

# Mathematics, Optimization, Statistics, and Machine Learning

Sunghee Yun

December 30, 2019



# Contents

<b>I</b>	<b>Mathematics</b>	<b>7</b>
<b>1</b>	<b>Calculus</b>	<b>9</b>
1.1	Basics	10
1.2	Multivariate functions	10
1.3	Chain rule	11
1.4	Integration	11
<b>2</b>	<b>Convex analysis</b>	<b>13</b>
2.1	Convex function	14
2.1.1	First order condition	14
2.1.2	Second order condition	17
<b>3</b>	<b>Linear Algebra</b>	<b>19</b>
3.1	Vector space	20
3.2	Eigenvalues	20
3.2.1	Basic definitions	20
3.2.2	Symmetric matrices	21
3.3	Positive definiteness	21
3.4	Matrix norms	22
<b>II</b>	<b>Optimization</b>	<b>23</b>
<b>4</b>	<b>Convex Optimization</b>	<b>25</b>
4.1	Mathematical optimization problem	26
4.2	Convex optimization problem	27
4.3	Duality	28
4.3.1	The Lagrange dual problem	28
4.3.1.1	Examples	28
4.3.2	Interpretations	29
4.3.2.1	Max-min characterization of weak and strong duality	29
4.3.2.2	Saddle-point interpretation	30
4.4	Unconstrained minimization	31
4.4.1	Gradient descent method	31

4.4.1.1	Examples . . . . .	31
<b>5</b>	<b>Portfolio optimization</b>	<b>33</b>
5.1	Problem formulation . . . . .	34
5.1.1	A portfolio optimization problem . . . . .	34
<b>III</b>	<b>Statistics</b>	<b>37</b>
<b>6</b>	<b>Statistics Basics</b>	<b>39</b>
6.1	Correlation coefficients . . . . .	39
6.2	Transformation of a random variable via a function . . . . .	39
6.2.1	Scale random variable . . . . .	39
6.2.2	Multivariate random variable . . . . .	41
6.2.3	Data Examples . . . . .	41
<b>7</b>	<b>Various distributions</b>	<b>43</b>
7.1	Log-normal distribution . . . . .	43
7.1.1	Some statistics . . . . .	44
7.1.2	Parameter estimation . . . . .	46
<b>8</b>	<b>Bayesian Statistics</b>	<b>47</b>
8.1	Bayesian Theorem . . . . .	47
8.2	Bayesian Inference . . . . .	47
8.3	Conjugate prior . . . . .	47
8.3.1	Bernoulli distribution . . . . .	47
8.3.2	Gaussian distribution . . . . .	48
<b>9</b>	<b>Information Theory</b>	<b>49</b>
9.1	Basics . . . . .	49
9.1.1	Entropy . . . . .	49
9.1.2	Mutual Information . . . . .	49
9.1.3	Relative Entropy (Kullback–Leibler divergence) . . . . .	49
<b>IV</b>	<b>Machine Learning</b>	<b>51</b>
<b>10</b>	<b>Machine Learning Basics</b>	<b>53</b>
10.1	Optimal Predictor . . . . .	54
10.2	Bias and Variance . . . . .	54
10.3	Maximum Likelihood Estimation . . . . .	56
10.4	Maximum a Posteriori Estimation . . . . .	58
10.5	Bayesian prior update . . . . .	59
<b>11</b>	<b>Optimization for Machine Learning</b>	<b>61</b>
11.1	Gradient method . . . . .	60
11.2	Stochastic gradient method . . . . .	60

