

# Machine Learning Notebook

Cong Bao

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	About this Notebook . . . . .	1
1.2	Policy of Use . . . . .	1
<b>2</b>	<b>Mathematics Basics</b>	<b>2</b>
2.1	Probability . . . . .	2
2.1.1	Basic Rules . . . . .	2
2.1.2	Common Probability Distributions . . . . .	4
2.2	Linear Algebra . . . . .	6
2.3	Calculus . . . . .	6
2.4	Informatics . . . . .	6
2.5	Optimization . . . . .	6
<b>3</b>	<b>Machine Learning Basics</b>	<b>7</b>
3.1	Regularization . . . . .	7
3.1.1	Under-fitting & Over-fitting . . . . .	7
3.1.2	Bias & Variance . . . . .	7
3.1.3	Vector Norm . . . . .	7
3.1.4	Penalize Complexity . . . . .	8
3.2	Cross-Validation . . . . .	8
3.3	Bayesian Learning . . . . .	8
3.3.1	Bayes' Rule Terminology . . . . .	8
3.3.2	Maximum Likelihood . . . . .	9
3.3.3	Maximum a Posterior (MAP) . . . . .	9
3.3.4	Bayesian Approach . . . . .	9
3.3.5	Example: Univariate Normal Distribution . . . . .	10
3.3.6	Example: Categorical Distribution . . . . .	12
3.4	Machine Learning Models . . . . .	14
3.4.1	Learning and Inference . . . . .	14

3.4.2	Three Types of Model . . . . .	15
3.4.3	Example: Regression . . . . .	16
3.4.4	Example: Classification . . . . .	17
<b>4</b>	<b>Regression</b>	<b>18</b>
4.1	Linear Regression . . . . .	18
4.2	Non-linear Regression . . . . .	18
4.3	Logistic Regression . . . . .	18
<b>5</b>	<b>Support Vector Machines</b>	<b>19</b>
<b>6</b>	<b>EM Algorithm</b>	<b>20</b>
<b>7</b>	<b>Boosting</b>	<b>21</b>
<b>8</b>	<b>Decision Tree &amp; Random Forest</b>	<b>22</b>
<b>9</b>	<b>Graphical Models &amp; Markov Network</b>	<b>23</b>
	<b>Bibliography</b>	<b>24</b>
<b>A</b>	<b>Statistical Assessment</b>	<b>25</b>
A.1	Hypothesis Testing . . . . .	25
A.2	Confidence Intervals . . . . .	25

# Chapter 1

## Introduction

### 1.1 About this Notebook

### 1.2 Policy of Use

# Chapter 2

## Mathematics Basics

### 2.1 Probability

#### 2.1.1 Basic Rules

**Three Axioms of Probability** Let  $\Omega$  be a sample space. A probability assigns a real number  $P(X)$  to each event  $X \subseteq \Omega$  in such a way that

1.  $P(X) \geq 0, \forall X$
2. If  $X_1, X_2, \dots$  are pairwise disjoint events ( $X_1 \cap X_2 = \emptyset, i \neq j, i, j = 1, 2, \dots$ ), then  $P(\bigcup_{i=1}^{\infty} X_i) = \sum_{i=1}^{\infty} P(X_i)$ . (This property is called countable additivity.)
3.  $P(\Omega) = 1$

**Joint Probability** The probability both event A and B occur.  $P(X, Y) = P(X \cap Y)$ .

**Marginalization** The probability distribution of any variable in a joint distribution can be recovered by integrating (or summing) over the other variables.

1. For continuous r.v.  $P(x) = \int P(x, y) dy ; P(y) = \int P(x, y) dx$ .
2. For discrete r.v.  $P(x) = \sum_y P(x, y) ; P(y) = \sum_x P(x, y)$ .
3. For mixed r.v.  $P(x, y) = \sum_w \int P(w, x, y, z) dz$ , where  $w$  is discrete and  $z$  is continuous.

**Conditional Probability**  $P(X = x|Y = y)$  is the probability  $X = x$  occurs given the knowledge  $Y = y$  occurs. Conditional probability can be extracted from joint probability that

$$P(x|y = y^*) = \frac{P(x, y = y^*)}{\int P(x, y = y^*) dx} = \frac{P(x, y = y^*)}{P(y = y^*)}$$

Usually, the formula is written as  $P(x|y) = \frac{P(x, y)}{P(y)}$ .

