked at the plots. I like ISOMAP with 2 neighbours because it clearly separates some sensible groups. One question though is that giant panda is away from the rest of the bears.

5 Very-Short Answer Questions

Rubric: {reasoning:10}

1. Why is the kernel trick often better than explicitly transforming your features into a new space?

   Answer: The number of new features may be huge, which would make everything computationally infeasible. The kernel version is fast.

2. Why is the kernel trick more popular for SVMs than with logistic regression?

   Answer: The speed of making predictions only depends on the number of support vectors instead of n, and this can be much smaller.

3. What is the key advantage of stochastic gradient methods over gradient descent methods?

   Answer: Low iteration cost (independent of n).

4. Does stochastic gradient descent with a fixed α converge to the minimum of a convex function in general?

   Answer: No, because of the variance in the gradients, the iterates will keep oscillating around the minimum, but never attain it.

5. What is the difference between multi-label and multi-class classification?

   Answer: In multi-class classification there is one "true" label, whereas in multi-label several of the labels (or none) can be correct.

6. What is the difference between MLE and MAP?

   Answer: In MAP we maximize the posterior, which includes a prior (corresponding to a regularizer).

7. Linear regression with one feature and PCA with 2 features (and $k = 1$) both find a line in a two-dimensional space. Do they find the same line? Briefly justify your answer.

   Answer: No, in linear regression we only minimize the squared distances in one dimension (to the target y) while in PCA we care about the squared distances in both dimensions (another minor difference is that PCA assumes centered data).

8. Are the vectors minimizing the PCA objective unique? Briefly justify your answer

   Answer: No, they are non-unique for a lot reasons: the ordering, the scale, and rotations. Even when accounting for these, you can flip the signs.

9. Name two methods for promoting sparse solutions in a linear regression model that result in convex problems.

   Answer: Non-negativity constraint and L1-regularization.

10. Can we use the normal equations to solve non-negative least squares problems?

    Answer: No, they don't respect the non-negativity constraints.