<b>12 Bayesian Network</b>	<b>61</b>
<b>13 Collaborative Filtering</b>	<b>63</b>
13.1 Item-based Collaborative Filtering	64
13.1.1 Rating matrix modeling for menu personalization for mobile shopping app	64
13.1.1.1 Rating matrix modeling	65
13.1.2 Value augmentation based on Bayesian MAP	66
13.1.3 Similarity measure among items	66
13.1.3.1 Cosine similarity	67
13.1.3.2 Cosine similarity when prior distribution is used	67
13.1.3.3 Correlation coefficient similarity	68
13.1.4 Data value transformation	68
13.1.4.1 TFIDF (or tf-idf)	69
13.1.4.2 Okapi BM25 transformation	69
13.1.5 Recommendation based on item similarities	69
13.2 Collaborative Filtering using Matrix Factorization	70
13.2.1 Problem definition and formulations	70
13.2.2 Solution methods	72
13.2.2.1 Matrix factorization via singular value decomposition (SVD)	72
13.2.2.2 Matrix factorization via gradient descent (GD) method	73
13.2.2.3 Matrix factorization via alternating gradient descent (GD) method	75
13.2.2.4 Matrix factorization via stochastic gradient descent (SGD) method	75
13.2.2.5 Matrix factorization via alternating least-squares (ALS)	76
13.2.2.6 Weighted matrix factorization via alternating least-squares (ALS)	78
13.2.3 Collaborative filtering for implicit feedback dataset	78
13.2.3.1 Regularization coefficient conversion	79
13.3 Appendix	81
13.3.1 Linear algebra	81
13.3.1.1 Singular value decomposition (SVD)	81
13.3.1.2 Singular value decomposition as rank- $k$ approximation	82
<b>14 Time Series Anomaly Detection</b>	<b>83</b>
14.1 Real-Time Anomaly Detection	84
14.1.1 Computing Anomaly Likelihood	84
<b>15 Reinforcement Learning</b>	<b>87</b>
15.1 Finite Markov decision processes	88
15.1.1 Markov property	88
15.1.2 Policy	89
15.1.3 Return	89
15.1.4 State value function and action value function	90
15.2 Bellman equation	90
15.2.1 Bellman equations	90
15.2.2 Bellman optimality equations	91
15.3 Dynamic programming	92
15.3.1 Policy evaluation (prediction)	92
15.3.2 Policy iteration	93

15.3.3	Value iteration . . . . .	93
15.4	Monte Carlo methods . . . . .	93
15.4.1	Monte Carlo prediction . . . . .	95
15.4.2	Monte Carlo control . . . . .	96
15.4.3	Monte Carlo control without exploring starts . . . . .	96
15.4.4	Off-policy prediction via important sampling . . . . .	98
15.4.5	Off-policy Monte Carlo control . . . . .	98
15.5	Temporal-difference learning . . . . .	98
15.5.1	TD prediction . . . . .	98
15.5.2	Sarsa: on-policy TD Control . . . . .	100
15.5.3	Q-learning: off-policy TD control . . . . .	100
15.5.4	Maximization bias and double learning . . . . .	101
15.6	$n$ -step bootstrapping . . . . .	101
15.6.1	$n$ -step TD prediction . . . . .	102
15.6.2	$n$ -step Sarsa . . . . .	106
15.6.3	$n$ -step off-policy learning . . . . .	108
15.7	Planning and learning with tabular methods . . . . .	108
15.7.1	Dyna: integrated planning, acting, and learning . . . . .	108
15.8	On-policy Prediction with Approximation . . . . .	108
15.9	On-policy Control with Approximation . . . . .	108
15.10	Off-policy Methods with Approximation . . . . .	111
15.11	Eligibility Traces . . . . .	111
15.11.1	The $\lambda$ -return . . . . .	111
15.11.2	$TD(\lambda)$ . . . . .	112
15.11.3	Why $TD(\lambda)$ approximates the off-line $\lambda$ -return algorithm? . . . . .	113
15.11.4	Sarsa( $\lambda$ ) . . . . .	118
15.11.5	Tabular methods using eligibility traces . . . . .	118
15.12	Appendix: conditional probability and expected value . . . . .	120

**Part I**

**Mathematics**





# Chapter 1

# Calculus



## Chapter 2

# Convex analysis



## Chapter 3

# Linear Algebra

# Part II

# Optimization



## Chapter 4

# Convex Optimization



## Chapter 5

# Portfolio optimization



# Part III

## Statistics





**Part IV**

**Machine Learning**



## Chapter 10

# Machine Learning Basics



## 10.1 Optimal Predictor

Consider a regression problem where we predict  $Y \in \mathbf{R}^m$  given  $X \in \mathbf{R}^n$ . We want to design a predictor  $g : \mathbf{R}^n \rightarrow \mathbf{R}^m$  so that  $g(X) \sim Y$  in some statistical sense. We first show that  $g(X) = \mathbf{E}(Y|X)$  is the optimal predictor (or estimator) in least-mean-square sense.

