Pavel Stepanov HW02

PCA:

1. PCA can be explained from two different perspectives. What are the two perspectives explained in class?

a) Maximizing Variance: PCA finds the directions (principal components) along which the data has maximum variance. (geometric perspective)

b) Minimizing reconstruction error: PCA finds a lower-dimensional representation that minimizes the mean squared distance between the data points and their projections.

2. The first principal direction is the direction in which the projections of the data points have the largest variance in the input space. We use λ1 to represent the first/largest eigenvalue of the covariance matrix, w1 to denote the corresponding principal vector/direction (w1 has unit length i.e., L2 norm is 1), μ to represent the sample mean, and x to represent a data point. The deviation of x from the mean μ is x – μ.

The forward transform, y = PCA(x), is implemented in sk-learn with "whiten=True".

x = data point

μ = mean

λ_1 = first/largest eigenvalue

w_1 = first principal direction (eigenvector)

y = result of forward transform (PCA(x))

(1) what is the scalar-projection of the deviation x – μ in the direction of w1?

$$\boxed{\text{Scalar Projection} = (x-\mu)\cdot w_1}$$

* how much of the deviation $(x-\mu)$ aligns with principal direction w_1.

(2) what is the first component of y ?

$$\boxed{y_1 = (x-\mu)\cdot w_1}$$

where y_1 is the scalar projection from part 1

(3) assuming y only has one component, then we do inverse transform to recover the input

$$\tilde{x} = PCA^{-1}(y)$$

Compute $\tilde{x}$ using μ_1, y_1, λ_1 and w_1

$$\boxed{\rightarrow \tilde{x} = \mu + y_1 \sqrt{\lambda_1}\, w_1}$$

* first principal direction.

(4) assuming x and y have the same number of elements, and we do inverse transform to recover the input

$$\tilde{x} = PCA^{-1}(y)$$

what is the value of $x - \tilde{x}$?

So, if x and y is the same number of component, the inverse transform should exactly recover the original point:

$$\boxed{x - \tilde{x} = 0}$$

→ the value is 0.

Maximum Likelihood Estimation and NLL loss
(This is a general method to estimate parameters of a PDF using data samples)

3. Maximum Likelihood Estimation when the PDF is an exponential distribution.
We have N i.i.d. (independently and identically distributed) data samples {x1, x2, x3, ... , xN} generated from
a PDF that is assumed to be an exponential distribution.
xn ∈ R + for n = 1 to N, which means they are positive scalars.
This is the PDF:

$$f(x) = \begin{cases} \lambda e^{-\lambda x} & \text{for } x \geq 0 \\ 0 & \text{Otherwise} \end{cases}$$

(1) write the NLL loss function: it is a function of the parameter λ

$$f(x;\lambda) = \lambda e^{-\lambda x}, \quad \text{for } x \geq 0$$

data sample (likelihood of observing) $\{x_1, x_2 \dots, x_N\}$ is the product of the individual probabilities:

$$L(\lambda) = \prod_{n=1}^{N} f(x_n;\lambda) = \prod_{n=1}^{N} \lambda e^{-\lambda x_n}$$

Then lets take a $\log(L(\lambda))$.

$$\log(L(\lambda)) = \sum_{n=1}^{N} \log\left(\lambda e^{-\lambda x_n}\right)$$

$$\hookrightarrow \quad \log(L(\lambda)) = \sum_{n=1}^{N} \left(\log\lambda - \lambda x_n\right)$$

$$\hookrightarrow \quad \log(L(\lambda)) = N\log(\lambda) - \lambda \sum_{n=1}^{N} x_n$$

Negative log-likelihood (NLL) function is the negative of the log-likelihood:

$$NLL(\lambda) = \boxed{-\log L(\lambda)} = -N\log(\lambda) + \lambda \sum_{n=1}^{N} x_n$$

So we have: $\boxed{NLL(\lambda) = -N\log(\lambda) + \lambda \sum_{n=1}^{N} x_n}$

(2) take the derivative of the loss with respect to λ, and set the result to 0.
After some calculations, you will obtain an equation about λ =******

Lets take a derivative:

$$\frac{d}{d\lambda}\left(-N\log\lambda+\lambda\sum_{n=1}^{N}x_n\right)$$

$$\frac{d}{d\lambda}NLL(\lambda)=-\frac{N}{\lambda}+\sum_{n=1}^{N}x_n$$

$$-\frac{N}{\lambda}+\sum_{n=1}^{N}x_n=0$$

$$\frac{N}{\lambda}=\sum_{n=1}^{N}x_n$$

$$\boxed{\lambda=\frac{N}{\sum_{n=1}^{N}x_n}}$$

So we made a conclusion that λ is the reciprocol of the mean of the data.