**Product Rule** The formula can be rearranged as  $P(x, y) = P(x|y) P(y) = P(y|x) P(x)$ .  
In case of multiple variables

$$\begin{aligned} P(w, x, y, z) &= P(w, x, y|z) P(z) \\ &= P(w, x|y, z) P(y|z) P(z) \\ &= P(w|x, y, z) P(x|y, z) P(y|z) P(z) \end{aligned}$$

**Independence** If two variables  $x$  and  $y$  are independent, then r.v.  $x$  tells nothing about r.v.  $y$  (and vice-versa)

$$\begin{aligned} P(x|y) &= P(x) \\ P(y|x) &= P(y) \\ P(x, y) &= P(x) P(y) \end{aligned}$$

**Baye's Rule** By rearranging formula in Product Rule, we have

$$\begin{aligned} P(y|x) &= \frac{P(x|y) P(y)}{P(x)} \\ &= \frac{P(x|y) P(y)}{\int P(x, y) dy} \\ &= \frac{P(x|y) P(y)}{\int P(x|y) P(y) dy} \end{aligned}$$

**Expectation** Expectation tells us the expected or average value of some function  $f(x)$ , taking into account the distribution of  $x$ .

$$\begin{aligned} \mathbf{E}[f(x)] &= \sum_x f(x) P(x) \\ \mathbf{E}[f(x)] &= \int f(x) P(x) dx \end{aligned}$$

Definition in two dimensions:  $\mathbf{E}[f(x, y)] = \iint f(x, y) P(x, y) dx dy$

Function $f(\bullet)$	Expectation
$x$	mean, $\mu_x$
$(x - \mu_x)^2$	variance
$(x - \mu_x)^3$	skew
$(x - \mu_x)^4$	kurtosis
$(x - \mu_x)(x - \mu_y)$	covariance of $x$ and $y$

Besides, Expectation has the following four rules

1. Expected value of a constant is the constant  $\mathbf{E}[\kappa] = \kappa$ .
2. Expected value of constant times function is constant times expected value of function  $\mathbf{E}[kf(x)] = k\mathbf{E}[f(x)]$ .
3. Expectation of sum of functions is sum of expectation of functions  $\mathbf{E}[f(x) + g(y)] = \mathbf{E}[f(x)] + \mathbf{E}[g(y)]$ .
4. Expectation of product of functions in variables  $x$  and  $y$  is product of expectations of functions if  $x$  and  $y$  are independent  $\mathbf{E}[f(x)g(y)] = \mathbf{E}[f(x)]\mathbf{E}[g(y)]$ ,  $x \perp y$ .

### 2.1.2 Common Probability Distributions

**Bernoulli** Bernoulli distribution describes situation where only two possible outcomes  $y = 0/y = 1$  or failure/success.

1.  $P(x) = \mathbf{Bern}_x[\lambda] = \lambda^x(1 - \lambda)^{1-x}$
2. univariate, discrete, binary
3.  $x \in \{0, 1\}$ ;  $\lambda \in [0, 1]$
4.  $\mathbf{E}[x] = \lambda$ ,  $\mathbf{Var}[x] = \lambda(1 - \lambda)$

**Beta** Beta distribution is the conjugate distribution to Bernoulli distribution.

1.  $P(\lambda) = \mathbf{Beta}_\lambda[\alpha, \beta] = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} \lambda^{\alpha-1}(1 - \lambda)^{\beta-1}$
2. univariate, continuous, unbounded
3.  $\lambda \in \mathbb{R}$ ;  $\alpha \in \mathbb{R}_+$ ,  $\beta \in \mathbb{R}_+$
4.  $\mathbf{E}[\lambda] = \frac{\alpha}{\alpha+\beta}$ ,  $\mathbf{Var}[\lambda] = \frac{\alpha\beta}{(\alpha+\beta)^2(\alpha+\beta+1)}$

**Categorical** Categorical distribution describes situation with  $K$  possible outcomes.

1.  $P(x) = \mathbf{Cat}_x[\boldsymbol{\lambda}]$ ,  $P(x = k) = \lambda_k$ ,  $P(\mathbf{x} = \mathbf{e}_k) = \prod_{j=1}^K \lambda_j^{x_j} = \lambda_k$
2. univariate, discrete, multi-valued
3.  $x \in \{1, 2, \dots, K\}$ ;  $\lambda_k \in [0, 1]$  where  $\sum_k \lambda_k = 1$
4.  $\mathbf{E}[x_i] = \lambda_i$ ,  $\mathbf{Var}[x_i] = \lambda_i(1 - \lambda_i)$ ,  $\mathbf{Cov}[x_i, x_j] = -\lambda_i\lambda_j$  ( $i \neq j$ )

**Dirichlet** Dirichlet distribution is the conjugate distribution to categorical distribution.

1.  $P(\boldsymbol{\lambda}) = \mathbf{Dir}_\lambda[\boldsymbol{\alpha}] = \frac{\Gamma(\sum_{k=1}^K \alpha_k)}{\prod_{k=1}^K \Gamma(\alpha_k)} \prod_{k=1}^K \lambda_k^{\alpha_k-1}$
2. multivariate, continuous, bounded, sums to one
3.  $\boldsymbol{\lambda} = [\lambda_1, \lambda_2, \dots, \lambda_K]^\top$ ,  $\lambda_k \in [0, 1]$ ,  $\sum_{k=1}^K \lambda_k = 1$ ;  $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \dots, \alpha_K]$ ,  $\alpha_k \in \mathbb{R}_+$

$$4. \mathbf{E}[\lambda_i] = \frac{\alpha_i}{\sum_k \alpha_k}, \mathbf{Var}[\lambda_i] = \frac{\alpha_i(\sum_k \alpha_k - \alpha_i)}{(\sum_k \alpha_k)^2(\sum_k \alpha_k + 1)}, \mathbf{Cov}[\lambda_i, \lambda_j] = \frac{-\alpha_i \alpha_j}{(\sum_k \alpha_k)^2(\sum_k \alpha_k + 1)} \quad (i \neq j)$$