We define  $g^* : \mathbf{R}^n \rightarrow \mathbf{R}^m$  where  $g(x) = \mathbf{E}(Y|X = x)$ . Then

$$\begin{aligned}
 \mathbf{E}\|g(X) - Y\|_2^2 &= \mathbf{E}\|g(X) - g^*(X) + g^*(X) - Y\|_2^2 \\
 &= \mathbf{E}\|g(X) - g^*(X)\|_2^2 + \mathbf{E}\|g^*(X) - Y\|_2^2 + 2\mathbf{E}(g(X) - g^*(X))^T(g^*(X) - Y) \\
 &= \mathbf{E}\|g(X) - g^*(X)\|_2^2 + \mathbf{E}\|g^*(X) - Y\|_2^2 + 2\mathbf{E}_X \mathbf{E}_Y ((g(X) - g^*(X))^T(g^*(X) - Y)|X) \\
 &= \mathbf{E}\|g(X) - g^*(X)\|_2^2 + \mathbf{E}\|g^*(X) - Y\|_2^2 + 2\mathbf{E}_X (g(X) - g^*(X))^T \mathbf{E}_Y (g^*(X) - Y|X) \\
 &= \mathbf{E}\|g(X) - g^*(X)\|_2^2 + \mathbf{E}\|g^*(X) - Y\|_2^2 + 2\mathbf{E}_X (g(X) - g^*(X))^T (g^*(X) - \mathbf{E}(Y|X)) \\
 &= \mathbf{E}\|g(X) - g^*(X)\|_2^2 + \mathbf{E}\|g^*(X) - Y\|_2^2 \geq \mathbf{E}\|g^*(X) - Y\|_2^2.
 \end{aligned}$$

Therefore  $g^*(X)$  is the optimal predictor for  $Y$  in the least-mean-square sense.

## 10.2 Bias and Variance

In §10.1, we proved that  $g^*(X) = \mathbf{E}(Y|X)$  is the optimal predictor (or estimator) in the least-mean-square sense. However, unless we have the full knowledge of the joint probability distribution of  $X$  and  $Y$ , *i.e.*,  $p(X, Y)$ , or know  $\mathbf{E}(Y|X = x)$  as a function of  $x$ , it is not possible to obtain  $g^*$ .

Here we assume that we obtain the predictor for  $Y$  given  $X$  from a dataset  $D$  where

$$D = \{(x_1, y_1), \dots, (x_N, y_N)\} \subseteq \mathbf{R}^n \times \mathbf{R}^m. \quad (10.1)$$

Now suppose that we have a predictor  $g(\cdot; D) : \mathbf{R}^n \rightarrow \mathbf{R}^m$ , which depends on  $D$ . Now let  $\mathcal{D}$  denote the random variable for this data set, *i.e.*,

$$\mathcal{D} = \{(X_1, Y_1), \dots, (X_N, Y_N)\} \subseteq \mathbf{R}^n \times \mathbf{R}^m. \quad (10.2)$$

---

<sup>1</sup>Note that strictly speaking,  $\mathcal{D}$  is *not* a set since the order of  $(x_i, y_i) \in \mathbf{R}^n \times \mathbf{R}^m$  matters, *i.e.*, if the order is changed, we generally have different predictor, and we are allowed to have identical data point. Thus, we should say  $\mathcal{D}$  is a (ordered) list of points,  $(x_i, y_i) \in \mathbf{R}^n \times \mathbf{R}^m$ .

Then the mean square error of this predictor can be decomposed as following.

