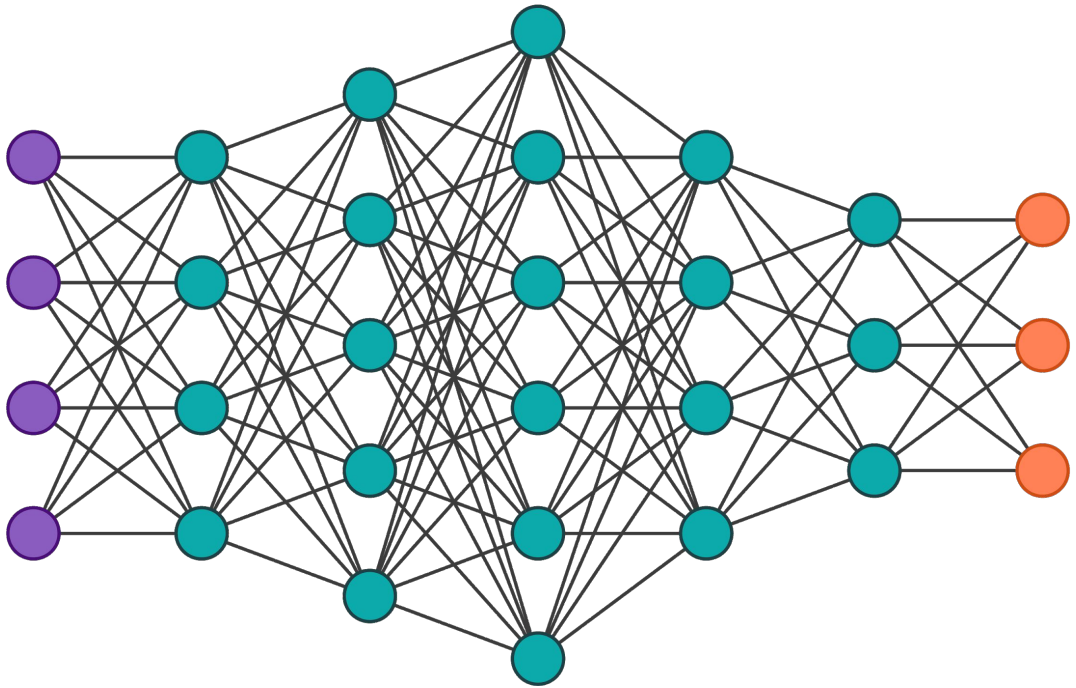
Chapter 2: Machine Learning, An Algorithmic Perspective