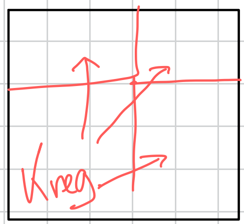
4. Maximum Likelihood Estimation when the PDF is histogram-like.

A histogram-like PDF $f(x)$ is defined on a 1-dimensional (1D) space that is divided into fixed regions/intervals. So, $f(x)$ takes constant value h_i in the i-th region. There are K regions. Thus, $\{h_1, h_2, \ldots, h_K\}$ is the set of (unknown) parameters of the PDF. Also, $\sum_{i=1}^{K} h_i \Delta_i = 1$, where Δ_i is the width of the i-th region.

Now, we have a dataset of N samples $\{x_1, x_2, x_3, \ldots, x_N\}$, and N_i is the number of samples in the i-th region. The task is to find the best parameters of the PDF using the samples.

(1) write the loss function: it is a function of the parameters

Task	Error type	Loss function	Note
Regression	Mean-squared error	$\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$	Easy to learn but sensitive to outliers (MSE, L2 loss)
	Mean absolute error	$\frac{1}{n}\sum_{i=1}^{n}\lvert y_i - \hat{y}_i\rvert$	Robust to outliers but not differentiable (MAE, L1 loss)
Classification	Cross entropy = Log loss	$-\frac{1}{n}\sum_{i=1}^{n}[y_i \log(\hat{y}_i) + (1-y_i)\log(1-\hat{y}_i)] =$	Quantify the difference between two probability



1D

4.3 Maximum likelihood estimation (MLE)

Maximum likelihood estimation (MLE) applies to a wide variety of problems.[6] Since it is the most common method for estimating discrete choice models and discrete choice models are central to the discussion of accounting choice, we focus the discussion of MLE around discrete choice models.

4.3.1 Parameter estimation

The most common method for estimating the parameters of discrete choice models is maximum likelihood. Recall the likelihood is defined as the joint density for the parameters of interest β conditional on the data X_t. For binary choice models and $Y_t = 1$ the contribution to the likelihood is $F(X_t\beta)$, and for $Y_t = 0$ the contribution to the likelihood is $1 - F(X_t\beta)$ where these are combined as binomial draws. Hence,

$$L(\beta|X) = \prod_{t=1}^{n} F(X_t\beta)^{Y_t}\left[1 - F(X_t\beta)\right]^{1-Y_t}$$

The log-likelihood is

$$\ell(\beta|X) \equiv logL(\beta|X) = \sum_{t=1}^{n} Y_t log(F(X_t\beta)) + (1-Y_t)\, log(1 - F(X_t\beta))$$

PDF takes constant value:

$h_i \to$ i-region.

N-data-samples

We know that N_i It is number of samples that in this h_i regions (how many of the total)

- So we need to estimate best value for $h_1 \dots h_K$ are using these data samples:

- We also need to consider that the total probability of PDF is 1, and we have: $\sum_{i=1}^{K} h_i \Delta_i = 1$

size of each i-th region

Likelihood of observing the data samples is the product of the probabilities of each region:

$$L(h_1 \dots, h_K) = \prod_{i=1}^{K} h_i^{N_i}$$

Same as in ex3: let's take a log and we will find NLL:

$$\log(L(h_1 \dots, h_K)) = -\sum_{i=1}^{K} N_i \log(h_i)$$

$$= NLL(h_1 \dots, h_K) = -\sum_{i=1}^{K} N_i \log(h_i)$$

We can't minimize NLL directly due to condition of the PDF: $\sum_{i=1}^{K} h_i \Delta_i = 1$

If we minimize this we will not be able to get some of probability $= 1$:

So let's define new way which ensures that our minimization will respect PDF property:

Method of Lagrange Multipliers

1. Solve the following system of equations.

$$\nabla f(x,y,z) = \lambda \, \nabla g(x,y,z)$$
$$g(x,y,z) = k$$

2. Plug in all solutions, (x,y,z), from the first step into $f(x,y,z)$ and identify the minimum and maximum values, provided they exist and $\nabla g \neq \vec{0}$ at the point.