$$\begin{aligned}
\mathbf{E}_{X,Y,\mathcal{D}} \|g(X; \mathcal{D}) - Y\|_2^2 &= \mathbf{E}_{X,Y,\mathcal{D}} \|g(X; \mathcal{D}) - g^*(X) + g^*(X) - Y\|_2^2 \\
&= \mathbf{E}_{X,Y,\mathcal{D}} \|g(X; \mathcal{D}) - g^*(X)\|_2^2 + \mathbf{E}_{X,Y,\mathcal{D}} \|g^*(X) - Y\|_2^2 \\
&\quad + 2\mathbf{E}_{X,Y,\mathcal{D}} (g(X; \mathcal{D}) - g^*(X))^T (g^*(X) - Y) \\
&= \mathbf{E}_{X,\mathcal{D}} \|g(X; \mathcal{D}) - g^*(X)\|_2^2 + \mathbf{E}_{X,Y} \|g^*(X) - Y\|_2^2 \\
&\quad + 2\mathbf{E}_{X,\mathcal{D}} \mathbf{E}_Y ((g(X; \mathcal{D}) - g^*(X))^T (g^*(X) - Y) | X, \mathcal{D}) \\
&= \mathbf{E}_{X,\mathcal{D}} \|g(X; \mathcal{D}) - g^*(X)\|_2^2 + \mathbf{E}_{X,Y} \|g^*(X) - Y\|_2^2 \\
&\quad + 2\mathbf{E}_{X,\mathcal{D}} (g(X; \mathcal{D}) - g^*(X))^T \mathbf{E}_Y (g^*(X) - Y | X, \mathcal{D}) \\
&= \mathbf{E}_{X,\mathcal{D}} \|g(X; \mathcal{D}) - g^*(X)\|_2^2 + \mathbf{E}_{X,Y} \|g^*(X) - Y\|_2^2 \\
&\quad + 2\mathbf{E}_{X,\mathcal{D}} (g(X; \mathcal{D}) - g^*(X))^T \mathbf{E}_Y (g^*(X) - Y | X) \\
&= \mathbf{E}_{X,\mathcal{D}} \|g(X; \mathcal{D}) - g^*(X)\|_2^2 + \mathbf{E}_{X,Y} \|g^*(X) - Y\|_2^2 \\
&\quad + 2\mathbf{E}_{X,\mathcal{D}} (g(X; \mathcal{D}) - g^*(X))^T (g^*(X) - \mathbf{E}_Y(Y | X)) \\
&= \mathbf{E}_{X,\mathcal{D}} \|g(X; \mathcal{D}) - \mathbf{E}_{\mathcal{D}} g(X; \mathcal{D}) + \mathbf{E}_{\mathcal{D}} g(X; \mathcal{D}) - g^*(X)\|_2^2 + \mathbf{E}_{X,Y} \|g^*(X) - Y\|_2^2 \\
&= \mathbf{E}_{X,\mathcal{D}} \|g(X; \mathcal{D}) - \mathbf{E}_{\mathcal{D}} g(X; \mathcal{D})\|_2^2 + \mathbf{E}_{X,\mathcal{D}} \|\mathbf{E}_{\mathcal{D}} g(X; \mathcal{D}) - g^*(X)\|_2^2 + \mathbf{E}_{X,Y} \|g^*(X) - Y\|_2^2 \\
&\quad + 2\mathbf{E}_{X,\mathcal{D}} (g(X; \mathcal{D}) - \mathbf{E}_{\mathcal{D}} g(X; \mathcal{D}))^T (\mathbf{E}_{\mathcal{D}} g(X; \mathcal{D}) - g^*(X)) \\
&= \mathbf{E}_{X,\mathcal{D}} \|g(X; \mathcal{D}) - \mathbf{E}_{\mathcal{D}} g(X; \mathcal{D})\|_2^2 + \mathbf{E}_X \|\mathbf{E}_{\mathcal{D}} g(X; \mathcal{D}) - g^*(X)\|_2^2 + \mathbf{E}_{X,Y} \|g^*(X) - Y\|_2^2 \\
&\quad + 2\mathbf{E}_X \mathbf{E}_{\mathcal{D}} ((g(X; \mathcal{D}) - \mathbf{E}_{\mathcal{D}} g(X; \mathcal{D}))^T (\mathbf{E}_{\mathcal{D}} g(X; \mathcal{D}) - g^*(X)) | X) \\
&= \mathbf{E}_{X,\mathcal{D}} \|g(X; \mathcal{D}) - \mathbf{E}_{\mathcal{D}} g(X; \mathcal{D})\|_2^2 + \mathbf{E}_X \|\mathbf{E}_{\mathcal{D}} g(X; \mathcal{D}) - g^*(X)\|_2^2 + \mathbf{E}_{X,Y} \|g^*(X) - Y\|_2^2 \\
&\quad + 2\mathbf{E}_X \mathbf{E}_{\mathcal{D}} (g(X; \mathcal{D}) - \mathbf{E}_{\mathcal{D}} g(X; \mathcal{D}))^T (\mathbf{E}_{\mathcal{D}} g(X; \mathcal{D}) - g^*(X)) \\
&= \mathbf{E}_{X,\mathcal{D}} \|g(X; \mathcal{D}) - \mathbf{E}_{\mathcal{D}} g(X; \mathcal{D})\|_2^2 + \mathbf{E}_X \|\mathbf{E}_{\mathcal{D}} g(X; \mathcal{D}) - g^*(X)\|_2^2 + \mathbf{E}_{X,Y} \|g^*(X) - Y\|_2^2 \\
&\quad + 2\mathbf{E}_X (\mathbf{E}_{\mathcal{D}} g(X; \mathcal{D}) - \mathbf{E}_{\mathcal{D}} g(X; \mathcal{D}) | X)^T (\mathbf{E}_{\mathcal{D}} g(X; \mathcal{D}) - g^*(X)) \\
&= \mathbf{E}_{X,\mathcal{D}} \|g(X; \mathcal{D}) - \mathbf{E}_{\mathcal{D}} g(X; \mathcal{D})\|_2^2 + \mathbf{E}_X \|\mathbf{E}_{\mathcal{D}} g(X; \mathcal{D}) - g^*(X)\|_2^2 + \mathbf{E}_{X,Y} \|g^*(X) - Y\|_2^2.
\end{aligned}$$

