SHORT LECTURES ON
MATHEMATICS/PHYSICS/ALGORITHM
2022-2

Lessons from Curve Fitting

Bao-Jun Cai, Institute of Innovation, 6/1/2022

In this lecture, several fundamental concepts of machine learning are introduced via the classical curve fitting problem, e.g., the learning model, parameter space, regularization term, over-fitting, training and testing errors, loss function, normal equation, model complexity, and the trade-off between bias and variance of the learning model. Moreover, the very basic ideas of Bayesian calculation are given using examples, like the prior, the likelihood function and the posterior for the parameter(s) to be estimated. The reduction of the variance of the parameter(s) by generation of data samples is also discussed. The lecture is organized as follows: In section I we introduce the basic concepts of model fitting using the linear regression as the example. Sections II and III generate the linear regression to the general nonlinear situation, where a few fundamental concepts of machine learning are given like the trade-off between bias and variance. Section IV is devoted to the regularization method, which could effectively handle the over-fitting problem. we discuss in section V several advanced theoretical issues related to the parameter estimation, such as the design matrix, the basis function, the geometrical meaning of the least squares, and the generalized learning model with data weights. Sections VI and VII give the overall picture of the Bayesian calculation and the maximum likelihood estimation and in section VIII we use the method to revisit the linear regression. Section IX discusses the variance reduction during the Bayesian learning process as data samples generates. Section X is devoted to the discussion on the correlations between/among the learning parameters together with the error-bar estimation problem. In sections XI and XII the concepts/techniques of/from the central limit theorem and the law of large numbers are briefly introduced. Finally sections XIII and XIV briefly introduce the concept of singular value decomposition from the viewpoints of linear equations and best-fit approximations. The role play by examples is emphasized throughout.

I. START: LINEAR FITTING MODEL

Assume that one has m data points $(x^{(i)}, y^{(i)})$ with $i = 1 \sim m$, here the physical or the realistic relation between $x^{(i)}$ and $y^{(i)}$ is assumed to be linear, e.g., the relation between the velocity v and the acceleration a as $v = at$. Since there exists measurement errors, the measured or the experimental relation between $x^{(i)}$ and $y^{(i)}$ is not exactly linear. In this case, one can still use linear regression to obtain the model parameters or the learning parameters from the noisy data.

To this end, we assume that the model is linear and denote it by $f_{\vec{\theta}}(x) = ax + b$ with a and b two parameters to be determined by the linear regression, the parameters are collectively denoted by $\vec{\theta} = (a, b)$. In order to obtain the model parameter $\vec{\theta}$, one needs to minimize the error between the model prediction for the data $x^{(i)}$, i.e., $f_{\vec{\theta}}(x^{(i)})$, and the measurement $y^{(i)}$. One of the frequently-used error is the squared loss, i.e., $(f_{\vec{\theta}}(x^{(i)}) - y^{(i)})^2$, here $\epsilon^{(i)} = f_{\vec{\theta}}(x^{(i)}) - y^{(i)}$ is called the algebraic distance between the model prediction and the measurement (which could take either positive or negative values), see Fig. 1 for the sketch of the algebraic distance. The total loss is the sum of the error of all samples,

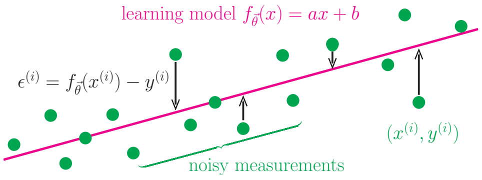

learning model $f_{\vec{\theta}}(x) = ax + b$
$\epsilon^{(i)} = f_{\vec{\theta}}(x^{(i)}) - y^{(i)}$
$(x^{(i)}, y^{(i)})$
noisy measurements

Fig. 1: Sketch of the linear regression problem.

$$J(\vec{\theta}) = \frac{1}{2}\sum_{i=1}^{m}\left[f_{\vec{\theta}}(x^{(i)}) - y^{(i)}\right]^2. \tag{1.1}$$

The factor 1/2 is irrelevant here. It should be pointed out that J is a function of $\vec{\theta}$ or equivalently of a and b, instead of $x^{(i)}$ or $y^{(i)}$.

In order to obtain the parameters a and b, one needs to minimize the function J. Since J is a convex function (like the parabolic function x^2), the minimization of J is reduced to $\partial J/\partial a = 0$ and $\partial J/\partial b = 0$, these two equations determine the optimized parameters a^* and b^*. By expanding the loss $J(\vec{\theta})$, we have

$$J(\vec{\theta}) = \frac{m}{2}\left[\langle x^2\rangle a^2 + b^2 + \langle y^2\rangle + 2\langle x\rangle ab - 2\langle xy\rangle a - 2\langle y\rangle b\right], \tag{1.2}$$

where the data sample averages are defined by,

$$\langle x\rangle = \frac{1}{m}\sum_{i=1}^{m} x^{(i)},\ \ \langle y\rangle = \frac{1}{m}\sum_{i=1}^{m} y^{(i)}, \tag{1.3}$$

$$\langle x^2\rangle = \frac{1}{m}\sum_{i=1}^{m} x^{(i),2},\ \ \langle y^2\rangle = \frac{1}{m}\sum_{i=1}^{m} y^{(i),2},\ \ \langle xy\rangle = \frac{1}{m}\sum_{i=1}^{m} x^{(i)}y^{(i)}, \tag{1.4}$$

which could be calculated once the measurement is available. After some straightforward calculations, one obtains

$$a^* = \frac{\langle x\rangle\langle y\rangle - \langle xy\rangle}{\langle x\rangle^2 - \langle x^2\rangle},\ \ b^* = \frac{\langle x\rangle\langle xy\rangle - \langle x^2\rangle\langle y\rangle}{\langle x\rangle^2 - \langle x^2\rangle}. \tag{1.5}$$

In fact, if $y \sim x$, the numerator of $b^* \sim \langle x\rangle\langle x^2\rangle - \langle x^2\rangle\langle x\rangle = 0$.

The coefficients a^* could be written as $a^* = \mathrm{cov}[x,y]/\mathrm{var}[x]$, and consequently the optimal prediction for the output is

$$y^* = \mathrm{E}[y] + \frac{\mathrm{cov}[x,y]}{\mathrm{var}[x]}\left[x - \mathrm{E}[x]\right]. \tag{1.6}$$

The mean square error (MSE) of observation is define as $\Delta^* = \mathrm{E}[y - y^*]^2 = \mathrm{var}[y][1 - \rho^2[x,y]]$, where $\rho[x,y]$ is the correlation between x and y. Consequently, the larger (in absolute value) the correlation coefficient the smaller the MSE of observation. In particular if $|\rho(x,y)| = 1$ then $\Delta^* = 0$, on the other hand if x's and y's are uncorrelated, $\Delta^* = \mathrm{var}[y]$ and $y^* = \mathrm{E}[y]$. We simulate the model and prepare the data as follows: Assume that the physical model is given by $y = 3x$, i.e., ideally $a_{\mathrm{phys}} = 3$ and $b_{\mathrm{phys}} = 0$. The data is generated through $y^{(i)} = a'x^{(i)}$ with $a' = a_{\mathrm{phys}} \pm \phi$ where ϕ denotes the noise, e.g., ϕ is a zero-mean random variable with Gaussian distribution $\phi \sim \mathcal{N}(0, \sigma^2)$, and consequently $a' = \mathcal{N}(a_{\mathrm{phys}}, \sigma^2) = \mathcal{N}(3, \sigma^2)$. In our calculations, we fix $\sigma^2 = 0.3$. In addition, the data sample is generated by using this a' and the $x^{(i)}$ uniformly distributed within 0 and 5, i.e., $x^{(i)} \sim \mathrm{Unif}[0,5]$. In Fig. 2 we show the simulated samples ($m = 50$), the fitting line $a^*x + b^*$ and the physical model used by randomly running the code. It is clearly demonstrated that the "learned model $a^*x + b^*$" has some deviation from the physical model (here $3x$). Specifically, b^* is not exactly equal to zero.

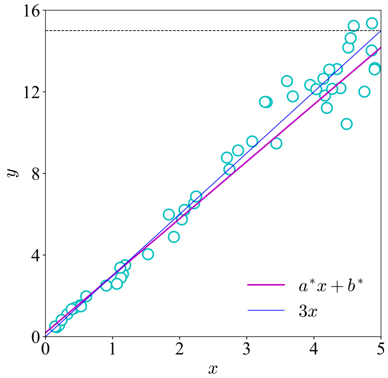


Fig. 2: Simulated samples, fitting line $a^*x + b^*$ and the physical model, the number of samples is $m = 50, a_{\mathrm{phys}} = 3, b_{\mathrm{phys}} = 0$.

EXERCISE 1: Define error of the point $(x^{(i)}, y^{(i)})$ to the line $y = ax + b$ as $d_{\perp}^{(i)}(a,b) = |ax^{(i)} + b - y^{(i)}|/\sqrt{a^2+1}$ or $d_{\mathrm{x}}^{(i)}(a,b) = |ax^{(i)} + b - y^{(i)}|/a$, the loss function could be obtained as $J_{\perp}(a,b) = 2^{-1}\sum_{i=1}^{m} d_{\perp}^{(i),2}(a,b)$ or $J_{\mathrm{x}}(a,b) = 2^{-1}\sum_{i=1}^{m} d_{\mathrm{x}}^{(i),2}(a,b)$. Derive the equations for determining the parameters a and b. If a is large, $d_{\perp} \approx d_{\mathrm{x}}$. Numerically solve the optimization problems.

We can independently run the simulation for k times and obtain the a^* and b^* for k times. Consequently, the k-average of the a^* and b^* are,

$$\overline{a^*}(k,m) \equiv \frac{1}{k}\sum_{j=1}^{k} a^{*,(j)},\ \ \overline{b^*}(k,m) \equiv \frac{1}{k}\sum_{j=1}^{k} b^{*,(j)}. \tag{1.7}$$

The corresponding k-dependence of the $\overline{a^*}$ and $\overline{b^*}$ is shown in Fig. 3 where $m = 10$ is fixed for each simulation. As the k increases the k-

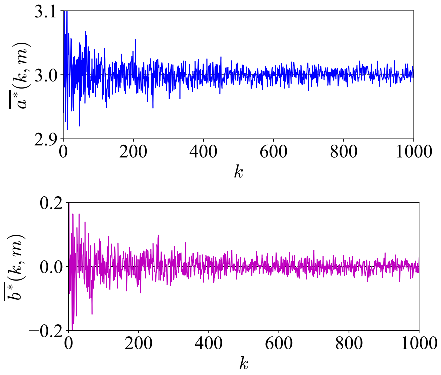


Fig. 3: k-dependence of $\overline{a^*}(k,m)$ and $\overline{b^*}(k,m)$, $m=10$ is fixed.

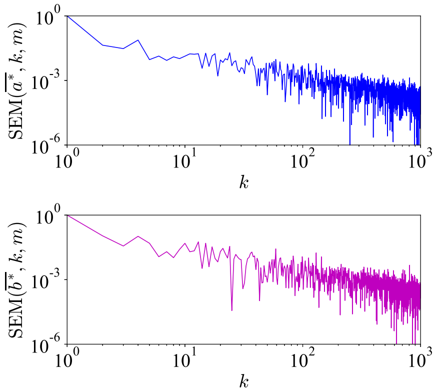


Fig. 4: k-dependence of SEM$(\overline{a^*},k,m)$ and SEM$(\overline{b^*},k,m)$, $m=10$ is fixed.

average of the learning parameters eventually approach to the physical values. Similarly, by defining the standard error in the means (SEM),

$$\mathrm{SEM}(\overline{f^*},k,m)=\sqrt{\frac{1}{k}\frac{1}{k-1}\sum_{j=1}^{k}\left(f^{*,(j)}-\overline{f^*}\right)^2},\quad f=a,b, \tag{1.8}$$

one could study, e.g., the large-k behavior of the overall errors encapsulated in the learning parameters. We show in Fig. 4 the k-dependence of the SEM for $\overline{a^*}$ and $\overline{b^*}$ while fixing $m=10$. From the figure it is clearly indicated that in the log-log plot the overall tendency is quasi-linear.

II. NONLINEAR GENERALIZATION: CONCEPTS

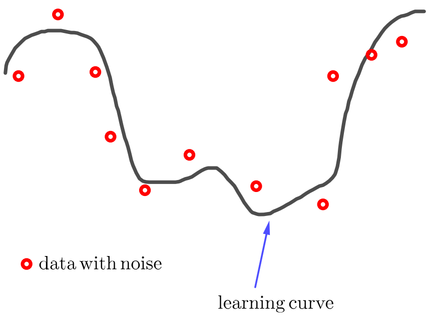


Fig. 5: Sketch of the nonlinear fitting problem.

The linear model $f_{\vec{\theta}}(x)=ax+b$ studied in the last section is simple and convenient to implement, however it is sometime too simple to capture other features encapsulated in the data, e.g., the irregularity and/or the nonlinearity, see Fig. 5 for an example. It is certain that the linear learning model could hardly work for these situations, where one needs to develop new learning techniques encapsulating the nonlinearity of the features. A few basic concepts are necessarily needed to be introduced. There is a physical model, denoted by $f_{\mathrm{phys}}(\mathbf{x})$ (generally the input data $\mathbf{x}$ has the vector nature), which is unknown in advance and maybe even very complicated. Although the physical model $f_{\mathrm{phys}}(\mathbf{x})$ is unknown, there exist relevant data generated by the model, and the data always has measurement noise (also unknown). We denote the data point as $(\mathbf{x}^{(i)},y^{(i)})$, and since there exists noise in the output $y^{(i)}$ generally $y^{(i)}\neq f_{\mathrm{phys}}(\mathbf{x}^{(i)})$. For simplicity here we assume the output is a scalar. Besides the physical model, a learning model denoted by $f_{\mathbf{w}}(\mathbf{x})$ is often introduced with $\mathbf{w}$ a set of parameters characterizing the learning model. Based on a fixed learning model, one could make a prediction on each input data $\mathbf{x}^{(i)}$ to obtain $f_{\mathbf{w}}(\mathbf{x}^{(i)})$. The learning model is an effective approximation of the physical model since the latter is generally very complicated. Generally, the prediction $f_{\mathbf{w}}(\mathbf{x}^{(i)})$ will be different from the measured output data $y^{(i)}$, and the basic task in machine learning and/or data analysis/mining is to minimize the difference between the measurement and the prediction. Consequently, an error (cost/loss) function emerges, which characterizes the above difference.

A very frequently-used error function is given by

$$J(\mathbf{w})=\frac{1}{2}\sum_{i=1}^{m}\left[f_{\mathbf{w}}(\mathbf{x}^{(i)})-y^{(i)}\right]^2, \tag{2.1}$$

i.e., the sum of the difference between the prediction ($f_{\mathbf{w}}(\mathbf{x}^{(i)})$) and the measurement ($y^{(i)}$), which has already been used in the linear learning model, see (1.1). This type of optimization is called the least-squares (LS). As similar as in the linear fitting problem, $J(\mathbf{w})$ is function of the learning parameter $\mathbf{w}$, and not a function of the measurements $(\mathbf{x}^{(i)},y^{(i)})$. The next task is to minimize the error function $J(\mathbf{w})$, and since generally $J(\mathbf{w})$ is a convex function of $\mathbf{w}$, the minimization of it is reduced to the condition $\partial J(\mathbf{w})/\partial\mathbf{w}=0$, or equivalently,

$$\partial J(\mathbf{w})/\partial w_j=0,\quad j=0,1,2,\cdots,n, \tag{2.2}$$

where one assumes that there are totally $n+1$ parameters namely w_0 to w_n. It is necessary to point out that we use two letters "n" and "m" to represent the number of the learning parameters and the number of data points, respectively. Specifically, if the input data has the scalar nature (i.e., x instead of $\mathbf{x}$), a very general learning model in machine learning is the polynomial of order n, i.e.,

$$f_{\mathbf{w}}(x)=w_0+w_1x+w_2x^2+\cdots+w_nx^n=\sum_{j=0}^{n}w_jx^j, \tag{2.3}$$

here the learning parameter consists of $n+1$ scalars, i.e., w_0 to w_n. One uses the measured data points $(x^{(i)},y^{(i)})$ to study these parameters, which is then called "learning from data". Although $f_{\mathbf{w}}(x)$ is nonlinear in x, it is still linear in the parameters w_j. In this sense, we also call $f_{\mathbf{w}}(x)$ the linear model. The second order derivative of the loss function with respect to w_j is similarly given

$$\begin{aligned}\frac{\partial^2 J}{\partial w_j^2}&=\frac{\partial}{\partial w_j}\frac{\partial}{\partial w_j}\left[\frac{1}{2}\sum_{i=1}^{m}\left[f_{\mathbf{w}}(\mathbf{x}^{(i)})-y^{(i)}\right]^2\right]\\&=\sum_{i=1}^{m}\left(\frac{\partial}{\partial w_j}f_{\mathbf{w}}(x^{(i)})\right)^2+\sum_{i=1}^{m}\left(f_{\mathbf{w}}(x^{(i)})-y^{(i)}\right)\frac{\partial^2}{\partial w_j^2}f_{\mathbf{w}}(x^{(i)}).\end{aligned} \tag{2.4}$$

If the learning model is linear then the second term is zero at the optimal parameter, leading to

$$\frac{\partial^2 J}{\partial w_j^2}=\sum_{i=1}^{m}\left(\frac{\partial}{\partial w_j}f_{\mathbf{w}}(x^{(i)})\right)^2, \tag{2.5}$$

which is always positive. Even the learning model is nonlinear, we can still prove that $\partial^2 J/\partial w_j^2>0$, using the normal equation given later.

III. BIAS-VARIANCE DECOMPOSITION: CALCULATIONS

We solve the nonlinear curve fitting problem in details to demonstrate the important features of machine learning, i.e., how could one "learn from data", and what to learn? According to Eq. (2.2), one can obtain

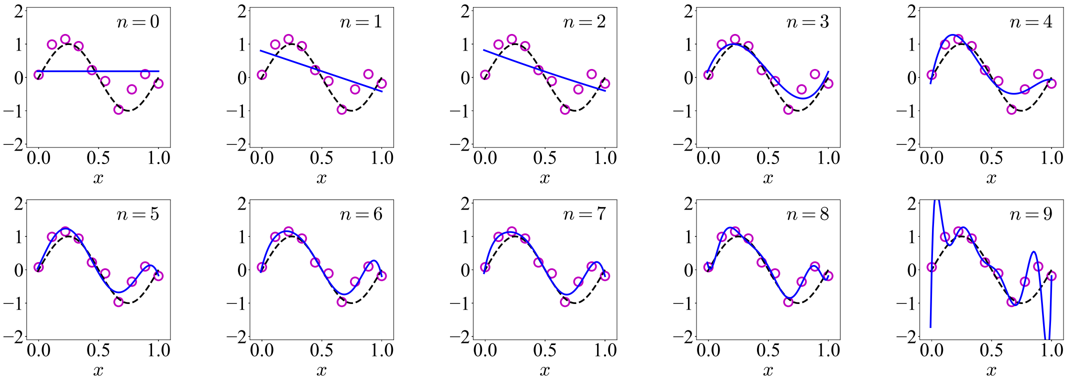


Fig. 6: Learning processes with different n.

	$n=0$	$n=1$	$n=2$	$n=3$	$n=4$	$n=5$	$n=6$	$n=7$	$n=8$	$n=9$
w_0^*	0.19	0.79	0.81	0.23	−0.00	0.10	0.08	0.08	0.08	0.08
w_1^*	∅	−1.21	−1.32	8.11	16.93	7.42	13.99	16.69	−9.48	145.11
w_2^*	∅	∅	0.11	−24.74	−69.30	10.00	−71.10	−115.61	412.07	−3158.52
w_3^*	∅	∅	∅	16.57	88.03	−136.10	212.92	476.91	−3503.21	28658.38
w_4^*	∅	∅	∅	∅	−35.73	221.18	−453.31	−1199.76	13815.03	−137275.68
w_5^*	∅	∅	∅	∅	∅	−102.76	497.20	1585.76	−29673.59	381953.29
w_6^*	∅	∅	∅	∅	∅	∅	−199.99	−989.95	35479.27	−638534.33
w_7^*	∅	∅	∅	∅	∅	∅	∅	225.70	−22102.39	631648.54
w_8^*	∅	∅	∅	∅	∅	∅	∅	∅	5582.02	−340299.59
w_9^*	∅	∅	∅	∅	∅	∅	∅	∅	∅	76862.54

Tab. 1: Optimal parameters $w_j^*, j = 0 \sim n$ in different learning models.

the equations for determining the model parameters $w_0, w_1, w_2, \cdots, w_n$. Specifically, we have

$$\begin{aligned}\frac{\partial J}{\partial w_j} &= \sum_{i=1}^m \left(f_{\mathbf{w}}(x^{(i)}) - y^{(i)}\right)\frac{\partial}{\partial w_j} f_{\mathbf{w}}(x^{(i)})\\ &= \sum_{i=1}^m f_{\mathbf{w}}(x^{(i)})x^{(i),j} - \sum_{i=1}^m y^{(i)}x^{(i),j} = \sum_{i=1}^m\sum_{j'=0}^n w_{j'}x^{(i),j'+j} - \sum_{i=1}^m y^{(i)}x^{(i),j}\\ &= \sum_{j'=0}^n w_{j'}\sum_{i=1}^m x^{(i),j'+j} - \sum_{i=1}^m y^{(i)}x^{(i),j} = m\left[\sum_{j'=0}^n w_{j'}\langle x^{j'+j}\rangle - \langle x^j y\rangle\right],\end{aligned} \tag{3.1}$$

and the relevant equation is given by setting it to be zero, i.e.,

$$\sum_{j'=0}^n w_{j'}\langle x^{j'+j}\rangle = \langle x^j y\rangle, \quad j = 0 \sim n. \tag{3.2}$$

Consequently, we have

$$\begin{cases} \langle 1\rangle w_0 + \langle x\rangle w_1 + \langle x^2\rangle w_2 + \cdots + \langle x^n\rangle w_n = \langle y\rangle,\\ \langle x\rangle w_0 + \langle x^2\rangle w_1 + \langle x^3\rangle w_2 + \cdots + \langle x^{n+1}\rangle w_n = \langle xy\rangle,\\ \vdots,\\ \langle x^n\rangle w_0 + \langle x^{n+1}\rangle w_1 + \langle x^{n+2}\rangle w_2 + \cdots + \langle x^{2n}\rangle w_n = \langle x^n y\rangle,\end{cases} \tag{3.3}$$

where $\langle 1\rangle = m^{-1}\sum_{i=1}^m 1 = 1$, and

$$\langle x^k\rangle = \frac{1}{m}\sum_{i=1}^m x^{(i),k} = \frac{1}{m}\left(x^{(1),k} + x^{(2),k} + \cdots + x^{(m),k}\right), \tag{3.4}$$

$$\langle x^k y\rangle = \frac{1}{m}\sum_{i=1}^m x^{(i),k}y^{(i)} = \frac{1}{m}\left(x^{(1),k}y^{(1)} + x^{(2),k}y^{(2)} + \cdots + x^{(m),k}y^{(m)}\right). \tag{3.5}$$

As a special case, consider $n = 1$, i.e., for the linear fitting problem, Eq. (3.3) becomes $\langle 1\rangle w_0 + \langle x\rangle w_1 = \langle y\rangle$ and $\langle x\rangle w_0 + \langle x^2\rangle w_1 = \langle xy\rangle$. In addition, Eq. (3.3) could be rewritten in the form, $\mathbf{Fw} = \mathbf{G}$, where

$$\mathbf{F} = \begin{pmatrix} \langle 1\rangle & \langle x\rangle & \cdots & \langle x^n\rangle\\ \langle x\rangle & \langle x^2\rangle & \cdots & \langle x^{n+1}\rangle\\ \vdots & \vdots & \ddots & \vdots\\ \langle x^n\rangle & \langle x^{n+1}\rangle & \cdots & \langle x^{2n}\rangle \end{pmatrix}, \quad \mathbf{w} = \begin{pmatrix} w_0\\ w_1\\ \vdots\\ w_n\end{pmatrix}, \quad \mathbf{G} = \begin{pmatrix}\langle y\rangle\\ \langle xy\rangle\\ \vdots\\ \langle x^n y\rangle\end{pmatrix}. \tag{3.6}$$

Note that the $(n+1)\times(n+1)$ matrix $\mathbf{F}$ is symmetric, namely $F_{ij} = F_{ji}$. The symmetry properties of $\mathbf{F}$ is useful for calculating its inverse.

