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}`.

Also, **READ THIS**: Like in a2, you'll need to grab the data from the course website. FYI: this happens because I'm using the GitHub API in a fairly silly way, which limits individual files to 1 MB each.

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

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

The MAP estimation with this prior is

$$f(w) = -\sum_{j=1}^{d} log(p(w_j)) = -\sum_{j=1}^{d} log(\frac{\lambda}{2}exp(-\lambda|w_j|)) = \sum_{j=1}^{d} log(\frac{\lambda}{2} - \lambda|w_j|) = (const) + \lambda||w||_1$$

Therefore, the regularizer changes to $\lambda||w||_1$.

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

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

The MAP estimation with this likelihood is

$$f(w) = -\sum_{i=1}^{n} log(p(y_i|x_i, w)) = -\sum_{i=1}^{n} log(\frac{1}{2} exp(-|w^T x_i - y_i|)) = (const) + \sum_{i=1}^{n} |w^T x_i - yi| = (const) + ||Xw - y||_1$$

Therefore, the data fitting term changes to $||Xw - y||_1$.

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

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

The MAP estimation with this likelihood is

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

$$= (const) + \sum_{i=1}^{n} \frac{(w^T x_i - y_i)^2}{2\sigma^2} = (const) + \frac{||Xw - y||^2}{2\sigma^2}$$

Therefore, the data fitting term changes to $\frac{||Xw-y||^2}{2\sigma^2}$.

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

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

The MAP estimation with this likelihood is

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

$$= (const) + \frac{1}{2}\sum_{i=1}^{n} \frac{(w^T x_i - y_i)^2}{\sigma_i^2} = (const) + \frac{1}{2}(Xw - y)^T Z(Xw - y)$$

where Z is a diagonal matrix with $1/\sigma_i^2$ along the diagonals.

Therefore, the data fitting term changes to $\frac{1}{2}(Xw - y)^T Z(Xw - y)$.

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?

   Firstly, we need to center the data. x_1 is already centered, which has a mean of 0.
   For x_2, mean is 1. After the centering, the second column becomes $-2, -1, 0, 1, 2$, which is same as x_1.
   Therefore normalized first principle component W_1 would be $(\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}})$.

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

   Firstly, we need to center the new point (3,3). Then it becomes (3-0, 3-1)=(3,2).

   Then $z = 3/\sqrt{2} + 2/\sqrt{2} = 5/\sqrt{2}$.
   $\hat{x} = zW_1 + \mu = \frac{5}{\sqrt{2}}(\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}) + (0, 1) = (\frac{5}{2}, \frac{7}{2})$.

   Then the reconstruction error is $\sqrt{(3 - 2.5)^2 + (3 - 3.5)^2} = \frac{1}{\sqrt{2}}$.

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

   Same process as above.
   Centering: (3,4)-(0,1) = (3,3). $z = 3/\sqrt{2} + 3/\sqrt{2} = 6/\sqrt{2}$.
   $\hat{x} = zW_1 + \mu = \frac{6}{\sqrt{2}}(\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}) + (0, 1) = (3, 4)$.

   The reconstruction error is zero.

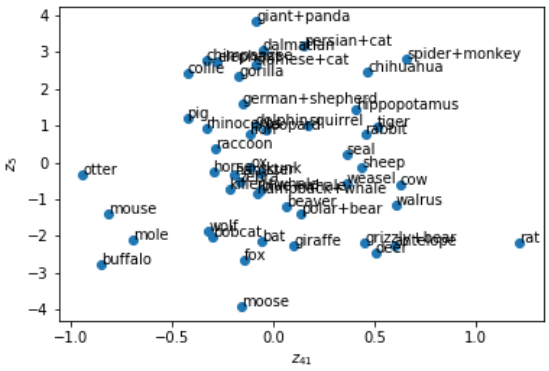
2.2 Data Visualization

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.

See code in code/main.py (question '2.2').
If we set $k = 20$, the scatterplot with two randomly chosen features from k features is as follows:



2.3 Data Compression

Rubric: {reasoning:2}

1. How much of the variance is explained by our 2-dimensional representation from the previous question?
   See code in code/main.py (question '2.3').

   85.95% variance is explained by our 2-dimensional representation from the previous question ($k = 20$).