Note that we use the fact that  $\mathbf{E}_{\mathcal{D}} g(X; \mathcal{D})$  is a function of  $X$  only (hence does not depend on  $X$ ).

In the last equation, the first term is called the *variance* since it is the expected value (with respect to  $X$ ) of variance of the predictor,  $g(X; \mathcal{D})$ , with respect to the dataset,  $\mathcal{D}$ . It represents the extent to which the prediction varies around its expected value. The second term is the expected value of the square of the bias where the bias is defined to be the difference between the expected value of prediction with respect to dataset and the optimal prediction. The second term itself is sometimes called *bias*. The third term is called *noise* since it is caused by the intrinsic noise residing in  $Y$  which cannot be reduced even with the optimal predictor (in least-mean-square sense).

The following equation summarizes these three quantities.

$$\begin{aligned}
&\mathbf{E}_{X,Y,\mathcal{D}} \|g(X; \mathcal{D}) - Y\|_2^2 \\
&= \underbrace{\mathbf{E}_{X,\mathcal{D}} \|g(X; \mathcal{D}) - \mathbf{E}_{\mathcal{D}} g(X; \mathcal{D})\|_2^2}_{\text{variance}} + \underbrace{\mathbf{E}_X \|\mathbf{E}_{\mathcal{D}} g(X; \mathcal{D}) - g^*(X)\|_2^2}_{\text{bias}} + \underbrace{\mathbf{E}_{X,Y} \|g^*(X) - Y\|_2^2}_{\text{noise}}.
\end{aligned} \tag{10.3}$$

In general, we do not know the optimal predictor; if we knew, we would not need to train our in

the first place. Thus we can only estimate  $g(X; \mathcal{D}) - \mathbf{E}_{\mathcal{D}}g(X; \mathcal{D})$  and  $\mathbf{E}_{\mathcal{D}}g(X; \mathcal{D}) - Y$ . The mean square error can also be expressed in these two quantities as follows.