Chen Hu, Yi Hu & Sophie Sadler

2.1 TERMINOLOGY

$\mathbf{x} = (x_1, x_2, \ldots, x_m)$



$\mathbf{y} = \mathbf{y}(\mathbf{x}, \mathbf{W})$

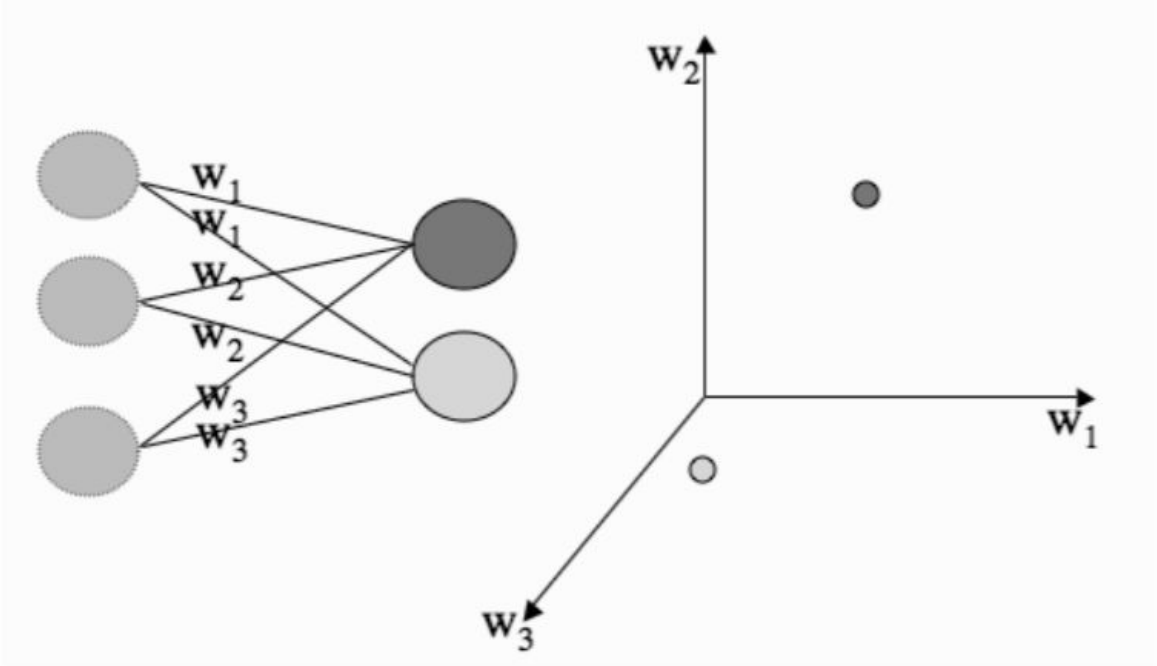
Targets, **t**

Error, E

Weights, **W**

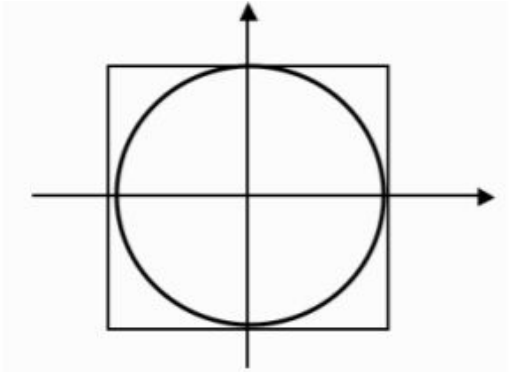
2.1.1 WEIGHT SPACE

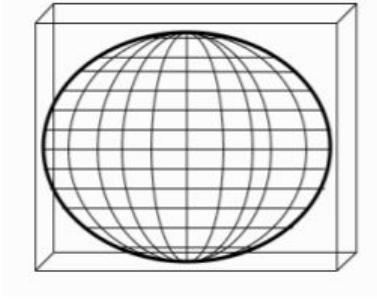
- Plot the input data.
- Plot the weights for each neuron.
- Then we can calculate distance between inputs and neurons.



2.1.2 CURSE OF DIMENSIONALITY

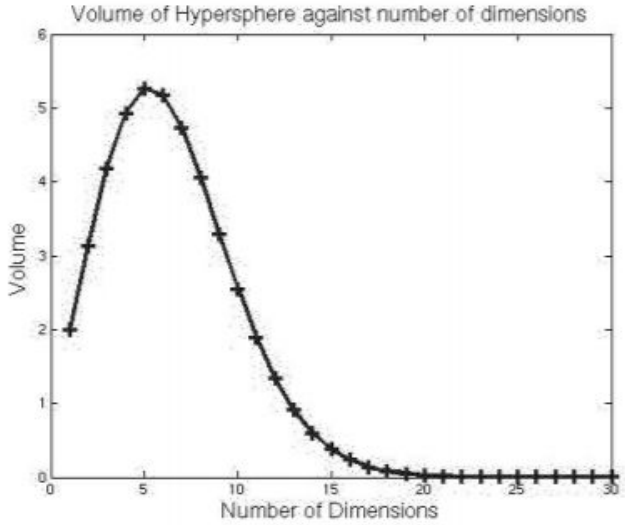
As the number of dimensions grows, the volume of the unit hypersphere doesn't grow with it.





Dimension	Volume
1	2.0000
2	3.1416
3	4.1888
4	4.9348
5	5.2636
6	5.1677
7	4.7248
8	4.0587
9	3.2985
10	2.5502