The constant, λ, is called the **Lagrange Multiplier**.

Notice that the system of equations from the method actually has four equations, we just wrote the system in a simpler form. To see this let's take the first equation and put in the definition of the gradient vector to see what we get.

$$\langle f_x, f_y, f_z \rangle = \lambda \langle g_x, g_y, g_z \rangle = \langle \lambda g_x, \lambda g_y, \lambda g_z \rangle$$

In order for these two vectors to be equal the individual components must also be equal. So, we actually have three equations here.

$$f_x = \lambda g_x \qquad f_y = \lambda g_y \qquad f_z = \lambda g_z$$

2) $$L(h_1, \dots, h_k, \lambda) = -\sum_{i=1}^{k} N_i \log h_i + \lambda\left(\sum_{i=1}^{k} h_i \Delta_i - 1\right)$$

Now lets see what will we have:

let's take a derivative $\frac{dL}{dh_i} = -\frac{N_i}{h_i} + \lambda \Delta_i$

$$-\frac{N_i}{h_i} + \lambda \Delta_i = 0$$ (solve for h_i)

$$-\frac{1}{h_i} + \frac{\lambda \Delta_i}{N_i} = 0$$

$$-\frac{1}{h_i} = -\frac{\lambda \Delta_i}{N_i}$$

$$h_i^{-1} = \frac{\lambda \Delta_i}{N_i}$$ (reciprocal both sides)

$$\Rightarrow \quad h_i = \frac{N_i}{\lambda \Delta_i}$$

→ Now let's try to obtain best parameter:

- the best approach can see here is to find for λ:

as we know $h_i = \dfrac{N_i}{\lambda \Delta_i}$ let's plug this into: $\sum_{i=1}^{k} h_i \Delta_i = 1$

$$\sum_{i=1}^{K} \frac{N_i}{\lambda \cancel{\Delta_i}} \cdot \cancel{\Delta_i} = 1$$

$$\sum_{i=1}^{K} \frac{N_i}{\lambda} = 1 \quad \Rightarrow \quad \frac{1}{\lambda} \sum_{i=1}^{K} N_i = 1 \quad \Big| \times \lambda$$

move

constant

$$\lambda = \sum_{i=1}^{K} N_i$$

* total numbers data points: N

$\lambda = N$

Final Step:
combine

we have: $\rightarrow h_i = \frac{N_i}{\lambda \Delta_i}$

$\rightarrow \lambda = N$

$$\boxed{h_i = \frac{N_i}{N \Delta_i}}$$

Bayes:

5. Bayes classifier has the minimum classification error assuming we know the true p(x|y) and p(y).
However, for many applications, reaching the minimum classification error may not be the best objective.
Now, let consider the application explained in the lecture: there are two classes, class-0 and class-1.

We need to design differentiable loss function $L_n(w)$

True Lables / Predicted Probability:

- $y_n \in \{0,1\}$ true for patent n.
- $\hat{y}_n = f(x_n; w)$ is predicted prob. of class 1 (aneurysms)

$\hat{y}_n \approx 0$ (no repture)

$\hat{y}_n \approx 1$ (repture)

Cost: True(0) $\rightarrow$ if (1) $\rightarrow$ $(100 - t_n) \times \varepsilon$ (surgery)

True(1) $\rightarrow$ if (0) $\rightarrow$ $(100 - t_n)$ (death)

True label y_n	Predicted Label $\hat{y}_n$	Cost for the patient-n
0	0	0
1	1	0
0	1	$(100-t_n)\times\varepsilon$
1	0	$100-t_n$

$\hat{y}_n = f(x_n; w)$ (classification model)

Let's combine this labels:

$$L_n(w) = \boxed{(1-y_n)\,\hat{y}_n\,(100-t_n)}\,\varepsilon \times \boxed{(1-y_n)\,\hat{y}_n\,(100-t_n)}$$

$$\frac{\partial L_n(w)}{dw} = \frac{\partial L_n(w)}{d\hat{y}_n} \times \frac{d\hat{y}_n}{dw}$$

$$\frac{\partial L_n(w)}{dw} = \begin{cases} (1-y_n)(100-t_n) \times \varepsilon & \text{If } y_n = 0 \\ -y_n \times (100-t_n) & \text{If } y_n = 1 \end{cases}$$