$$\begin{aligned}
\mathbf{E}_{X,Y,\mathcal{D}}\|g(X; \mathcal{D}) - Y\|_2^2 &= \mathbf{E}_{X,Y,\mathcal{D}}\|g(X; \mathcal{D}) - \mathbf{E}_{\mathcal{D}}g(X; \mathcal{D}) + \mathbf{E}_{\mathcal{D}}g(X; \mathcal{D}) - Y\|_2^2 \\
&= \mathbf{E}_{X,Y,\mathcal{D}}\|g(X; \mathcal{D}) - \mathbf{E}_{\mathcal{D}}g(X; \mathcal{D})\|_2^2 + \mathbf{E}_{X,Y,\mathcal{D}}\|\mathbf{E}_{\mathcal{D}}g(X; \mathcal{D}) - Y\|_2^2 \\
&\quad + 2\mathbf{E}_{X,Y,\mathcal{D}}(g(X; \mathcal{D}) - \mathbf{E}_{\mathcal{D}}g(X; \mathcal{D}))^T(\mathbf{E}_{\mathcal{D}}g(X; \mathcal{D}) - Y) \\
&= \mathbf{E}_{X,\mathcal{D}}\|g(X; \mathcal{D}) - \mathbf{E}_{\mathcal{D}}g(X; \mathcal{D})\|_2^2 + \mathbf{E}_{X,Y}\|\mathbf{E}_{\mathcal{D}}g(X; \mathcal{D}) - Y\|_2^2 \\
&\quad + 2\mathbf{E}_{X,Y}\mathbf{E}_{\mathcal{D}}((g(X; \mathcal{D}) - \mathbf{E}_{\mathcal{D}}g(X; \mathcal{D}))^T(\mathbf{E}_{\mathcal{D}}g(X; \mathcal{D}) - Y)|X, Y) \\
&= \mathbf{E}_{X,\mathcal{D}}\|g(X; \mathcal{D}) - \mathbf{E}_{\mathcal{D}}g(X; \mathcal{D})\|_2^2 + \mathbf{E}_{X,Y}\|\mathbf{E}_{\mathcal{D}}g(X; \mathcal{D}) - Y\|_2^2 \\
&\quad + 2\mathbf{E}_{X,Y}\mathbf{E}_{\mathcal{D}}(g(X; \mathcal{D}) - \mathbf{E}_{\mathcal{D}}g(X; \mathcal{D}))^T(\mathbf{E}_{\mathcal{D}}g(X; \mathcal{D}) - Y) \\
&= \mathbf{E}_{X,\mathcal{D}}\|g(X; \mathcal{D}) - \mathbf{E}_{\mathcal{D}}g(X; \mathcal{D})\|_2^2 + \mathbf{E}_{X,Y}\|\mathbf{E}_{\mathcal{D}}g(X; \mathcal{D}) - Y\|_2^2 \\
&\quad + 2\mathbf{E}_{X,Y}(\mathbf{E}_{\mathcal{D}}g(X; \mathcal{D}) - \mathbf{E}_{\mathcal{D}}g(X; \mathcal{D}))^T(\mathbf{E}_{\mathcal{D}}g(X; \mathcal{D}) - Y) \\
&= \mathbf{E}_{X,\mathcal{D}}\|g(X; \mathcal{D}) - \mathbf{E}_{\mathcal{D}}g(X; \mathcal{D})\|_2^2 + \mathbf{E}_{X,Y}\|\mathbf{E}_{\mathcal{D}}g(X; \mathcal{D}) - Y\|_2^2.
\end{aligned}$$

Equating the last equation with (10.3) yields

$$\mathbf{E}_{X,Y,\mathcal{D}}\|g(X; \mathcal{D}) - Y\|_2^2 = \underbrace{\mathbf{E}_{X,\mathcal{D}}\|g(X; \mathcal{D}) - \mathbf{E}_{\mathcal{D}}g(X; \mathcal{D})\|_2^2}_{\text{variance}} + \underbrace{\mathbf{E}_{X,Y}\|\mathbf{E}_{\mathcal{D}}g(X; \mathcal{D}) - Y\|_2^2}_{\text{bias} + \text{noise}} \quad (10.4)$$

Therefore in reality, we can only obtain the sum of the bias and noise (not separately) unless we know the quantity of the noise.

### 10.3 Maximum Likelihood Estimation

Suppose that  $X \in \mathbf{R}^n$  and  $Y \in \mathbf{R}^m$  are random variables representing inputs (or independent variables or predictors or features) and outputs (or dependent variables or responses). We want to find a parameterized model to predict  $Y$  from  $X$ .

We consider the parameter  $\theta \in \Theta$  where  $\Theta \subset \mathbf{R}^l$  is the set of all possible parameter values, and the model,  $g : \mathbf{R}^n \times \mathbf{R}^l \rightarrow \mathbf{R}^m$ , such that  $g(X; \theta)$  is close to  $Y$  in some statistical sense.

We further assume that the conditional probability of  $Y$  given  $g(X; \theta)$  can be characterized by  $\beta \in \mathcal{B}$ , *i.e.*,  $p(Y|g(X; \theta))$  is a function of  $\beta$  where  $\mathcal{B}$  is the set of all possible values for  $\beta$ . To express that  $p(Y|g(X; \theta))$  is a function of  $\beta$ , we will use a notation,  $p(Y|X; \theta, \beta)$ , for  $p(Y|g(X; \theta))$ .