EXERCISE 2: Derive the analytical expressions for w_j's in the case of $n = 2$ and $n = 3$, write down the explicit form of $\mathbf{F}^{-1}$.

EXERCISE 3: Prove $\mathbf{F}$ could be written as $\vec{\Phi}^{\mathrm{T}}\vec{\Phi}$ where the element of the matrix $\vec{\Phi}$ is $\phi_{ji} = \phi_j(x^{(i)})$ with $j = 0 \sim n, i = 1 \sim m$, see (5.4).

We similarly prepare the data to be used in the simulation. Here the physical model is adopted as $f_{\mathrm{phys}}(x) = \sin(2\pi x)$ with $0 \le x \le 1$, and we generate total 10 data points uniformly distributed within this range, i.e., $x^{(i)} = i/9, i = 0, 1, 2, \cdots, 9$, or, $x^{(i)} = 0, 1/9, 2/9, \cdots, 8/9, 1$. The output data $y^{(i)}$ is obtained from the physical model by including a noise, i.e., $y^{(i)} = \sin(2\pi x^{(i)}) + a^{(i)}$, here $a^{(i)} \sim \mathrm{Unif}[-\ell, \ell]$ is a uniformly distributed random number. We adopt $\ell = 0.8$ for the simulation. After solving the equation $\mathbf{Fw} = \mathbf{G}$, one obtains the parameter $\mathbf{w}^*$.

Results for a series of learning model $f_{\mathbf{w}^*}(x)$ with different parameter n are shown in Fig. 6. Let's discuss starting with the case "$n = 0$", now the learning model is $f_{\mathbf{w}}(x) = w_0$ and in this case the optimized parameter w_0^* is just the mean of the output $y^{(i)}$, namely $w_0^* = (y^{(1)} + y^{(2)} + \cdots + y^{(10)})/10$. Next, we have re-obtained the linear fitting result if $n = 1$. There are several novel and important features shown in Fig. 6. Firstly, at small n the learning curve (blue line) predicts bad for the data points, and the prediction becomes better and better when n increases. In the limit situation of $n = 9$, the prediction can perform perfectly on the data points (there is no difference between the measurement $y^{(i)}$ and the prediction $f_{\mathbf{w}}(x^{(i)})$ at these points). However, the learning curve becomes stranger at larger n while it is much smoother at smaller

n. The smoothness and the strangeness characterize two important aspects of the learning model: When the learning curve is smoother we call it has a smaller variance, while when the curve is closer to the measurements we call it has a smaller bias. Thus, when n is small, the learning model has a small variance and a large bias and the learning model has a large variance and a small bias when n is large. This phenomenon is sometimes called the "no-free-lunch theorem",[1] in the sense that one could not obtain a learning model with both small variance and bias. It is a very general feature of all the learning problems in data analysis. The "no-free-lunch theorem" is also called the bias-variance trade-off or the bias-variance decomposition.

The learning model with large n is very complicated and has a very strong power of fitting data but limited power of predicting new data. We thus often call n the complexity of the model. In our example, $n \approx 3 \sim 6$, is reasonable. In Tab. 1, the values of the optimal $\mathbf{w}_j^*$ are shown in models with different n. One of the important features is that as the model complexity n increases the magnitude of w_j^*'s eventually increase. It is very dangerous in the sense that in order to finally obtain a naturally prediction on the output the large terms $w_j^* x^j$ are added, and this phenomenon is called fine-tuning. One of the popular to avoid the fine-tuning is through the regularization term introduced into the model, see discussion given in section IV.

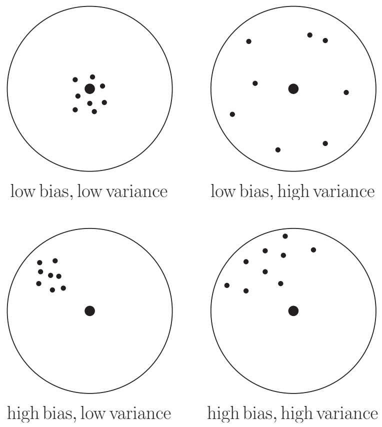


Fig. 7: Patterns of bias and variance.

The bias-variance decomposition is a general result in data fitting problems, independent of the learning model adopted, see Fig. 7 for four popular patterns of bias and variance. If the physical model for the quantity x is as before denoted as $f_{\text{phys}}(x)$, and the output generated by x is $y = f_{\text{phys}}(x) + a$, where a is a noise with mean $\mathrm{E}[a] = 0$ and variance $\mathrm{var}[a]$. In addition, the learning model is denoted by $\widehat{f}(x)$ without introducing the learning parameter $\mathbf{w}$. For each testing data $\overline{x}$, the output given by the learning model is consequently $\widehat{f}(\overline{x})$. After some straightforward derivations we could obtain the mean of the square of the difference between the physical model and the learning prediction $\Delta = \mathrm{E}[(f_{\text{phys}}(\overline{x}) + a - \widehat{f}(\overline{x}))^2]$, which characterizes the goodness of the learning model,

$$\begin{aligned}
\Delta =& \mathrm{E}\left[\left(f_{\text{phys}}(\overline{x}) + a - \widehat{f}(\overline{x})\right)^2\right] \\
=& \mathrm{E}\left[f_{\text{phys}}^2(\overline{x}) + a^2 + \widehat{f}^2(\overline{x}) - 2 f_{\text{phys}}(\overline{x})\widehat{f}(\overline{x})\right] \\
=& \mathrm{E}\left[a^2\right] + \mathrm{E}\left[f_{\text{phys}}^2(\overline{x})\right] + \mathrm{E}\left[\widehat{f}^2(\overline{x})\right] - 2\mathrm{E}\left[f_{\text{phys}}(\overline{x})\widehat{f}(\overline{x})\right] \\
=& \mathrm{var}[a] + \left[\mathrm{E}\left[\widehat{f}(\overline{x})\right] - f_{\text{phys}}(\overline{x})\right]^2 + \mathrm{E}\left[\left[\widehat{f}(\overline{x}) - \mathrm{E}\left[\widehat{f}(\overline{x})\right]\right]^2\right], \qquad (3.7)
\end{aligned}$$

the meaning of each term is clear:

(a) The noise $\mathrm{var}[a]$ can't be reduced once the physical model is fixed.

(b) $\mathrm{E}[\widehat{f}(\overline{x})] - f_{\text{phys}}(\overline{x})$ is the bias between the learning model $\widehat{f}$ (characterized by its mean $\mathrm{E}[\widehat{f}(\overline{x})]$) and the physical model f_{phys}.

(c) $\mathrm{E}[[\widehat{f}(\overline{x}) - \mathrm{E}[\widehat{f}(\overline{x})]]^2]$ is the variance of the learning model $\widehat{f}$ on the testing sample.

thus

$$\mathrm{E}\left[\left(f_{\text{phys}}(\overline{x}) + a - \widehat{f}(\overline{x})\right)^2\right] = \mathrm{var}[a] + \left[\text{bias of } \widehat{f}(\overline{x})\right]^2 + \text{variance of } \widehat{f}(\overline{x}). \qquad (3.8)$$

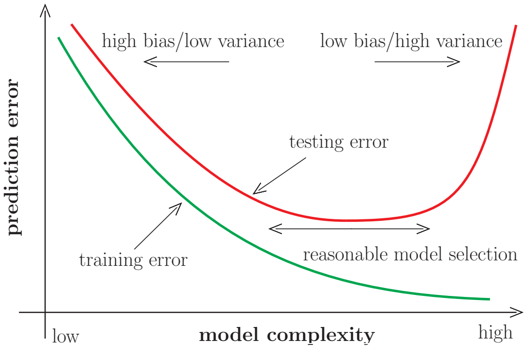


Fig. 8: Sketch of the bias-variance decomposition.

The relation (3.8) is often called the bias-variance decomposition of the learning process, which partially explains that if the bias of the learning is small then the variance is correspondingly large and vice versa, since the noise $\mathrm{var}[a]$ is generally a constant once the physical model is fixed. See Fig. 8 for the sketch of the bias-variance decomposition of the learning model. One can see that as the model complexity increases the training error generally decreases, however the testing error first decreases and then increases once again.[2]

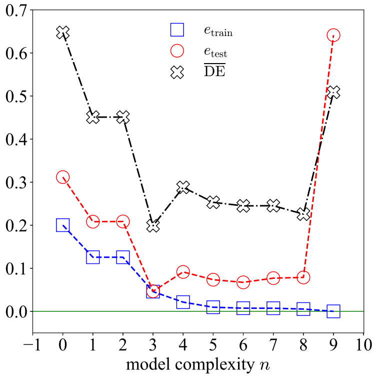


Fig. 9: Training error e_{train}, testing error e_{test} and the integration error $\overline{\text{DE}}$.

In order to characterize the bias-variance decomposition in a more qualitatively manner, we define two errors both based on Eq. (2.1). We already have $m = 10$ training data samples, we define the training error per data sample e_{train} as

$$me_{\text{train}} \equiv \frac{1}{2}\sum_{i=1}^{m}\left[f_{\mathbf{w}^*}(x^{(i)}) - y^{(i)}\right]^2, \qquad (3.9)$$

where the optimized $\mathbf{w}^*$ is used. For $n = 9$, the e_{train} will be zero since the learning model can exactly pass through all the training data points. Besides these already existed training data, one could randomly generate another m' data points (different from the training data) according to $f_{\text{phys}}(x^{(i')}) + a^{(i')}$ where both $x^{(i')}$ and $a^{(i')}$ are random numbers, and define the testing error per data sample as

$$e_{\text{test}} \equiv \frac{1}{m'}\left[\frac{1}{2}\sum_{i=1}^{m'}\left[f_{\mathbf{w}^*}(x^{(i')}) - y^{(i')}\right]^2\right]. \qquad (3.10)$$

It is reasonable to expect that when $n = 9$ the testing error would be larger than that in the case $n = 3$.

Selecting a model with a certain n is called model selection. A very simple scheme to select a reasonable n is to select the learning model with a smallest training error, a smallest testing error, or a total error

[1] D. Wolpert and W. Macready, *No Free Lunch Theorems for Optimization*, IEEE Transactions on Evolutionary Computation **1**, 67 (1997).

[2] In fact, there is no prior that the total error should be decomposed into the variance and the bias. A natural question is that could the bias and the variance be small simultaneously? Problems like these are at the center of modern deep learning theory. For example, in some model studies, the "double descent" for the prediction error is found, indicating that the testing error could be reduced even to be very small in the over-parametrized region. See, e.g., M. Belkin *et al.*, *Reconciling Modern Machine Learning Practice and the Classical Biasvariance Trade-off*, PNAS, **116**, 15849 (2019). Deep understanding on the current neural networks is an important and exciting problem, we have no attempts to review the status on this issue, the following papers, e.g., introduce some relevant improvements, H.W. Lin, M. Tegmark, and D. Rolnick, *Why Does Deep and Cheap LearningWork So Well?*, J. Stat. Phys. **168**, 1223 (2017), here the idea of effective field theories was applied; C. Beny, *Deep learning and the Renormalization Group*, arXiv:1301.3124 (2013), in this work the renormalization group technique was used.

defined as the weighted sum of the training and the testing error, i.e., $e_{\text{total}} = he_{\text{train}} + (1-h)e_{\text{test}}$, where $0 \le h \le 1$. Taking $h = 1/2$ means that the training error and the testing error have the same weights, otherwise they have different weights. Another scheme characterizing the learning model on the prediction power is to calculate the average (overall) deviation between the physical model $f_{\text{phys}}(x)$ and the learning model $f_{\mathbf{w}}(x)$, for our example, we have the following integration error,

$$\overline{\text{DE}} = \int_0^1 \left| f_{\text{phys}}(x) - f_{\mathbf{w}^*}(x)\right| \mathrm{d}x. \tag{3.11}$$

In Fig. 9 the above three errors are shown as functions of the model complexity where $m' = 5$ is adopted. For the current learning problem we find that $n \approx 3 \sim 6$ is a reasonable choice.

IV. REGULARIZATION

As discussed above and shown in Fig. 9 if the model complexity n is selected reasonably, e.g., $n = 3$ or $n = 4$, the learning curve has both low training and testing errors. On the other hand, if n is very small, e.g., $n = 0$, the learning model $f_{\mathbf{w}^*}(x) = w_0^*$ has both large training and testing errors. If n is too large like $n = 9$ the learning model has zero training error. However the testing error is now extremely large, indicating that the model has very poor prediction power although it could pass through the training data perfectly. In addition, in the small n case (e.g., $n = 0$ or $n = 1$), the learning model is very poor in grasping the training data, and we call the situation is under-fitting. On the other hand, in the large n case (e.g., $n = 9$), the learning model is very strong in grasping the training data (but very poor in predicting new samples), and we say the situation is over-fitting. Either under-fitting or over-fitting is bad. The overall performance of the training error and the testing error is characterized by the prediction error.

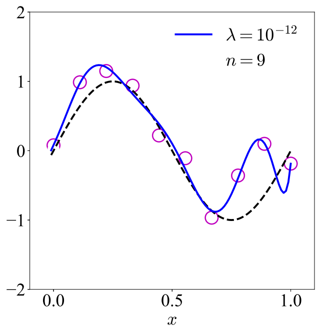


Fig. 10: Effects of the regularization on the polynomial regression.

We introduce the regularization method to deal with the over-fitting problem. Besides the original loss function $2^{-1}\sum_{i=1}^m [f_{\mathbf{w}}(x^{(i)}) - y^{(i)}]^2$, we introduce an extra term $\lambda g(\mathbf{w})$ into it, i.e.,

$$J(\mathbf{w}) = \frac{1}{2}\sum_{i=1}^m \left[f_{\mathbf{w}}(x^{(i)}) - y^{(i)}\right]^2 + \lambda g(\mathbf{w}), \tag{4.1}$$

where $\lambda > 0$ is a parameter put by hand and $g(\mathbf{w})$ is a function of the learning parameter $\mathbf{w}$, which is also positive. We call $\lambda g(\mathbf{w})$ the regularization term to the loss function $J(\mathbf{w})$. There are many different forms of the regularization term and different form has different realistic meaning. Here we adopt the following regularization term,

$$\lambda g(\mathbf{w}) = \frac{1}{2}\lambda \mathbf{w}^{\mathrm{T}}\mathbf{w} = \frac{1}{2}\lambda\left(w_0^2 + w_1^2 + \cdots + w_n^2\right), \tag{4.2}$$

which is called the ℓ-2 regularization term. Another important regularization term is the ℓ-1 form, defined as $\lambda\|\mathbf{w}\|_1$. The ℓ-1 regularization is often used to guarantee the sparsity of the learning model.

The equation determining the parameter $w_0, w_1, \cdots, w_n$ in the presence of the regularization terms could be derived similarly as

$$\begin{pmatrix} \langle 1\rangle & \langle x\rangle & \cdots & \langle x^n\rangle \\ \langle x\rangle & \langle x^2\rangle & \cdots & \langle x^{n+1}\rangle \\ \vdots & \vdots & \ddots & \vdots \\ \langle x^n\rangle & \langle x^{n+1}\rangle & \cdots & \langle x^{2n}\rangle \end{pmatrix}\begin{pmatrix} w_0 \\ w_1 \\ \vdots \\ w_n\end{pmatrix} = \begin{pmatrix}\langle y\rangle \\ \langle xy\rangle \\ \vdots \\ \langle x^n y\rangle\end{pmatrix} - \lambda\begin{pmatrix} w_0 \\ w_1 \\ \vdots \\ w_n\end{pmatrix}, \tag{4.3}$$

or $(\mathbf{F} + \lambda\mathbf{1})\mathbf{w} = \mathbf{G}$, where $\mathbf{1}$ is the $(n+1)\times(n+1)$ unit matrix.

In Fig. 10 the effects of the regularization term with $\lambda = 10^{-12}$ are shown based on the 9th-order polynomial learning model. It is found that a tiny $\lambda = 10^{-12}$ essentially makes the overall behavior of the learning curve more regular and smoother, and one can expect as λ increases the curve becomes much smoother. It should be pointed out once again that the regularization term $\lambda g(\mathbf{w})$ is not encapsulated in the data points themselves, instead it is put by hand. In this sense the λ-term more or less characterizes people's belief in the data, i.e., if λ is small the original data points are treated more importantly, and on the other hand, if λ is large it means that one does not believe the original data points. A natural question is what will happen in this large λ limit. For example, by taking $\lambda = 1$ the w_j^*'s in the $n = 9$ model are found to be $0.44, -0.28, -0.31, -0.20, -0.10, -0.02, 0.03, 0.06, 0.08$ and 0.09, respectively. The case $\lambda = \infty$ corresponds to the situation that one totally un-believe the original data points, since now

$$\lambda g(\mathbf{w}) \gg \frac{1}{2}\sum_{i=1}^m \left[f_{\mathbf{w}}(x^{(i)}) - y^{(i)}\right]^2, \tag{4.4}$$

and the learning model is approximately reduced to be $f_{\mathbf{w}}(x) = \lambda g(\mathbf{w})$ and if $g(\mathbf{w}) = \mathbf{w}^{\mathrm{T}}\mathbf{w}$ then the optimal $\mathbf{w}^*$ is essentially $\mathbf{0}$ without doubt. One can easily find that as $\log\lambda \to -\infty$, i.e., the limit without using the regularization term, the training loss tends to zero since when $n = 9$ the learning model could perfectly pass through all the data. In the meanwhile the testing loss is fixed at a nonzero constant. Under the opposite limit named $\log\lambda \to \infty$, the learning model naturally becomes zero, and consequently either the training loss or the testing loss is fixed at constants which are determined by the simulated data points used.

V. NORMAL EQUATION

Let's discuss the polynomial learning model in some more details. In our polynomial learning model, $f_{\mathbf{w}}(x) = w_0 + w_1 x + w_2 x^2 + \cdots + w_n x^n$, which could be rewritten in the form[3]

$$f_{\mathbf{w}}(x) = \mathbf{w}^{\mathrm{T}}\vec{\phi}(x) = \vec{\phi}^{\mathrm{T}}(x)\mathbf{w} \tag{5.1}$$

with $\mathbf{w} = (w_0, w_1, w_2, \cdots, w_n)^{\mathrm{T}}, \vec{\phi}(x) = (1, x, x^2, \cdots, x^n)^{\mathrm{T}}$, i.e., $\mathbf{w} \in \mathrm{R}^{(n+1)\times 1} \equiv \mathrm{R}^{n+1}$ is a column vector (column vector is thin and tall) and its transpose $\mathbf{w}^{\mathrm{T}} \in \mathrm{R}^{1\times(n+1)}$ is a row vector (row vector is fat and short). The jth component of the vector $\vec{\phi}$ is x^j with $j = 0 \sim n$. Denoting $\phi_j(x) = x^j$, then the function $\vec{\phi}(x)$ could be written as

$$\vec{\phi}(x) = \left(\phi_0(x), \phi_1(x), \cdots, \phi_n(x)\right)^{\mathrm{T}}, \quad \phi_j(x) = x^j, \quad j = 0 \sim n. \tag{5.2}$$

In the polynomial learning model, each component $\phi_j(x)$ takes the form x^j. However, in more general situations, the component $\phi_j(x)$ is free to take other forms, e.g., $\phi_j(x) = \sin(jtx)$ with t a constant. We call the functions $\phi_j(x)$'s the basis functions.

Adopting the basis function representation, the loss function without regularization term is given by

$$J(\mathbf{w}) = \frac{1}{2}\sum_{i=1}^m \left[f_{\mathbf{w}}(x^{(i)}) - y^{(i)}\right]^2, \quad f_{\mathbf{w}}(x^{(i)}) = \mathbf{w}^{\mathrm{T}}\vec{\phi}(x^{(i)}). \tag{5.3}$$

By introducing the matrix $\vec{\Phi} \in \mathrm{R}^{m\times(n+1)}$ as follows,

$$\vec{\Phi} = \begin{pmatrix} 1 & \phi_1(x^{(1)}) & \cdots & \phi_n(x^{(1)}) \\ 1 & \phi_1(x^{(2)}) & \cdots & \phi_n(x^{(2)}) \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \phi_1(x^{(m)}) & \cdots & \phi_n(x^{(m)})\end{pmatrix}, \tag{5.4}$$

[3] One has $\mathbf{a}^{\mathrm{T}}\mathbf{b} = \mathbf{b}^{\mathrm{T}}\mathbf{a}$ since this quantity is a scalar, i.e., $\sum_{i=1}^d a_i b_i$ with d the dimension of the vector $\mathbf{a}$ or $\mathbf{b}$.

and $\mathbf{y}=(y^{(1)},y^{(2)},\cdots,y^{(m)})^{\mathrm{T}}\in\mathrm{R}^{m}$, we can rewrite the loss function as,

$$J(\mathbf{w})=\frac{1}{2}\left(\vec{\Phi}\mathbf{w}-\mathbf{y}\right)^{\mathrm{T}}\left(\vec{\Phi}\mathbf{w}-\mathbf{y}\right), \tag{5.5}$$

We call the matrix $\vec{\Phi}$ the design matrix with its component given by $\phi_j(x^{(i)})$. The derivative of $J(\mathbf{w})$ with respect to $\mathbf{w}$ is $\partial J(\mathbf{w})/\partial\mathbf{w}=\vec{\Phi}^{\mathrm{T}}\vec{\Phi}\mathbf{w}-\vec{\Phi}^{\mathrm{T}}\mathbf{y}$, then taking $\partial J(\mathbf{w})/\partial\mathbf{w}$ to be zero gives the normal equation,

$$\mathbf{w}^{*}=\left(\vec{\Phi}^{\mathrm{T}}\vec{\Phi}\right)^{-1}\vec{\Phi}^{\mathrm{T}}\mathbf{y}. \tag{5.6}$$

After the regularization term $2^{-1}\lambda\mathbf{w}^{\mathrm{T}}\mathbf{w}$ is included into the loss function $J(\mathbf{w})$, the optimized solution is changed to be $\mathbf{w}^{*}=(\vec{\Phi}^{\mathrm{T}}\vec{\Phi}+\lambda\mathbf{1})^{-1}\vec{\Phi}^{\mathrm{T}}\mathbf{y}$.