**Univariate Normal** Univariate normal distribution describes single continuous variable.

1.  $P(x) = \mathbf{Norm}_x[\mu, \sigma^2] = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$
2. univariate, continuous, unbounded
3.  $x \in \mathbb{R}; \mu \in \mathbb{R}, \sigma^2 \in \mathbb{R}_+$
4.  $\mathbf{E}[x] = \mu, \mathbf{Var}[x] = \sigma^2$

**Normal Inverse Gamma** Normal inverse gamma distribution is a conjugate distribution to univariate normal distribution.

1.  $P(\mu, \sigma^2) = \mathbf{NormInvGam}_{\mu, \sigma^2}[\alpha, \beta, \gamma, \delta] = \frac{\sqrt{\gamma}}{\sqrt{2\pi\sigma^2}} \frac{\beta^\alpha}{\Gamma(\alpha)} \left(\frac{1}{\sigma^2}\right)^{\alpha+1} \exp\left(-\frac{2\beta + \gamma(\delta - \mu)^2}{2\sigma^2}\right)$
2. bivariate, continuous,  $\mu$  unbounded,  $\sigma^2$  bounded below
3.  $\mu \in \mathbb{R}, \sigma^2 \in \mathbb{R}_+; \alpha \in \mathbb{R}_+, \beta \in \mathbb{R}_+, \gamma \in \mathbb{R}_+, \delta \in \mathbb{R}$
4.  $\mathbf{E}[\mu] = \delta, \mathbf{E}[\sigma^2] = \frac{\beta}{\alpha-1} \quad (\alpha > 1), \mathbf{Var}[\mu] = \frac{\beta}{(\alpha-1)\gamma} \quad (\alpha > 1), \mathbf{Var}[\sigma^2] = \frac{\beta^2}{(\alpha-1)^2(\alpha-2)} \quad (\alpha > 2), \mathbf{Cov}[\mu, \sigma^2] = 0 \quad (\alpha > 1)$

**Multivariate Normal** Multivariate normal distribution describes multiple continuous variables. It takes two parameters: a vector containing mean position  $\boldsymbol{\mu}$ , and a symmetric positive definite covariance matrix  $\boldsymbol{\Sigma}$ .

1.  $P(\mathbf{x}) = \mathbf{Norm}_x[\boldsymbol{\mu}, \boldsymbol{\Sigma}] = \frac{1}{\sqrt{\det(2\pi\boldsymbol{\Sigma})}} \exp\left(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu})\right)$
2. multivariate, continuous, unbounded
3.  $\mathbf{x} \in \mathbb{R}^K; \boldsymbol{\mu} \in \mathbb{R}^K, \boldsymbol{\Sigma} \in \mathbb{R}^{K \times K}$  (positive semi-definite matrix)
4.  $\mathbf{E}[\mathbf{x}] = \boldsymbol{\mu}, \mathbf{Var}[\mathbf{x}] = \boldsymbol{\Sigma}$

**Normal Inverse Wishart** Normal inverse wishart distribution is a conjugate distribution to multivariate normal distribution.

1.  $P(\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \mathbf{NormInvWis}_{\boldsymbol{\mu}, \boldsymbol{\Sigma}}[\alpha, \boldsymbol{\Psi}, \gamma, \boldsymbol{\delta}]$   
 $= \frac{\gamma^{D/2} |\boldsymbol{\Psi}|^{\alpha/2} |\boldsymbol{\Sigma}|^{-\frac{\alpha+D+2}{2}}}{(2\pi)^{D/2} 2^{(\alpha\boldsymbol{\Sigma})/2} \Gamma_D(\alpha/2)} \exp\left(-\frac{1}{2}(\text{Tr}(\boldsymbol{\Psi}\boldsymbol{\Sigma}^{-1}) + \gamma(\boldsymbol{\mu} - \boldsymbol{\delta})^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu} - \boldsymbol{\delta}))\right)$
2. multivariate,  $\boldsymbol{\mu}$  unbounded,  $\boldsymbol{\Sigma}$  square, positive definite
3.  $\boldsymbol{\mu} \in \mathbb{R}^K, \boldsymbol{\Sigma} \in \mathbb{R}^{K \times K}; \alpha \in \mathbb{R}_{>D-1}, \boldsymbol{\Psi} \in \mathbb{R}^{K \times K}, \gamma \in \mathbb{R}_+, \boldsymbol{\delta} \in \mathbb{R}^K$



## **2.2 Linear Algebra**

## **2.3 Calculus**

## **2.4 Informatics**

## **2.5 Optimization**

# Chapter 3

## Machine Learning Basics

### 3.1 Regularization

#### 3.1.1 Under-fitting & Over-fitting

**Under-fitting** If  $N > D$  (e.g. 30 data points, 2 dimensions) we have more equations than unknowns: over-determined system. Input-output relations can only hold approximately.

