به نام خدا


# تمرین درس یادگیری آماری

# سری دوم

**امیدرضا داودنیا**

**پاییز۱۴۰۲**

## ۱. (۲۰ نمره) پاسخ کوتاه

به سوالات زیر به صورت کوتاه پاسخ دهید:

- تفاوت بین forward selection و backward selection را برای انتخاب متغیرها توضیح دهید. آیا به ازای تعداد مشخصی متغیر مورد نظر، مجموعه متغیرهای این دو روش یکسان خواهد شد؟

Forward selection and backward selection are two different methods used for variable selection in statistical models. Their primary differences lie in how they pick variables for the model:

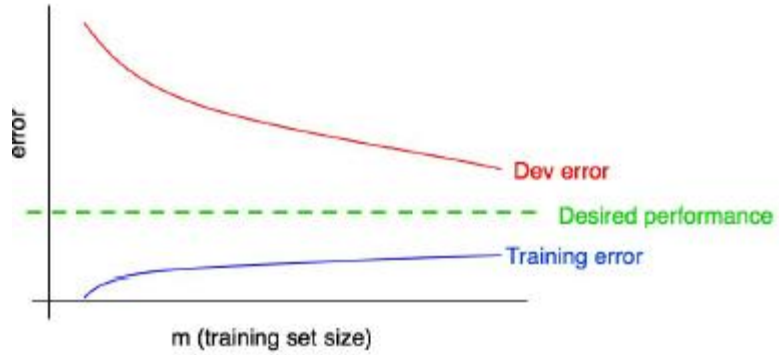
- Forward Selection: In this method, you start with no variables and iteratively add in variables that improve the model the most, until no significant improvement can be made.
- Backward Selection: This approach begins with all variables included in the model. You iteratively remove the variable that improves the model the most by its removal, until no significant improvement can be made by further removal.

Whether these two methods end up selecting the same variables for a given number of desirable variables is not guaranteed. It depends on how those variables interact with each other and the outcome variable in the model. Therefore, the resulting sets of variables from the forward and backward selection techniques might be different. This is one of the reasons why the choice of method depends on the specific context and the characteristics of the data.

- شما در حال طراحی مدلی بر روی یک دیتاست با تعداد ۱۰۰۰ ویژگی برای یک تسک رگرسیون (regression) هستید. در ابتدا مدل خود را بر روی ۱۰۰ نمونه آموزش می‌دهید و مشاهده می‌کنید که با وجود همگرا شدن آموزش، خطای آموزش بر روی این نمونه‌ها زیاد است. پس در ادامه تصمیم می‌گیرید که شبکه خود را این‌بار روی ۱۰۰۰۰ نمونه آموزش دهید. آیا روش شما برای حل این مشکل صحیح است؟ اگر بلی، محتمل‌ترین نتایج مدل خود را در این حالت توضیح دهید. اگر خیر، راه‌حلی برای رفع این مشکل بیان کنید.

Increasing the number of training examples can improve the performance of your model, especially if it's currently suffering from high bias or underfitting. Underfitting often occurs when your model is too simple to capture the complexities within the data, which can result in high training error.

In your case, by training on a larger dataset (10000 samples instead of 100), you're providing your model with more examples from which it can learn. This could help improve its performance, especially if the added examples cover parts of the feature space not well represented in your initial training set.




Increasing the amount of data used for model training generally influences the train error, test error, and development (dev) error as follows:

**Train Error**: Adding more data usually helps to improve the model's learning capacity up to a point. When a model is trained with a small amount of data, it's easier for it to fit the training data too closely (overfit), which can lead to high training error when tested on new data. As more data is fed into the model, it can generalize better, potentially decreasing the training error. However, if the model is too simple to capture the complexity in the added data, the training error might initially increase.

**Test Error**: Regarding the test error, adding more quality, diverse data into training typically helps to reduce it. This is because a more diverse set of examples often leads to better generalizations, resulting in a model that performs better on unseen data. But again, this is up to a point. If the new data is noisy, redundant, or unrepresentative of the broader problem space, adding more of it may not lead to further reductions in test error.

**Dev Error**: The dev set (also known as the validation set) is used during the model's development process to tune hyperparameters, select the best model, etc. Increasing the amount of data in the training set can, as in the test set's case, help the model to better generalize, hence decreasing the dev error. But the same concerns apply here as well.

In essence, adding more data can be beneficial for improving the model's performance, but the data's quality, diversity, and representativeness of the problem space must be assured. After a certain point, simply adding more data

might not yield significant improvements. It's also important to note a balance has to be struck between the complexity of the model and the amount of data - an overly complex model can overfit a large dataset, and an overly simple model may underfit it.

However, just increasing the number of training examples may not solve your problem if your model is not complex enough to capture the underlying patterns in the data. You should also consider other approaches:

Increasing the model complexity: Adding more features or using a more complex model can help reduce the bias and fit the training data better.

Regularization: This is a technique used to prevent overfitting. Regularization adds a penalty term to the loss function that increases with the complexity of the model.