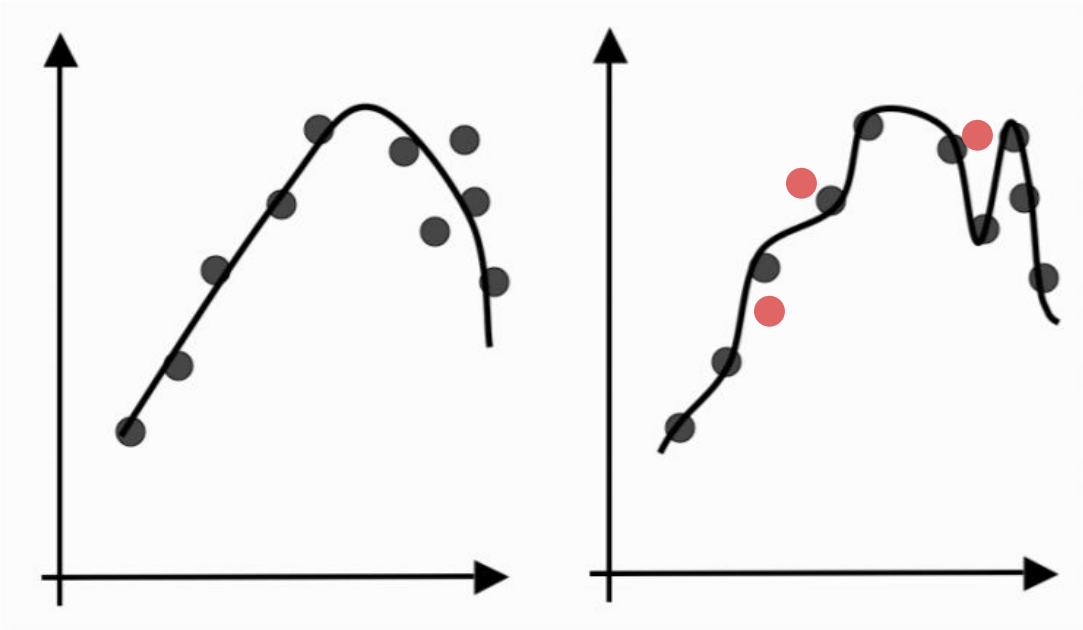
$$v_n = (2\pi/n)v_{n-2}$$



2.2 TESTING MACHINE LEARNING ALGORITHMS

- **Our aim**: to make a model which is good at predicting the outputs.
- So, we can compare *outputs* of the model to the *targets* of the training dataset.
- However, we also want to make good predictions for data not present in the training set.
- So, we use a test dataset.

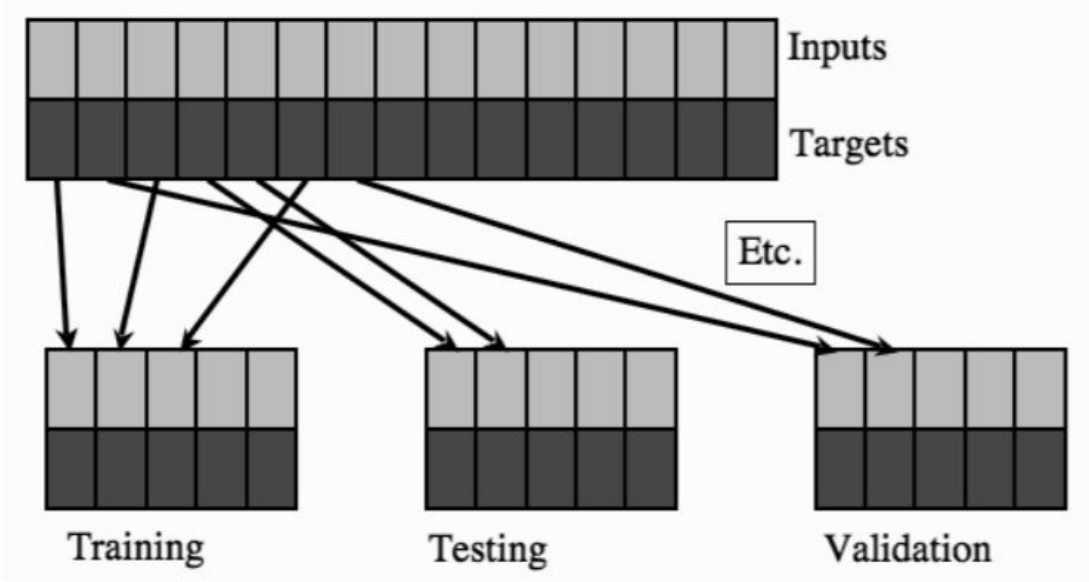
2.2.1 OVERFITTING



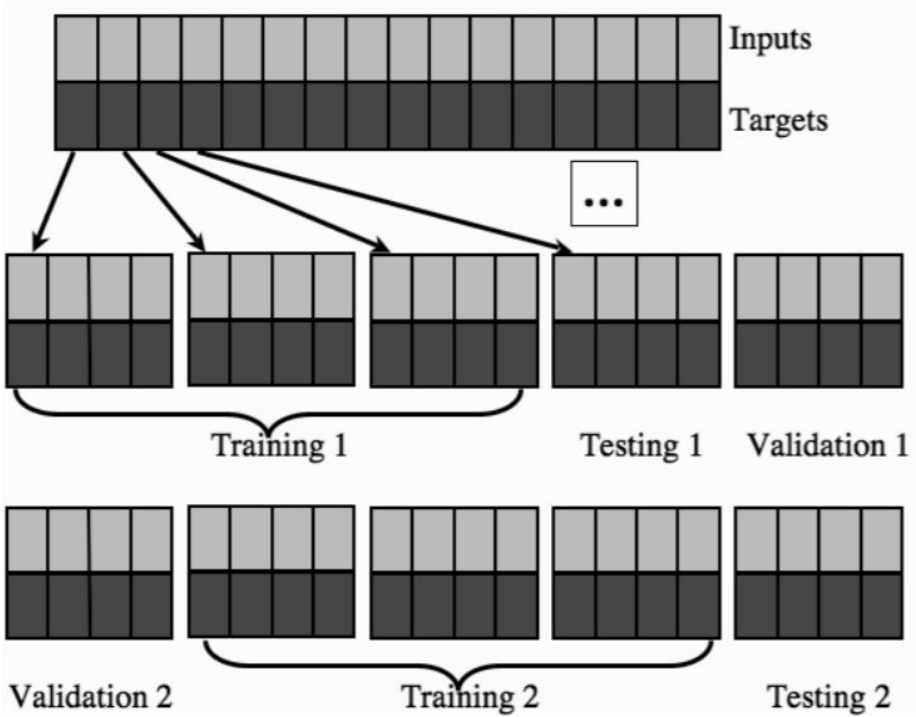
To prevent this, we use a third dataset: the validation set.
This is known as cross-validation.