**Over-fitting** If  $N < D$  (e.g. 30 points, 15265 dimensions) we have more unknowns than equations: under-determined system. Input-output equations hold exactly, but we are simply memorizing data.

#### 3.1.2 Bias & Variance

**High Bias & Low Variance** A rigid model's (low complexity) performance is more predictable in the test set but the model may not be good even on the training set.

**Low Bias & High Variance** A flexible model (high complexity) approximates the target function well in the training set but can “overtrain” and have poor performance on the test set.

#### 3.1.3 Vector Norm

**L1, (“Manhattan”) norm**  $\|\mathbf{w}\|_1 = \sum_{d=1}^D |w_d|$

**L2, (“Euclidean”) norm**  $\|\mathbf{w}\|_2 = \sqrt{\sum_{d=1}^D w_d^2} = \sqrt{\langle \mathbf{w}, \mathbf{w} \rangle} = \sqrt{\mathbf{w}^\top \mathbf{w}}$

**Lp norm,  $p > 1$**   $\|\mathbf{w}\|_p = \left( \sum_{d=1}^D w_d^p \right)^{1/p}$

### 3.1.4 Penalize Complexity

In linear regression, the residual vector is  $\epsilon = \mathbf{y} - \Psi\mathbf{w}$ . The loss function is  $L(\mathbf{w}) = \epsilon^\top \epsilon$ . We add a complexity term  $R(\mathbf{w}) = \|\mathbf{w}\|_2^2 = \mathbf{w}^\top \mathbf{w}$  to the loss function. Hence, the original loss function becomes  $L(\mathbf{w}) = \epsilon^\top \epsilon + \lambda \mathbf{w}^\top \mathbf{w}$ .

Without regularization, the loss function is  $L(\mathbf{w}) = \epsilon^\top \epsilon$ . Let  $\nabla L(\mathbf{w}^*) = 0$ , we have  $\mathbf{w}^* = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y}$ .

With L2-regularization, the loss function is  $L(\mathbf{w}) = \epsilon^\top \epsilon + \lambda \mathbf{w}^\top \mathbf{w}$ . Let  $\nabla L(\mathbf{w}^*) = 0$ , we have  $\mathbf{w}^* = (\mathbf{X}^\top \mathbf{X} + \lambda \mathbf{I})^{-1} \mathbf{X}^\top \mathbf{y}$ . The additional  $\lambda \mathbf{I}$  makes the data matrix more robust to calculate inversion.

## 3.2 Cross-Validation

We can select hyperparameters with (cross-)validation. Cross-validation excludes part of the training data from parameter estimation, and use them only to predict the test error.

K-fold cross validation: split data set into K folds and each time train on (K-1) folds and valid on the remaining fold until all folds have been used as validation fold. The cross-validation error is the average of K validation errors. We pick hyperparameters that minimize cross-validation error.

## 3.3 Bayesian Learning

### 3.3.1 Bayes' Rule Terminology

Bayes' Rule:

$$P(y|x) = \frac{P(x|y) P(y)}{\int P(x|y) P(y) dy}$$

**Prior**  $P(y)$  what we know about  $y$  before seeing  $x$ . In parameters learning we choose prior that is conjugate to likelihood.

**Likelihood**  $P(x|y)$  propensity for observing a certain value of  $x$  given a certain value of  $y$ .

**Posterior**  $P(y|x)$  what we know about  $y$  after seeing  $x$ . Posterior must have same form as conjugate prior distribution.

**Evidence**  $\int P(x|y) P(y) dy$  a constant to ensure that the LHS is a valid distribution. Posterior must be a distribution which implies that evidence equals to a constant  $\kappa$  from conjugate relation.

### 3.3.2 Maximum Likelihood

**Fitting** As the name suggests we find the parameters under which the data  $\mathbf{x}_{1...I}$  are most likely. Here, we have assumed that data was independent.

$$\begin{aligned}\hat{\boldsymbol{\theta}} &= \operatorname{argmax}_{\boldsymbol{\theta}} P(\mathbf{x}_{1...I}|\boldsymbol{\theta}) \\ &= \operatorname{argmax}_{\boldsymbol{\theta}} \prod_{i=1}^I P(\mathbf{x}_i|\boldsymbol{\theta})\end{aligned}$$

**Predictive Density** Evaluate new data point  $\mathbf{x}^*$  under probability distribution  $P(\mathbf{x}^*|\hat{\boldsymbol{\theta}})$  with best parameters.