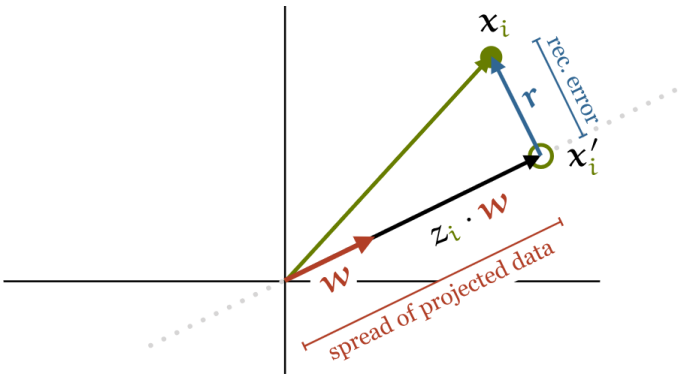
Cross-validation: It helps in understanding how well our model will generalize to unseen data.

Remember, blindly increasing the number of training examples or the complexity of the model can lead to overfitting, where the model performs well on training data but poorly on unseen data. A balanced approach is necessary.

- هر چه بردارهای ویژه‌ای از ماتریس کواریانس که برای کاهش ابعاد از طریق PCA استفاده می‌کنیم دارای مقدار ویژه‌ی بزرگتری باشند، خطای بازسازی کمتر می‌شود.  دلیل این موضوع را به صورت خلاصه توضیح دهید.

The reason for this phenomenon lies in the nature of PCA and eigenvectors. The principal component analysis (PCA) transforms the data into a new coordinate system such that the greatest variance lies on the first axis (the first principal component), the second greatest variance on the second axis, and so on. This transformation is done using the eigenvectors of the covariance matrix, which define these new axes.

Eigenvectors with higher eigenvalues capture a larger amount of the variance in the original data.




Thus, if we use these higher-valued eigenvectors in the PCA transformation, we retain more of the original data's variation, resulting in a reduced reconstruction error. That's because less information is lost during the dimensionality reduction.

The aim of Principal Component Analysis (PCA) is essentially to project a high-dimensional dataset onto a smaller subspace, while retaining as much "interesting" structure (i.e., variance) in the data as possible. This is accomplished through the use of eigenvectors and eigenvalues.

So lets dive into question,The covariance matrix of a dataset is a matrix where each element represents the covariance between two features. The covariance between two features is calculated as follows:

$Cov(X, Y) = \Sigma[(X_i - \mu_x) * (Y_i - \mu_y)] / (n - 1)$

where X_i and Y_i are individual data points, μ_x and μ_y are the means of X and Y, and n is the number of data points.

We can then find the eigenvectors and eigenvalues of this covariance matrix. The eigenvectors represent directions or components in the feature space, while the eigenvalues represent the magnitude of these directions. In other words, an eigenvector with a high eigenvalue indicates a direction in the feature space where there is a lot of variance—in other words, a lot of "interesting" structure.

When we perform dimensionality reduction using PCA, we can choose to preserve only those eigenvectors (i.e., principal components) associated with the largest eigenvalues because they explain the most variance in the data.

The reconstruction error is calculated as follows:

$RE = 1/n * \Sigma(X_i - \hat{X}_i)^2$

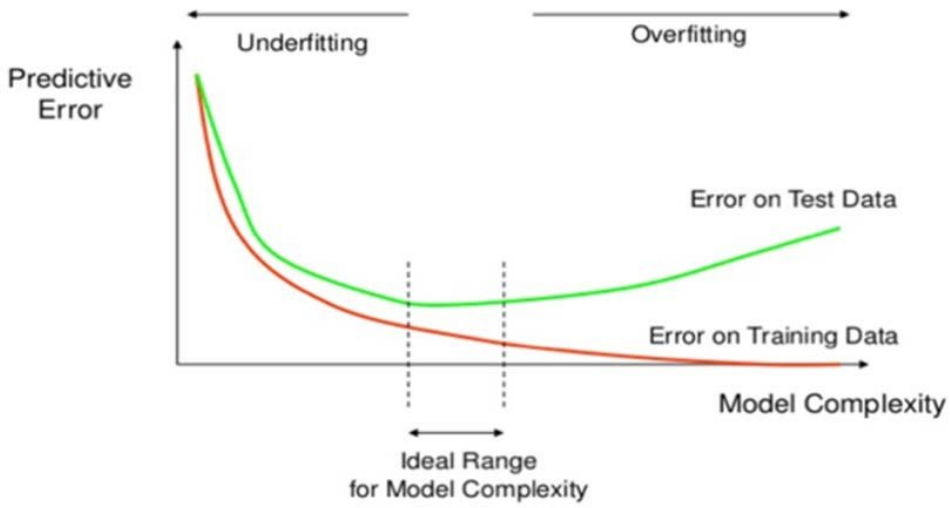
where X_i is the original data and X_hat_i is the reconstructed data from the reduced dimensions.

The reconstructed data X_hat is obtained from the eigenvectors with large eigenvalues. When these dominant eigenvectors are kept, most of the original information (variance) in the data is preserved, hence the reconstruction is more accurate leading to a lower reconstruction error.

In summary, by using the higher-valued eigenvectors for PCA, we are capturing the directions in the data which contain the most information and this in turn makes the reconstruction from the dimension-reduced data quite accurate thereby reducing the reconstruction error.