   If we set k from 1 to d, the scatterplot with two randomly chosen features from k features is as follows:

```
The explained variance for k = 0 is 0.0
The explained variance for k = 1 is 0.172068784206
The explained variance for k = 2 is 0.301938151559
The explained variance for k = 3 is 0.38779248564
The explained variance for k = 4 is 0.448603643724
The explained variance for k = 5 is 0.505901629487
The explained variance for k = 6 is 0.547276982595
The explained variance for k = 7 is 0.585398960028
The explained variance for k = 8 is 0.617603543313
The explained variance for k = 9 is 0.648976968964
The explained variance for k = 10 is 0.677631818695
The explained variance for k = 11 is 0.70487886459
The explained variance for k = 12 is 0.728123614956
The explained variance for k = 13 is 0.749463372961
The explained variance for k = 14 is 0.768656384913
The explained variance for k = 15 is 0.786414723091
The explained variance for k = 16 is 0.803268605954
The explained variance for k = 17 is 0.818977714812
The explained variance for k = 18 is 0.833276601932
The explained variance for k = 19 is 0.84694284698
The explained variance for k = 20 is 0.859548494878
The explained variance for k = 21 is 0.871694502552
The explained variance for k = 22 is 0.883343134899
The explained variance for k = 23 is 0.893873164222
The explained variance for k = 24 is 0.903883779453
The explained variance for k = 25 is 0.913079413282
The explained variance for k = 26 is 0.921370344259
The explained variance for k = 27 is 0.929162536238
The explained variance for k = 28 is 0.936565639624
The explained variance for k = 29 is 0.943518775501
```

```
The explained variance for k = 30 is 0.949451545973
The explained variance for k = 31 is 0.954545631958
The explained variance for k = 32 is 0.959381032
The explained variance for k = 33 is 0.963935136177
The explained variance for k = 34 is 0.968272123225
The explained variance for k = 35 is 0.972363946369
The explained variance for k = 36 is 0.975990321736
The explained variance for k = 37 is 0.979028145496
The explained variance for k = 38 is 0.981788590046
The explained variance for k = 39 is 0.984496653467
The explained variance for k = 40 is 0.986784747391
The explained variance for k = 41 is 0.988994155846
The explained variance for k = 42 is 0.991084552709
The explained variance for k = 43 is 0.993001145212
The explained variance for k = 44 is 0.994780254208
The explained variance for k = 45 is 0.996281566712
The explained variance for k = 46 is 0.997675301124
The explained variance for k = 47 is 0.998589192389
The explained variance for k = 48 is 0.99936352408
The explained variance for k = 49 is 1.0
The explained variance for k = 50 is 1.0
The explained variance for k = 51 is 1.0
The explained variance for k = 52 is 1.0
The explained variance for k = 53 is 1.0
The explained variance for k = 54 is 1.0
The explained variance for k = 55 is 1.0
The explained variance for k = 56 is 1.0
The explained variance for k = 57 is 1.0
The explained variance for k = 58 is 1.0
The explained variance for k = 59 is 1.0
The explained variance for k = 60 is 1.0
The explained variance for k = 61 is 1.0
```

```
The explained variance for k = 62 is 1.0
The explained variance for k = 63 is 1.0
The explained variance for k = 64 is 1.0
The explained variance for k = 65 is 1.0
The explained variance for k = 66 is 1.0
The explained variance for k = 67 is 1.0
The explained variance for k = 68 is 1.0
The explained variance for k = 69 is 1.0
The explained variance for k = 70 is 1.0
The explained variance for k = 71 is 1.0
The explained variance for k = 72 is 1.0
The explained variance for k = 73 is 1.0
The explained variance for k = 74 is 1.0
The explained variance for k = 75 is 1.0
The explained variance for k = 76 is 1.0
The explained variance for k = 77 is 1.0
The explained variance for k = 78 is 1.0
The explained variance for k = 79 is 1.0
The explained variance for k = 80 is 1.0
The explained variance for k = 81 is 1.0
The explained variance for k = 82 is 1.0
The explained variance for k = 83 is 1.0
The explained variance for k = 84 is 1.0
```

2. How many PCs are required to explain 50% of the variance in the data?

   When $k >= 5$, at least 50% of the variance is explained.