### 3.3.3 Maximum a Posterior (MAP)

**Fitting** As the name suggests we find the parameters which maximize the posterior probability  $P(\boldsymbol{\theta}|\mathbf{x}_{1...I})$ . Again we have assumed that data was independent.

$$\begin{aligned}\hat{\boldsymbol{\theta}} &= \operatorname{argmax}_{\boldsymbol{\theta}} P(\boldsymbol{\theta}|\mathbf{x}_{1...I}) \\ &= \operatorname{argmax}_{\boldsymbol{\theta}} \frac{P(\mathbf{x}_{1...I}|\boldsymbol{\theta}) P(\boldsymbol{\theta})}{P(\mathbf{x}_{1...I})} \\ &= \operatorname{argmax}_{\boldsymbol{\theta}} \frac{\prod_{i=1}^I P(\mathbf{x}_i|\boldsymbol{\theta}) P(\boldsymbol{\theta})}{P(\mathbf{x}_{1...I})}\end{aligned}$$

Since the denominator does not depend on the parameters we can instead maximize

$$\hat{\boldsymbol{\theta}} = \operatorname{argmax}_{\boldsymbol{\theta}} \prod_{i=1}^I P(\mathbf{x}_i|\boldsymbol{\theta}) P(\boldsymbol{\theta})$$

**Predictive Density** Evaluate new data point  $\mathbf{x}^*$  under probability distribution with MAP parameters  $P(\mathbf{x}^*|\hat{\boldsymbol{\theta}})$

### 3.3.4 Bayesian Approach

**Fitting** Compute the posterior distribution over possible parameter values using Bayes' rule. Principle: There are many values that could have explained the data. Instead of picking one set of parameters, try to capture all of the possibilities.

$$P(\boldsymbol{\theta}|\mathbf{x}_{1...I}) = \frac{\prod_{i=1}^I P(\mathbf{x}_i|\boldsymbol{\theta}) P(\boldsymbol{\theta})}{P(\mathbf{x}_{1...I})}$$

**Predictive Density** (a) Each possible parameter value makes a prediction. (b) Some parameters more probable than others.

$$P(\mathbf{x}^*|\mathbf{x}_{1...I}) = \int P(\mathbf{x}^*|\boldsymbol{\theta}) P(\boldsymbol{\theta}|\mathbf{x}_{1...I}) d\boldsymbol{\theta}$$

Make a prediction that is an infinite weighted sum (integral) of the predictions for each parameter value ( $P(\mathbf{x}^*|\boldsymbol{\theta})$ ), where weights are the probabilities ( $P(\boldsymbol{\theta}|\mathbf{x}_{1...I})$ ).

### 3.3.5 Example: Univariate Normal Distribution

#### Maximum Likelihood

Likelihood given by normal distribution pdf:

$$P(x|\mu, \sigma^2) = \mathbf{Norm}_x[\mu, \sigma^2] = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x - \mu)^2}{2\sigma^2}\right)$$

Apply maximum likelihood:

$$\begin{aligned} \hat{\mu}, \hat{\sigma}^2 &= \operatorname{argmax}_{\mu, \sigma^2} P(\mathbf{x}_{1...I}|\mu, \sigma^2) \\ &= \operatorname{argmax}_{\mu, \sigma^2} \prod_{i=1}^I P(x_i|\mu, \sigma^2) \\ &= \operatorname{argmax}_{\mu, \sigma^2} \prod_{i=1}^I \mathbf{Norm}_{x_i}[\mu, \sigma^2] \\ &= \operatorname{argmax}_{\mu, \sigma^2} \sum_{i=1}^I \log \mathbf{Norm}_{x_i}[\mu, \sigma^2] \\ &= \operatorname{argmax}_{\mu, \sigma^2} \left( -\frac{I}{2} \log 2\pi - \frac{I}{2} \log \sigma^2 - \frac{1}{2} \sum_{i=1}^I \frac{(x_i - \mu)^2}{\sigma^2} \right) \end{aligned}$$

Let  $\nabla L(\hat{\mu}, \hat{\sigma}^2) = 0$ , we have the solution:

$$\begin{aligned} \hat{\mu} &= \frac{\sum_{i=1}^I x_i}{I} \\ \hat{\sigma}^2 &= \sum_{i=1}^I \frac{(x_i - \hat{\mu})^2}{I} \end{aligned}$$

#### Maximum a Posterior

Likelihood given by normal distribution pdf:

$$P(x|\mu, \sigma^2) = \mathbf{Norm}_x[\mu, \sigma^2] = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x - \mu)^2}{2\sigma^2}\right)$$

Prior given by normal inverse gamma distribution pdf:

$$P(\mu, \sigma^2) = \mathbf{NormInvGam}_{\mu, \sigma^2}[\alpha, \beta, \gamma, \delta] = \frac{\sqrt{\gamma}}{\sqrt{2\pi\sigma^2}} \frac{\beta^\alpha}{\Gamma(\alpha)} \left(\frac{1}{\sigma^2}\right)^{\alpha+1} \exp\left(-\frac{2\beta + \gamma(\delta - \mu)^2}{2\sigma^2}\right)$$