EXERCISE 4: Prove the identities, $\partial\mathbf{a}^{\mathrm{T}}\mathbf{b}/\partial\mathbf{a}=\mathbf{b}$ and $\partial\mathbf{a}^{\mathrm{T}}\mathbf{M}\mathbf{a}/\partial\mathbf{a}=\mathbf{a}^{\mathrm{T}}(\mathbf{M}+\mathbf{M}^{\mathrm{T}})$, where $\mathbf{a},\mathbf{b}\in\mathrm{R}^{d\times1}\equiv\mathrm{R}^{d},\mathbf{M}\in\mathrm{R}^{d\times d}$.

It should be point out that the regularization term introduced into the least-squares plays an important role in situations where the matrix $\vec{\Phi}^{\mathrm{T}}\vec{\Phi}$ is singular, i.e., $\det(\vec{\Phi}^{\mathrm{T}}\vec{\Phi})=0$. Choosing a large λ indicates that the original data information is put at the secondary position. Let's give more analysis on the normal equation. Assume that we want to find the minimum of the quadratic objective function $K(\mathbf{w})=2^{-1}\mathbf{w}^{\mathrm{T}}\vec{\Phi}\mathbf{w}-\mathbf{y}^{\mathrm{T}}\mathbf{w}$, the gradient of $K(\mathbf{w})$ is given by $\vec{\Phi}\mathbf{w}-\mathbf{y}$. In order to find the optimal $\mathbf{w}^{*}$ one naturally needs to solve this equation, i.e., $\vec{\Phi}\mathbf{w}=\mathbf{y}$. However due to some reasons, e.g., there exist more equations than unknowns (the number of rows of $\vec{\Phi}$ is larger than that of columns), this equation may have no solutions, i.e., the system is over-determined. It is often the origin of the situation $\det(\vec{\Phi}^{\mathrm{T}}\vec{\Phi})=0$ aforementioned. Hence we can not expect to a solution of $\vec{\Phi}\mathbf{w}=\mathbf{y}$ and may instead try to change the problem to solving the least-squares problem, $\min_{\mathbf{w}}(\vec{\Phi}\mathbf{w}-\mathbf{y})^{\mathrm{T}}(\vec{\Phi}\mathbf{w}-\mathbf{y})$. This objective function is just the $J(\mathbf{w})$, and the solution is given by that of the normal equation.

Clarifying the connection between the normal equation and the loss function $K(\mathbf{w})$ is useful for understanding the important role played by the design matrix. Since $\phi_0=1$, without loosing generality the design matrix could be written as

$$\vec{\Phi}\sim\begin{pmatrix}\phi_1(x^{(1)}) & \phi_2(x^{(1)}) & \cdots & \phi_n(x^{(1)})\\ \phi_1(x^{(2)}) & \phi_2(x^{(2)}) & \cdots & \phi_n(x^{(2)})\\ \vdots & \vdots & \ddots & \vdots\\ \phi_1(x^{(m)}) & \phi_2(x^{(m)}) & \cdots & \phi_n(x^{(m)})\end{pmatrix}, \tag{5.7}$$

where each column forms a vector $\vec{\varphi}_j=(\phi_j(x^{(1)}),\cdots,\phi_j(x^{(m)}))^{\mathrm{T}}\in\mathrm{R}^{m}$ with $j=1\sim n$ (the difference between n and $n+1$ is essentially irrelevant for the following discussion). On the other hand, each row $\vec{\phi}(x^{(i)})$ has dimension n. Under the assumption that the model complexity n is smaller than the data number m, the n-vector $\vec{\phi}(x^{(i)})$ may span a sub-space S with dimension m. Denote $\mathbf{t}$ as an m-vector with its ith component given by $f_{\mathbf{w}}(x^{(i)})$, i.e., $\mathbf{t}=(f_{\mathbf{w}}(x^{(1)}),\cdots,f_{\mathbf{w}}(x^{(m)}))^{\mathrm{T}}$. Since the vector $\mathbf{t}$ is some linear combination of the basis $\vec{\varphi}_j$, it could be at any point in the n-dimensional space. Under these considerations, the loss function $J(\mathbf{w})\sim\sum_{i=1}^{m}[y^{(i)}-f_{\mathbf{w}}(x^{(i)})]^2=(y^{(1)}-t_1)^2+\cdots+(y^{(m)}-t_m)^2$ is the Euclidean distance between $\mathbf{y}$ and $\mathbf{t}$. The least-squares searching for $\mathbf{w}$ is consequently to find a vector $\mathbf{t}$ in the sub-space S in order to make the distance between $\mathbf{t}$ and $\mathbf{y}$ be smallest.

It is useful to notice that the dimension of the matrix $\vec{\Phi}^{\mathrm{T}}\vec{\Phi}$ is $n+1$, which may possibly be far smaller than the data number m, making the solving of the normal equation possible. It of course should depend on the algorithms like the gradient descent to search the optimal parameter if the model complexity n is a very large number which hinders the direct inverse of the matrix $\vec{\Phi}^{\mathrm{T}}\vec{\Phi}$. In this case, one just uses the information of the gradient of $J(\mathbf{w})$, namely, $\mathbf{w}\leftarrow\mathbf{w}-\epsilon(\vec{\Phi}^{\mathrm{T}}\vec{\Phi}\mathbf{w}-\vec{\Phi}^{\mathrm{T}}\mathbf{y})$, to update the learning parameter $\mathbf{w}$. Here ϵ is the learning rate or the step size of the gradient descent search. As we study the learning problems in this lecture based on the polynomial model or the basis function has the form of x^j, based on which there exists the closed form for the optimal parameter, i.e., the one given by the normal equation.

In more general situations in which the basis function takes other forms, there exists no closed form for the learning parameter. In these situations one should necessary use the optimization algorithms to do the search. The search is composed of the following consecutive steps: (a) Firstly initialize the learning parameter $\mathbf{w}$; (b) Randomly select a data sample $(x^{(i)},y^{(i)})$; (c) Update the learning parameter according to $\mathbf{w}\leftarrow\mathbf{w}-\epsilon_i\nabla J^{(i)}(\mathbf{w})$, where the $\nabla J^{(i)}(\mathbf{w})$ is the gradient associated with the selected data sample, i.e., $\nabla J^{(i)}(\mathbf{w})=-\vec{\phi}(x^{(i)})(y^{(i)}-f_{\mathbf{w}}(x^{(i)}))$ where the n-vector $\vec{\phi}$ is constructed from the row of the design matrix; and (d) recursively do the search to the fulfill termination condition. Here one selects the data sample in a stochastic manner in order to reduce the calculation task in the gradient of the loss function at each step. We call this gradient descent the stochastic gradient descent (SGD), which plays a central role in modern large-scale optimization problems.

VI. BAYESIAN CONSIDERATION: EXAMPLES

We assume there are two random events denoted by A and B. Denoting $P(A|B)$ the probability of A under the condition that the random event B occurs, one calls $P(A|B)$ the conditional probability of A under B (the conditional probability density could be defined in a similar manner). Consequently, the joint probability of A and B, i.e., the random event A and B occur simultaneously, is obtained via

$$P(A,B)=P(A|B)P(B). \tag{6.1}$$

It is obvious that the joint probability could be obtained as $P(A,B)=P(B|A)P(A)$, i.e., $P(A,B)$ is the product of $P(B|A)$ and $P(A)$ with the former the conditional probability of B under A. Consequently, one obtains

$$P(A|B)=\frac{P(B|A)P(A)}{P(B)}, \tag{6.2}$$

by equating the two formulae for $P(A,B)$, (6.2) is called the Bayes' theorem. If the random events A and B are independent with each other in the sense the occurrence of B does not affect that of event A, i.e., $P(A|B)=P(A)$ or $P(B|A)=P(B)$, one has $P(A,B)=P(A)P(B)$. In this case the two events are independent with each other.

We write down the Bayes' theorem in the following form,

$$p(\mathbf{w}|\{\text{data}\})=\frac{p(\{\text{data}\}|\mathbf{w})p(\mathbf{w})}{p(\{\text{data}\})},\quad\text{or,}\quad p(\mathbf{w}|\mathcal{D})=\frac{p(\mathcal{D}|\mathbf{w})p(\mathbf{w})}{p(\mathcal{D})}, \tag{6.3}$$

where $\mathcal{D}$ or equivalently $\{\text{data}\}$ is called the data, which plays a central role in Bayesian analysis. Moreover, $\mathbf{w}$ is a parameter (or a set of parameters) or a hypothesis to be estimated/investigated. For instance, in the linear regression problem, it is the parameter appeared in the fitting function, $f_{\mathbf{w}}(x)=w_0+w_1x+w_2x^2+\cdots+w_nx^n=\sum_{j=0}^{n}w_jx^j$, i.e., $\mathbf{w}=(w_0,w_1,w_2,\cdots,w_n)^{\mathrm{T}}$. In addition, $p(\mathbf{w}|\{\text{data}\})$ is the posterior distribution for the parameter $\mathbf{w}$, i.e., it is the probability density distribution (pdf) for the parameter $\mathbf{w}$ after observing the data, representing people's understanding on the parameter $\mathbf{w}$ combining with the data. On the other hand, the $p(\mathbf{w})$ is the prior for the parameter $\mathbf{w}$, i.e., it is the probability (density distribution) without any data input. The prior plays a similar role as the posterior, however it may also reflect people's own understanding on the parameter $\mathbf{w}$. Furthermore, $p(\{\text{data}\}|\mathbf{w})$ is called the likelihood function, it characterizes the corresponding probability for the data with different parameters $\mathbf{w}$. It is necessary to point out that the likelihood is not a distribution function for the parameter $\mathbf{w}$, indicating that the integration of the likelihood over $\mathbf{w}$ does not give the result 1. Finally, the $p(\{\text{data}\})$ is a normalization constant guaranteeing the normalization of the posterior (called the evidence), i.e., $p(\{\text{data}\})=\int p(\{\text{data}\}|\mathbf{w})p(\mathbf{w})\mathrm{d}\mathbf{w}$. The normalization constant is not a function of the parameter $\mathbf{w}$, and could be safely omitted in analyzing problems involving the parameter $\mathbf{w}$. Consequently, the Bayes' theorem could be rewritten in the direct form, posterior ~ likelihood × prior.

Let us use an example to fix the basic idea of the Bayesian analysis. In this example one tries to check whether the given coin is fair or not.

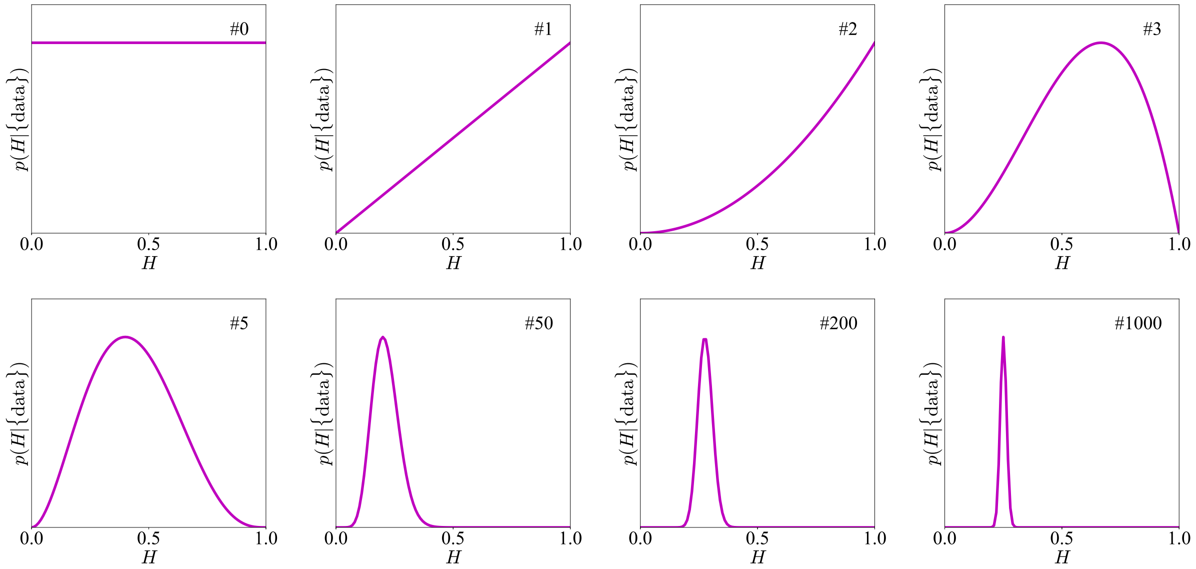


Fig. 11: Bayesian analysis for the example on throwing coins.

If it is fair that after a large number of throwings, the probability (or the frequency) of the head or the tail is roughly one half. Here we denote the event that the throwing result is the head as H. Naturally the H could take any value between 0 and 1, i.e., $0 \le H \le 1$. If $H = 0$, it strongly indicates that the coin has two tails and no head. On the other hand if the H is found to be 1, the coin has two heads and no tail. Similarly if the H takes the value, e.g., 0.2, then it is believed that the coin is biased towards the tail. Before throwing the coin, in fact one knows nothing about the coin, and in this situation one could assume the following prior,

$$p(H) = \left\{ \begin{array}{ll} 1, & 0 \le H \le 1, \\ 0, & \text{otherwise}. \end{array} \right. \tag{6.4}$$

Actually this prior is some unrealistic in the sense only two outcomes are possible in the experiment. Or one could artificially treat the coin has two heads and no tail and consequently $p(H) = 1$ if $H = 1$ and zero otherwise in this case. Actually the effects of the prior on the hypothesis H decays as the number of experiments increases. Next we need to specify the likelihood appeared in the Bayesian analysis. For the current example on throwing a coin, for example, if a total number throwings is n and the probability that the head appears is denoted as r, the likelihood could be described using the binomial distribution, i.e., $p(\{\text{data}\}|H) \sim H^r(1-H)^{n-r}$, where the constant in front of the likelihood originated from the binomial distribution is omitted here for simplicity. Consequently, the posterior is given as $p(H|\{\text{data}\}) \sim p(\{\text{data}\}|H) \sim H^r(1-H)^{n-r}$, and the meaning of it is very obvious: As the number of throwing the coin increases the posterior is updated according to each result of the throwing, which is used to explore the parameter H.

EXERCISE 5: What is mode of the posterior $p(H|\{\text{data}\})$? Denote it as H_{m}, calculate the $\mathrm{d}^2p(H|\{\text{data}\})/\mathrm{d}H^2$ at H_{m}.

EXERCISE 6: Prove the two relations $\mathrm{E}[w] = \mathrm{E}[\mathrm{E}[w|x]]$ and $\mathrm{var}[w] = \mathrm{E}[\mathrm{var}[w|x]] + \mathrm{var}[\mathrm{E}[w|x]]$.

The relevant Bayesian analysis for the coin throwing example is shown in Fig. 11, let us explain the figure in some details. Before the first throwing since we know nothing about the coin (whether it is fair or not, or whether it has two heads or two tails, etc.), the probability for the outcome H is assumed to be a constant, i.e., anything on the H has equal probability. After obtaining the first outcome (here it is the head), one strengthens the confidence that the coin is head-biased while weakens the confidence that the coin has other (unrealistic) values, see the second panel shown in Fig. 11. Moreover, after the first outcome of the head, it is certainly that there is no probabilities that the coin has two tails, and consequently the probability for $H = 0$ descends to zero immediately. The posterior after the first outcome is proportional to H. Now if the second throwing still gives a head as the outcome, then the posterior becomes proportional to H^2, and the confidence on the head is strengthened once again, see the third panel of the figure. The difference between the third panel and the second panel is that although the overall shape for the steps is similar, the posterior in the third panel favors a large H above 0.5. Specifically, since the posterior after one (two) head(s) is $2H_0$ $(3H_0^2)$ according to $p(H_0|\text{data}) = H_0/\int_0^1 \mathrm{d}H p(H|\text{data})$, where H_0 is a certain number between 0 and 1, one finds that for, e.g., $H_0 = 5/6$, $p(5/6|\text{one head}) = 5/3 < p(5/6|\text{two heads}) = 25/12$, and the posteriors in these two steps are equal at $H_0 = 2/3$ and the one in the third panel is greater than this in the second panel for $H_0 > 2/3$. Furthermore, if the third throwing gives a tail, the posterior is then proportional to $(1-H)H^2$, etc. After the outcome of a tail, the probability that the coin has two heads suddenly becomes zeros as shown in the fourth panel of the figure. However the most possibility (i.e., the mode of the posterior) is still larger than 0.5 since now there are essentially two heads over one tail. The mode (at 2/3) is easily to found from the posterior ($\sim (1-H)H^2$). By throwing more and more times one could obtain more and more accurate information on the fairness of the coin, see the remaining panels in Fig. 11. It is essentially to point out at this time that in the Bayesian analysis all the inference on the hypothesis H is based on the observations/experiments. For instance, after a large number of throwings, the final inference on the H in the current experiment is found to be around 0.25, i.e., the coin is unfair and is tail-biased. Different experiments may lead to very different results, and in this sense the Bayesian analysis is experiment-based, and it is this nature the Bayesian analysis is widely adopted in experiment-guiding subjects, e.g., in astrophysics and cosmology, and in medical science. Another interesting point shown in Fig. 11 is that as the number of throwings increases the posterior tends to concentrate at the probable value, i.e., the uncertainties/fluctuations around the probable value decrease, indicating the inference/prediction becomes more and more accurate. Finally if one uses a different prior for the H here, the posterior will become similar as Fig. 11, demonstrating the effects of the prior are floating by the data.

VII. MAXIMUM LIKELIHOOD ESTIMATION

If the parameter $\mathbf{w}$ is obtained by taking the maximum value of the likelihood, we call the relevant method the maximum likelihood (ML) estimation. In order to make the discussion transparent and simple here we adopt that the data is generated from an independently identity distribution (IID), e.g., by some Gaussian distribution, i.e., $x^{(i)} \sim \mathcal{N}(\mu, \sigma^2)$

with $i = 1 \sim m$. The aim of the current discussion is to estimate the mean μ and the variance σ^2 of the Gaussian distribution via these data. One collectively denotes these data as $\mathbf{X} = (x^{(1)}, x^{(2)}, \cdots, x^{(m)})^{\mathrm{T}}$, and since the data samples are IID distributed, one naturally has the relation $p(\mathbf{X}|\mu,\sigma^2) = \prod_{i=1}^m \mathcal{N}(x^{(i)}|\mu,\sigma^2)$, it is the function of the μ and σ^2 to be estimated, and thus the likelihood. Maximizing the likelihood is equivalent to maximizing the log-likelihood since the logarithmic function is monotonic,

$$\log p(\mathbf{X}|\mu,\sigma^2) = -\frac{1}{2\sigma^2}\sum_{i=1}^m (x^{(i)} - \mu)^2 - \frac{1}{2} m \log \sigma^2 - \frac{1}{2} m \log(2\pi). \tag{7.1}$$

By calculating the derivatives of $\log p(\mathbf{X}|\mu,\sigma^2)$ with respect to μ and σ^2 and then taking them to be zero leads to

$$\widehat{\mu}_{\mathrm{ML}} = \frac{1}{m}\sum_{i=1}^m x^{(i)}, \quad \widehat{\sigma}^2_{\mathrm{ML}} = \frac{1}{m}\sum_{i=1}^m \left(x^{(i)} - \widehat{\mu}_{\mathrm{ML}}\right)^2, \tag{7.2}$$

here $\widehat{\mu}_{\mathrm{ML}}$ and $\widehat{\sigma}^2_{\mathrm{ML}}$ are the sample mean and sample variance. We can compute the mean of these sampled quantities. Noticing that when $i \neq j$, we have $\mathrm{E}[x^{(i)}x^{(j)}] = \mathrm{E}[x^{(i)}]\mathrm{E}[x^{(j)}] = \mu^2$, and when $i = j$, $\mathrm{E}[x^{(i)}x^{(j)}] = \mathrm{E}[x^{(i),2}] = \mathrm{var}[x] + \mathrm{E}^2[x^{(i)}] = \mu^2 + \sigma^2$, and thus $\mathrm{E}[x^{(i)}x^{(j)}] = \mu^2 + \delta_{ij}\sigma^2$. Consequently, $\mathrm{E}[\widehat{\mu}_{\mathrm{ML}}] = \mu$, and similarly

$$\begin{aligned}
\mathrm{E}[\widehat{\sigma}^2_{\mathrm{ML}}] \equiv \left\langle \widehat{\sigma}^2_{\mathrm{ML}} \right\rangle &= \left\langle \frac{1}{m}\sum_{j=1}^m \left(x^{(j)} - \frac{1}{m}\sum_{i=1}^m x^{(i)} \right)^2 \right\rangle \\
&= \frac{1}{m}\sum_{j=1}^m \left[\left\langle x^{(i),2} \right\rangle - \frac{2}{m}\sum_{i=1}^m \left\langle x^{(i)}x^{(j)} \right\rangle + \frac{1}{m^2}\sum_{i,i'=1}^m \left\langle x^{(i)}x^{(i')} \right\rangle \right] \\
&= \frac{1}{m}\sum_{j=1}^m \left[\mu^2 + \sigma^2 - \frac{2}{m}\left(m\mu^2 + \sigma^2\right) + \frac{1}{m^2}\sum_{i=1}^m \left(m\mu^2 + \sigma^2\right) \right] \\
&= \frac{1}{m}\sum_{j=1}^m \left(1 - \frac{1}{m}\right)\sigma^2 = \left(1 - \frac{1}{m}\right)\sigma^2.
\end{aligned} \tag{7.3}$$

These results show that the mean of the mean is the same as the mean of the Gaussian distribution, while the mean of the variance is different from the distribution variance by a factor $1 - 1/m$, i.e., the variance is under-estimated. The estimation (here the ML) is thus biased.

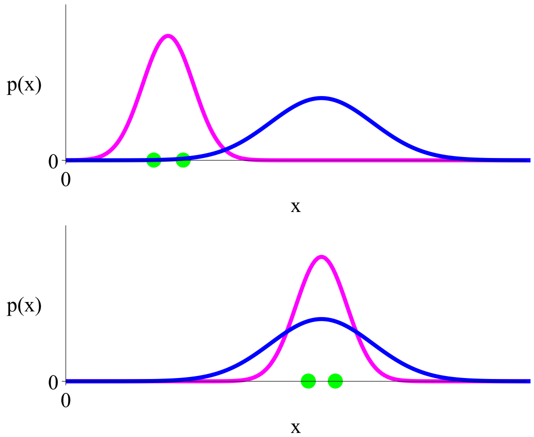


Fig. 12: Bias for the variance, here blue lines correspond to the real Gaussian distribution and the magenta lines corresponds to the biased estimation.

If one introduces a new variance estimator, i.e.,

$$\widetilde{\sigma}^2 = \frac{m}{m-1}\widehat{\sigma}^2_{\mathrm{ML}} = \frac{1}{m-1}\sum_{i=1}^m \left(x^{(i)} - \widehat{\mu}_{\mathrm{ML}}\right)^2, \tag{7.4}$$

then the new $\widetilde{\sigma}^2$ is unbiased. This is the reason why we defined the SEM in (1.8) by including a factor $(k-1)^{-1}$. Fig. 12 gives the sketch of the bias of the variance estimation. Here two data points are generated IID from the Gaussian distribution in each set and the distribution mean of the original distribution could be reconstructed by these data points. However the variance is under-estimated, i.e., the width of the magenta lines is systematically narrower than that of the blue lines. If the data number becomes large, the biasness weakens eventually.