2.2.2 TRAINING, TEST AND VALIDATION SETS

- 50:25:25 if lots of data is available
- 60:20:20 if not
- If really short on data, K-fold cross-validation can be used.



2.2.2 TRAINING, TEST AND VALIDATION SETS



2.2.3 THE CONFUSION MATRIX

- Used for classification problems.
- Targets down the side, outputs along.

	Outputs		
	C_1	C_2	C_3
C_1	5	1	0
C_2	1	4	1
C_3	2	0	4

2.2.4 ACCURACY METRICS

True Positives	False Positives
False Negatives	True Negatives

$$\text{Sensitivity} = \frac{\#TP}{\#TP + \#FN}$$

$$\text{Specificity} = \frac{\#TN}{\#TN + \#FP}$$

$$\text{Precision} = \frac{\#TP}{\#TP + \#FP}$$

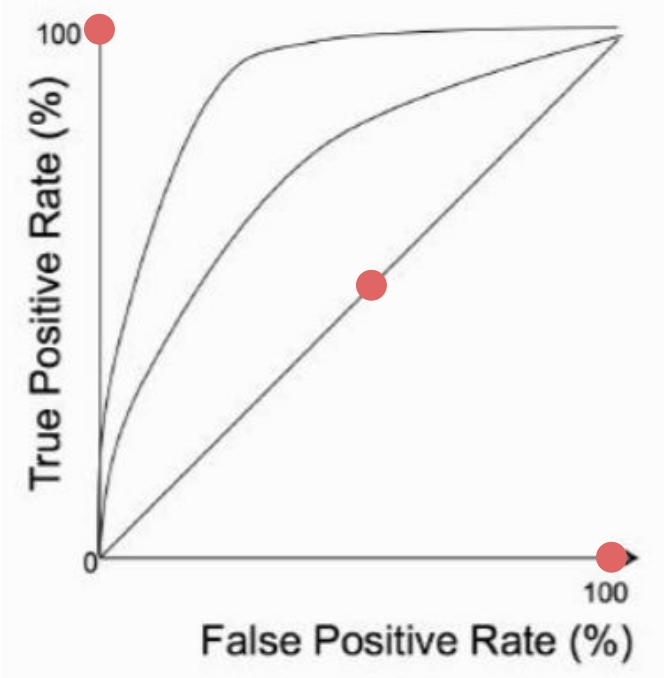
$$\text{Recall} = \frac{\#TP}{\#TP + \#FN}$$

$$\text{Accuracy} = \frac{\#TP + \#TN}{\#TP + \#FP + \#TN + \#FN}$$

$$F_1 = 2\frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}} = \frac{\#TP}{\#TP + (\#FN + \#FP)/2}$$

2.2.5 THE ROC CURVE

(Receiver Operator Characteristic)



- A single run of the classifier produces a single point on the plot.
- The key to getting a curve is using cross-validation.
- You can compare 2 classifiers by comparing their ROC curves.

2.2.6 UNBALANCED DATASETS

- So far, we have assumed there are exactly the same number of positive and negative examples.
- If there aren't, we can instead used balanced accuracy: the sum of specificity and sensitivity, divided by 2.
- However, there is also Matthew's Correlation Coefficient:

$$MCC = \frac{\#TP \times \#TN - \#FP \times \#FN}{\sqrt{(\#TP + \#FP)(\#TP + \#FN)(\#TN + \#FP)(\#TN + \#FN)}}$$

2.3 Probabilities

- **the conditional probability P(X|C)**: the probability of X given C.
- **the joint probability P(C,X)**: the probability of C and X.

School	British	Foreign	Sum
Boy	75	25	100
Girl	45	55	100
Sum	120	80	200

A: the girl student; B: the British student.