Apply maximum a posterior:

$$\begin{aligned} \hat{\mu}, \hat{\sigma}^2 &= \underset{\mu, \sigma^2}{\operatorname{argmax}} \prod_{i=1}^I P(x_i | \mu, \sigma^2) P(\mu, \sigma^2) \\ &= \underset{\mu, \sigma^2}{\operatorname{argmax}} \prod_{i=1}^I \mathbf{Norm}_{x_i}[\mu, \sigma^2] \mathbf{NormInvGam}_{\mu, \sigma^2}[\alpha, \beta, \gamma, \delta] \\ &= \underset{\mu, \sigma^2}{\operatorname{argmax}} \left( \sum_{i=1}^I \log \mathbf{Norm}_{x_i}[\mu, \sigma^2] + \log \mathbf{NormInvGam}_{\mu, \sigma^2}[\alpha, \beta, \gamma, \delta] \right) \end{aligned}$$

Let  $\nabla L(\hat{\mu}, \hat{\sigma}^2) = 0$ , we have the solution:

$$\begin{aligned} \hat{\mu} &= \frac{\sum_{i=1}^I x_i + \gamma\delta}{I + \gamma} \\ \hat{\sigma}^2 &= \frac{\sum_{i=1}^I (x_i - \mu)^2 + 2\beta + \gamma(\delta - \mu)^2}{I + 3 + 2\alpha} \end{aligned}$$

## Bayesian Approach

Compute the posterior distribution using Bayes' rule:

$$\begin{aligned} P(\mu, \sigma^2 | x_{1...I}) &= \frac{\prod_{i=1}^I P(x_i | \mu, \sigma^2) P(\mu, \sigma^2)}{P(x_{1...I})} \\ &= \frac{\prod_{i=1}^I \mathbf{Norm}_{x_i}[\mu, \sigma^2] \mathbf{NormInvGam}_{\mu, \sigma^2}[\alpha, \beta, \gamma, \delta]}{P(x_{1...I})} \\ &= \frac{\kappa(\alpha, \beta, \gamma, \delta, x_{1...I}) \mathbf{NormInvGam}_{\mu, \sigma^2}[\tilde{\alpha}, \tilde{\beta}, \tilde{\gamma}, \tilde{\delta}]}{P(x_{1...I})} \\ &= \mathbf{NormInvGam}_{\mu, \sigma^2}[\tilde{\alpha}, \tilde{\beta}, \tilde{\gamma}, \tilde{\delta}] \end{aligned}$$

where

$$\begin{aligned} \tilde{\alpha} &= \alpha + \frac{I}{2} \\ \tilde{\beta} &= \frac{\sum_i x_i^2}{2} + \beta + \frac{\gamma\delta^2}{2} - \frac{(\gamma\delta + \sum_i x_i)^2}{2(\gamma + I)} \\ \tilde{\gamma} &= \gamma + I \\ \tilde{\delta} &= \frac{\gamma\delta + \sum_i x_i}{\gamma + I} \end{aligned}$$

Take weighted sum of predictions from different parameter values:

$$\begin{aligned}
P(x^*|x_{1...I}) &= \iint P(x^*|\mu, \sigma^2) P(\mu, \sigma^2|x_{1...I}) d\mu d\sigma \\
&= \iint \mathbf{Norm}_{x^*}[\mu, \sigma^2] \mathbf{NormInvGam}_{\mu, \sigma^2}[\tilde{\alpha}, \tilde{\beta}, \tilde{\gamma}, \tilde{\delta}] d\mu d\sigma \\
&= \iint \kappa(\alpha, \beta, \gamma, \delta, x_{1...I}) \mathbf{NormInvGam}_{\mu, \sigma^2}[\check{\alpha}, \check{\beta}, \check{\gamma}, \check{\delta}] d\mu d\sigma \\
&= \kappa(\alpha, \beta, \gamma, \delta, x_{1...I}) \iint \mathbf{NormInvGam}_{\mu, \sigma^2}[\check{\alpha}, \check{\beta}, \check{\gamma}, \check{\delta}] d\mu d\sigma \\
&= \kappa(\alpha, \beta, \gamma, \delta, x_{1...I}) \\
&= \frac{1}{\sqrt{2\pi}} \frac{\sqrt{\tilde{\gamma}} \tilde{\beta}^{\tilde{\alpha}} \Gamma(\check{\alpha})}{\sqrt{\check{\gamma}} \check{\beta}^{\check{\alpha}} \Gamma(\check{\alpha})}
\end{aligned}$$

where

$$\begin{aligned}
\check{\alpha} &= \tilde{\alpha} + \frac{1}{2} \\
\check{\beta} &= \frac{x^{*2}}{2} + \tilde{\beta} + \frac{\tilde{\gamma} \tilde{\delta}^2}{2} - \frac{(\tilde{\gamma} \tilde{\delta} + x^*)^2}{2(\tilde{\gamma} + 1)} \\
\check{\gamma} &= \tilde{\gamma} + 1
\end{aligned}$$

### 3.3.6 Example: Categorical Distribution

#### Maximum Likelihood

Likelihood given by categorical distribution pdf:

$$P(x|\boldsymbol{\lambda}) = \mathbf{Cat}_x[\boldsymbol{\lambda}] = \prod_{j=1}^K \lambda_j^{x_j} = \lambda_k$$