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,

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

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:

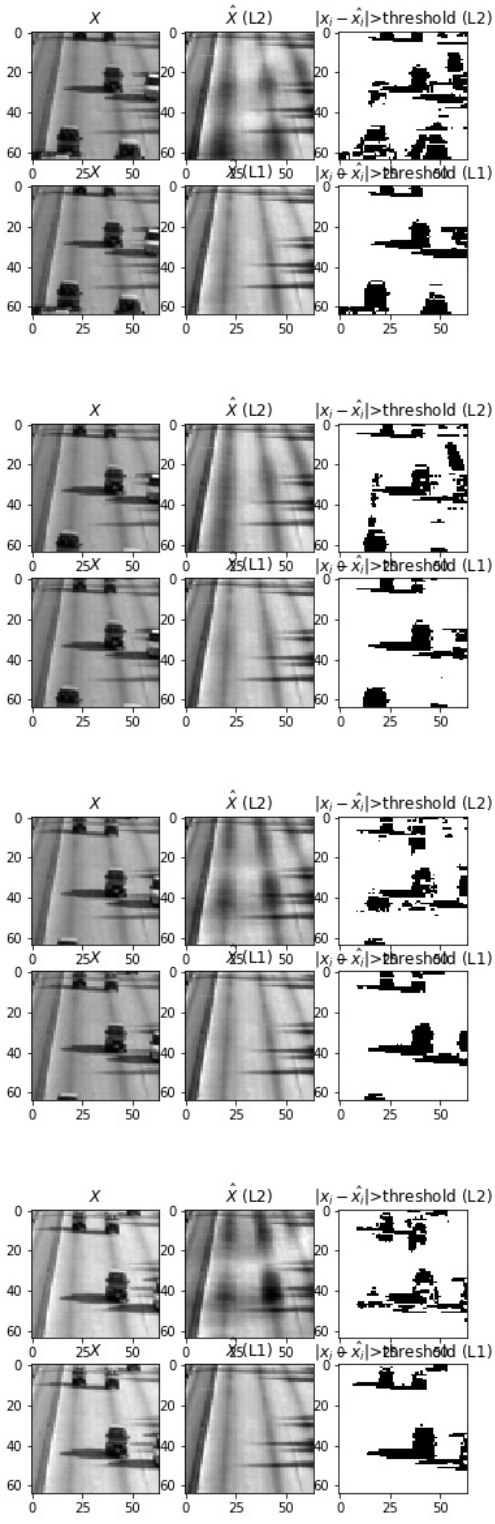
$$|\alpha| \approx \sqrt{\alpha^2 + \epsilon},$$

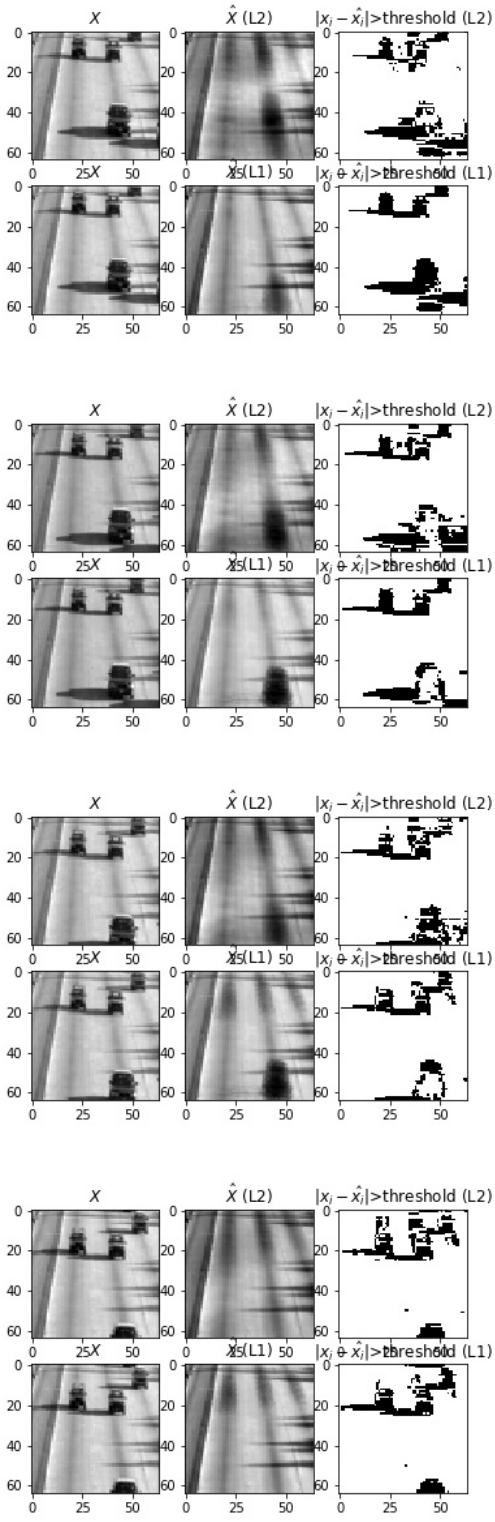
where ϵ controls the accuracy of the approximation (a typical value of ϵ is 0.0001).