EXERCISE 7: Is the estimation based on d_{x} or $d_{\perp}$ (equivalently J_{x} or $J_{\perp}$) in EXERCISE 1 biased or unbiased? Investigate by simulations.

VIII. LINEAR CURVE FITTING REVISITED

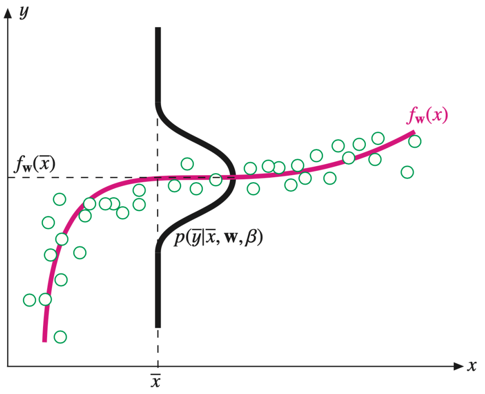

y
$f_{\mathrm{w}}(x)$
$f_{\mathrm{w}}(\overline{x})$
$p(\overline{y}|\overline{x},\mathbf{w},\beta)$
x
$\overline{x}$

Fig. 13: Probabilistic interpretation of the curve fitting problem.

Now we use the viewpoint of probability and the ML consideration to review the curve fitting problem. In this problem one needs to calculate some type of the error function and minimize this error function and in the mean while the model could predict a reasonable value for the output $\overline{y}$ as a new input $\overline{x}$ is generated. In order to arrive this aim, one assets a Gaussian distribution for the output y associated with each input x, and the mean of this Gaussian distribution is given by the learning model for the data $f_{\mathbf{w}}(x)$. Consequently,

$$p(\overline{y}|\overline{x},\mathbf{w},\beta) = \mathcal{N}(\overline{y}|f_{\mathbf{w}}(\overline{x}),\beta^{-1}), \tag{8.1}$$

here the β is called the precision parameter which is just the inverse of the variance, see the sketch of Fig. 13 for its probabilistic meaning.

We use the training data $(x^{(i)}, y^{(i)})$, or the $(\mathbf{X},\mathbf{y})$ to learn the model parameters $\mathbf{w}$ and β via the ML method. Since all the data samples are obtained IID from the distribution (8.1), the likelihood function for the parameters $\mathbf{w}$ and β is given by $p(\mathbf{y}|\mathbf{X},\mathbf{w},\beta) = \prod_{i=1}^m \mathcal{N}(y^{(i)}|f_{\mathbf{w}}(x^{(i)}),\beta^{-1})$, with the logarithmic of which as,

$$\log p(\mathbf{y}|\mathbf{X},\mathbf{w},\beta) = -\frac{\beta}{2}\sum_{i=1}^m \left[f_{\mathbf{w}}(x^{(i)}) - y^{(i)}\right]^2 + \frac{m\log\beta}{2}, \tag{8.2}$$

where the irrelevant term $-(m/2)\log(2\pi)$ was omitted. Maximizing the likelihood is equivalent to minimizing its opposite. Since in the curve fitting problems one also needs to minimize some type of the error function, the negative log-likelihood is in some sense equivalent to the error function. We write out the negative log-likelihood function as

$$J(\mathbf{w}) \sim -\log p(\mathbf{y}|\mathbf{X},\mathbf{w},\beta)\sum_{i=1}^m \left[f_{\mathbf{w}}(x^{(i)}) - y^{(i)}\right]^2 - \frac{1}{2}m\log\beta. \tag{8.3}$$

It demonstrates the squared error function (with least-squares) originates from maximizing the likelihood function under the Gaussian.

Calculating the derivative of $\log p(\mathbf{y}|\mathbf{X},\mathbf{w},\beta)$ with respect to $\mathbf{w}$ and taking it to be zero, one obtains the ML estimation for $\mathbf{w}$ as $\widehat{\mathbf{w}}_{\mathrm{ML}}$. Similarly taking derivative of $\log p(\mathbf{y}|\mathbf{X},\mathbf{w},\beta)$ with respect to β, one obtains

$$\frac{1}{\widehat{\beta}_{\mathrm{ML}}} = \frac{1}{m}\sum_{i=1}^m \left[f_{\widehat{\mathbf{w}}_{\mathrm{ML}}}(x^{(i)}) - y^{(i)}\right]^2. \tag{8.4}$$

It is necessary to point out that one could first obtain the estimate $\widehat{\mathbf{w}}_{\mathrm{ML}}$ and then the $\widehat{\beta}_{\mathrm{ML}}$ since in the Gaussian model these two parameters are decoupled. In general situations one needs to take derivatives of the parameters simultaneously. After obtaining the $\widehat{\mathbf{w}}_{\mathrm{ML}}$ and $\widehat{\beta}_{\mathrm{ML}}$, one could estimate the output $\overline{y}$ and the result is given by putting the ML estimation into (8.1), i.e., $p(\overline{y}|\overline{x},\widehat{\mathbf{w}}_{\mathrm{ML}},\widehat{\beta}_{\mathrm{ML}}) = \mathcal{N}(\overline{y}|f_{\widehat{\mathbf{w}}_{\mathrm{ML}}}(\overline{x}),\widehat{\beta}^{-1}_{\mathrm{ML}})$.

If one introduces the prior for the parameter $\mathbf{w}$, e.g., another Gaussian distribution like

$$p(\mathbf{w}|\alpha) = \mathcal{N}(\mathbf{w}|\mathbf{0}, \alpha^{-1}\mathbf{1}) = \left(\frac{\alpha}{2\pi}\right)^{(n+1)/2} \exp\left(-\frac{\alpha}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w}\right), \tag{8.5}$$

where α is the precision parameter for the prior and $n+1$ is, e.g., the order of the polynomial for nonlinear fitting. The posterior for $\mathbf{w}$ is obtained as the product of the likelihood and the prior according Bayes' theorem, i.e.,

$$p(\mathbf{w}|\mathbf{X}, \mathbf{y}, \alpha, \beta) \sim p(\mathbf{y}|\mathbf{X}, \mathbf{w}, \beta) p(\mathbf{w}|\alpha). \tag{8.6}$$

In order to obtain the optimal estimation for $\mathbf{w}$ one should find the maximum (local or global) of the posterior and the process is called the maximum-a-poster (MAP) optimization. Specifically, the MAP estimation is equivalent to the minimum of the following function,

$$\frac{\beta}{2}\left[\sum_{i=1}^{m}\left[f_{\mathbf{w}}(x^{(i)}) - y^{(i)}\right]^2 + \frac{\alpha}{\beta}\mathbf{w}^{\mathrm{T}}\mathbf{w}\right]. \tag{8.7}$$

It is the (squared) error function including the regularization term characterized by the coefficient $\lambda = \alpha/\beta$.

In order to give the prediction on $\bar{y}$ one needs to do integration over the parameter $\mathbf{w}$, and the result is given by (here the dependence on α and β is suppressed for simplicity),

$$p(\bar{y}|\bar{x}, \mathbf{X}, \mathbf{y}) = \int p(\bar{y}|\bar{x}, \mathbf{w}) p(\mathbf{w}|\mathbf{X}, \mathbf{y}) \mathrm{d}\mathbf{w}, \tag{8.8}$$

where $p(\bar{y}|\bar{x}, \mathbf{w})$ is given by (8.1) and $p(\mathbf{w}|\mathbf{X}, \mathbf{y})$ by (8.6). For the general nonlinear curve fitting problem these expressions could be wrote in the analytical form, i.e., $p(\bar{y}|\bar{x}, \mathbf{X}, \mathbf{y}) = \mathcal{N}(\bar{y}|m(\bar{x}), s^2(\bar{x}))$, with

$$m(\bar{x}) = \beta\vec{\phi}^{\mathrm{T}}(\bar{x})\mathbf{S}\sum_{i=1}^{m}\vec{\phi}(x^{(i)})y^{(i)}, \quad s^2(\bar{x}) = \beta^{-1} + \vec{\phi}^{\mathrm{T}}(\bar{x})\mathbf{S}\vec{\phi}(\bar{x}), \tag{8.9}$$

$$\mathbf{S}^{-1} = \alpha\mathbf{1} + \beta\sum_{i=1}^{m}\vec{\phi}(x^{(i)})\vec{\phi}^{\mathrm{T}}(x^{(i)}), \tag{8.10}$$

where the components of the vector $\vec{\phi}(x)$ is given by $\phi_j(x) = x^j$.

The likelihood for the learning parameter $\mathbf{w}$ is given by

$$p(\mathbf{y}|\mathbf{X}, \mathbf{w}, \beta) = \prod_{i=1}^{m}\mathcal{N}(y^{(i)}|\mathbf{w}^{\mathrm{T}}\vec{\phi}(\mathbf{x}^{(i)}), \beta^{-1}), \tag{8.11}$$

where $\mathbf{X}$ and $\mathbf{y}$ are the input and output datasets, respectively. In order to make the discussion be clear and simplicity, the prior of the parameter $\mathbf{w}$ is also adopted as a Gaussian, i.e., $p(\mathbf{w}) = \mathcal{N}(\mathbf{w}|\mathbf{m}_0, \mathbf{S}_0)$. Using the matrix splitting and transformation formulas, one obtains the posterior for the parameter $\mathbf{w}$, i.e., $p(\mathbf{w}|\mathbf{y}) = \mathcal{N}(\mathbf{w}|\mathbf{m}_m, \mathbf{S}_m)$, with

$$\mathbf{m}_m = \mathbf{S}_m\left(\mathbf{S}_0^{-1}\mathbf{m}_0 + \beta\vec{\Phi}^{\mathrm{T}}\mathbf{y}\right), \quad \mathbf{S}_m^{-1} = \mathbf{S}_0^{-1} + \beta\vec{\Phi}^{\mathrm{T}}\vec{\Phi}. \tag{8.12}$$

Moreover, the precision parameter β here is assumed known in prior. Furthermore, by introducing the precision parameter α for the prior namely $\mathbf{S}_0 = \alpha^{-1}\mathbf{1}$, one has

$$\mathbf{m}_m = \mathbf{S}_m\left(\mathbf{S}_0^{-1}\mathbf{m}_0 + \beta\vec{\Phi}^{\mathrm{T}}\mathbf{y}\right) \to \beta\mathbf{S}_m\vec{\Phi}^{\mathrm{T}}\mathbf{y} = \left(\vec{\Phi}^{\mathrm{T}}\vec{\Phi}\right)^{-1}\vec{\Phi}^{\mathrm{T}}\mathbf{y}, \tag{8.13}$$

the last relation is under the assumption $\alpha \to 0$, i.e., the prior has little information on the parameter $\mathbf{w}$. This is the familiar least-squares solution, i.e., the ML estimation under the Gaussian distribution, see (8.3). In addition, if there is no data ($m = 0$) the posterior naturally reduces to the prior. Finally, each time a new data sample arrives the posterior should be updated by absorbing the effects of the current data sample, and this posterior pdf acts as the prior before the next new data sample arrives. In this sense, the parameter $\mathbf{w}$ is updated step by step which is one of the main features of the Bayesian inference. In the following, the prior in the calculation for the $\mathbf{w}$ is taken as $p(\mathbf{w}|\alpha) = \mathcal{N}(\mathbf{w}|\mathbf{0}, \alpha^{-1}\mathbf{1})$, and consequently $\mathbf{m}_m = \beta\mathbf{S}_m\vec{\Phi}^{\mathrm{T}}\mathbf{y}, \mathbf{S}_m^{-1} = \alpha\mathbf{1} + \beta\vec{\Phi}^{\mathrm{T}}\vec{\Phi}$, and the logarithm of the posterior is the conventional least-squares error including the regularization terms,

$$\log p(\mathbf{w}|\mathbf{y}) = -\frac{\beta}{2}\sum_{i=1}^{m}\left[f_{\mathbf{w}}(\mathbf{x}^{(i)}) - y^{(i)}\right]^2 - \frac{\alpha}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w} + \text{terms independent of } \mathbf{w}, \tag{8.14}$$

where the regularization coefficient $\lambda \leftrightarrow \alpha/\beta$.

As a basic example, we study the Bayesian linear regression problem in more details. The physical model is adopted as $f_{\mathrm{phys}}(x) = w_0' + w_1'x$ where the two ground truth parameters given by $w_0' = -0.2$ and $w_1' = 0.5$. The data is still simulated by the uniform random numbers, i.e., $x^{(i)} \sim \mathrm{Unif}[-1,1]$, the output $y^{(i)}$ is created via the physical model value $f_{\mathrm{phys}}(x^{(i)})$ together with a stochastic fluctuation $a_\delta \sim \mathcal{N}(0, 0.5^2)$, i.e., $y^{(i)} \sim f_{\mathrm{phys}}(x^{(i)}) + a_\delta$. In addition, the learning model is a line $f_{w_0,w_1}(x) = w_0 + w_1x$ with two effective learning parameters w_0 and w_1. The aim of the Bayesian inference is to make inference on these two parameters as data samples generated. Moreover the precision parameter of the likelihood is fixed at the value of $\beta = (1/0.5)^2 = 4$ while that of the prior is fixed as $\alpha = 2.0$, and the calculated results are shown in Fig. 14 using total $m = 50$ total data samples. In the figure the left column is the results for the likelihood function, the middle column for the prior and/or the posterior, and the right column for the data. There are total six linear curves randomly generated from the prior for w_0 and w_1 in the first line of the right panel, and since at the beginning of the simulation there is no data generated these lines are very irregular, i.e., they are totally random. Shift to the second line where the first data sample is generated indicated by the green circle, now the left column of this line is the likelihood for the parameter. By multiplying this likelihood by the prior shown in the first line one obtains the corresponding posterior, i.e., the middle panel shown in the second line. At this point if one generates again six linear curves according to the posterior, one can easily find that these lines are closer to the first data sample already generated. The reason is really simple since the data effects are considered in the Bayesian inference framework. Next after the second data sample is generated which is similar marked as a green circle in the right panel of the third line, the corresponding likelihood is shown in the left panel of this line and the posterior is obtained again by the product of the likelihood and the prior named the posterior of the last step. By generating once again six curves via this posterior one can further find that these curves are even closer to the two data samples. The simulation could be updated data by data. As the number of data samples increases the peak (mode) of the posterior, i.e., the MAP estimation for the learning parameters, becomes more and more sharp, eventually to be consistent with the realistic point (w_0', w_1') indicated in the figure by the magenta cross. With very large m, e.g., $m = 50$ or 100, the six curves approach to the physical model. The above process gives the main features of a typical Bayesian inference on the learning problem, i.e., the learning becomes better and better as more and more data samples arrive.

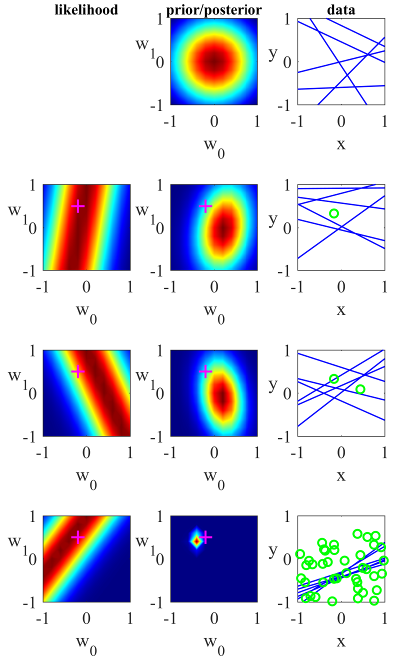


Fig. 14: Bayesian inference for the learning parameters w_0 and w_1 in the linear curve fitting problem with total $m = 50$ data samples.

IX. VARIANCE REDUCTION VIA DATA GENERATION

In many circumstances, not only the estimation on the learning parameter is important, but also the prediction on the output when new data sample is generated. When the new data $\mathbf{x}^{(m+1)}$ arrives the output $y^{(m+1)}$ is updated according to

$$p(y^{(m+1)}|\mathbf{y},\alpha,\beta)=\int p(y^{(m+1)}|\mathbf{x}^{(m+1)},\mathbf{w},\beta)p(\mathbf{w}|\mathbf{y},\alpha,\beta) \tag{9.1}$$

where the dependence on the dataset $\mathbf{X}$ is suppressed here for simplicity. This is the convolution of two Gaussians, one obtains

$$p(y^{(m+1)}|\mathbf{y},\alpha,\beta)=\mathcal{N}(y^{(m+1)}|\mathbf{m}_m^{\mathrm{T}}\vec{\phi}(\mathbf{x}^{(m+1)}),\sigma_m^2(\mathbf{x}^{(m+1)})), \tag{9.2}$$

where $\sigma_m^2(\mathbf{x})=\beta^{-1}+\vec{\phi}^{\mathrm{T}}(\mathbf{x})\mathbf{S}_m\vec{\phi}(\mathbf{x})$. The meaning of $\sigma_m^2(\mathbf{x})$ is very clear: The first term is the noise carried by the data sample itself characterized by the precision parameter β, while the second term is the uncertainty on the learning parameter characterized by the covariance matrix $\mathbf{S}_m$. It is useful to notice that since the noise and the parameter $\mathbf{w}$ is independent with each other, the covariance is additive between them. Moreover, the posterior becomes narrower as the new data sample is obtained, see Fig. 14, indicating $\sigma_{m+1}^2(\mathbf{x})\le\sigma_m^2(\mathbf{x})$. Specifically, the second term approaches to zero under $m\to\infty$ since the width of the posterior of $\mathbf{w}$ approaches to zero, i.e., $\mathbf{S}_m\to 0$, and $\sigma_m^2(\mathbf{x}^{(m+1)})$ is characterized by β.

Since the relation $\sigma_{m+1}^2(\mathbf{x})\le\sigma_m^2(\mathbf{x})$ is extremely important, let's prove it. Firstly let us notice that when the data "$m+1$" is generated, the likelihood is $\sim\mathcal{N}(y^{(m+1)}|f_{\mathbf{w}}(\mathbf{x}^{(m+1)}),\beta^{-1})$, and the posterior is the product of the likelihood and the prior, the relevant exponential factor in the posterior is thus given by

$$(\mathbf{w}-\mathbf{m}_m)^{\mathrm{T}}\mathbf{S}_m^{-1}(\mathbf{w}-\mathbf{m}_m)+\beta\left(y^{(m+1)}-\mathbf{w}^{\mathrm{T}}\vec{\phi}(\mathbf{x}^{(m+1)})\right)^2, \tag{9.3}$$

from which one easily finds

$$\mathbf{S}_{m+1}^{-1}=\mathbf{S}_m^{-1}+\beta\vec{\phi}(\mathbf{x}^{(m+1)})\vec{\phi}^{\mathrm{T}}(\mathbf{x}^{(m+1)}), \tag{9.4}$$

which is similar as the relation $\mathbf{S}_m^{-1}=\mathbf{S}_0^{-1}+\beta\vec{\Phi}^{\mathrm{T}}\vec{\Phi}$ obtained previously, see (8.12). One also obtains the update for the mean of the data namely $\mathbf{m}_{m+1}=\mathbf{S}_{m+1}(\mathbf{S}_m^{-1}\mathbf{m}_m+\beta\vec{\phi}(\mathbf{x}^{(m+1)})y^{(m+1)})$. Consequently, we obtain the following results (where $\vec{\phi}_{m+1}=\vec{\phi}(\mathbf{x}^{(m+1)})$),[4]

$$\mathbf{S}_{m+1}=\mathbf{S}_m-\frac{\beta\mathbf{S}_m\vec{\phi}_{m+1}\vec{\phi}_{m+1}^{\mathrm{T}}\mathbf{S}_m}{1+\beta\vec{\phi}_{m+1}^{\mathrm{T}}\mathbf{S}_m\vec{\phi}_{m+1}}, \tag{9.5}$$

$$\sigma_m^2(\mathbf{x})-\sigma_{m+1}^2(\mathbf{x})=\left\|\vec{\phi}^{\mathrm{T}}(\mathbf{x})\mathbf{S}_m\vec{\phi}(\mathbf{x})\right\|^2\Big/\left(\frac{1}{\beta}+\vec{\phi}_{m+1}^{\mathrm{T}}\mathbf{S}_m\vec{\phi}_{m+1}\right). \tag{9.6}$$

Since the covariance matrix $\mathbf{S}_m$ is positive definite, the above expression is always larger than zero as long as new data is generated, i.e., the variance is always reduced as more and more data samples generate.

EXERCISE 8 :For a d-dimensional Gaussian $\mathbf{x}\sim\mathcal{N}(\mathbf{x}|\vec{\mu},\vec{\Sigma})$, one can decompose $\mathbf{x}$ into two parts as $\mathbf{x}_a$ and $\mathbf{x}_b$, i.e., $\mathbf{x}=(\mathbf{x}_a,\mathbf{x}_b)^{\mathrm{T}}$ where the dimension of $\mathbf{x}_a$ is s. Consequently, the mean could be decomposed as $\vec{\mu}=(\vec{\mu}_a,\vec{\mu}_b)^{\mathrm{T}}$, and similarly for the weight matrix,

$$\vec{\Sigma}=\begin{pmatrix}\vec{\Sigma}_{aa} & \vec{\Sigma}_{ab}\\ \vec{\Sigma}_{ba} & \vec{\Sigma}_{bb}\end{pmatrix}. \tag{9.7}$$

According to the symmetry of the covariance-variance matrix, i.e., $\vec{\Sigma}=\vec{\Sigma}^{\mathrm{T}}$, the matrices $\vec{\Sigma}_{aa}$ and $\vec{\Sigma}_{bb}$ are also symmetric, and moreover $\vec{\Sigma}_{ab}=\vec{\Sigma}_{ba}^{\mathrm{T}}$. The inverse of the covariance matrix is called the precision matrix and is denoted by $\vec{\Lambda}=\vec{\Sigma}^{-1}$ with its component $\vec{\Lambda}_{kk'}$, here $k,k'=a,b$.

(a) The first conclusion is on the conditional probability $p(\mathbf{x}_a|\mathbf{x}_b)$. By denoting the mean and covariance matrix of this distribution as $\vec{\mu}_{a|b}$ and $\vec{\Sigma}_{a|b}$, respectively, and treating the $\mathbf{x}_b$ as the data and only the $\mathbf{x}_a$ as the variable, one obtains by selecting the quadratic terms involving the variable $\mathbf{x}_a$, and from which the expression for the covariance matrix could be obtained, i.e., $\vec{\Sigma}_{a|b}=\vec{\Lambda}_{aa}^{-1}$. Prove the relations, $\vec{\mu}_{a|b}=\vec{\mu}_a+\vec{\Sigma}_{ab}\vec{\Sigma}_{bb}^{-1}\left(\mathbf{x}_b-\vec{\mu}_b\right),\vec{\Sigma}_{a|b}=\vec{\Sigma}_{aa}-\vec{\Sigma}_{ab}\vec{\Sigma}_{bb}^{-1}\vec{\Sigma}_{ba}$.