P(B) = 120/200 = 0.6
P(A,B) = 45/200 = 0.225
P(A|B) = P(A,B) / P(B) = 0.225/0.6 = 0.375

2.3 The Bayes' Rule

The joint probability: $P(C_i, X_j)$ The class-conditional probability: $P(X_j|C_i)$

$P(C_i|X_j)$

$$P(C_i, X_j) = P(X_j|C_i)P(C_i), \quad (2.10)$$

$$P(C_i, X_j) = P(C_i|X_j)P(X_j). \quad (2.11)$$

$$P(X_j|C_i)P(C_i) = P(C_i|X_j)P(X_j)$$

The Bayes' Rule:

$$P(C_i|X_j) = \frac{P(X_j|C_i)P(C_i)}{P(X_j)}. \quad (2.12)$$

$$P(X_k) = \sum_i P(X_k|C_i)P(C_i). \quad (2.13)$$

e.g. $P(X_k) = P(X_k|C_1)*P(C_1) + P(X_k|C_2)*P(C_2) + P(X_k|C_3)*P(C_3)$

2.3 maximum a posterior or MAP

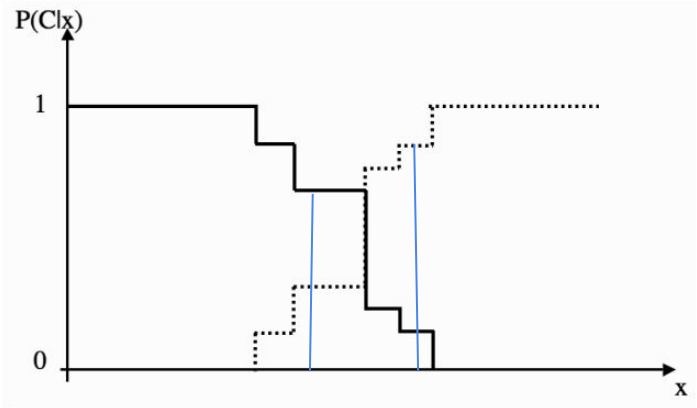
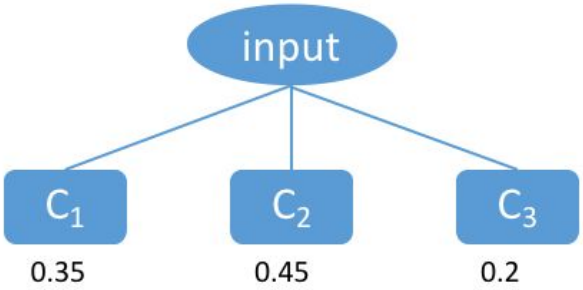


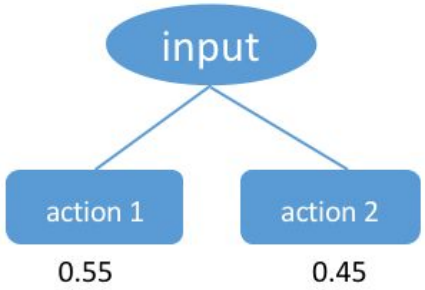
FIGURE 2.12 The posterior probabilities of the two classes C_1 and C_2 for feature x.

$$P(C_i|\mathbf{x}) > P(C_j|\mathbf{x}) \quad \forall\ i \neq j, \tag{2.14}$$

2.3 the Bayes' Optimal Classification

Example: Suppose that there are three possible output classes, and for a particular input the posterior probabilities of the classes are $P(C_1|x) = 0.35$, $P(C_2|x) = 0.45$, $P(C_3|x) = 0.2$. The inputs are the results of a blood test, the three classes are different possible diseases, and the output is whether or not to treat with a particular antibiotic. If the class is C1 or C3 then we do action 1 (treating), and if the class is C_2 then we do action 2 (no treating).





2.3 Minimising Risk

	C_1	C_2	C_3
C_1	0	r_{12}	r_{13}
C_2	r_{21}	0	r_{23}
C_3	r_{31}	r_{32}	0

Example: assume a patient has a disease of C_2, and posterior probabilities of the classes are $P(C_1|x) = 0.35$, $P(C_2|x) = 0.45$, $P(C_3|x) = 0.2$. In this case, the risk summation is:

$$R = r_{21} \times P(C_1|x) + 0 \times P(C_2|x) + r_{23} \times P(C_3|x)$$

2.3 The Naïve Bayes' Classifier

- assumption: the elements of feature vector are conditionally independent of each other, given the classification.

$$P(X_j^1 = a_1, X_j^2 = a_2, \ldots, X_j^n = a_n | C_i)$$

$$P(X_j^1 = a_1|C_i) \times P(X_j^2 = a_2|C_i) \times \ldots \times P(X_j^n = a_n|C_i) = \prod_k P(X_j^k = a_k|C_i), \quad (2.15)$$

$$P(C_i|X_j) = \frac{P(X_j|C_i)P(C_i)}{P(X_j)}. = \frac{P(C_i)\prod_k P(X_j^k = a_k|C_i).}{\sum_i P(X_{ij}|C_i)P(C_i).}$$

2.3 An Example

What to do in the evening?