- خطای روی داده‌های آموزش و تست را در دو حالت overfitting و underfitting مقایسه کنید.

Overfitting and underfitting are two common problems that affect the accuracy of a machine learning model.




Overfitting: Overfitting occurs when your model is too complex, causing it to fit the training data too perfectly and failing to generalize well to new unseen data. In this case, the error on the training data would be very low because the model has basically “memorized” the training data. However, when we test the model on new data (the test set), the error would be high because the model can’t adapt well to the new, unseen data.

Underfitting: Underfitting, on the other hand, occurs when your model is too simple to capture all the relevant patterns in the data. In this scenario, the model performs poorly on both the training data (gives a high training error)

and on new, unseen data (yields a high testing error). It can't generalize well because it hasn't learned enough from the training data in the first place.

In summary, the goal in machine learning is to find a balance between overfitting and underfitting. This is typically achieved through techniques like cross-validation, regularisation, choosing the right complexity for your model, and getting more training data if possible.

## ۲. (۳۵ نمره) رگرسیون خطی، تخمین ML و تخمین MAP

همانطور که از درس می‌دانید، در یک مدل رگرسیون خطی با ویژگی‌های $x_i$ داریم:

$$y = \sum_{i=1}^{p} w_i x_i + \epsilon = w^T x + \epsilon$$

در صورتی که نویز موجود دارای توزیع $\epsilon \sim \mathcal{N}(0, \sigma^2)$ باشد، مشخصا خواهیم داشت:

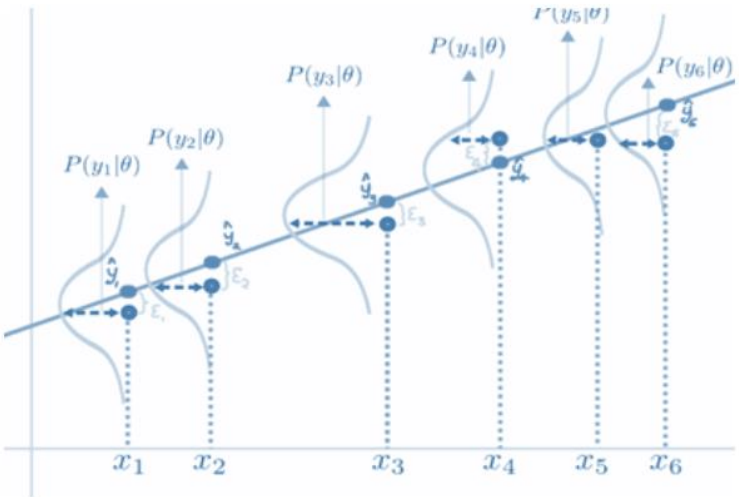
$$y|x, w \sim \mathcal{N}(w^T x, \sigma^2)$$

با در نظر گرفتن تمام نمونه‌های آموزشی می‌توان این عبارت را برای همه آن‌ها بنویسیم و در نتیجه به صورت برداری خواهیم داشت:

$$Y|X, w \sim \mathcal{N}(Xw, \sigma^2 I_n)$$

که در عبارت بالا $y \in \mathbb{R}^n$, $X \in \mathbb{R}^{n \times p}$ و $w \in \mathbb{R}^p$ می‌باشد.

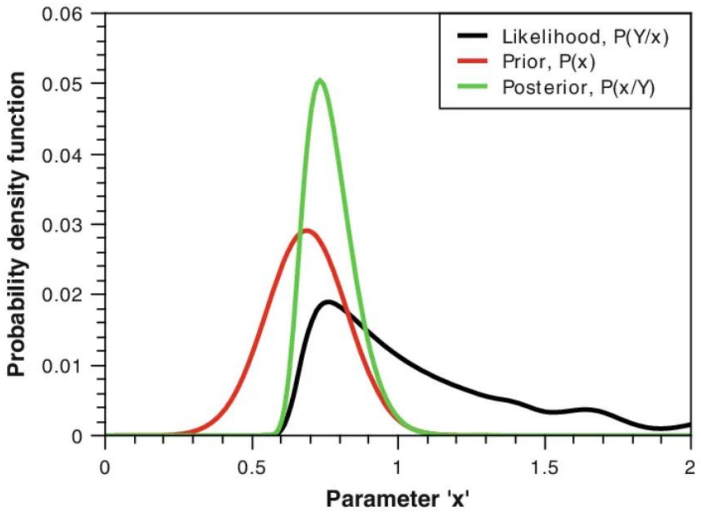
الف) توزیع بالا به چه معناست؟ برای راهنمایی می‌توانید از شکل زیر کمک بگیرید.




**نکته:** در ۳ بخش بعدی جواب خود را به صورت یک مسئله بهینه‌سازی کمترین مربعات (که می‌تواند همراه با یک جمله regularizer باشد) بنویسید و نیازی به محاسبه $\hat{w}_{MAP}$ و $\hat{w}_{ML}$ نیست.