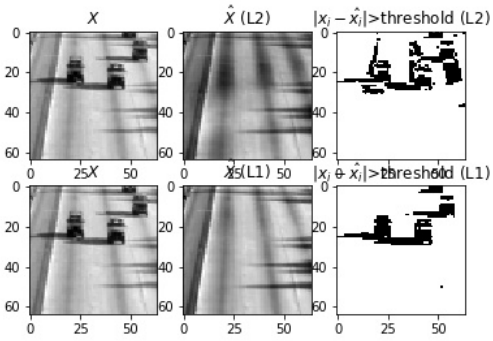
See code in code/pca.py.

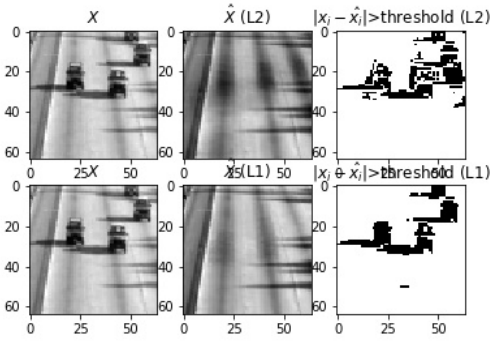
What we need to modify is the objective functions and corresponding derivatives.

The "subtraction" is much clear for cars and highway. After replacement of the L2-norm with L1-norm, the figures do much better in distinction of objects and background. The 10 results is as follows:


X
$\hat{X}$ (L2)
$|x_i - \hat{x}_i|$>threshold (L2)
X
$\hat{X}$ (L1)
$|x_i - \hat{x}_i|$>threshold (L1)


X
$\hat{X}$ (L2)
$|x_i - \hat{x}_i|$>threshold (L2)
X
$\hat{X}$ (L1)
$|x_i - \hat{x}_i|$>threshold (L1)


X
$\hat{X}$ (L2)
$|x_i - \hat{x}_i|$>threshold (L2)
X
$\hat{X}$ (L1)
$|x_i - \hat{x}_i|$>threshold (L1)

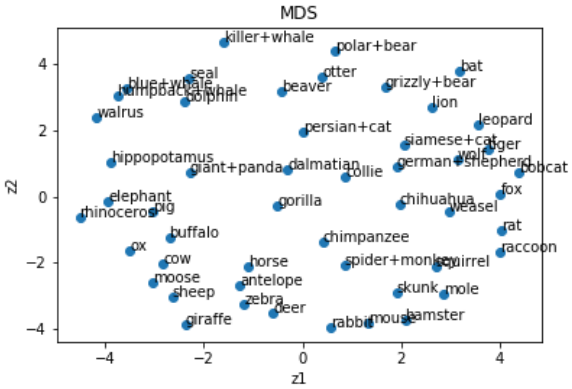

X
$\hat{X}$ (L2)
$|x_i - \hat{x}_i|$>threshold (L2)
X
$\hat{X}$ (L1)
$|x_i - \hat{x}_i|$>threshold (L1)

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.