Apply maximum likelihood:

$$\begin{aligned}
\hat{\boldsymbol{\lambda}} &= \underset{\boldsymbol{\lambda}}{\operatorname{argmax}} \prod_{i=1}^I P(x_i|\boldsymbol{\lambda}) & s.t. \sum_k \lambda_k &= 1 \\
&= \underset{\boldsymbol{\lambda}}{\operatorname{argmax}} \prod_{i=1}^I \mathbf{Cat}_{x_i}[\boldsymbol{\lambda}] & s.t. \sum_k \lambda_k &= 1 \\
&= \underset{\boldsymbol{\lambda}}{\operatorname{argmax}} \prod_{k=1}^K \lambda_k^{N_k} & s.t. \sum_k \lambda_k &= 1 \\
&= \underset{\boldsymbol{\lambda}}{\operatorname{argmax}} \sum_{k=1}^K N_k \log \lambda_k & s.t. \sum_k \lambda_k &= 1
\end{aligned}$$

Here,  $N_k$  represents the number of times the data is classified in class  $k$ . As before, we will instead optimize log probability. Since there is a constraint *s.t.*  $\sum_k \lambda_k = 1$ , we use Lagrange multiplier to reconstruct the loss function.

$$L(\boldsymbol{\lambda}) = \sum_{k=1}^K N_k \log \lambda_k + v \left( \sum_{k=1}^K \lambda_k - 1 \right)$$

Let  $\nabla L(\boldsymbol{\lambda}, v) = 0$ , we have the solution:

$$\hat{\lambda}_k = \frac{N_k}{\sum_{m=1}^K N_m}$$

### Maximum a Posterior

Likelihood given by categorical distribution pdf:

$$P(x|\boldsymbol{\lambda}) = \mathbf{Cat}_x[\boldsymbol{\lambda}] = \prod_{j=1}^K \lambda_j^{x_j} = \lambda_k$$

Prior given by Dirichlet distribution pdf:

$$P(\boldsymbol{\lambda}) = \mathbf{Dir}_{\boldsymbol{\lambda}}[\boldsymbol{\alpha}] = \frac{\Gamma(\sum_{k=1}^K \alpha_k)}{\prod_{k=1}^K \Gamma(\alpha_k)} \prod_{k=1}^K \lambda_k^{\alpha_k - 1}$$

Apply maximum a posterior:

$$\begin{aligned} \hat{\boldsymbol{\lambda}} &= \underset{\boldsymbol{\lambda}}{\operatorname{argmax}} \prod_{i=1}^I P(x_i|\boldsymbol{\lambda}) P(\boldsymbol{\lambda}) & \text{s.t. } \sum_k \lambda_k &= 1 \\ &= \underset{\boldsymbol{\lambda}}{\operatorname{argmax}} \mathbf{Cat}_{x_i}[\boldsymbol{\lambda}] \mathbf{Dir}_{\boldsymbol{\lambda}}[\boldsymbol{\alpha}] & \text{s.t. } \sum_k \lambda_k &= 1 \\ &= \underset{\boldsymbol{\lambda}}{\operatorname{argmax}} \prod_{k=1}^K \lambda_k^{N_k} \prod_{k=1}^K \lambda_k^{\alpha_k - 1} & \text{s.t. } \sum_k \lambda_k &= 1 \\ &= \underset{\boldsymbol{\lambda}}{\operatorname{argmax}} \prod_{k=1}^K \lambda_k^{N_k + \alpha_k - 1} & \text{s.t. } \sum_k \lambda_k &= 1 \\ &= \underset{\boldsymbol{\lambda}}{\operatorname{argmax}} \sum_{k=1}^K (N_k + \alpha_k - 1) \log \lambda_k & \text{s.t. } \sum_k \lambda_k &= 1 \end{aligned}$$

The loss function is very similar to maximum likelihood (same when the prior is uniform, i.e.  $\alpha_{1\dots k} = 1$ ). Take derivative with Lagrange multiplier, we have the solution:

$$\hat{\lambda}_k = \frac{N_k + \alpha_k - 1}{\sum_{m=1}^K (N_m + \alpha_m - 1)}$$



## Bayesian Approach

Compute the posterior distribution using Bayes' rule:

$$\begin{aligned}
 P(\boldsymbol{\lambda}|x_{1...I}) &= \frac{\prod_{i=1}^I P(x_i|\boldsymbol{\lambda}) P(\boldsymbol{\lambda})}{P(x_{1...I})} \\
 &= \frac{\prod_{i=1}^I \text{Cat}_{x_i}[\boldsymbol{\lambda}] \text{Dir}_{\boldsymbol{\lambda}}[\boldsymbol{\alpha}]}{P(x_{1...I})} \\
 &= \frac{\kappa(\boldsymbol{\alpha}, x_{1...I}) \text{Dir}_{\boldsymbol{\lambda}}[\tilde{\boldsymbol{\alpha}}]}{P(x_{1...I})} \\
 &= \text{Dir}_{\boldsymbol{\lambda}}[\tilde{\boldsymbol{\alpha}}]
 \end{aligned}$$