So, these methods don't give you enough information about the distribution of the coefficients, whereas in Bayesian we estimate the posterior distribution of the parameters. Hence, in this case, the output is not a single value but a probability density/mass function.

$$P(\theta|D) = \frac{P(D|\theta)P(\theta)}{P(D)}$$




Therefore, our linear regression model is given by:

$$p(y \mid \boldsymbol{x}, \boldsymbol{\theta}) = \mathcal{N}(y \mid \boldsymbol{x}^\top \boldsymbol{\theta}, \sigma^2)$$
$$y = \boldsymbol{x}^\top \boldsymbol{\theta} + \epsilon\,,$$

In the context of this model, the probability density function is referred to as the likelihood, and the overall uncertainty in the model arises from the observation noise, epsilon.

From this formulation, it becomes evident that our objective is to identify the optimal parameter vector, θ, which defines the most accurate predictor function. We will establish the equations for linear regression by approaching parameter estimation from the perspective of Bayesian statistics.

Estimating Parameters

Given our training set $\mathbb{D}$, comprising inputs $\mathbb{X}$ and targets $\mathbb{Y}$, the mathematical representation of the likelihood for our training set is as follows:

$$p(\mathbb{Y} \mid \mathbb{X}, \boldsymbol{\theta}) = p(y_1, \ldots, y_N \mid \boldsymbol{x_1}, \ldots, \boldsymbol{x_N}, \boldsymbol{\theta})$$

^

Formulated in this manner, our data is assumed to be independent and identically distributed (i.i.d.). From a probabilistic perspective, this assumption allows us to decompose the aforementioned equation into a product of individual data samples.

Approaching parameter estimation, the training process endeavors to determine a singular point value for each parameter based on our training set. Within this post, we will elucidate parameter estimation through two methodologies: Maximum Likelihood Estimation (MLE) and Maximum a Posteriori (MAP).

$$p(\mathbb{Y} \mid \mathbb{X}, \boldsymbol{\theta}) = \prod_{n=1}^{N} p(y_n \mid \boldsymbol{x}_n, \boldsymbol{\theta}) = \prod_{n=1}^{N} \mathcal{N}(y_n \mid \boldsymbol{x}_n^T \boldsymbol{\theta}, \sigma^2)$$

ب) تخمین ML را برای w بدست بیاورید.
این مسئله معادل با کدام حالت روش رگرسیون است؟

In the context of Maximum Likelihood Estimation (MLE), the parameters $\theta_{mle}$ are obtained by maximizing the likelihood function:

$$\boldsymbol{\theta_{mle}} = \arg\max_{\theta} p(\mathbb{Y} \mid \mathbb{X}, \boldsymbol{\theta})$$

From an intuitive standpoint, maximizing likelihood implies adjusting the parameter vector to systematically enhance the likelihood of the predictor function producing the observation y for each corresponding input vector x — precisely the desired outcome.

Rather than employing the product form of the likelihood, we opt for a log transformation for two compelling reasons: to simplify derivative calculations and to circumvent underflows arising from the multiplication of numerous small decimal values. Since the log function is a strictly monotonic increasing function, it preserves the optimization critical points. Furthermore, as we are accustomed to minimizing cost functions, let's reframe this as a minimization problem as well. Thus:

$$-\log p(\mathbb{Y} \mid \mathbb{X}, \boldsymbol{\theta}) = -\log \prod_{n=1}^{N} p(y_n \mid \boldsymbol{x}_n, \boldsymbol{\theta}) = -\sum_{n=1}^{N} \log p(y_n \mid \boldsymbol{x}_n, \boldsymbol{\theta})$$

Here lies the crucial concept of "minimizing negative log likelihood" in machine learning problems! Furthermore, if we regard our problem formulation of likelihood as a Gaussian variable, we can deduce:

$$-\log p(\mathbb{Y} \mid \mathbb{X}, \boldsymbol{\theta}) = \frac{1}{2\sigma^2} \sum_{n=1}^{N} (y_n - \boldsymbol{x}_n^T \boldsymbol{\theta})^2$$
$$= \frac{1}{2\sigma^2} (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\theta})^T (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\theta}) = \frac{1}{2\sigma^2} \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\theta}\|^2$$

When dealing with linear regression, expressing our likelihood in terms of a Gaussian distribution makes maximizing this function akin to minimizing the squared error. Quite fascinating, wouldn't you agree? This also elucidates why we opt for Mean Squared Error (MSE) loss during gradient descent.

An analytical solution is within reach for this predicament. Given that the loss function is quadratic in θ, it implies the existence of a sole global optimum. Consequently, pinpointing this optimum involves calculating its gradient and setting it to zero:

$$\frac{\partial(\frac{1}{2\sigma^2}\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\theta}\|^2)}{\partial \boldsymbol{\theta}} = \frac{1}{\sigma^2}(-\boldsymbol{y}^T\boldsymbol{X} + \boldsymbol{\theta}^T\boldsymbol{X}^T\boldsymbol{X}) = 0$$
$$\iff \boldsymbol{\theta}_{mle} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y}$$

So based on what we learned in class this is simple linear regression without lambda parameter or lambda equal to 0.