Now suppose that we have observed  $N$  independent data sample,  $\{(x_i, y_i)\}_{i=1}^N$  where  $x_i \in \mathbf{R}^n$  and  $y_i \in \mathbf{R}^m$ . We want to find  $\theta \in \Theta$  which maximizes the probability of this event, *i.e.*,

$$p((X_1, Y_1) = (x_1, y_1), \dots, (X_N, Y_N) = (x_N, y_N)) \quad (10.5)$$

For notational convenience, we define two random variables,  $\tilde{X} \in \mathbf{R}^{n \times N}$  and  $\tilde{Y} \in \mathbf{R}^{m \times N}$ , such that

$$\tilde{X} = \begin{bmatrix} X_1 & \dots & X_N \end{bmatrix} \quad (10.6)$$

and

$$\tilde{Y} = \begin{bmatrix} Y_1 & \dots & Y_N \end{bmatrix}. \quad (10.7)$$

We also defined  $\tilde{x} \in \mathbf{R}^{n \times N}$  and  $\tilde{y} \in \mathbf{R}^{m \times N}$ , such that

$$\tilde{x} = \begin{bmatrix} x_1 & \dots & x_N \end{bmatrix} \quad (10.8)$$

and

$$\tilde{y} = \begin{bmatrix} y_1 & \dots & y_N \end{bmatrix}. \quad (10.9)$$

Then (10.5) becomes

$$p(\tilde{X} = \tilde{x}, \tilde{Y} = \tilde{y}). \quad (10.10)$$

Bayes's theorem and the independence among  $(X_i, Y_i)$  imply

$$\begin{aligned} p(\tilde{X} = \tilde{x}, \tilde{Y} = \tilde{y}) &= \prod_{i=1}^N p(X_i = x_i, Y_i = y_i) \\ &= \prod_{i=1}^N p(X_i = x_i | Y_i = y_i; \theta, \beta) p(X_i = x_i) \\ &\propto \prod_{i=1}^N p(X_i = x_i | Y_i = y_i; \theta, \beta). \end{aligned}$$

Here  $p(X_i = x_i | Y_i = y_i; \theta, \beta)$  is called *likelihood function*.

The maximum likelihood estimation (MLE) (or learning) is to find  $\theta$  (and  $\beta$ ) which maximizes this probability. Since the log function is a strictly increasing function, taking log on this probability does give us the same results when maximizing it.

$$\log p(\tilde{X} = \tilde{x}, \tilde{Y} = \tilde{y}) = \sum_{i=1}^N \log p(X_i = x_i | Y_i = y_i; \theta, \beta) + \text{constant} \quad (10.11)$$

In many cases, this log form is preferable due to its numerical property.

The optimal value of  $\theta$  for the maximum likelihood estimation (when  $\beta$  is fixed) can be written as

$$\theta_{\text{ML}}(\beta) = \operatorname{argmax}_{\theta \in \Theta} \sum_{i=1}^N \log p(X_i = x_i | Y_i = y_i; \theta, \beta). \quad (10.12)$$

Note that the solution is a function of  $\beta$  when it is fixed. We sometimes want to find the optimal value for  $\beta$ , too. In this case, we have

$$(\theta_{\text{ML}}, \beta_{\text{ML}}) = \operatorname{argmax}_{(\theta, \beta) \in \Theta \times \mathcal{B}} \sum_{i=1}^N \log p(X_i = x_i | Y_i = y_i; \theta, \beta). \quad (10.13)$$

Note that both of these are point estimation, *i.e.*, we want to find one value for  $\theta$  (and  $\beta$ ) to maximize the log likelihood function.

## 10.4 Maximum a Posteriori Estimation

In MLE, we regard  $\theta$  as a deterministic variable, which is called Frequentist perspective.

However, we sometimes have prior knowledge of the distribution of  $\theta$  (or belief about  $\theta$ ). In this situation, we want to find probability distribution of  $\theta$  after observing some evidence, *e.g.*, the  $N$  data sample we have observed,  $\{(x_i, y_i)\}_{i=1}^N$ .

The natural way of finding the distribution of  $\theta$  after observing this evidence is to evaluate the condition probability of  $\theta$  given  $\tilde{x}$  and  $\tilde{y}$ , *i.e.*,

$$p(\theta|\tilde{X} = \tilde{x}, \tilde{Y} = \tilde{y}). \quad (10.14)$$

Note that  $\theta$  is considered to be a *random* variable unlike in MLE case.

Here we introduce a new parameter  $\alpha \in \mathcal{A}$  that characterizes the distribution of  $\theta$  where  $\mathcal{A}$  is the set of all the possible values of  $\alpha$ .