Deadline?	Is there a party?	Lazy?	Activity
Urgent	Yes	Yes	Party
Urgent	No	Yes	Study
Near	Yes	Yes	Party
None	Yes	No	Party
None	No	Yes	Pub
None	Yes	No	Party
Near	No	No	Study
Near	No	Yes	TV
Near	Yes	Yes	Party
Urgent	No	No	Study

You have **deadlines looming**, but them are **no urgent**, there is **no party** on, and that you are currently **lazy**:

$$P(C_i|X_j) = \frac{P(X_j|C_i)P(C_i)}{P(X_j)}.$$

$$= \frac{P(X_j^1 = a_1|C_i) \times P(X_j^2 = a_2|C_i) \times \ldots \times P(X_j^n = a_n|C_i)}{\sum_i P(X_k|C_i)P(C_i).}$$

1) P(Party) × P(Near | Party) × P(No Party | Party) × P(Lazy | Party)=$\frac{5}{10} \times \frac{2}{5} \times \frac{0}{5} \times \frac{3}{5} = 0$

2) P(Study) × P(Near | Study) × P(No Party | Study) × P(Lazy | Study)=$\frac{3}{10} \times \frac{1}{3} \times \frac{3}{3} \times \frac{1}{3} = \frac{1}{30}$

3) P(Pub) × P(Near | Pub) × P(No Party | Pub) × P(Lazy | Pub)=$\frac{1}{10} \times \frac{0}{1} \times \frac{1}{1} \times \frac{1}{1} = 0$

4) P(TV) × P(Near | TV) × P(No Party | TV) × P(Lazy | TV)=$\frac{1}{10} \times \frac{1}{1} \times \frac{1}{1} \times \frac{1}{1} = \frac{1}{10}$

›(Party|near (not urgent) deadline, no party, lazy) =$\frac{0}{0+\frac{1}{30}+0+\frac{1}{10}} = 0$

›(Study|near (not urgent) deadline, no party, lazy)=$\frac{\frac{1}{30}}{0+\frac{1}{30}+0+\frac{1}{10}} = \frac{1}{4}$

P(Pub|near (not urgent) deadline, no party, lazy) =$\frac{0}{0+\frac{1}{30}+0+\frac{1}{10}} = 0$

P(TV|near (not urgent) deadline, no party, lazy) =$\frac{\frac{1}{10}}{0+\frac{1}{30}+0+\frac{1}{10}} = \boxed{\frac{3}{4}}$

2.4 SOME BASIC STATISTICS

Averages

Mean: add up all the data points and divide by the number of points;

Median: sort the dataset according to size and find the point in the middle - $O(n \log n)$;

*Randomized Find Median with High Probability** - $O(n)$

Mode: the most common value, pick the most frequent one.

* https://www.cc.gatech.edu/~vigoda/6550/Notes/Lec2.pdf

2.4 SOME BASIC STATISTICS

Expectation

Definition (discrete):

$$E[X] = \sum_x x p_X(x)$$

Interpretations:

- Center of gravity of Probability Mass Function
- Average in large number of repetitions

2.4 SOME BASIC STATISTICS

Expectation

Let X be a random variable, and let $Y = g(X)$, to compute the expectation of Y, instead of

$$E[Y] = \sum_y y p_Y(y)$$

We can easily do

$$E[Y] = \sum_x g(x) p_X(x)$$

2.4 SOME BASIC STATISTICS

Expectation

Properties:

If α, β are constants, then $\mathrm{E}[\alpha] = \alpha$, $\mathrm{E}[\alpha X + \beta] = \alpha \mathrm{E}[X] + \beta$.

If X, Y are random variables, we have $\mathrm{E}[X + Y] = \mathrm{E}[X] + \mathrm{E}[Y]$.

Only for independent random variables, $\mathrm{E}[XY] = \mathrm{E}[X]\mathrm{E}[Y]$.

2.4 SOME BASIC STATISTICS

Variance

Definition:

$$Var(X) = E[(X - E[X])^2]$$

Interpretation:

- Measures the average squared distance from the mean

2.4 SOME BASIC STATISTICS

Variance

Properties:

$$\begin{aligned} Var(X) &= E[(X - E[X])^2] \\ &= \sum_x (x - E[X])^2 p_X(x) \\ &= E[X^2] - (E[X])^2 \end{aligned}$$

If α, β are constants, then $\mathrm{Var}(\alpha X + \beta) = \alpha^2 \mathrm{Var}(X)$.