پ) فرض کنید برای پارامترهای w یک توزیع اولیه (Prior) در نظر می‌گیریم؛ به طوریکه $w \sim \mathcal{N}(۰, \lambda^۲ I_p)$.
تخمین MAP را برای w بدست آورید.
این مسئله معادل با کدام حالت روش رگرسیون است؟

MLE stands out as an excellent technique for estimating parameters in linear regression problems. However, it exhibits vulnerability to overfitting, particularly evident in the realm of polynomial regression. When dealing with numerous parameters, signifying a substantial representation capacity within our prediction function, MLE tends to produce parameters with significant magnitudes. This results in a function that wildly oscillates and meticulously passes through each data point, yielding a suboptimal representation of our

data-generating function. In machine learning, our goal is not only to minimize errors but also to achieve robust generalization.

To counter the impact of unwieldy parameter values, we can introduce a prior distribution, denoted as p(θ), on the parameters. This prior distribution encodes the plausible range of values. This approach aligns with Bayes' theorem:

$$p(\boldsymbol{\theta} \mid \mathbb{X}, \mathbb{Y}) = \frac{p(\mathbb{Y} \mid \mathbb{X}, \boldsymbol{\theta}) p(\boldsymbol{\theta})}{p(\mathbb{Y} \mid \mathbb{X})}$$

Within MAP estimation, the focus shifts from maximizing the likelihood to maximizing the posterior function, denoted as p(θ | 𝕏, 𝕐). This function encapsulates information from both the likelihood and the prior distribution. To uncover the solution, we employ the same logarithmic transformation:

$$\log p(\boldsymbol{\theta} \mid \mathbb{X}, \mathbb{Y}) = \log p(\mathbb{Y} \mid \mathbb{X}, \boldsymbol{\theta}) + \log p(\boldsymbol{\theta}) + \text{const}$$

We treat everything constant that is independent of the parameters. Once more:

$$\boldsymbol{\theta_{map}} \in \arg\min\{-\log p(\mathbb{Y} \mid \mathbb{X}, \boldsymbol{\theta}) - \log p(\boldsymbol{\theta})\}$$

We designate the prior as a conjugate Gaussian distribution, implying that the posterior distribution will also adopt a Gaussian form (refer to information on Conjugacy distributions). Thus:

$$p(\boldsymbol{\theta}) = \mathcal{N}(\mathbf{0}, b^2 \boldsymbol{I})$$

Deriving the gradient and equating it to zero (while preserving the conditions of positive definiteness):

$$\frac{d \log p(\boldsymbol{\theta} \mid \mathbb{X}, \mathbb{Y})}{d\boldsymbol{\theta}} = -\frac{1}{\sigma^2}(\boldsymbol{\theta}^T \boldsymbol{X}^T \boldsymbol{X} - \boldsymbol{y}^T \boldsymbol{X}) + \frac{1}{b^2}\boldsymbol{\theta}^T = 0$$

$$\iff \boldsymbol{\theta_{map}} = (\boldsymbol{X}^T \boldsymbol{X} + \frac{\sigma^2}{b^2}\boldsymbol{I})^{-1} \boldsymbol{X}^T \boldsymbol{y}$$

As we learned in class this is regularized linear regression or ridge regression!

ج) حال توزیع اولیه را تغییر می‌دهیم. فرض کنید که هر یک از وزن‌ها دارای توزیع $w_i \sim Laplace(0, \lambda)$ باشند. تخمین MAP را برای w بدست آورید.
این مسئله معادل با کدام حالت روش رگرسیون است؟

It appears that since the weights are deterministic, the formulation of the problem must be as follows:

$\varepsilon_i \sim Laplace(0.\lambda) = \frac{1}{2\lambda}\exp(-\frac{|\varepsilon_i|}{\lambda})$,with $\lambda > 0$

So we can rewrite ;

$$\beta = armmax_\beta \; log \prod_i p(y_i|x_i;\beta) = armmax_\beta \sum_i \log p(y_i|x_i;\beta)$$

$$= armmax_\beta \sum_i \log\left(\frac{1}{2\lambda}\exp\left(-\frac{|y_i - x_i^T\beta|}{\lambda}\right)\right)$$

$$= armmax_\beta \sum_i \log\left(\frac{1}{2\lambda} - \frac{|y_i - x_i^T\beta|}{\lambda}\right)$$

$$= armmax_\beta \sum_i -\frac{|y_i - x_i^T\beta|}{\lambda} = armmax_{\beta>0} = \sum_i |y_i - x_i^T\beta|$$

Cost = Logistic Regression Cost + $\lambda \sum_i |\beta_i|$

As we learned in class this is equal to lasso Regression.

چ) تفاوت بین استفاده از این دو توزیع را از دیدگاه اثر آن‌ها بر روی اندازه $w_i$ ها به صورت خلاصه توضیح دهید.

**راهنمایی:**

$$Z \sim Laplace(۰, \lambda) \rightarrow f_Z(z) = \frac{۱}{۲\lambda} exp(-\frac{|z|}{\lambda})$$

$$Z \sim \mathcal{N}(\mu, \Sigma) \rightarrow f_Z(\mathbf{z}) = \frac{۱}{(۲\pi)^{n/۲}|\Sigma|^{۱/۲}} exp(-\frac{۱}{۲}(\mathbf{z}-\mu)^T\Sigma^{-۱}(\mathbf{z}-\mu))$$

Maximum Likelihood Estimation (MLE) and Maximum A Posteriori (MAP) estimation are both methods used to estimate parameters in statistical models, such as the weights (wi) in regression. Here's a brief overview of the key differences:

Objective Function:

MLE: Maximizes the likelihood function, which is the probability of observing the given data under a certain set of parameters.