(b) Assume that $p(\mathbf{x})=\mathcal{N}(\mathbf{x}|\vec{\mu},\vec{\Lambda}^{-1}),p(\mathbf{y}|\mathbf{x})=\mathcal{N}(\mathbf{y}|\mathbf{A}\mathbf{x}+\mathbf{b},\mathbf{L}^{-1})$. Prove $\mathrm{E}[\mathbf{y}]=\mathbf{A}\vec{\mu}+\mathbf{b}\mathrm{cov}[\mathbf{y}]=\mathbf{L}^{-1}+\mathbf{A}\vec{\Lambda}^{-1}\mathbf{A}^{\mathrm{T}}$, and show that $\mathrm{E}[\mathbf{x}|\mathbf{y}]=(\vec{\Lambda}+\mathbf{A}^{\mathrm{T}}\mathbf{L}\mathbf{A})^{-1}[\mathbf{A}^{\mathrm{T}}\mathbf{L}(\mathbf{y}-\mathbf{b})+\vec{\Lambda}\vec{\mu}],\mathrm{cov}[\mathbf{x}|\mathbf{y}]=(\vec{\Lambda}+\mathbf{A}^{\mathrm{T}}\mathbf{L}\mathbf{A})^{-1}$.

X. CORRELATIONS AND ERROR-BARS

We discuss the correlations among certain parameters estimated, e.g., from the maximum a posterior approach. We denote the posterior pdf collectively as $p(\{w_i\}|\mathcal{D})$ where $\mathcal{D}$ is the set of data samples, $\{w_i\}$ is the set of parameters. In addition, we denote the logarithm of the posterior pdf as Q, i.e., $Q(\{w_i\})=\log p(\{w_i\}|\mathcal{D})$. Firstly we investigate the situation where there are two parameters a and b in the problem, and denote the point corresponding to the maximum of the posterior as (a_0,b_0), which is determined conventionally via $[\partial Q/\partial a]_{a_0,b_0}=0$ and $[\partial Q/\partial b]_{a_0,b_0}=0$. Expand the logarithmic posterior around (a_0,b_0) to second order,

$$Q(a,b)\approx Q_0+\frac{1}{2}\Phi,\quad \Phi=\begin{pmatrix}\delta a & \delta b\end{pmatrix}\begin{pmatrix}A & C\\ C & B\end{pmatrix}\begin{pmatrix}\delta a\\ \delta b\end{pmatrix},\quad \vec{\Pi}=\begin{pmatrix}A & C\\ C & B\end{pmatrix}, \tag{10.1}$$

here the three quadratic terms characterize the width of the posterior and play the fundamental role in correlations between the parameters a and b. In addition, $Q_0\equiv Q(a_0,b_0)$ is a constant, and

$$A=\left.\frac{\partial^2 Q}{\partial a^2}\right|_{a_0,b_0},\quad B=\left.\frac{\partial^2 Q}{\partial b^2}\right|_{a_0,b_0},\quad C=\left.\frac{\partial^2 Q}{\partial a\partial b}\right|_{a_0,b_0}, \tag{10.2}$$

with $\delta a=a-a_0,\delta b=b-b_0$. The correlation between the parameters a and b is described by the properties of the matrix $\vec{\Pi}$.[5] Particularly, after solving the eigenvalue equation $\vec{\Pi}\mathbf{x}=\lambda\mathbf{x}$, one could obtain two eigenvalues λ_1 and λ_2. The equation $Q=\phi>0$ gives the semi-axes of the equal-probability surface of the ellipse, i.e., $(\phi/\lambda_j)^{1/2}$ with $j=1,2$. Naturally in order the point (a_0,b_0) is the maximum one of the posterior, one requires that the eigenvalues λ_j be negative, or $A<0,B<0$ and $AB>C^2$.

If one is only interested in knowing the properties of the parameter a, then the effects of b could be integrated out as $p(a|\mathcal{D})=\int p(a,b|\mathcal{D})\mathrm{d}b$. Under the quadratic approximation (10.1), this integration could be done analytically, leading to the posterior for a as,

$$p(a|\mathcal{D})\sim\exp\left(\frac{AB-C^2}{2B}(a-a_0)^2\right),\quad \sigma_a=\sqrt{\frac{-B}{AB-C^2}}, \tag{10.3}$$

where the second expression gives the the error-bar of the a parameter. A similar expression for σ_b could also be easily obtained. In fact, the standard deviation or the root-mean-square error of a, namely σ_a could be also written in the form, $\sigma_a^2=\langle\delta a^2\rangle=\int\delta a^2 p(a,b|\mathcal{D})\mathrm{d}a\mathrm{d}b$, where $\delta a^2=(\delta a)^2$. Similarly, the covariance between a and b can be wrote out, $\sigma_{ab}^2=\langle\delta a\delta b\rangle=\int\delta a\delta b p(a,b|\mathcal{D})\mathrm{d}a\mathrm{d}b$. Basically, $|\sigma_{ab}^2|\le[\sigma_a^2\sigma_b^2]^{1/2}$.

After working out the integration, we obtain

$$\sigma_{ab}^2=\frac{C}{AB-C^2}. \tag{10.4}$$

Combining it with the expressions for σ_a^2 and σ_b^2, we can find

$$\begin{pmatrix}\sigma_a^2 & \sigma_{ab}^2\\ \sigma_{ab}^2 & \sigma_b^2\end{pmatrix}=\frac{1}{AB-C^2}\begin{pmatrix}-B & C\\ C & -A\end{pmatrix}=-\vec{\Pi}^{-1}. \tag{10.5}$$

In fact, this is the covariance matrix of the parameters a and b. In the special situation where $C=0$, i.e., $\sigma_{ab}^2=0$, the inferred parameters a and

[4] The following relation is used (where $\mathbf{A}$ is a matrix and $\mathbf{a}$ a vector)

$$\left(\mathbf{A}+\mathbf{a}\mathbf{a}^{\mathrm{T}}\right)^{-1}=\mathbf{A}^{-1}-\frac{(\mathbf{A}^{-1}\mathbf{a})(\mathbf{a}^{\mathrm{T}}\mathbf{A}^{-1})}{1+\mathbf{a}^{\mathrm{T}}\mathbf{A}^{-1}\mathbf{a}}.$$

[5] See, e.g., D. Sivia and J. Skilling, *Data Analysis: A Bayesian Tutorial*, Oxford, 2006, Chap.3.

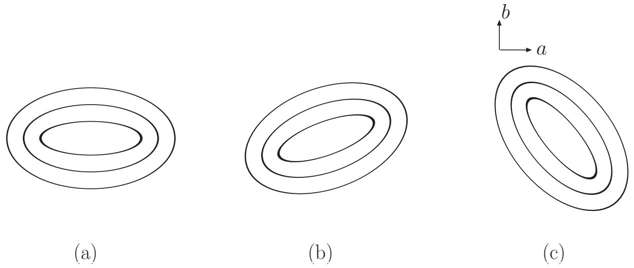


Fig. 15: Correlation between a and b for three typical cases.

b are uncorrelated. Consequently, the principal directions of the corresponding posterior will be parallel to the coordinate axes, see the panel (a) of Fig. 15. Moreover, as the magnitude of the coefficient C increases, the posterior becomes more and more skew and elongated, reflecting the growth of the correlation between the two parameters a and b, as shown in the panels (b) and (c) of Fig. 15. Here, the a and b is anti-correlated in the case (c) while positive-correlated in case (b). In the extreme situation where the value of C equals to $\pm\sqrt{AB}$, the elliptical contours are infinitely wide in one direction and oriented at an angle of $\pm\arctan\sqrt{A/B}$ with respect to the a axis. Although the error-bars of a and b will be extremely huge, the large off-diagonal elements of the covariance matrix still tells that one combination of the parameters is meaningful. More precisely, if the covariance is positive then the posterior will be very broad in the direction $b = ka$ where $k = \sqrt{A/B}$, and fairly narrow perpendicular to it. Actually, one has $b - ka \approx$ const., indicating that data contain much information about the sum $b - ka$ instead of the difference $b + a/k$. Similar conclusion could be cast if the covariance is negative, i.e., the posterior in this case will be very broad in the direction $b = -ka$, narrow perpendicular to it, and one has $b + ka \approx$ const., indicating that data contain much information on $b + ka$.

EXERCISE 9: For the logarithmic likelihood function (7.1), compute the second derivatives of the $\log p(\mathbf{X}|\mu,\sigma^2)$ with respective to μ and σ^2 and confirm that the relations $A < 0, B < 0$ and $AB > C^2$ are fulfilled. Is the coefficient C zero? Assume that the priors of the parameters μ and σ^2 are simply constants.

EXERCISE 10: Work out the in-front constant of $p(a|\mathcal{D})$ in (10.3).

In situations where there are more than two parameters, the above discussion could be straightforwardly generalized, e.g., the quadratic approximation for the logarithmic posterior takes the form,

$$Q(\mathbf{w}) \approx Q(\mathbf{w}_0) + \frac{1}{2}\sum_{i,j=1}^{n}\left.\frac{\partial^2 Q}{\partial w_i \partial w_j}\right|_{\mathbf{w}_0}\left(w_i - w_i^0\right)\left(w_j - w_j^0\right) + \cdots, \qquad (10.6)$$

where n is the dimension of the parameter vector $\mathbf{w} \in \mathrm{R}^n$. Consequently,

$$p(\mathbf{w}|D) = p(\{w_i\}|\mathcal{D}) \sim \exp\left[\frac{1}{2}(\mathbf{w}-\mathbf{w}_0)^{\mathrm{T}}\Delta Q(\mathbf{w}_0)(\mathbf{w}-\mathbf{w}_0)\right], \qquad (10.7)$$

here $\Delta L \equiv \nabla^2 L$ is the symmetric $n \times n$ matrix of the second derivatives, with its ij-component given by $\partial^2 Q/\partial w_i \partial w_j$.

For the Gaussian noise, we can use the samples to estimate its characteristic parameters, e.g., the mean μ of the noise. Here, we have $p(\mu|\mathcal{D}) = \int_0^\infty p(\mu,\sigma|\mathcal{D})\mathrm{d}\sigma$, where $\mathcal{D} = \{x^{(i)}\}$ and the integrand can be expressed as a product of the likelihood and the prior. Adopting the prior as the constant, one then naturally has (7.1), and consequently,

$$p(\mu|\mathcal{D}) \sim \int_0^\infty \vartheta^{m-2}\exp\left[-\frac{\vartheta^2}{2}\sum_{i=1}^{m}(x^{(i)}-\mu)^2\right]\mathrm{d}\vartheta, \quad \vartheta = 1/\sigma. \qquad (10.8)$$

Scaling $\varphi = \vartheta[\sum_{i=1}^m (x^{(i)}-\mu)^2]^{1/2}$ gives $p(\mu|\mathcal{D}) \sim [\sum_{i=1}^m (x^{(i)}-\mu)^2]^{-(m-1)/2}$, and

$$\left.\frac{\mathrm{d}Q}{\mathrm{d}\mu}\right|_{\mu_0} = \frac{(m-1)\sum_{i=1}^m (x^{(i)}-\mu)}{\sum_{i=1}^m (x^{(i)}-\mu)^2} = 0, \quad Q = \log p(\mu|\mathcal{D}), \qquad (10.9)$$

solving it gives $\mu_0 = m^{-1}\sum_{i=1}^m x^{(i)}$, i.e., the ML solution $\widehat{\mu}_{\mathrm{ML}}$. In other words, the optimal estimation for μ is still given by the arithmetic average of the samples. Differentiating Q for a second time and evaluating it at the corresponding maximum value μ_0, one obtains

$$\left.\frac{\mathrm{d}^2 Q}{\mathrm{d}\mu^2}\right|_{\mu_0} = -\frac{m(m-1)}{\sum_{i=1}^m (x^{(i)}-\mu)^2}. \qquad (10.10)$$

Since the error-bar for the best estimate is given by the inverse of the square root of the minus the second derivative (as inferring from the basic formula for the 1d Gaussian), we can obtain our estimation for the mean as

$$\mu = \mu_0 \pm \frac{s}{\sqrt{m}}, \quad s^2 = \frac{1}{m-1}\sum_{i=1}^m \left(x^{(i)} - \mu_0\right)^2. \qquad (10.11)$$

Here s^2 is simply the unbiased variance estimator $\widetilde{\sigma}^2$ of (7.4). In this sense, we re-explain the factor $(m-1)^{-1}$ adopted in (7.4).

When discussing the linear curve fitting problem in the starting part of the current lecture, we assume that the noise produced on the simulated data has the same error-bar σ. However, this restriction is limited-useful and could be generalized to different error-bars for each data sample $\{\sigma^{(i)}\}$, i.e., the likelihood for the parameters a and b takes the following form,

$$p(\mathcal{D}|\vec{\theta}) = \frac{1}{\sqrt{2\pi\sigma_i^2}}\exp\left[-\frac{(f_{\vec{\theta}}(x^{(i)}) - y^{(i)})^2}{2\sigma^{(i),2}}\right], \quad \vec{\theta} = (a,b), \quad \mathcal{D} = \{y^{(i)}\}, \qquad (10.12)$$

where $f_{\vec{\theta}}(x)$ is the learning model, see (1.1). Under the independence assumption of the samples, the total likelihood function is obtained as,

$$p(\mathcal{D}|\vec{\theta}) \sim e^{-\chi^2/2}, \quad \chi^2 = \sum_{i=1}^m \left(\frac{f_{\vec{\theta}}(x^{(i)}) - y^{(i)}}{\sigma^{(i)}}\right)^2. \qquad (10.13)$$

Adopting the notation J for the (negative) logarithmic posterior,

$$J(\vec{\theta}) = -\log p(\vec{\theta}|D) = \frac{1}{2}\chi^2 + \text{const.} \sim \frac{1}{2}\sum_{i=1}^m \left(\frac{f_{\vec{\theta}}(x^{(i)}) - y^{(i)}}{\sigma^{(i)}}\right)^2, \qquad (10.14)$$

see (1.1), and (10.14) is the weighted least squares. Based on (10.14), one can find the optimal a and b appeared in the fitting model $y = ax + b$.

EXERCISE 11: Work out the closed form for the optimal a^* and b^*.

XI. CENTRAL LIMIT THEOREM

Let us ask a question that if m IID random samples are generated from a distribution with the mean μ and the variance σ^2, what is the distribution for the quantity

$$X = \frac{\bar{x}_m - \mu}{\sigma/\sqrt{m}}, \quad \bar{x}_m = \frac{1}{m}\sum_{i=1}^m x^{(i)}. \qquad (11.1)$$

Mathematically, we have the relation $\lim_{m\to\infty} P(a \le X \le b) = \Phi(b) - \Phi(a)$, with $\Phi(x) = [1/\sqrt{2\pi}]\int_{-\infty}^x \exp(-x^2/2)\mathrm{d}x$ the cdf of the normal Gaussian.

We prove the central limit theorem via calculating the generating function of the variable X, i.e., it is given by the formula $e^{t^2/2}$. After introducing the new variable $y^{(i)} = (x^{(i)} - \mu)/\sigma$, one has

$$X = \frac{1}{\sqrt{m}}\sum_{i=1}^m \frac{x^{(i)} - \mu}{\sigma} = \frac{1}{\sqrt{m}}\sum_{i=1}^m y^{(i)}. \qquad (11.2)$$

Since the mean of the variable $y^{(i)}$ is zero and the variance is unit, and by considering the relation

$$\mathrm{E}[e^{tx}] = \mathscr{M}_x(t) = 1 + t\mu_1 + t^2\mu_2/2! + t^3\mu_3/3! + \cdots, \qquad (11.3)$$

one obtains the generating function for the $y^{(i)}$ as $\mathscr{M}_{y^{(i)}}(t) = 1 + 2^{-1}t^2 + \cdots$.

Now, we have $\mathscr{M}'_{y^{(i)}}(t) = t, \mathscr{M}''_{y^{(i)}}(t) = 1$, indicating that,

$$\begin{aligned}
\mathscr{M}_X(t) &= \int e^{Xt} p(X)\mathrm{d}X = \int \exp\left(\frac{t}{\sqrt{m}}\sum_{i=1}^m y^{(i)}\right) p\left(\frac{1}{\sqrt{m}}\sum_{i=1}^m y^{(i)}\right)\mathrm{d}X \\
&= \prod_{i=1}^m \int \exp\left(\frac{ty^{(i)}}{\sqrt{m}}\right) \prod_{i=1}^m p\left(\frac{y^{(i)}}{\sqrt{m}}\right) \prod_{i=1}^m \mathrm{d}\left(\frac{y^{(i)}}{\sqrt{m}}\right) \\
&= \prod_{i=1}^m \left[\int \exp\left(\frac{ty^{(i)}}{\sqrt{m}}\right) p\left(\frac{y^{(i)}}{\sqrt{m}}\right) \mathrm{d}\left(\frac{y^{(i)}}{\sqrt{m}}\right)\right] \\
&= \prod_{i=1}^m \left[\int \exp\left(\frac{ty^{(i)}}{\sqrt{m}}\right) \sqrt{m}\, p\left(y^{(i)}\right) \frac{1}{\sqrt{m}}\mathrm{d}y^{(i)}\right] \\
&= \left[M_{y^{(i)}}\left(\frac{t}{\sqrt{m}}\right)\right]^m = \left(1 + \frac{t^2}{2}\frac{1}{m} + \cdots\right)^m,
\end{aligned} \tag{11.4}$$

where one uses the independence of the random samples with their pdf given by $p(x,y) = p(x)p(y)$, together with the basic relation $p(z) = p(y)|\partial y/\partial z|$ with $z = y/\sqrt{m}$ and $\partial y/\partial z = \sqrt{m}$. Consequently $p(y^{(i)}/\sqrt{m}) = \sqrt{m}p(y^{(i)})$, and thus $\lim_{m\to\infty}\mathscr{M}_X(t) = e^{t^2/2}$, furnishing the proof. The quantity $\bar{x}_m$ has the distribution approximately as $\mathcal{N}(\mu, \sigma^2/m)$.

The center limit theorem is relevant for the discussion on the posterior distribution when the data number is enough large. Actually, one could prove that the posterior $p(w|x)$ always takes the form

$$\left(\frac{w - \mathrm{E}[w|x]}{\sqrt{\mathrm{var}[w|x]}}\middle| x\right) \to \mathcal{N}(0,1). \tag{11.5}$$

Let us prove the posterior takes the form of Gaussian when the sample number m is large. For the scalar dataset $x = (x^{(1)}, \cdots, x^{(m)})$, assume that the real physical model describing it is given by $f(x)$, and the prior for w is denoted as $p(w)$. Moreover, the likelihood for w is denoted as $p(x|w) = \prod_{i=1}^m p(x^{(i)}|w)$ by assuming that the data is IID. The deviation between the likelihood and the real distribution is

$$\mathrm{KL}\left(f(x^{(i)})\|p(x^{(i)}|w)\right) = \mathrm{E}_{f(x^{(i)})}\left[\log\left(\frac{f(x^{(i)})}{p(x^{(i)}|w)}\right)\right], \tag{11.6}$$

which is called the Kullback–Leibler (KL) divergence.

EXERCISE 12: Prove that the KL divergence is unsymmetric with respect to f and p. Moreover, prove that the KL divergence is non-negative based on the convexity of $-\log x$ via Jensen's inequality which says that for a convex function $f(x)$, one has $\mathrm{E}[f(x)] \geq f(\mathrm{E}[x])$.

Denote w_0 the minimum point of the KL divergence, i.e., the value minimizing the KL term. In addition one assumes that the adopted likelihood is reasonable in the sense that there exists a real parameter w that the real model matches the likelihood, i.e., $f(x^{(i)}) = p(x^{(i)}|w)$. Under this circumstance the KL term takes its minimum at this real parameter and the w could be naturally denoted as w_0. We firstly need to prove $p(w = w_0|x) \to 1$ as $m \to \infty$. The conclusion is equivalent to for all $w \neq w_0$, the corresponding probability approaches to zero under large m. Consider,

$$\log\left(\frac{p(w|x)}{p(w_0|x)}\right) = \log\left(\frac{p(w)}{p(w_0)}\right) + \sum_{i=1}^m \log\left(\frac{p(x^{(i)}|w)}{p(x^{(i)}|w_0)}\right). \tag{11.7}$$

If w and w_0 are treated as fixed and $x^{(i)} \sim f$, the second term on the right hand side is the sum of the IID samples, and each one could be written,[6]

$$\mathrm{E}\left[\log\left(\frac{p(x^{(i)}|w)}{p(x^{(i)}|w_0)}\right)\right] = \mathrm{KL}(w_0) - \mathrm{KL}(w). \tag{11.8}$$

Consequently, if $w = w_0$ the above expression is zero while otherwise it is negative (since w_0 is its minimum point). If $w \neq w_0$, the second term on the right hand side of the previous equation could be expressed as sum of the m random variables with a negative mean, i.e., if $m \to \infty$, the sum also approaches to negative infinity. It means that $p(w|x)/p(w_0|x) \to 0$ as $m \to \infty$, and thus $p(w|x) \to 0$ under $m \to \infty$. On the other hand since the total probability is normalized, one obtains that $p(w_0|x) \to 1$.

Next, denote the mode of the posterior as $\widehat{w}$, we should prove that when $m \to \infty$ as $\widehat{w} \to w_0$ (normality). In order to do that, we expand

$$\begin{aligned}
\log p(w|x) \approx{}& \log p(\widehat{w}|x) + \frac{1}{2}(w - \widehat{w})^2 \left[\frac{\mathrm{d}^2}{\mathrm{d}w^2}\log p(w|x)\right]_{w=\widehat{w}} \\
&+ \frac{1}{6}(w - \widehat{w})^3 \left[\frac{\mathrm{d}^3}{\mathrm{d}w^3}\log p(w|x)\right]_{w=\widehat{w}} + \cdots,
\end{aligned} \tag{11.9}$$

where the first term is independent of w and the second term can be rewritten,

$$\left[\frac{\mathrm{d}^2}{\mathrm{d}w^2}\log p(w|x)\right]_{w=\widehat{w}} = \frac{\mathrm{d}^2}{\mathrm{d}w^2}\log p(\widehat{w}) + \sum_{i=1}^m \left[\frac{\mathrm{d}^2}{\mathrm{d}w^2}\log p(x^{(i)}|w)\right]_{w=\widehat{w}}, \tag{11.10}$$

here the first term is constant (independent of w) and the second term is the sum of each IID sample with negative mean. If one has $f(x) = p(x|w_0)$ for some w_0, then each term owns the mean value $-J(w_0)$ with,

$$J(w) = \mathrm{E}\left[\left(\frac{\mathrm{d}\log p(x|w)}{\mathrm{d}w}\right)^2 \middle| w\right] = -\mathrm{E}\left[\frac{\mathrm{d}^2\log p(x|w)}{\mathrm{d}w^2}\middle| w\right], \tag{11.11}$$

which is called the Fisher information for w. Consequently,

$$\begin{aligned}
\left[\frac{\mathrm{d}^2}{\mathrm{d}w^2}\log p(w|x)\right]_{w=\widehat{w}} &= \frac{\mathrm{d}^2}{\mathrm{d}w^2}\log p(\widehat{w}) + \sum_{i=1}^m \left[\frac{\mathrm{d}^2}{\mathrm{d}w^2}\log p(x^{(i)}|w)\right]_{w=\widehat{w}} \\
&\sim -mJ(w_0) + \text{constant}.
\end{aligned} \tag{11.12}$$

Similarly, one could prove that the high order terms in the expansion have lower increasing speed compared with the number m, indicating as the data number m increases only the second order contribution is kept with the corresponding variance $(mJ(w_0))^{-1}$.