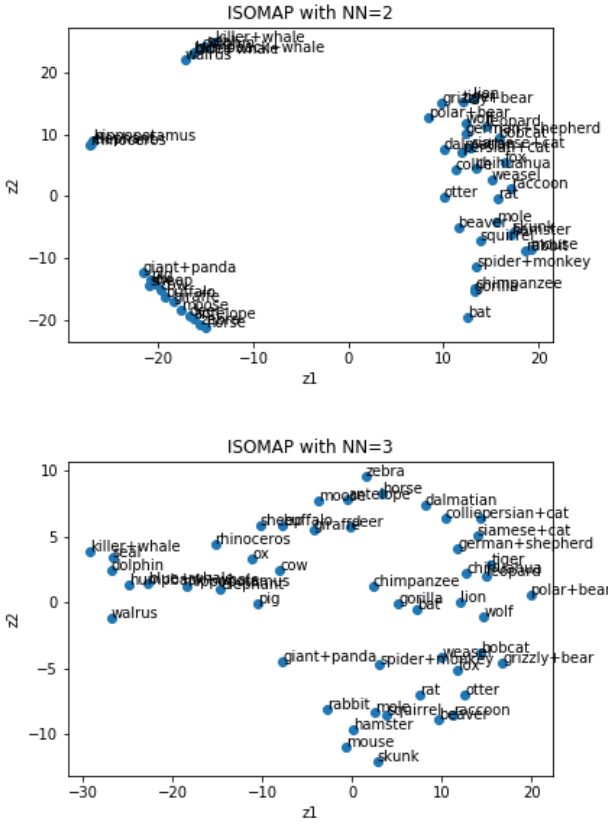
Note: when we say 2 nearest neighbours, we mean the two closest neigbours excluding the point itself. This is the opposite convention from what we used in KNN at the start of the course.

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.)

See code in /code/manifold.py

In codes, we change the distance function from Euclidean distance to geodesic distance.

The results with 2 and 3-nearest neighbours are as follows.



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?

PCA is a kind of linear dimensionality reduction method that projects data from high dimension to low dimension. It is to achieve the solution with the most covariance and orthogonality. But it cannot be used to solve non-linear models.

MDS is a non-parametric latent-factor model not focusing on data but the distance between examples.
Thus it is always used for low-dimensional data visualization.

When it comes to non-linear or curved structures, ISOMAP works. ISOMAP is latent-factor model for visualizing data on manifolds. It applies geodesic distance and MDS to realize the dimensionality reduction.

In this animal data set, we think ISOMAP does the best job.

The relationship between different animals are non-linear. Hence, PCA does not work well for this data set.
As for MDS, because it uses the Euclidean distance, for animals, it is not such reasonable.
We know that related animals tend to share similar traits, so we can assume animals live on a low-dimensional manifold. ISOMAP makes more sense.

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: Because explicitly transforming features into a new space takes very high time complexity while using kernel trick can directly get K rather than Z so that the time complexity could be efficiently reduced.

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

   Answer: Kernel trick with logistic regression is of too high time complexity. Since SVM uses a sparse model in which only those support vectors make contributions when predicting, SVM has less computationally cost.

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

   Answer: As stochastic gradient descent does not use all examples, it is independent of n thus the iteration cost is lower.

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

   Answer: No. It will show a convergence trend at the beginning but will be only up to a certain point and oscillate around the minimum.

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

   Answer: In multi-label classification, there can be several correct labels but in multi-class classification, there is only one correct label.

6. What is the difference between MLE and MAP?

   Answer: MAP stands for maximum a posteriori probability. Thus MAP maximizes the likelihood $p(D|w)$ times the prior $p(w)$.

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. Linear regression only considers one dimension squared distance. Linear regression minimizes vertical squared distance and only cares about predicting y_i.

   While PCA finds line $'W'$ minimizing squared distance in both dimensions. Also PCA assumes centered data and predicts features.

8. Are the vectors minimizing the PCA objective unique? Briefly justify your answer.
   Answer: No. The label switching, scaling and rotations all can make it non-unique.

9. Name two methods for promoting sparse solutions in a linear regression model that result in convex problems.
   Answer: L1-regularization and non-negativity constraint.

10. Can we use the normal equations to solve non-negative least squares problems?
    Answer: No. Normal equations do not consider the non-negative constraint.