Since the data samples are assumed to be independent, the Bayes' theorem implies that

$$\begin{aligned} p(\theta|\tilde{X} = \tilde{x}, \tilde{Y} = \tilde{y}; \alpha, \beta) &= p(\tilde{X} = \tilde{x}, \tilde{Y} = \tilde{y}|\theta; \alpha, \beta)p(\theta; \alpha, \beta)/p(\tilde{X} = \tilde{x}, \tilde{Y} = \tilde{y}) \\ &= p(\tilde{X} = \tilde{x}, \tilde{Y} = \tilde{y}|\theta; \beta)p(\theta; \alpha)/p(\tilde{X} = \tilde{x}, \tilde{Y} = \tilde{y}) \\ &\propto p(\tilde{X} = \tilde{x}, \tilde{Y} = \tilde{y}|\theta; \beta)p(\theta; \alpha) \\ &= p(\theta; \alpha) \prod_{i=1}^N p(X_i = x_i, Y_i = y_i|\theta; \beta) \\ &= p(\theta; \alpha) \prod_{i=1}^N p(Y_i = y_i|X_i = x_i, \theta; \beta)p(X_i = x_i|\theta; \beta) \\ &= p(\theta; \alpha) \prod_{i=1}^N p(Y_i = y_i|X_i = x_i, \theta; \beta)p(X_i = x_i) \\ &\propto p(\theta; \alpha) \prod_{i=1}^N p(Y_i = y_i|X_i = x_i, \theta; \beta) \end{aligned}$$

The maximum a posteriori (MAP) estimation is to find  $\theta$  (when  $\beta$  is fixed) which maximizes this posteriori probability. Thus, the MAP solution can be expressed as

$$\theta_{\text{MAP}}(\alpha, \beta) = \underset{\theta \in \Theta}{\operatorname{argmax}} \left( \log p(\theta; \alpha) + \sum_{i=1}^N \log p(Y_i = y_i|X_i = x_i, \theta; \beta) \right) \quad (10.15)$$

where  $p(Y_i = y_i|X_i = x_i, \theta; \beta)$  is called *likelihood function*.

Note the difference between the likelihood function,  $p(Y_i = y_i|X_i = x_i; \theta, \beta)$ , in (10.12) and the likelihood function,  $p(Y_i = y_i|X_i = x_i, \theta; \beta)$ , in (10.15) where  $\theta$  in (10.12) is an optimization variable and  $\theta$  in (10.15) is a variable for a random variable.

## 10.5 Bayesian prior update

Note that both MLE and MAP estimation is a point estimation, *i.e.*, to find one solution that maximizes some probability.

However, the posterior probability  $p(\theta|\tilde{X} = \tilde{x}, \tilde{Y} = \tilde{y}; \alpha, \beta)$  can be used to update the prior probability.

In Bayesian probability theory, if the posterior distributions are in the same probability distribution family as the prior probability distribution, the prior and posterior are then called *conjugate distributions*, and the prior is called a *conjugate prior* for the likelihood function.

In this case, we can update the prior by updating  $\alpha$ . Suppose that we have initial prior,  $\alpha^{(0)} \in \mathcal{A}$ . After we observing first data samples,  $(\tilde{x}^{(1)}, \tilde{y}^{(1)})$ , we evaluate the posterior probability  $p(\theta|\tilde{X} = \tilde{x}^{(1)}, \tilde{Y} = \tilde{y}^{(1)}; \alpha^{(0)}, \beta)$  which can be characterized by some  $\theta^+$  due to conjugate distribution assumption. We let  $\alpha^{(1)}$  be this updated parameter. We can repeat this process every time we observe new set of data samples. This process can be expressed as

$$\alpha^{(0)} \xrightarrow{\tilde{x}^{(1)}, \tilde{y}^{(1)}} \alpha^{(1)} \xrightarrow{\tilde{x}^{(2)}, \tilde{y}^{(2)}} \alpha^{(2)} \xrightarrow{\tilde{x}^{(3)}, \tilde{y}^{(3)}} \dots \quad (10.16)$$

We can see this process as the one similar to what happens in our brain. A simplified version of explaining human learning process is to update its prior knowledge whenever it observes new evidence. For example, if one has observed that when it rains, the temperature is high, his prior knowledge is that rain  $\rightarrow$  high temperature.



## Chapter 11

# Optimization for Machine Learning



## Chapter 12

# Bayesian Network

## Chapter 13

# Collaborative Filtering

## Chapter 14

# Time Series Anomaly Detection



## Chapter 15

# Reinforcement Learning