EXERCISE 13: Given the transform $\phi = t(\theta)$, prove the corresponding transform for the Fisher information is $J^{1/2}(\phi) = J^{1/2}(\theta)|\mathrm{d}\theta/\mathrm{d}\phi|$.

XII. LAW OF LARGE NUMBERS: RANDOMNESS

Another very useful concept/technique in statistics is the law of large numbers, here we briefly introduce the very basic concept of it. Assume that $x^{(i)}$ with $i = 1 \sim m$ are m IID samples from some distribution, one then has

$$P\left(\left|\frac{x^{(1)} + x^{(2)} + \cdots + x^{(m)}}{m} - \mathrm{E}[x]\right| \geq \epsilon\right) \leq \frac{\mathrm{var}[x]}{m\epsilon^2} \tag{12.1}$$

for some positive number ϵ. In order to prove the law of large numbers, one needs the following two inequalities,

(a) For any non-negative random x and $a > 0$, $P(x \geq a) \leq \mathrm{E}[x]/a$.

(b) For any random x and $c > 0$, $P(|x - \mathrm{E}[x]| \geq c) \leq \mathrm{var}[x]/c^2$.

They are called Markov's and Chebyshev's inequalities, respectively, and hold both for discrete and continuous random numbers, here we prove them under the continuous case. For a continuous non-negative random variable x with pdf p, one has

$$\begin{aligned}
\mathrm{E}[x] &= \int_0^\infty xp(x)\mathrm{d}x = \int_0^a xp(x)\mathrm{d}x + \int_a^\infty xp(x)\mathrm{d}x \\
&\geq \int_a^\infty xp(x)\mathrm{d}x \geq a\int_a^\infty p(x)\mathrm{d}x = aP(x \geq a),
\end{aligned} \tag{12.2}$$

and consequently leading to the Markov's inequality. In order to prove the Chebyshev's inequality, note that $y = |x - \mathrm{E}[x]|^2$ is a non-negative random variable with $\mathrm{E}[y] = \mathrm{var}[x]$, so the Markov's inequality naturally leads to Chebyshev's inequality. The Markov's inequality bounds the tail of a distribution using only information about the mean while the Chebyshev's inequality also uses the variance of the distribution.

[6] A. Gelman, J.B. Carlin, H.S. Stern, D.B. Dunson, A. Vehtari, and D.B. Rubin, *Bayesian Data Analysis*, 3rd, 2014, Taylor and Francis Group, Appendix B.

Now we prove the law of large numbers. Since $x^{(i)}$ is IID,

$$\mathrm{E}\left[\frac{x^{(1)}+x^{(2)}+\cdots+x^{(m)}}{m}\right]=\frac{1}{m}\sum_{i=1}^{m}\mathrm{E}[x]=\mathrm{E}[x], \tag{12.3}$$

and thus

$$\begin{aligned}&P\left(\left|\frac{x^{(1)}+x^{(2)}+\cdots+x^{(m)}}{m}-\mathrm{E}[x]\right|\geq\epsilon\right)\\=&P\left(\left|\frac{x^{(1)}+x^{(2)}+\cdots+x^{(m)}}{m}-\mathrm{E}\left[\frac{x^{(1)}+x^{(2)}+\cdots+x^{(m)}}{m}\right]\right|\geq\epsilon\right).\end{aligned} \tag{12.4}$$

By using Chebyshev's inequality (and the relation $\mathrm{var}[ax]=a^2\,\mathrm{var}[x]$),

$$\begin{aligned}&P\left(\left|\frac{x^{(1)}+x^{(2)}+\cdots+x^{(m)}}{m}-\mathrm{E}\left[\frac{x^{(1)}+x^{(2)}+\cdots+x^{(m)}}{m}\right]\right|\geq\epsilon\right)\\\leq&\frac{1}{\epsilon^2}\,\mathrm{var}\left[\frac{x^{(1)}+x^{(2)}+\cdots+x^{(m)}}{m}\right]=\frac{1}{m^2\epsilon^2}\sum_{i=1}^{m}\mathrm{var}[x]=\frac{\mathrm{var}[x]}{m\epsilon^2}.\end{aligned} \tag{12.5}$$

Similarly, by applying the Markov's inequality to the random variable $|x-\mu|^k$, one has

$$P(|x-\mu|\geq t)\leq\frac{\mathrm{E}[(x-\mu)^k]}{t^k},\quad t>0,\quad \mu=\mathrm{E}[x]. \tag{12.6}$$

On the other hand, suppose x has the momentum generating function well defined near zero, i.e., $\phi(\lambda)=\mathscr{M}_{x-\mu}(\lambda)$ exists for all $|\lambda|\leq b$ for some constant $b>0$, then for any $\lambda\in[0,b]$ one has by applying the Markov's inequality to the variable $y=e^{\lambda(x-\mu)}$,

$$P(x-\mu\geq t)=P\left(e^{\lambda(x-\mu)}\geq e^{\lambda t}\right)\leq \mathrm{E}[e^{\lambda(x-\mu)}]/e^{\lambda t}. \tag{12.7}$$

If the random variable x is Gaussian, then it gives $P(x\geq\mu+t)\leq\exp(-t^2/2\sigma^2)$ where $t\geq 0$, it is called the Chernoff's bound for the Gaussian random variable. If the random variable x has the property $\mathrm{E}[e^{\lambda(x-\mu)}]\leq e^{\sigma^2\lambda^2/2}, \lambda\in\mathrm{R}$, x is called sub-Gaussian with parameter σ.

EXERCISE 14: Show that for any constant $c\geq 1$, there exist distributions for which Chebyshev's inequality is tight.

EXERCISE 15: Consider the pdf $p(x=0)=1-1/a$ and $p(x=a)=1/a$. Plot the pdf that x is greater than or equal to a as a function of a for the bound given by the Markov's inequality applied to x^2 and x^4.

Consider an example, let $\mathbf{x}$ and $\mathbf{y}$ be two d-dimensional random point whose coordinates are each selected from a zero mean and unit variance Gaussian. Notice that $\mathrm{E}[|\mathbf{x}|^2]=\sum_{i=1}^{d}\mathrm{E}[x_i^2]=d\,\mathrm{var}[x_i]=d$, so the mean squared distance of a point from the center is d. Moreover,[7]

$$\begin{aligned}\|\mathbf{x}-\mathbf{y}\|^2&=\sum_{i=1}^{d}(x_i-y_i)^2=\sum_{i=1}^{d}\left(\mathrm{E}[x_i^2]+\mathrm{E}[y_i^2]-2\mathrm{E}[x_i]\mathrm{E}[y_i]\right)\\&=\sum_{i=1}^{d}\left(\mathrm{var}[x_i]+\mathrm{var}[y_i]-2\mathrm{E}[x_i]\mathrm{E}[y_i]\right)=2d.\end{aligned} \tag{12.8}$$

These relations together indicate that the random d-dimensional $\mathbf{x}$ and $\mathbf{y}$ must be approximately orthogonal since $|\mathbf{x}-\mathbf{y}|^2\approx\mathbf{x}^2+\mathbf{y}^2$. If one scales these random points to be unit length and call $\mathbf{x}$ the north pole, much of the surface area of the unit ball lies near the equator. An important property of the high-dimensional objects is that most of their volume is near the surface. Consider a ball B with radius r in d-dimensions, if one shrinks the radius of the ball by an amount ϵ to produce another ball B' with radius $r(1-\epsilon)$, then $\mathrm{vol}(B')/\mathrm{vol}(B)=(1-\epsilon)^d\leq e^{-\epsilon d}$, where one uses the inequality $1-x\leq e^{-x}$. Fixing ϵ and letting $d\to\infty$, the above quantity rapidly approaches to zero. It means the nearly all of the volume of B must be in the portion of B that does not belong to the region B'. We denote S the unit ball in dimension d.

An immediate consequence of the above observation is that at least a $1-e^{-\epsilon d}$ fraction of the volume of the unit ball is concentrated in a small annulus of width ϵ at the boundary. Particularly, most of the volume of the d-dimensional unit ball is contained in an annulus of width $\mathcal{O}(1/d)$ near the boundary. See Fig. 16 for the sketch of the prediction. If the radius is r then the annulus width is on the order $\mathcal{O}(r/d)$. Moreover, one can show that most of the volume of the unit ball in high dimensions is concentrated near its "equator". More specifically, for any unit-length vector $\mathbf{a}$ defining the "north pole", most of the volume of the unit ball lies in the thin slab of points whose inner product with $\mathbf{a}$ has magnitude $\mathcal{O}(d^{-1/2})$. In order to show this fact it suffices by symmetry to fix the $\mathbf{a}$ to be the first coordinate vector. Specifically we want to show that most of the volume of the unit ball has $x_1=\mathcal{O}(d^{-1/2})$. Thus we can show that two random numbers (points) in the unit ball are with high probability nearly orthogonal.

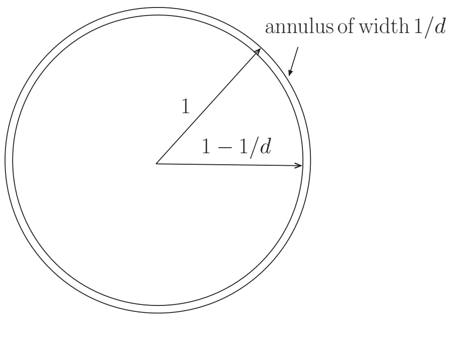


Fig. 16: Most volume of the d-ball is contained in an annulus of width $\mathcal{O}(1/d)$.

EXERCISE 16: For a d-dimensional circular cylinder of radius r and height h, what is the surface area and what is the volume $V(d)$? Similarly, show the volume of a unit d-ball is $V(d)=2\pi^{d/2}/d\Gamma(d/2)$.

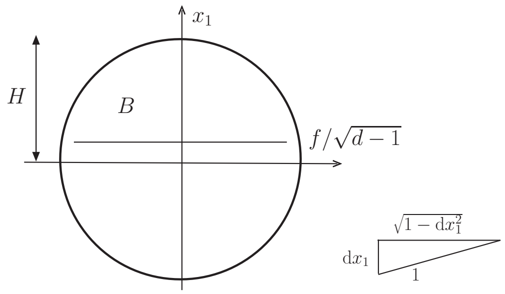


Fig. 17: Most of the volume of the upper hemisphere in dimension d is below the plane defined by $x_1=f/\sqrt{d-1}$.

By symmetry we need to prove that at most a $(2/f)e^{-f^2/2}$ fraction of the half of the ball with $x_1\geq 0$ has $x_1\geq f/\sqrt{d-1}$ where $f\geq 1$ is a constant and here we consider $d\geq 3$. Let B denote the portion of the ball with $x_1\geq f/\sqrt{d-1}$ and H denote the upper hemisphere, see Fig. 17. The idea is to show that the ratio of the volume of B to the volume of H goes to zero by calculating an upper bound on the vol$B$ and a lower bound on the vol(H) and proving that,

$$\frac{\mathrm{vol}(B)}{\mathrm{vol}(H)}\leq\frac{\text{upper bound vol}(B)}{\text{lower bound vol}(H)}\leq\frac{2}{f}\exp\left(-\frac{f^2}{2}\right). \tag{12.9}$$

In order to calculate the volume of B, integrate an incremental volume that is a disk of width $\mathrm{d}x_1$ and whose face is ball of dimension $d-1$ with radius $\sqrt{1-x_1^2}$. The surface area of the disk is $(1-x_1^2)^{d/2-1/2}V(d-1)$ and the volume above the slice is

$$\mathrm{vol}(B)=\int_{f/\sqrt{d-1}}^{1}\left(1-x_1^2\right)^{\frac{d-1}{2}}V(d-1)\mathrm{d}x_1\leq\frac{V(d-1)}{f\sqrt{d-1}}\exp\left(-\frac{f^2}{2}\right). \tag{12.10}$$

The volume of the hemisphere below the plane $x_1=1/\sqrt{d-1}$ is a lower bound on the entire volume and it is at least that of a cylinder of height $1/\sqrt{d-1}$ and radius $\sqrt{1-1/(d-1)}$. The volume of the cylinder is,

$$V(d-1)\left(\frac{d-2}{d-1}\right)^{d/2-1/2}\cdot\frac{1}{\sqrt{d-1}}. \tag{12.11}$$

[7] For a general reference, see, A. Blum, J. Hopcroft, and R. Kannan, *Foundations of Data Science*, Cambridge, 2020.

By using the fact $(1-x)^y \ge 1 - yx$ for $y \ge -1$, and the volume of the cylinder is at least $V(d-1)/2\sqrt{d-1}$ for $d \ge 3$. Here we use the above technique involving the cylinder is for obtaining the lower limit containing the factor $V(d-1)$, which could be canceled by the same one included in the upper limit fro the volume of B. After combining these two limits, one obtains

$$\text{ratio} \le \frac{V(d-1)}{f\sqrt{d-1}}\exp\left(-\frac{f^2}{2}\right)\bigg/\frac{V(d-1)}{2\sqrt{d-1}} = \frac{2}{f}\exp\left(-\frac{f^2}{2}\right). \tag{12.12}$$

For $f \ge 1$ and $d \ge 3$ at least a fraction of $1-(2/f)e^{-f^2/2}$ of the volume of the d-dimensional unit ball has $|x_1| \le f/\sqrt{d-1}$, i.e., almost all of the mass is near the equator. It should be point out that the range of f depends on d, i.e., one must have $f/\sqrt{d-1} \le 1$ or $f \le \sqrt{d-1}$. In the limit $d \to \infty$ one has $V(d) \to 0$. From the ratio $V(d+1)/V(d)$ one finds that the decreasing on the $V(d)$ grows faster and faster as d increases. In addition, the surface area of the unit ball is given by $S(d) = \partial V(d)/\partial r = 2\pi^{d/2}r^{d-1}/\Gamma(d/2)$ and consequently $S(d)/V(d) = d/r$, with the latter being d for the unit ball. It shows that as $d \to \infty$, all most all of the volume is near the surface.

EXERCISE 17: Consider a sphere of radius R in d-dimensions together with the concentric hypercube of side $2R$, so that the sphere touches the hypercube at the centers of each of its sides. Prove that the ratio of the volume of the sphere to the volume of the hypercube is given by $\pi^{d/2}/d2^{d-1}\Gamma(d/2)$. By using Stirling's formula for the Γ function, i.e., $\Gamma(x+1) \approx (2\pi)^{1/2}e^{-x}x^{x+1/2}$ for $x \gg 1$, show that as $d \to \infty$ the above ratio approaches to zero. Show also that the ratio of the distance from the center of the hypercube to one of the corners, divided by the perpendicular distance to one of the sides, is $\sqrt{d}$.

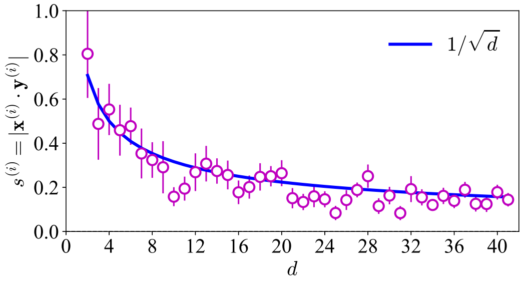


Fig. 18: Cosine value $s^{(i)}$ between two random vectors on the unit ball.

Now we draw two points randomly from the unit ball, with high probability their corresponding vectors will be nearly orthogonal to each other. If one defines the vector in the direction of the first point as the north pole, with high probability the second will have a projection of only $\pm\mathcal{O}(1/\sqrt{d})$ in this direction thus their inner product will be $\pm\mathcal{O}(1/\sqrt{d})$. It implies that with high probability the angle between the two vectors will be $\pi \pm \mathcal{O}(1/\sqrt{d})$. Fig. 18 gives the simulation result on $s^{(i)} = |\mathbf{x}^{(i)}\cdot\mathbf{y}^{(i)}|$ where a total 10 independent drawings (i.e., $i = 1 \sim 10$) for each d are performed, the blue curve corresponds to the theoretical prediction $\sim 1/\sqrt{d}$. We can formulate this conclusion in more accurate form: If we draw m points at random in the unit ball, with high probability all points will be close to unit length and each pair of them will be orthogonal, i.e., with probability $1-\mathcal{O}(1/m)$, we have

$$\|\mathbf{x}^{(i)}\| \ge 1 - \frac{2\log m}{d}, \quad \mathbf{x}^{(i),\mathrm{T}}\mathbf{x}^{(j)} = \mathbf{x}^{(i)}\cdot\mathbf{x}^{(j)} \le \sqrt{\frac{6\log n}{d-1}}, \tag{12.13}$$

for $i \ne j$. The prove is straightforward. The probability $\|\mathbf{x}^{(i)}\| \le 1-\epsilon$ is less than $e^{-\epsilon d}$, thus

$$P\left(\|\mathbf{x}^{(i)}\| \le 1 - \frac{2\log m}{d}\right) \le \exp\left(-\frac{2\log m}{d}\cdot d\right) = \frac{1}{m^2}. \tag{12.14}$$

By the union bound the probability there exists an i such $\|\mathbf{x}^{(i)}\| < 1 - 2\log m/d$ is at most $1/m$. Moreover, for a component of a Gaussian vector the probability $|\mathbf{x}^{(i)}| > f/\sqrt{d-1}$ is at most $(2/f)e^{-f^2/2}$. There are $m(m-1)/2$ pairs i and j and for each pair if we define $\mathbf{x}^{(i)}$ as the north pole the probability that the projection of $\mathbf{x}^{(j)}$ onto to the north pole is more than $[\log m/(d-1)]^{1/2}$ is at most $\mathcal{O}(e^{-6\log m/2}) = \mathcal{O}(m^{-3})$. Thus the inner product condition is violated with probability at most $\mathcal{O}(m(m-1)m^{-3}) \sim \mathcal{O}(1/m)$. In fact one can even use the above result to show the volume of the unit ball approaches to zero without the explicit formula for $V(d)$. More specifically, consider a small box centered at the origin of side length $2f/\sqrt{d-1}$. Then for $f = 2\sqrt{\log d}$ this box contains over half of the volume of the ball. On the other hand, the volume of this box clearly goes to zero as d goes to infinity, since it volume is $\mathcal{O}([\log d/(d-1)]^{d/2})$. Consequently the volume of the ball goes to zero as well. With $f = 2\sqrt{\log d}$, the fraction of the volume of the ball with $|x_1| \ge f/\sqrt{d-1}$ is at most

$$\frac{2}{f}\exp\left(-\frac{f^2}{2}\right) = \frac{1}{\sqrt{\log d}}e^{-2\log d} = \frac{1}{d^2\sqrt{\log d}} < \frac{1}{d^2}. \tag{12.15}$$

Since this is true for each of the d dimensions, by a union bound at most a $\mathcal{O}(1/d) \le 1/2$ fraction of the volume of the ball lies outside the cube, furnishing the proof. It seems strange how it can be that nearly all the points in the unit ball are very close to the surface and yet in the meanwhile nearly all the points are in a box of side length $\mathcal{O}([\log d/(d-1)]^{1/2})$. The ingredient is to remember that points on the surface of the ball satisfy $x_1^2 + \cdots + x_d^2 = 1$, so for each each coordinate i, a typical value will be $\pm\mathcal{O}(1/\sqrt{d})$. Actually it is often help to think of picking a random point on the sphere as very similar to picking a random point of the form $(\pm 1/\sqrt{d}, \pm 1/\sqrt{d}, \cdots, \pm 1/\sqrt{d})$.

We now consider generating points uniformly at random on the surface of the unit ball. For the two-dimensional version of generating points on the circumference of the unit-radius circle, we can independently generate each coordinate uniformly at random from the interval $[-1,1]$ and then project each point onto the unit circle, see Fig. 19. However, the distribution is not uniform since more points fall on a line from the origin to a vertex of the square than fall on a line from the origin to the midpoint of an edge of the square due to the difference in length. To solve this problem one could discard all points outside the unit circle and project the remaining points onto the circle. In higher dimensions this method does not work since the fraction of points that fall inside the ball drops to zero and all of the points would be thrown away. In this situation the solution is to generate a point each of whose coordinates is an independent Gaussian variable. In particular, generate $x_1, x_2, \cdots, x_d$ using a zero-mean and unit-variance Gaussian on the real line, e.g., via the Box–Muller method.[8] Thus the probability density of the data sample $\mathbf{x}$ is given by

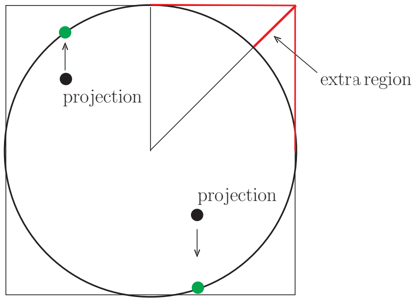


Fig. 19: Drawing random points on the circle.

$$p(\mathbf{x}) = \frac{1}{(2\pi)^{d/2}}\exp\left(-\frac{x_1^2 + x_2^2 + \cdots + x_d^2}{2}\right) \tag{12.16}$$

and is spherically symmetric. Normalizing the vector $\mathbf{x}$ to a unit vector gives a distribution that is uniformly over the surface of the sphere. Notice that once the vector is normalized its coordinates are no longer statistically independent. To generate a point $\mathbf{z}$ uniformly over the ball (both surface and the interior), scale the point $\mathbf{x}/\|\mathbf{x}\|$ generated on the surface by a scalar $\rho \in [0,1]$. The next question is what should the distribution of ρ be as a function of the radial length r, namely the func-

[8] G. Box and M. Muller, *A Note on the Generation of Random Normal Deviates*, The Annals of Mathematical Statistics, **29**, 610 (1958).

tion $\rho(r)$? The answer is that $\rho(r)$ is proportional to r^{d-1} in dimension d. Solving the integration $\int_0^1 cr^{d-1}\mathrm{d}r = 1$ gives the constant $c = d$. Another way to see this formally is that the volume of the radius r ball in d dimension is $r^d V(d)$ where $V(d)$ is the volume for unit ball, see EXERCISE 18. The density at radius r is exactly $(\mathrm{d}/\mathrm{d}r)(r^d V(d)) = dr^{d-1}V(d)$. So pick $\rho(r) = dr^{d-1}$ for $0 \le r \le 1$ one can succeed in generating a point $\mathbf{z} = \rho\mathbf{x}/\|\mathbf{x}\| = dr^{d-1}\mathbf{x}/\|\mathbf{x}\|$ uniformly at random from the unit ball by using convenient spherical Gaussian. For the 2d case, one would expect that the random x and y components are given by $x = R\cos\phi$ and $y = R\sin\phi$ where $R \sim \mathrm{Unif}[0,1]$ and $\phi \sim \mathrm{Unif}[0,2\pi]$. However, the scaling function ρ takes the form $\rho(r) = 2r$, and the r-dependence of the ρ function tells that the correct sampling approach is $x = \sqrt{R}\cos\phi$ and $y = \sqrt{R}\sin\phi$. See Fig. 20 where the left panel is the result from the naive consideration while the right panel gives the correct sampling. It is obvious that in the left panel there are points near the center than the surface.

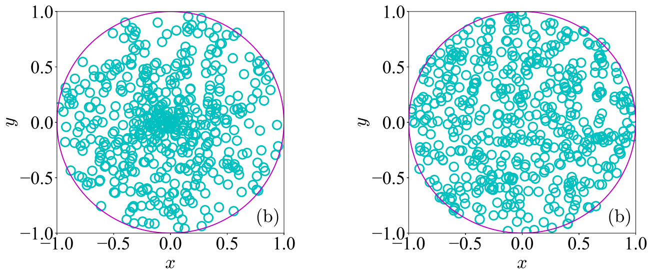


Fig. 20: Drawing random points on the 2d unit ball (circle).