MAP: Maximizes the posterior distribution, which incorporates both the likelihood and a prior distribution over the parameters.

Incorporation of Prior Information:

MLE: Assumes no prior knowledge about the parameters and relies solely on the likelihood of the observed data.

MAP: Incorporates prior beliefs about the parameters through a prior distribution. The prior distribution represents any existing knowledge or assumptions about the parameters before observing the data.

Formula:

MLE: θ^MLE = argmaxθ L(θ|data), where L(θ|data) is the likelihood function.

MAP: θ^MAP = argmaxθ (L(θ|data)·p(θ)), where p(θ) is the prior distribution.

Handling of Uncertainty:

MLE: Typically provides point estimates of parameters. It doesn't explicitly account for uncertainty in parameter estimates.

MAP: Provides a point estimate, but the incorporation of a prior allows for a more nuanced handling of uncertainty. The posterior distribution reflects a combination of the likelihood and prior, providing a more comprehensive view of parameter uncertainty.

In summary, MLE and MAP differ mainly in their treatment of prior information. MLE assumes no prior information and focuses solely on the likelihood of the data, while MAP incorporates prior beliefs into the estimation process. The choice between MLE and MAP often depends on the amount of prior knowledge available and the desire to regularize or guide parameter estimates based on that prior information.

## ۳. (۲۵ نمره) رگرسیون خطی Ridge

مدل رگرسیون خطی $y = X\beta + \epsilon$ که $X \in \mathbb{R}^{n\times p}$ با کمترین مربعات رگولایز شده $L_۲$ را در نظر بگیرید:

$$\min_{\beta} \|y - X\beta\|_۲^۲ + \lambda\|\beta\|_۲^۲$$

که $\lambda$ پارامتر رگولاریزاسیون است.

الف) فرم بسته $\hat{\beta}^{ridge}(\lambda)$ را بدست آورید.

$$\beta^{ridge} = argmin\{C_{ridge}\}$$

$$C_{ridge} = (y - \boldsymbol{X}\beta)^T(y - \boldsymbol{X}\beta) + \lambda\beta^T\beta = y^Ty - 2\beta^T\boldsymbol{X}^Ty + \beta^T(\boldsymbol{X}^T\boldsymbol{X} + \lambda\boldsymbol{I})\beta$$

$$\frac{\delta C_{ridge}}{\delta \beta} = -2\boldsymbol{X}^T y + 2(\boldsymbol{X}^T\boldsymbol{X} + \lambda \boldsymbol{I})\beta \triangleq 0$$

$$\hat{\beta}_{ridge} = (\boldsymbol{X}^T\boldsymbol{X} + \lambda \boldsymbol{I})^{-1}\boldsymbol{X}^T\boldsymbol{y}$$

ب) اگر $\epsilon \in \mathcal{N}(۰, \sigma^۲ I_n)$ باشد، ثابت کنید که کوواریانس $\hat{\beta}^{ridge}(\lambda)$ به شکل زیر است:

$$Cov(\hat{\beta}^{ridge}(\lambda)) = \sigma^۲(X^TX + \lambda I)^{-۱}X^TX(X^TX + \lambda I)^{-۱}$$

$$var(y|x) = var(\boldsymbol{X}b + \varepsilon|\boldsymbol{X}) = var(\epsilon) = \sigma^2\boldsymbol{I_n}$$

$$var\left(\hat{\beta}_{ridge}\middle|\boldsymbol{X}\right) = (\boldsymbol{X}^T\boldsymbol{X} + \lambda\boldsymbol{I})^{-1}\boldsymbol{X}^T var(y|\boldsymbol{X})[(\boldsymbol{X}^T\boldsymbol{X} + \lambda\boldsymbol{I})^{-1}\boldsymbol{X}^T]^{-1}$$

$$= (\boldsymbol{X}^T\boldsymbol{X} + \lambda\boldsymbol{I})^{-1}\boldsymbol{X}^T\sigma^2\boldsymbol{I_n}\boldsymbol{X}[(\boldsymbol{X}^T\boldsymbol{X} + \lambda\boldsymbol{I})^{-1}\boldsymbol{X}^T]^{-1} =$$

$$\sigma^2(\boldsymbol{X}^T\boldsymbol{X} + \lambda\boldsymbol{I})^{-1}\boldsymbol{X}^T\sigma^2\boldsymbol{X}[(\boldsymbol{X}^T\boldsymbol{X} + \lambda\boldsymbol{I})^{-1}\boldsymbol{X}^T]^{-1}$$