2.4 SOME BASIC STATISTICS

Covariance

Definition:

$$\mathrm{cov}(\{\mathbf{x}_i\}, \{\mathbf{y}_i\}) = E(\{\mathbf{x}_i\} - \boldsymbol{\mu})E(\{\mathbf{y}_i\} - \boldsymbol{\nu}), \text{ where } \mu = E[\{x_i\}], \nu = E[\{y_i\}]$$

Interpretation:

- Measures how statistically dependent the two variables are, eg. is there a relation between having a big X and having a big Y.

2.4 SOME BASIC STATISTICS

Covariance

Properties:

$$cov(X,Y) = \begin{cases} positive, & X,Y increases/decreases\ at\ the\ same\ time \\ 0 & X,Y are\ uncorrelated \\ negative & one\ goes\ up\ while\ the\ other\ goes\ down \end{cases}$$

2.4 SOME BASIC STATISTICS

Covariance Matrix

Definition:

$$\boldsymbol{\Sigma} = \begin{pmatrix} E[(\mathbf{x}_1 - \boldsymbol{\mu}_1)(\mathbf{x}_1 - \boldsymbol{\mu}_1)] & E[(\mathbf{x}_1 - \boldsymbol{\mu}_1)(\mathbf{x}_2 - \boldsymbol{\mu}_2)] & \dots & E[(\mathbf{x}_1 - \boldsymbol{\mu}_1)(\mathbf{x}_n - \mu_n)] \\ E[(\mathbf{x}_2 - \boldsymbol{\mu}_2)(\mathbf{x}_1 - \boldsymbol{\mu}_1)] & E[(\mathbf{x}_2 - \boldsymbol{\mu}_2)(\mathbf{x}_2 - \boldsymbol{\mu}_2)] & \dots & E[(\mathbf{x}_2 - \boldsymbol{\mu}_2)(\mathbf{x}_n - \boldsymbol{\mu}_n)] \\ \dots & \dots & \dots & \dots \\ E[(\mathbf{x}_n - \boldsymbol{\mu}_n)(\mathbf{x}_1 - \boldsymbol{\mu}_1)] & E[(\mathbf{x}_n - \boldsymbol{\mu}_n)(\mathbf{x}_2 - \boldsymbol{\mu}_2)] & \dots & E[(\mathbf{x}_n - \boldsymbol{\mu}_n)(\mathbf{x}_n - \boldsymbol{\mu}_n)] \end{pmatrix}$$

where x_i is a column vector describing the elements of the i-th variable, μ_i is their mean.

Interpretation:

- The level to which each pair of variables vary together.

2.4 SOME BASIC STATISTICS

Covariance Matrix

Example:

```
In [1]:
1 import numpy as np

In [2]:
1 X = np.array([[0, 2], [1, 1], [2, 0]])
2 X

Out[2]: array([[0, 2],
               [1, 1],
               [2, 0]])

In [3]:
1 X.T

Out[3]: array([[0, 1, 2],
               [2, 1, 0]])

In [4]:
1 np.cov(X.T)

Out[4]: array([[ 1., -1.],
               [-1.,  1.]])
```

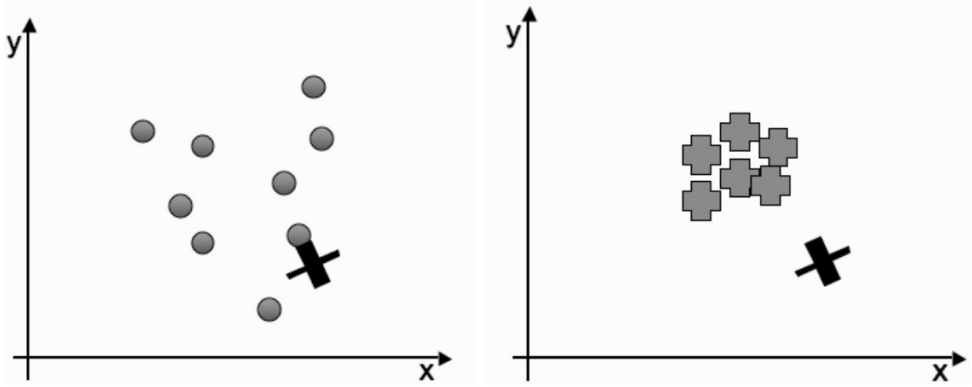
$$cov(x_0, x_1) = E[(x_0 - \mu_0)(x_1 - \mu_1)] * \frac{n}{(n-1)}$$

```
In [5]:
1 x0 = X.T[0]
2 x1 = X.T[1]
3
4 ((x0 - x0.mean()) * (x1 - x1.mean())).mean() * len(x0) / (len(x0) - 1)

Out[5]: -1.0
```

2.4 SOME BASIC STATISTICS

Mahalanobis Distance



Definition:

$$D_M(\mathbf{x}) = \sqrt{(\mathbf{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu})}$$

Interpretation:

- A measure of the distance between a sample point and the distribution.