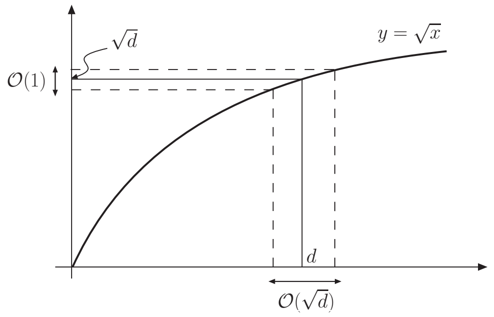


Fig. 21: Concentration of the norm of a random vector $\mathbf{x}$ in R^d.

A one-dimensional Gaussian has its mass close to the origin. However as the dimension d increases new thing emerges. The d-dimensional spherical Gaussian with zero mean and variance σ^2 in each direction has the density function $p(\mathbf{x}) = [2\pi\sigma^2]^{-d/2}e^{-\|\mathbf{x}\|^2/2\sigma^2}$. The value of the density is maximum at the origin but there is very little volume there. To see it, by taking $\sigma^2 = 1$ and integrating the probability density over a unit ball centered at the origin yields almost zero mass, since the volume of such a ball is negligible. In fact one needs to increase the radius of the ball to nearly $\sqrt{d}$ before there is a significant volume and hence significant probability mass. If one increases the radius much beyond $\sqrt{d}$, the integral barely increases even though the volume increases, since the probability density is dropping off at a much higher rate. Intuitively, notice that $\mathrm{E}[\mathbf{x}^2] \approx d$, so the mean squared distance of a point from the center is d. We often call the square root of the mean squared distance $\sqrt{d}$ as the radius of the Gaussian. Let us show these features by doing some mathematics. The probability density function over a thin shell of radius r and width ϵ is $p(r)\epsilon$ where $p(r)$ is

$$p(r) = \frac{S(d)r^{d-1}}{(2\pi\sigma^2)^{d/2}}e^{-r^2/2\sigma^2}, \qquad (12.17)$$

where $S(d)$ is the surface of the unit ball in dimensions d. Taking the derivative of $p(r)$ with respect to r, one obtains $\mathrm{d}p(r)/\mathrm{d}r = p(r)[(d-1/r) - r/\sigma^2]$, which gives the optimal r as $r^* = \sqrt{d-1}\sigma \approx \sqrt{d}\sigma$ with the latter being effective for large d. Expanding the $p(r)$ near r^* gives $p(r^*+\epsilon)$ as,

$$p(r^*)\exp\left[-\frac{(r^*+\epsilon)^2}{2\sigma^2} + (d-1)\log(r^*+\epsilon)\right] \approx p(r^*)e^{-\epsilon^2/\sigma^2}, \qquad (12.18)$$

using the expansion $\log(1+x) \approx x - x^2/2$. These formulae show that r^* is a maximum of the radial probability density and also that $p(r)$ decays exponentially away from its maximum value at r^* with length scale σ. The above discussion establishes the interesting relation for large d,

$$\sqrt{d \pm \mathcal{O}(\sqrt{d})} \approx \sqrt{d} \pm \mathcal{O}(1), \qquad (12.19)$$

i.e., while $\|\mathbf{x}\|^2$ deviates by $\mathcal{O}(\sqrt{d})$ around d, $\|\mathbf{x}\|$ deviates by $\mathcal{O}(1)$ (i.e., a constant) around $\sqrt{d}$, see Fig. 21 for an illustration of this result.

EXERCISE 18: Sample uniformly in the 2d/3d unit sphere. Plot the density distribution (12.17) as a function of r for fixed σ and large d.

One can actually separate Gaussians where the centers are much closer by adopting the singular value decomposition algorithm. In fact it is the basic idea of all the dimension reduction techniques. The projection $\mathbf{f}: \mathrm{R}^d \to \mathrm{R}^s$ is as follows: Pick s Gaussian vectors $\mathbf{a}^{(1)}, \mathbf{a}^{(2)}, \cdots, \mathbf{a}^{(s)}$ in R^d with unit-variance coordinates. For any vector $\mathbf{b}$, define the projection $\mathbf{f}(\mathbf{b})$ as $\mathbf{f}(\mathbf{b}) = (\mathbf{a}^{(1)}\cdot\mathbf{b}, \mathbf{a}^{(2)}\cdot\mathbf{b}, \cdots, \mathbf{a}^{(s)}\cdot\mathbf{b})^{\mathrm{T}}$. The projection function $\mathbf{f}$ is the vector of inner products of $\mathbf{b}$ with the $\mathbf{a}^{(i)}$. It could be shown that with high probability that $\|\mathbf{f}(\mathbf{b})\| \approx \sqrt{s}\|\mathbf{b}\|$. For any two vectors $\mathbf{b}_1$ and $\mathbf{b}_2$, we have $\mathbf{f}(\mathbf{b}_1 - \mathbf{b}_2) = \mathbf{f}(\mathbf{b}_1) - \mathbf{f}(\mathbf{b}_2)$. Thus in order to estimate the distance $\|\mathbf{b}_1 - \mathbf{b}_2\|$ between two vectors in R^d, it is suffices to calculate $\|\mathbf{f}(\mathbf{b}_1) - \mathbf{f}(\mathbf{b}_2)\| = \|\mathbf{f}(\mathbf{b}_1 - \mathbf{b}_2)\|$ in the s-dimensional space, since the factor of $\sqrt{s}$ is known. The reason that distances increase when we project to a lower-dimensional space is that the vector $\mathbf{a}^{(i)}$ is not unit length. We state the above description in more accurate form: Let $\mathbf{b}$ be a fixed vector in R^d and let $\mathbf{f}$ be defined as above. There exists constant $f > 0$ such that

$$P\left(\left|\|\mathbf{f}(\mathbf{b})\| - \sqrt{s}\|\mathbf{b}\|\right| \ge \epsilon\sqrt{s}\|\mathbf{b}\|\right) \le 3e^{-fs\epsilon^2}, \quad 0 < \epsilon < 1, \qquad (12.20)$$

where the probability is taken over the random draws of the vector $\mathbf{a}^{(i)}$ used to construct the function $\mathbf{f}$. This conclusion is called the random projection theorem. Without of loss of generality we may assume that $\mathbf{b}$ is a unit vector. The sum of independent normally distributed real variables is also normally distributed where the mean and the variance are the sums of the individual means and the variances. Since $\mathbf{a}^{(i)}\cdot\mathbf{b} = \sum_{j=1}^d a_j^{(i)}b_j$, the random variable $\mathbf{a}^{(i)}\cdot\mathbf{b}$ has Gaussian density with zero mean and unit variance, more specifically one has,

$$\mathrm{var}\left[\mathbf{a}^{(i)}\cdot\mathbf{b}\right] = \mathrm{var}\left[\sum_{j=1}^{d} a_j^{(i)}b_j\right] = \sum_{j=1}^{d} b_j^2\,\mathrm{var}\left[a_j^{(i)}\right] = \sum_{j=1}^{d} b_j^2 = 1. \qquad (12.21)$$

In addition since $\mathbf{a}^{(1)}\cdot\mathbf{b}, \cdots, \mathbf{a}^{(s)}\cdot\mathbf{b}$ are independent Gaussian random variables, $\mathbf{f}(\mathbf{b})$ is a random vector from a s-dimensional spherical Gaussian with unit variance in each direction, and via the Gaussian annulus theorem one can prove the random projection theorem.

The random projection theorem establishes the fact that the probability of the length of the projection of a single vector differing significantly from its expectation value is exponentially small in s, namely the dimension of the target subspace. By a union bound the probability that any of $\mathcal{O}(m^2)$ pairwise differences $\|\mathbf{a}^{(i)} - \mathbf{a}^{(j)}\|$ among the m vectors $\mathbf{a}^{(i)}$ differs significantly from their expected values is small, provided that $s \ge 3\log m/f\epsilon^2$. The random projection preserves all relative pairwise distances between points in a set of m points with high probability, i.e., for any $0 < \epsilon < 1$ and any integer m, let $s \ge 3\log m/f\epsilon^2$ with f the constant number introduced above, for any set of m points in R^d the random projection $\mathbf{f}: \mathrm{R}^d \to \mathrm{R}^s$ defined has the property that for all pairs of points $\mathbf{a}^{(i)}$ and $\mathbf{a}^{(j)}$, with probability at least $1 - 3/2m$ that,

$$(1-\epsilon)\sqrt{s}\left\|\mathbf{a}^{(i)} - \mathbf{a}^{(j)}\right\| \le \left\|\mathbf{f}(\mathbf{a}^{(i)}) - \mathbf{f}(\mathbf{a}^{(j)})\right\| \le (1+\epsilon)\sqrt{s}\left\|\mathbf{a}^{(i)} - \mathbf{a}^{(j)}\right\|. \qquad (12.22)$$

This result is called the Johnson–Lindenstrauss (JL) lemma, which could be directly proved by the random projection theorem. For the nearest neighbor problem, if the dataset has m_1 points and m_2 queries are expected during the lifetime of the algorithm, take $m = m_1 + m_2$ and project the dataset to a random s-dimensional space, for s as in the Johnson–Lindenstrauss lemma. On receiving a query project the query to the same subspace and compute nearby dataset points. The Johnson–Lindenstrauss lemma tells that with high probability this will yield the right answer. Note that the exponentially small in s probability is useful in making s dependent on $\log m$ instead of m.

EXERCISE 19: Prove the JL lemma via the random projection theorem, and prove the latter by the Gaussian annulus theorem.

EXERCISE 20: Generate 10^6 points on the surface of a 5d sphere and create a histogram of all distances between the pairs of points.

XIII. SINGULAR VALUE DECOMPOSITION: BASIS

The "singular value decomposition (SVD)" method plays a fundamental role in many optimization problems in machine learning issues. The introduction in this section is conceptual and is for practical use of the SVD instead of the numerical algorithms behind the decomposition.[9] For any real matrix $\mathbf{A} \in \mathrm{R}^{m\times d}$, the SVD of $\mathbf{A}$ is given by $\mathbf{A} = \mathbf{U}\vec{\Sigma}\mathbf{V}^{\mathrm{T}}$, where $\mathbf{U} \in \mathrm{R}^{m\times d}, \vec{\Sigma} \in \mathrm{R}^{d\times d}, \mathbf{V} \in \mathrm{R}^{d\times d}$. Moreover, the matrices $\mathbf{V}$ is always orthogonal in the sense that $\sum_{j=1}^{d} V_{jk}V_{jk'} = \delta_{kk'}, 1 \le k, k' \le d$, i.e., $\mathbf{V}^{\mathrm{T}}\mathbf{V} = \mathbf{1}$ and since the $\mathbf{V}$ is square it also indicates $\mathbf{V}\mathbf{V}^{\mathrm{T}} = \mathbf{1}$. On the other if $m \ge d$ the columns of $\mathbf{U}$ are also orthogonal,

$$\sum_{i=1}^{m} U_{ik}U_{ik'} = \delta_{kk'}, \quad 1 \le k, k' \le d, \tag{13.1}$$

i.e., $\mathbf{U}^{\mathrm{T}}\mathbf{U} = \mathbf{1}$. Furthermore for the situation $m < d$, the following two possibilities emerge: (a) The σ_j's for $j = m+1 \sim d$ are zero; or (b) The corresponding columns of $\mathbf{U}$ are zero and the relation (13.1) holds for $1 \le k, k' \le m$. See Fig. 22 for the sketch of the SVD, here the matrix $\mathbf{A}$ is assumed to have rank d. Instead if the rank of the matrix $\mathbf{A}$ has rank r, then $\mathbf{U} \in \mathrm{R}^{m\times r}, \vec{\Sigma} \in \mathrm{R}^{r\times r}, \mathbf{V} \in \mathrm{R}^{d\times r}$. Of course $r \le \min\{m, d\}$.

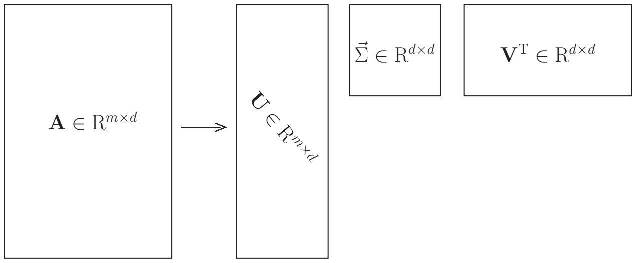

$\mathbf{A} \in \mathrm{R}^{m\times d}$
$\mathbf{U} \in \mathrm{R}^{m\times d}$
$\vec{\Sigma} \in \mathrm{R}^{d\times d}$
$\mathbf{V}^{\mathrm{T}} \in \mathrm{R}^{d\times d}$

Fig. 22: Sketch of the singular value decomposition.

The meanings of m and d could be demonstrated more obviously using the set of linear equations,

$$\begin{cases} a_1^{(1)}x_1 + a_2^{(1)}x_2 + \cdots + a_d^{(1)}x_d = b_1, \\ a_1^{(2)}x_1 + a_2^{(2)}x_2 + \cdots + a_d^{(2)}x_d = b_2, \\ \vdots \\ a_1^{(m)}x_1 + a_2^{(m)}x_2 + \cdots + a_d^{(m)}x_d = b_m, \end{cases} \tag{13.2}$$

i.e., there are totally d unknowns $x_1 \sim x_d$ with m equations. The above set of linear equations could be rewritten in the matrix form $\mathbf{A}\mathbf{x} = \mathbf{b}$, with

$$\mathbf{A} = \begin{pmatrix} a_1^{(1)} & a_2^{(1)} & \cdots & a_d^{(1)} \\ a_1^{(2)} & a_2^{(2)} & \cdots & a_d^{(2)} \\ \vdots & \vdots & \ddots & \vdots \\ a_1^{(m)} & a_2^{(m)} & \cdots & a_d^{(m)} \end{pmatrix}, \quad \mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_d \end{pmatrix}, \quad \mathbf{b} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{pmatrix}. \tag{13.3}$$

The the situation $m > d$ corresponds to that there are more equations than unknowns, i.e., the system is over-determined, and the one with $m < d$ corresponds to that there are few equations than unknown, i.e., the system is under-determined. If $m < d$, or if $m = d$ but the equations are degenerate (e.g., the problem contains the two equations $2x + y = 1$ and $4x + 2y = 2$), then there are effectively fewer equations than unknowns. In this case there can be either no solution, or else more than one solution vector $\mathbf{x}$. In the latter event, the solution space consists of a particular solution $\mathbf{x}_p$ added to any linear combination of (typically) $d - m$ vectors (which are said to be in the null-space of the matrix $\mathbf{A}$). The task of finding the solution space of $\mathbf{A}$ then naturally involves the SVD of $\mathbf{A}$. If there are more equations than unknowns, $m > d$, there is in general no solution vector $\mathbf{x}$ to equation (13.2), and it happens frequently, however, that the best "compromise" solution is sought the one that comes closest to satisfying all equations simultaneously. If closeness is defined in the least-squares sense, i.e., that the sum of the squares of the differences between the left and right hand sides of equation (13.2) be minimized, then the over-determined linear problem reduces to a (usually) solvable linear problem, called the linear least squares problem. The reduced set of equations to be solved can be written as $\mathbf{A}^{\mathrm{T}}\mathbf{A}\mathbf{x} = \mathbf{A}^{\mathrm{T}}\mathbf{b}$.

The equation $\mathbf{A}\mathbf{x} = \mathbf{b}$ defines $\mathbf{A}$ as a linear mapping from an d-dimensional vector space (for $\mathbf{x}$) to an m-dimensional one (for $\mathbf{b}$). But the map might be able to reach only a lesser-dimensional subspace of the full m-dimensional one. That subspace is called the range of the matrix $\mathbf{A}$. The dimension of the range is called the rank of $\mathbf{A}$, denoted by rank($\mathbf{A}$). The rank of $\mathbf{A}$ is equal to its number of linearly independent columns, and also (perhaps less obviously) to its number of linearly independent rows. If $\mathbf{A}$ is not identically zero, its rank is at least 1, and at most the minimum of m and d, i.e., $\min(m, d)$, which is an elementary conclusion from basic linear algebra. Sometimes there are nonzero vectors $\mathbf{x}$ that are mapped to zero by the matrix $\mathbf{A}$, i.e., $\mathbf{A}\mathbf{x} = \mathbf{0}$. The space of such vectors (a subspace of the d-dimensional space that the vector $\mathbf{x}$ lives in) is called the null-space of $\mathbf{A}$, and its dimension is called the nullity of the matrix $\mathbf{A}$, denoted by null($\mathbf{A}$), and that the nullity can have any value from zero to d. The rank-nullity theorem in matrix algebra states that for any matrix $\mathbf{A}$, the rank plus the nullity is d (the number of columns), i.e., $\mathrm{rank}(\mathbf{A}) + \mathrm{null}(\mathbf{A}) = d$. An important special case is $m = d$ such that the matrix $\mathbf{A}$ is square with dimension d. Moreover, if the rank of $\mathbf{A}$ is d, its maximum possible value, then the nullity of $\mathbf{A}$ is zero, i.e., the $\mathbf{A}$ is nonsingular and invertible: The equation $\mathbf{A}\mathbf{x} = \mathbf{b}$ has a unique solution for any vector $\mathbf{b}$, and only the zero vector is mapped to zero. The relation between the SVD and the null-space as well as the range of the matrix $\mathbf{A}$ is that: SVD explicitly constructs the orthogonal bases for the null-space and the range of a matrix. More specifically, the columns of $\mathbf{U}$ whose same-numbered elements σ_j are nonzero are an orthogonal set of basis vectors that span the range, and the columns of $\mathbf{V}$ whose same-numbered elements σ_j are zero are an orthogonal basis for the null-space.

For the situation $m = d$, i.e., the matrix $\mathbf{A}$ is square, the matrices $\mathbf{U}$ and $\mathbf{V}$ are also square, the SVD of $\mathbf{A}$ is simplified and its inverse is

$$\mathbf{A}^{-1} = \mathbf{V}\mathrm{diag}\left(1/\sigma_1, 1/\sigma_2, \cdots, 1/\sigma_d\right)\mathbf{U}^{\mathrm{T}}. \tag{13.4}$$

Here the σ_j's are actually the eigenvalues of the matrix $\mathbf{A}$. One of the main problems is that if one or several of the σ_j's are zero or very small near zero (It is necessary to remember that the condition number of a square matrix $\mathbf{A}$ is defined as the ratio between its maximum and minimum eigenvalues (in magnitude), i.e., $\kappa(\mathbf{A}) = \max_j |\sigma_j| / \min_j |\sigma_j|$, which becomes infinite or very large if σ_d is zero or very close to zero). We now discuss the solution of the equation $\mathbf{A}\mathbf{x} = \mathbf{b}$ in the situation with $\mathbf{A}$ a square matrix and $\mathbf{b}$ nonzero. The discussion is classified as,

(a) If the nonzero vector $\mathbf{b}$ is in the range of the matrix $\mathbf{A}$, the singular set of equations does have a solution $\mathbf{x}$ and in fact it has more than one solution since any vector in the null-space $\mathbf{x}'$ (any column of $\mathbf{V}$ with a corresponding zero σ_j) can be added to $\mathbf{x}$ in any linear combination, e.g., $\mathbf{x} + \mathbf{x}'$ is still a solution of $\mathbf{A}(\mathbf{x} + \mathbf{x}') = \mathbf{b}$. In fact one has $\mathbf{A}(\mathbf{x} + \mathbf{x}') = \mathbf{A}\mathbf{x} + \mathbf{A}\mathbf{x}' = \mathbf{A}\mathbf{x} = \mathbf{b}$. If we want to single out one particular member of this solution set of vectors as a representative,

[9] A general reference on this topic is, G. Golub and Van C. Loan, *Matrix Computations*, 2013, John Hopkins University.

we might want to pick the one with the smallest length $\|\mathbf{x}\|^2$. Here is how to find that vector using SVD: Simply replace $1/\sigma_j$ by zero if $\sigma_j = 0$, and then obtain the solution via,[10]

$$\mathbf{x} = \mathbf{V}\mathrm{diag}\left(1/\sigma_1, 1/\sigma_2, \cdots, 1/\sigma_d\right)\mathbf{U}^{\mathrm{T}}\mathbf{b} \tag{13.5}$$

from right to left. The reason is as follows:

$$\|\mathbf{x}+\mathbf{x}'\| = \left\|\mathbf{V}\vec{\Sigma}^{-1}\mathbf{U}^{\mathrm{T}}\mathbf{b}+\mathbf{x}'\right\| = \left\|\vec{\Sigma}^{-1}\mathbf{U}^{\mathrm{T}}\mathbf{b}+\mathbf{V}^{\mathrm{T}}\mathbf{x}'\right\|, \tag{13.6}$$

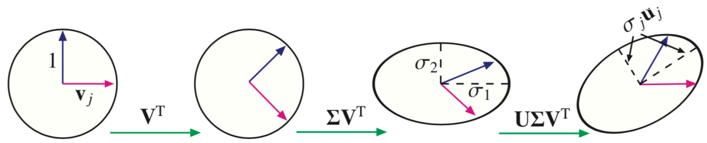
where the first equality follows from the SVD solution (13.5) and the second and third ones from the orthogonality of the matrix $\mathbf{V}$. If one examines the two terms that make up the summation on the right hand side, one immediately finds that the first one has nonzero j-components only where $\sigma_j \neq 0$ while the second one since $\mathbf{x}'$ is in the null-space has nonzero j-components only where $\sigma_j = 0$. Thus the minimum length is obtained when $\mathbf{x}' = \mathbf{0}$.

(b) On the other hand if $\mathbf{b}$ is not in the range of the singular matrix $\mathbf{A}$, then the set of $\mathbf{Ax} = \mathbf{b}$ has no solution. However, the expression (13.5) can still be used to construct an approximation "solution" vector $\mathbf{x}$, and this vector $\mathbf{x}$ does not exactly solve the equation $\mathbf{Ax} = \mathbf{b}$. But among all possible vectors $\mathbf{x}$, it does the closest possible job in the least-squares sense. In fact, one could prove that the expression (13.5) minimizes the difference between the $\mathbf{Ax}$ and $\mathbf{b}$, i.e., the (13.5) is the $\mathrm{argmin}_{\mathbf{x}}\,\|\mathbf{Ax}-\mathbf{b}\|$ (the quantity $\|\mathbf{Ax}-\mathbf{b}\|$ is called the residual of the solution). More specifically, suppose one modifies $\mathbf{x}$ of (13.5) by adding some arbitrary $\mathbf{x}'$, and then $\mathbf{Ax}-\mathbf{b}$ is modified by adding some $\mathbf{b}' = \mathbf{Ax}'$. Obviously the $\mathbf{b}'$ is in the range of the matrix $\mathbf{A}$, and

$$\begin{aligned}\|\mathbf{Ax}'-\mathbf{b}+\mathbf{b}'\| &= \left\|\left(\mathbf{U}\vec{\Sigma}\mathbf{V}^{\mathrm{T}}\right)\left(\mathbf{V}\vec{\Sigma}^{-1}\mathbf{U}^{\mathrm{T}}\mathbf{b}\right)-\mathbf{b}+\mathbf{b}'\right\| \\ &= \left\|\left(\vec{\Sigma}\vec{\Sigma}^{-1}-\mathbf{1}\right)\mathbf{U}^{\mathrm{T}}\mathbf{b}+\mathbf{U}^{\mathrm{T}}\mathbf{b}'\right\|, \end{aligned}\tag{13.7}$$

with $\vec{\Sigma}\vec{\Sigma}^{-1}-\mathbf{1}$ a diagonal matrix having nonzero j-components only for $\sigma_j = 0$, while $\mathbf{U}^{\mathrm{T}}\mathbf{b}'$ has nonzero j-components only for $\sigma_j \neq 0$, since $\mathbf{b}'$ lies in the range of the matrix $\mathbf{A}$. Therefore the minimum is obtained for $\mathbf{b}' = \mathbf{0}$, i.e., the (13.5) minimizes the residual $|\mathbf{Ax}-\mathbf{b}|$.