پ) اگر ماتریس X متعامد یکه (Orthonormal) باشد، رابطه بین $\hat{\beta}_j^{ridge}(\lambda)$ و $\hat{\beta}_j^{LS}(\lambda)$ (تخمین‌گر کمترین مربعات معمولی) را برای $j = ۱, ۲, ..., p$ بدست بیاورید. چه انتخابی برای $\lambda$، مقدار $\|\hat{\beta}^{ridge}(\lambda)\|_۲^۲$ را نصف مقدار $\|\hat{\beta}^{LS}(\lambda)\|_۲^۲$ می‌کند؟

$$\hat{\beta}_{ridge} = (\boldsymbol{I} + \lambda\boldsymbol{I})^{-1}\boldsymbol{X}^T y \,.because\ of\ orthogonality\ \boldsymbol{X}^T\boldsymbol{X} = \boldsymbol{I}$$

$$\hat{\beta}_{LS} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T y = (\boldsymbol{I})^{-1}\boldsymbol{X}^T y = \boldsymbol{X}^T y$$

$$\hat{\beta}_{ridge} = ((1+\lambda)\boldsymbol{I})^{-1}\boldsymbol{X}^T y = ((1+\lambda)\boldsymbol{I})^{-1}\hat{\beta}_{LS}$$

$$\hat{\beta}_{ridge} = \frac{1}{1+\lambda}\hat{\beta}_{LS}$$

$$\sum_{j=1}^{p}\hat{\beta}_j^{ridge^2} = \frac{1}{2}\sum_{j=1}^{p}\hat{\beta}_j^{LS^2}$$

$$\xrightarrow{for\ each\ element} \frac{1}{(1+\lambda)^2}\hat{\beta}_j^{LS^2} = \frac{1}{2}\hat{\beta}_j^{LS^2} \rightarrow \lambda = -1 \pm \sqrt{2} \rightarrow \lambda = \sqrt{2} - 1$$

## ۴. (۲۰ نمره) خطای بازسازی PCA

می‌خواهیم عمل PCA را انجام دهیم. هر نمونه $x_i \in \mathbb{R}^p$ به $z_i = V_{۱:k}^T x_i$ تصویر می‌شود. در اینجا $V_{۱:k}$ در واقع $[v_۱|v_۲|...|v_k]$ یا به عبارتی دیگر همان k مولفه اساسی اول است. می‌توانیم $x_i$ را از روی $z_i$ با استفاده از رابطه $\hat{x}_i = V_{۱:k}z_i$ بازسازی نماییم.

الف) نشان دهید

$$\|\hat{x}_i - \hat{x}_j\|_۲ = \|z_i - z_j\|_۲$$

we can rewtite the expression like this ;

$$\left\|\hat{\mathrm{x}}_{\mathrm{i}}-\hat{\mathrm{x}}_j\right\|_2^2=\left\|z_i-z_j\right\|_2^2$$

$$\left\|\hat{\mathrm{x}}_{\mathrm{i}}-\hat{\mathrm{x}}_j\right\|_2^2=\left\|\boldsymbol{V}_{1:k}z_i-\boldsymbol{V}_{1:k}z_j\right\|_2^2=\left((\boldsymbol{V}_{1:k})(z_i-z_j)\right)^T\left((\boldsymbol{V}_{1:k})(z_i-z_j)\right)$$

$$=\left(z_i-z_j\right)^T\boldsymbol{V}_{1:k}^T\boldsymbol{V}_{1:k}\left(z_i-z_j\right)=\left(z_i-z_j\right)^T\boldsymbol{I}\left(z_i-z_j\right)=\left\|z_i-z_j\right\|_2^2$$

With respect to orthogonality we can say the above equation is correct.

ب) نشان دهید خطای بازسازی برابر است با:

$$\sum_{i=1}^{n}\|x_i-\hat{x}_i\|_2^2=(n-1)\sum_{i=k+1}^{p}\lambda_i$$

Start with the definition of the reconstruction error:

$$\|x_i-\hat{x}_i\|^2=\left\|x_i-\boldsymbol{V}_{1:k}\boldsymbol{V}_{1:k}^Tx_i\right\|^2=(x_i-\boldsymbol{V}_{1:k}\boldsymbol{V}_{1:k}^Tx_i)^T(x_i-\boldsymbol{V}_{1:k}\boldsymbol{V}_{1:k}^Tx_i)$$

$$=x_i^Tx_i-x_i^T\boldsymbol{V}_{1:k}\boldsymbol{V}_{1:k}^Tx_i=x_i^Tx_i-\sum_{j=1}^{k}\left(x_i^Tv_j\right)^2$$

$second\ term\ is\ related\ to\ eigen\ values$

Sum over n-1 data points :

$$\frac{1}{n-1}\sum_{i=1}^{n}\|x_i-\hat{x}_i\|^2=\frac{1}{n-1}\sum_{i=1}^{n}(x_i^Tx_i-\sum_{j=1}^{k}\left(x_i^Tv_j\right)^2)$$

And we can rewrite the right side as

$$\frac{1}{n-1}\sum_{i=1}^{n}\|x_i-\hat{x}_i\|^2=\sum_{j=1}^{k}\lambda_j$$

provides valuable insights into the relationship between the reconstruction error (the left-hand side) and the eigenvalues of the covariance matrix (the right-hand side) within the framework of Principal Component Analysis (PCA).

Let's dissect what this formula communicates:

Reconstruction Error:

The left-hand side signifies the average squared reconstruction error across all data points. It gauges how effectively the data can be approximated using the first (k) principal components.