2.4 SOME BASIC STATISTICS

The Gaussian

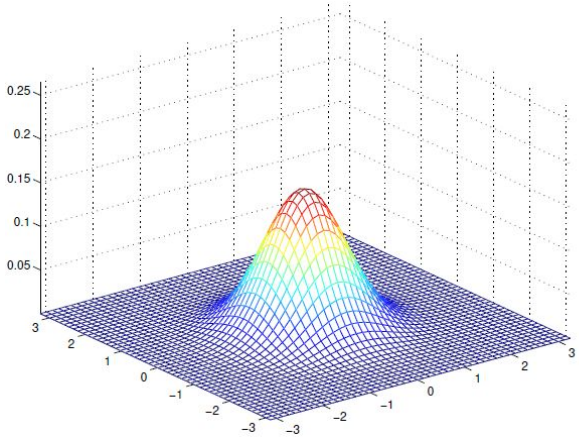
Normal Distribution:

$$p(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(\frac{-(x-\mu)^2}{2\sigma^2}\right)$$

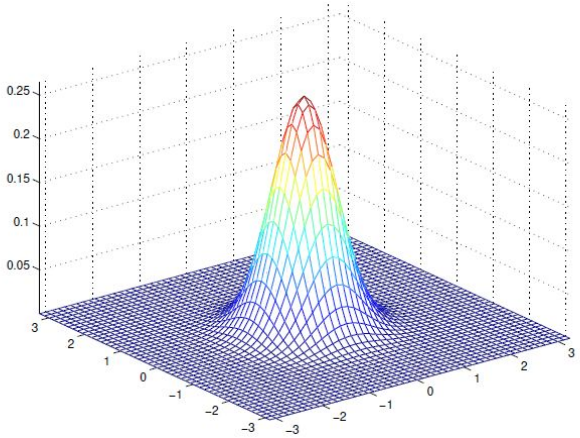
Multivariate Gaussian Distribution:

$$p(\mathbf{x}) = \frac{1}{(2\pi)^{d/2}|\mathbf{\Sigma}|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T\mathbf{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right)$$

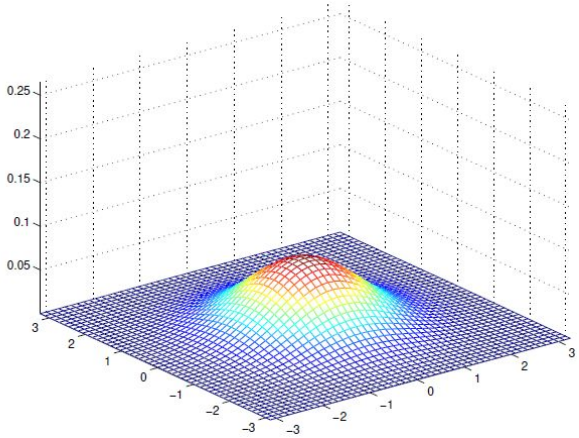
2.4 SOME BASIC STATISTICS



$\Sigma = I$



$\Sigma = 0.6I$



$\Sigma = 2I$

2.5 THE BIAS-VARIANCE TRADEOFF

Bias-Variance Decomposition

Assume that data points in our training/test set are all taken from a similar distribution

$$y_i = f(x_i) + \epsilon_i \qquad where\ E[\epsilon_i] = 0, Var(\epsilon_i) = \sigma^2$$

Based on the training set, we obtain an estimate $\hat{f}$, then for each sample *j* in the test set, our prediction for $y_j = f(x_j) + \epsilon_j$ is $\hat{f}(x_j)$. Now we compute the MSE on the test set.

2.5 THE BIAS-VARIANCE TRADEOFF

Bias-Variance Decomposition

Fixed number: $f(x)$

Random Variables: ϵ, $\hat{f}(x)$

$$
\begin{aligned}
MSE &= E[(y - \hat{f}(x))^2] \\
&= E[(\epsilon + f(x) - \hat{f}(x))^2] \\
&= E[\epsilon^2] + E[(f(x) - \hat{f}(x))^2] + E[2\epsilon(f(x) - \hat{f}(x))] \\
&= \sigma^2 + (E[f(x) - \hat{f}(x)])^2 + Var(f(x) - \hat{f}(x)) \\
&= \sigma^2 + (Bias\ \hat{f}(x))^2 + Var(\hat{f}(x))
\end{aligned}
$$

High Bias <-> Underfitting
High Variance <-> Overfitting
Large σ^2 <-> Noisy Data

Goal: Reduce the bias without increasing the variance.