XIV. SINGULAR VALUE DECOMPOSITION: BEST-FIT SCHEME



1
$\mathbf{v}_j$
$\mathbf{V}^{\mathrm{T}}$
σ_2
σ_1
$\Sigma\mathbf{V}^{\mathrm{T}}$
$\sigma_j\mathbf{u}_j$
$\mathbf{U}\Sigma\mathbf{V}^{\mathrm{T}}$

Fig. 23: Geometrical meaning of the SVD.

We discuss the geometrical meaning of the SVD as shown in Fig. 23 for the 2-dimensional situation. The original space is spanned by the sphere with axes $\mathbf{v}_j$ having unit length. When acted with $\mathbf{V}^{\mathrm{T}}$ the axes are rotated since the matrix $\mathbf{V}$ is orthogonal, however the relative relation between the two axes is unchanged. Next, when applying the matrix $\vec{\Sigma}$, the axes are elongated or shrank according to the magnitude of the singular values, consequently the sphere becomes an elliptic, here $\sigma_1 > \sigma_2 > 0$ is assumed. The action of the matrix $\mathbf{U}$ rotates the elliptic again and in the mean while the original axes $\mathbf{v}_j$ becomes $\sigma_j\mathbf{u}_j$, since the SVD of the matrix $\mathbf{A}$ could be rewritten as $\mathbf{AV} = \mathbf{V}\vec{\Sigma}$, or $\mathbf{Av}_j = \sigma_j\mathbf{u}_j$, in component. In other words the matrix $\mathbf{A}$ written in the SVD form transforms the $\mathbf{v}_j$ into $\sigma_j\mathbf{u}_j$, both of which are orthogonal, i.e., $\mathbf{v}_1 \perp \mathbf{v}_2$ and $\mathbf{u}_1 \perp \mathbf{u}_2$. Moreover, one also has $\mathbf{A}^{\mathrm{T}}\mathbf{u}_j = \sigma_j\mathbf{v}_j$, by combining these two one essentially obtains $\mathbf{A}^{\mathrm{T}}\mathbf{Av}_j = \sigma_j\mathbf{A}^{\mathrm{T}}\mathbf{u}_j = \sigma_j^2\mathbf{v}_j$. Using the vectors $\mathbf{u}_i$ and $\mathbf{v}_i$ the SVD of $\mathbf{A}$ is written in the form, $\mathbf{A} = \sum_{i=1}^{d}\sigma_i\mathbf{u}_i\mathbf{v}_i^{\mathrm{T}}$, with $\mathbf{u}_i \in \mathrm{R}^m$ and $\mathbf{v}_i \in \mathrm{R}^d$, where some of the σ_i's maybe zero. The shape of the matrix constructed on the right hand side is $\mathrm{R}^{m\times d}$, and the σ_i's play the role of the weights of the approximation. If a small number s is used instead of d, we obtain the s-rank approximation of $\mathbf{A}$ namely, $\mathbf{A}_{\mathrm{eff}} = \mathbf{A}_s \approx \sum_{i=1}^{s}\sigma_i\mathbf{u}_i\mathbf{v}_i^{\mathrm{T}}$. The rows of $\mathbf{A}_s$ are the projections of the rows of $\mathbf{A}$ onto the subspace spanned by the first r singular vectors of $\mathbf{A}$.

The SVD for matrix $\mathbf{A}$ could be rewritten to obtain the component of the matrix $\mathbf{A}$ as $a_j^{(i)} = \sum_{k=1}^{d}\sigma_k U_{ik}V_{jk}$. For squared matrix $\mathbf{A}$, we then have $\mathbf{A} = \sum_{j=1}^{d}\sigma_j\mathbf{u}_j\mathbf{u}_j^{\mathrm{T}}$. If one encounters a situation where most of the singular values σ_j of a matrix $\mathbf{A}$ are very small (in magnitude), then the $\mathbf{A}$ will be well approximated by only a few terms in the above summation, i.e., $r \to d$ with $r \ll d$, indicating that one has to store only a few columns of $\mathbf{U}$ and of $\mathbf{V}$ (with the same k indices) and one could be able to recover with good accuracy the whole matrix.

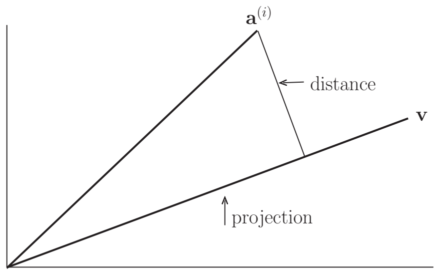


Fig. 24: Projection of the point $\mathbf{a}$ onto the line through the origin in the direction of $\mathbf{v}$.

It is the main mathematical principal behinds the principal component analysis, and the corresponding approximation is called the best-fit subspace. The SVD is very useful in this sense it could find a low-rank approximation to $\mathbf{A}$, and for any s the SVD of $\mathbf{A}$ gives the best rank-s approximation of $\mathbf{A}$ in a well-defined sense. Let's discuss it in more details. Consider projecting a point $\mathbf{a}^{(i)} = (a_1^{(i)}, \cdots, a_d^{(i)})^{\mathrm{T}} \in \mathrm{R}^d$ onto a line through the origin, see Fig. 24. Then

$$\begin{aligned} a_1^{(i),2} + \cdots + a_d^{(i),2} =& [\text{length of projection}]^2 \\ &+ [\text{distance of point to line}]^2, \end{aligned}\tag{14.1}$$

consequently,

$$\begin{aligned} [\text{distance of point to line}]^2 =& -[\text{length of projection}]^2 \\ &+ a_1^{(i),2} + \cdots + a_d^{(i),2}. \end{aligned}\tag{14.2}$$

Since $\sum_{i=1}^{m}(a_1^{(i),2} + a_2^{(i),2} + \cdots + a_d^{(i),2})$ is a constant independent of the line, minimizing the sum of squares of the distance to the line is equivalent to maximizing the sum of the squares of the lengths of the projections onto the line. Similarly for the best-fit subspace, maximizing the sum of the squared lengths of the projections onto the subspace minimizes the sum of the squared distances to the subspace. Thus we have two interpretations of the best-fit subspace. The first one is that it minimizes the sum of squared distances of the data points to it. This interpretation and its use are akin to the notion of least-squares method. The second interpretation of best-fit subspace is that it maximizes the sum of projections squared of the data points on it. This tells that the subspace contains the maximum content of data among all subspaces of the same dimension.

Consider the rows of $\mathbf{A}$ as m points in a d-dimensional space. Consider the best-fit line through the origin, and let $\mathbf{v}$ be a unit vector along this line. The length of the projection of $\mathbf{a}^{(i)}$, the ith row of $\mathbf{A}$ (by transposing), onto $\mathbf{v}$ is simply as $\|\mathbf{a}^{(i)}\cdot\mathbf{v}\|$. From this we see that the sum of the squared lengths of the projections is $\|\mathbf{Av}\|^2$. The best-fit line is the one maximizing $\|\mathbf{Av}\|^2$ and hence minimizing the sum of the squared distances of the points to the line. With this in mind, the first singular vector of $\mathbf{A}$ could be defined as,

$$\mathbf{v}_1 = \underset{\|\mathbf{v}\|=1}{\mathrm{argmax}}\,\|\mathbf{Av}\|. \tag{14.3}$$

[10] Eq. (13.5) is essentially very general in the sense that if no σ_j's are zero, it solves a non-singular system of linear equations. If some σ_j's are zero and their reciprocals are made zero (i.e., are zeroized), then it gives a "best" solution, and either the one of shortest length among many, or the one of minimum residual when no exact solution exists. Eq. (13.4) with the singular $1/\sigma_j$ zeroized is called the Moore–Penrose inverse or the pseudo-inverse of the matrix $\mathbf{A}$, denoted as $\mathbf{A}^{+}$.

The value $\sigma_1(\mathbf{A}) = \|\mathbf{A}\mathbf{v}_1\|$ is the the first singular value of $\mathbf{A}$. Note that $\sigma_1^2 = \sum_{i=1}^m (\mathbf{a}^{(i)} \cdot \mathbf{v}_1)^2$ is the sum of the squared lengths of the projections of the points onto the line determined by $\mathbf{v}_1$. If the data points were all either on a line or close to a line, intuitively $\mathbf{v}_1$ should give us the direction of that line. It is possible that data points are not close to one line, but lie close to a two-dimensional subspace or more generally a low-dimensional space. Suppose we have an algorithm for finding $\mathbf{v}_1$. How do we use this to find the best-fit two-dimensional plane or more generally the best-fit r-dimensional space? We have the following greedy algorithm: The second singular vector $\mathbf{v}_2$ is defined by the best-fit line perpendicular to $\mathbf{v}_1$, $\mathbf{v}_2 = \mathrm{argmax}_{\mathbf{v}\perp\mathbf{v}_1, \|\mathbf{v}\|=1} \|\mathbf{A}\mathbf{v}\|$. The $\sigma_2(\mathbf{A}) = \|\mathbf{A}\mathbf{v}_2\|$ is called the second singular value of the matrix $\mathbf{A}$. The third singular vector $\mathbf{v}_3$ and the third singular value are defined very similarly, namely $\mathbf{v}_3 = \mathrm{argmax}_{\mathbf{v}\perp\mathbf{v}_1,\mathbf{v}_2,\|\mathbf{v}\|=1} \|\mathbf{A}\mathbf{v}\|$, and $\sigma_3(\mathbf{A}) = \|\mathbf{A}\mathbf{v}_3\|$.

Note that the d-dimensional vector $\mathbf{A}\mathbf{v}_i$ is a list of lengths (with signs) of the projections of the rows of $\mathbf{A}$ onto $\mathbf{v}_i$. Think of $\|\mathbf{A}\mathbf{v}_i\| = \sigma_i(\mathbf{A})$ as the component of the matrix $\mathbf{A}$ along $\mathbf{v}_i$. For this interpretation to make sense, it should be true that adding up the squares of the components of $\mathbf{A}$ along each of the $\mathbf{v}_i$ gives the square of the "whole content of $\mathbf{A}$". This is indeed the situation and is the matrix analogy of decomposing a vector into its components along orthogonal directions. Consider one row $\mathbf{a}^{(j)}$, for instance, since $\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_d$ span the space of all rows of $\mathbf{A}$, one has $\mathbf{a}^{(j)} \cdot \mathbf{v} = 0$ for all $\mathbf{v}$ perpendicular to $\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_d$. Thus for each row $\mathbf{a}^{(j)}$, the $\sum_{i=1}^d (\mathbf{a}^{(j)} \cdot \mathbf{v}_i)^2 = \|\mathbf{a}^{(j)}\|^2$. Summing over all row j, one obtains,

$$\begin{aligned}\sum_{j=1}^m \left\|\mathbf{a}^{(j)}\right\|^2 &= \sum_{j=1}^m \sum_{i=1}^d \left(\mathbf{a}^{(j)} \cdot \mathbf{v}_i\right)^2 = \sum_{i=1}^d \sum_{j=1}^m \left(\mathbf{a}^{(j)} \cdot \mathbf{v}_i\right)^2 \\ &= \sum_{i=1}^d \|\mathbf{A}\mathbf{v}_i\|^2 = \sum_{i=1}^d \sigma_i^2(\mathbf{A}). \end{aligned} \tag{14.4}$$

On the other hand,

$$\sum_{j=1}^m \|\mathbf{a}^{(j)}\|^2 = \sum_{j=1}^m \sum_{k=1}^d a_k^{(j),2}, \tag{14.5}$$

the sum of squares of all entries of $\mathbf{A}$. Thus the sum of squares of the singular values of $\mathbf{A}$ is indeed the square of the "whole content of $\mathbf{A}$", the sum of squares of all the entries. This value is the Frobenius norm of the matrix $\mathbf{A}$, i.e., $\|\mathbf{A}\|_\mathrm{F}$. The vectors $\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_d$ are the right singular vectors of the matrix $\mathbf{A}$, and from the above discussion $\|\mathbf{A}\|_\mathrm{F}^2 = \sum_{i=1}^d \sigma_i^2$. The vectors $\mathbf{A}\mathbf{v}_i$ form a fundamental set of vectors and we normalize them to length 1 by $\mathbf{u}_i = \mathbf{v}_i/\sigma_i(\mathbf{A})$. In fact the $\mathbf{u}_i$ similarly maximizes $\|\mathbf{u}^\mathrm{T}\mathbf{A}\|$ over all $\mathbf{u}$ perpendicular to $\mathbf{u}_1, \cdots, \mathbf{u}_{i-1}$. These vectors are the left singular vectors, which are also orthogonal.

Denote i the smallest integer such that $\mathbf{u}_i$ is not orthogonal to some other $\mathbf{u}_j$ and assume $\mathbf{u}_i^\mathrm{T}\mathbf{u}_j = \delta > 0$. Next one defines the vector $\mathbf{v}_i' = (\mathbf{v}_i + \epsilon\mathbf{v}_j)/\|\mathbf{v}_i + \epsilon\mathbf{v}_j\|$ by $\epsilon > 0$. Consequently $\mathbf{A}\mathbf{v}_i' = (\sigma_i\mathbf{u}_i + \epsilon\sigma_j\mathbf{u}_j)/\sqrt{1+\epsilon^2}$, which has length at least as large as its component along $\mathbf{u}_i$,

$$\mathbf{u}_i^\mathrm{T}\left(\frac{\sigma_i\mathbf{u}_i + \epsilon\sigma_j\mathbf{u}_j}{\sqrt{1+\epsilon^2}}\right) = \frac{\sigma_i + \epsilon\delta\sigma_j}{\sqrt{1+\epsilon^2}} > \sigma_i + \epsilon\delta\sigma_j - \frac{1}{2}\epsilon^2\sigma_i - \frac{1}{2}\epsilon^3\delta\sigma_j, \tag{14.6}$$

which is greater than σ_i for sufficiently small ϵ, and this is a contradiction since $\mathbf{v}_i + \epsilon\mathbf{v}_j$ is orthogonal to $\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_{i-1}$ due to $j > i$. Using the fact that all the right singular vectors are also orthogonal with each other, we calculate the residual of the SVD approximation. By denoting $\mathbf{v}$ the top singular vector of $\mathbf{A} - \mathbf{A}_s$ and expressing it as a combination of $\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_d$, i.e., $\mathbf{v} = \sum_{j=1}^d c_j\mathbf{v}_j$, one has

$$\begin{aligned}\|(\mathbf{A} - \mathbf{A}_s)\mathbf{v}\|^2 &= \left\|\sum_{i=s+1}^d \sigma_i\mathbf{u}_i\mathbf{v}_i^\mathrm{T} \sum_{j=1}^d c_j\mathbf{v}_j\right\|^2 = \left\|\sum_{i=s+1}^d \sum_{j=1}^d \sigma_i c_j \mathbf{u}_i \delta_{ij}\right\|^2 \\ &= \left\|\sum_{i=s+1}^d c_i\sigma_i\mathbf{u}_i\right\|^2 = \sum_{i=s+1}^d c_i^2\sigma_i^2. \end{aligned} \tag{14.7}$$

The $\mathbf{v}$ maximizing this last quantity subject to the constraint that $\|\mathbf{v}\|^2 = \sum_{i=1}^d c_i^2 = 1$ occurs when $c_{s+1} = 1$ and the rest of the coefficients are zero. Consequently

$$\|\mathbf{A} - \mathbf{A}_s\|_2^2 = \sigma_{s+1}^2. \tag{14.8}$$

Very similarly one can prove that for the 2-norm of a matrix that $\|\mathbf{A} - \mathbf{A}_s\|_2 \le \|\mathbf{A} - \mathbf{B}\|_2$, where $\mathbf{B}$ is a matrix of rank at most s. The 2-norm for a matrix is simply defined as $\|\mathbf{A}\|_2 = \max_{\|\mathbf{x}\|\le 1} \|\mathbf{A}\mathbf{x}\| = \sigma_1(\mathbf{A})$.

EXERCISE 21: Prove for any matrix $\mathbf{A}$ that $\sigma_s \le \|\mathbf{A}\|_\mathrm{F}/\sqrt{s}$ and there exists a matrix $\mathbf{B}$ of rank at most s such that $\|\mathbf{A} - \mathbf{B}\|_2 \le \|\mathbf{A}\|_\mathrm{F}/\sqrt{s}$.

If the number of the data points m is large while that of the dimension d is reasonable, the original computing of the SVD of the matrix $\mathbf{A} \in \mathrm{R}^{m\times d}$ is perhaps time consuming. We can construct another matrix $\mathbf{B}$ by $\mathbf{B} = \mathbf{A}^\mathrm{T}\mathbf{A} \in \mathrm{R}^{d\times d}$, and use the orthogonality of the $\mathbf{u}_i$'s to make progress. More specifically,

$$\mathbf{B} = \mathbf{A}^\mathrm{T}\mathbf{A} = \left(\sum_i \sigma_i\mathbf{v}_i\mathbf{u}_i^\mathrm{T}\right)\left(\sum_j \sigma_j\mathbf{u}_j\mathbf{v}_j^\mathrm{T}\right) = \sum_i \sigma_i^2\mathbf{v}_i\mathbf{v}_i^\mathrm{T}. \tag{14.9}$$

The matrix $\mathbf{B}$ is square and symmetric, and has the same left-singular and right-singular vectors. Particularly, one has $\mathbf{B}\mathbf{v}_j = \sum_i \sigma_i^2\mathbf{v}_i\mathbf{v}_i^\mathrm{T}\mathbf{v}_j = \sigma_j^2\mathbf{v}_j$, i.e., the left singular vector $\mathbf{v}_j$ is the eigenvector of the matrix $\mathbf{B}$ corresponding to the eigenvalue σ_j^2. If the matrix $\mathbf{A}$ is itself square and symmetric, it will have the same right-singular and left-singular vectors, and in this case there is no need to computing the matrix $\mathbf{B}$. Next, we calculate the square of the matrix $\mathbf{B}$,

$$\mathbf{B}^2 = \left(\sum_i \sigma_i^2\mathbf{v}_i\mathbf{v}_i^\mathrm{T}\right)\left(\sum_j \sigma_j^2\mathbf{v}_j\mathbf{v}_j^\mathrm{T}\right) = \sum_{i,j} \sigma_i^2\sigma_j^2\mathbf{v}_i\mathbf{v}_i^\mathrm{T}\mathbf{v}_j\mathbf{v}_j^\mathrm{T} = \sum_i \sigma_i^4\mathbf{v}_i\mathbf{v}_i^\mathrm{T}. \tag{14.10}$$

In computing the sth power of the matrix $\mathbf{B}$, all the cross-product terms are zero, and $\mathbf{B}^s = \sum_{i=1}^d \sigma_i^{2s}\mathbf{v}_i\mathbf{v}_i^\mathrm{T}$. If $\sigma_1 > \sigma_2$, i.e., the largest singular value is unique, then the first term in the summation dominates and $\mathbf{B}^s \to \sigma_1^{2s}\mathbf{v}_1\mathbf{v}_1^\mathrm{T}$. This means a close estimation on the first singular vector $\mathbf{v}_1$ can be computed by simply taking the first column of $\mathbf{B}^s$ and normalizing it to be a unit vector.

The above scheme is useful only in principal. For example, assume that the size of $\mathbf{A}$ is very large, e.g., a $10^6 \times 10^6$ matrix with 10^8 non-zero elements. Although $\mathbf{A}$ in this case is a sparse matrix it does not mean the square of $\mathbf{A}$ should also be sparse, and in the even worse case the $\mathbf{B} = \mathbf{A}^2$ may have all 10^{12} elements non-zero. It is impossible to even write down $\mathbf{B}$, let alone compute the product $\mathbf{B}^2$. Even if $\mathbf{A}$ is moderate in size, computing matrix products is costly in time and space. Instead of computing $\mathbf{B}^s$, select randomly a vector $\mathbf{x}$ and compute the product $\mathbf{B}^s\mathbf{x} = \mathbf{A}^\mathrm{T}\mathbf{A}\cdots\mathbf{A}^\mathrm{T}\mathbf{A}\mathbf{x}$ from right to left. The vector $\mathbf{x}$ could be expressed in terms of the singular vectors of $\mathbf{B}$ augmented to a full orthogonal basis as $\mathbf{x} = \sum_{i=1}^d c_i\mathbf{v}_i$ assuming the last few singular values of $\mathbf{A}$ are zero. Then,

$$\mathbf{B}^s\mathbf{x} \approx \left(\sigma_1^{2s}\mathbf{v}_1\mathbf{v}_1^\mathrm{T}\right)\left(\sum_{i=1}^d c_i\mathbf{v}_i\right) = \sigma_1^{2s}c_1\mathbf{v}_1. \tag{14.11}$$

Normalizing the resulting vector gives the first singular vector of $\mathbf{A}$. In order to compute s singular vectors, one selects randomly a vector $\mathbf{x}$ and finds an orthonomal basis of the space spanned by $\mathbf{x}, \mathbf{A}\mathbf{x}, \mathbf{A}\mathbf{x}^2, \cdots, \mathbf{A}^{s-1}\mathbf{x}$. Then compute $\mathbf{A}$ times each of the basis vectors, and find and orthonomal basis for the space spanned by the resulting vectors. Intuitively, one has applied $\mathbf{A}$ to a subspace rather than a single vector. One repeatedly applies $\mathbf{A}$ to the subspace, calculating an orthonomal basis after each application to prevent the subspace collapsing to the one-dimensional subsapce spanned by the first singular vector. The process essentially converges to the first s singular vectors.

EXERCISE 22: If $\mathbf{A} \in \mathrm{R}^{m\times d}$ is a matrix with nonnegative elements and $\sigma_1(\mathbf{A}) = \mathbf{x}^\mathrm{T}\mathbf{A}\mathbf{y} = \sum_{i,j} a_j^{(i)}x^{(i)}y_j$ with $\|\mathbf{x}\| = \|\mathbf{y}\| = 1$. Zero out all $x^{(i)}$ less than $1/2\sqrt{m}$ and all y_j less than $1/2\sqrt{d}$. Estimate the loss.

EXERCISE 23: Use the power method to computer the SVD of the matrix $\mathbf{A}$ with $a_1^{(1)} = 1, a_2^{(1)} = 2, a_1^{(2)} = 3$ and $a_2^{(2)} = 4$.

The author would like to thank the colleagues of the INSTITUTE OF INNOVATION *at Shadow Creator for long-time support and help.*