Eigenvalues of the Covariance Matrix:

The right-hand side represents the sum of the first (k) eigenvalues of the covariance matrix. These eigenvalues quantify the amount of variance captured by each principal component.

Connection:

The formula establishes a connection between the reconstruction error and the eigenvalues. Specifically, it asserts that the average squared reconstruction error equals the sum of the eigenvalues associated with the first (k) principal components.

Interpretation:

In the context of PCA, the eigenvalues symbolize the amount of variance present in the data along each principal component. Consequently, the sum of the first (k) eigenvalues reflects the total variance retained by utilizing the first (k) principal components.

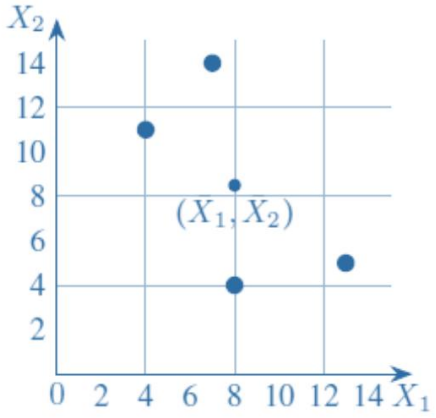
The formula implies that the reconstruction error is linked to the retained variance. As the number of principal components ((k)) increases, the reconstruction error typically decreases because more information is employed to represent the data. However, this reduction in error becomes less pronounced as (k) approaches the total number of features.

Dimensionality Reduction Trade-off:

The formula underscores the concept of a trade-off in dimensionality reduction. Opting for a smaller (k) reduces the dimensionality of the data but introduces reconstruction error, as not all available information is utilized. Striking a balance often involves choosing a value of (k) that captures a substantial portion of the total variance.

In summary, the formula establishes a quantitative relationship between the reconstruction error and the eigenvalues, providing valuable insights into the trade-off between dimensionality reduction and the amount of variance retained in the data. This understanding is crucial for grasping the impact of PCA on data representation and compression.'

مجموعه دادگان زیر را در نظر بگیرید:




If our points are ;

Original Data:

[ [ 4 11],[ 7 14],[ 8  4],[13  5] ]

Mean of Data:

[8.  8.5]

Subtract the mean of each variable from the data points to center them around the origin.

$X_{centered} = X - \bar{\bar{X}}$

Centered Data:

[[-4.  2.5],[-1.  5.5],[ 0. -4.5],[ 5. -3.5]]

Calculate the covariance matrix ;

$$S = \frac{1}{n-1} {X_{centered}}^T X_{centered}$$

Covariance Matrix:

[[ 14. -11.],[-11.  23.]]

Solve the characteristic equation

$\det(S - \lambda \boldsymbol{I}) \triangleq 0$

Eigenvalues:

[ 6.615 ,30.385]

Eigenvectors:

[[-0.830  0.557],[-0.557 -0.830]]

Normalize each eigenvector to unit length.

$$V_{normalized} = \frac{V}{|V|}$$

Normalized Eigenvectors:

[[-0.830  0.557],[-0.557 -0.830]]

Sort the eigenvalues in descending order and choose the corresponding eigenvectors as the principal components.

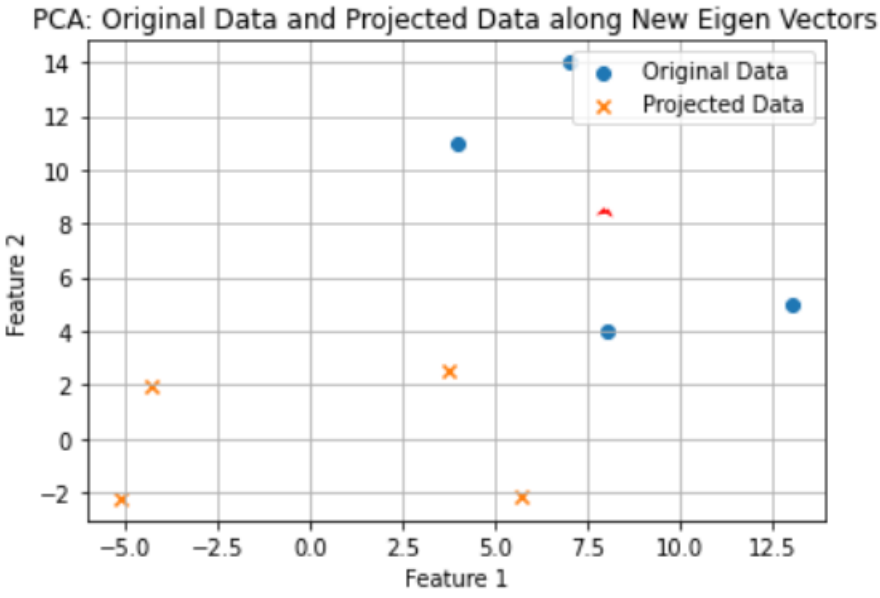
Projected Data=Xcentered×Principal Components

Principal Components:

[[ 0.557 -0.830],[-0.830 -0.557]]

Projected Data:

[[-4.305  1.927],[-5.123 -2.235],[ 3.736  2.508],[ 5.692 -2.200]]



The practical part is impleamented and well described in note book file.