Compute predictive distribution:

$$\begin{aligned}
 P(x^*|x_{1...I}) &= \int P(x^*|\boldsymbol{\lambda}) P(\boldsymbol{\lambda}|x_{1...I}) d\boldsymbol{\lambda} \\
 &= \int \text{Cat}_{x^*}[\boldsymbol{\lambda}] \text{Dir}_{\boldsymbol{\lambda}}[\tilde{\boldsymbol{\alpha}}] d\boldsymbol{\lambda} \\
 &= \int \kappa(x^*, \tilde{\boldsymbol{\alpha}}) \text{Dir}_{\boldsymbol{\lambda}}[\tilde{\boldsymbol{\alpha}}] d\boldsymbol{\lambda} \\
 &= \kappa(x^*, \boldsymbol{\alpha})
 \end{aligned}$$

## 3.4 Machine Learning Models

### 3.4.1 Learning and Inference

In real world problems, we usually have two tasks:

1. Observe measured data,  $\mathbf{x}$
2. Draw inferences from it about world,  $\mathbf{w}$

and

1. When the world state  $\mathbf{w}$  is *continuous*, we'll call this *regression*.
2. When the world state  $\mathbf{w}$  is *discrete*, we'll call this *classification*.

We want take observations  $\mathbf{x}$ , and return probability distribution  $P(\mathbf{w}|\mathbf{x})$  over possible worlds compatible with data. To solve this, we need

1. A *model* that mathematically relates the visual data  $\mathbf{x}$  to the world state  $\mathbf{w}$ . Model specifies family of relationships, particular relationship depends on parameter  $\theta$ .
2. A *learning algorithm* fits parameters  $\theta$  from paired training examples  $\mathbf{x}_i, \mathbf{w}_i$ .
3. An *inference algorithm* uses model to return  $P(\mathbf{w}|\mathbf{x})$  given new observation data  $\mathbf{x}$ .

### 3.4.2 Three Types of Model

We have three types of model:

1. Model contingency of the world on the data  $P(w|x)$ . (Discriminative Model)
2. Model joint occurrence of world and data  $P(x, w)$ . (Generative Model)
3. Model contingency of data on world  $P(x|w)$ . (Generative Model)

Within the three models, type 1 is called *Discriminative Model*. Type 2 and 3 are called *Generative Model*.

#### Model $P(w|x)$ - Discriminative

1.  $P(w|x, \theta) = \mathbf{Distrib}_w[f(x, \theta)]$
2. How to model: (a) Choose an appropriate form for  $P(w)$ . (b) Make parameters a function of  $x$ . (c) Function takes parameters  $\theta$  that define its shape.
3. Learning algorithm: Learn parameters  $\theta$  from training data  $x, w$ .
4. Inference algorithm: Just evaluate  $P(w|x)$

#### Model $P(x, w)$ - Generative

1.  $P(z|\theta) = \mathbf{Distrib}_z[\theta]$
2. How to model: (a) Concatenate  $x$  and  $w$  to make  $z = [x^\top, w^\top]^\top$ . (b) Model the pdf of  $z$ . (c) pdf takes parameters  $\theta$  that define its shape.
3. Learning algorithm: Learn parameters  $\theta$  from training data  $x, w$ .
4. Inference algorithm: Compute  $P(w|x)$  using Bayes' rule  $P(w|x) = \frac{P(x, w)}{P(x)} = \frac{P(x, w)}{\int P(x, w) dw}$ .

#### Model $P(x|w)$ - Generative

1.  $P(x|w, \theta) = \mathbf{Distrib}_x[f(w, \theta)]$
2. How to model: (a) Choose an appropriate form for  $P(x)$ . (b) Make parameters a function of  $w$ . (c) Function takes parameters  $\theta$  that define its shape.
3. Learning algorithm: Learn parameters  $\theta$  from training data  $x, w$ .
4. Define prior  $P(w)$  and then compute  $P(w|x)$  using Bayes' rule  $P(w|x) = \frac{P(x|w)P(w)}{\int P(x|w)P(w)dw}$ .

### 3.4.3 Example: Regression

Consider a simple case:

1. We make a univariate continuous measurement  $x$ .
2. Use this to predict a univariate continuous state  $w$ .

#### Model $P(w|x)$ - Discriminative

1.  $P(w|x, \theta) = \text{Norm}_w[\phi_0 + \phi_1 x, \sigma^2], \theta = \{\phi_0, \phi_1, \sigma^2\}$
2. How to model: (a) Choose normal distribution over  $w$ . (b) Make mean  $\mu$  linear function of  $x$  (variance constant). (c) Parameters are  $\phi_0$  (y-offset),  $\phi_1$  (slope),  $\sigma^2$  (variance). This model is called *linear regression*.
3. Learning algorithm: Learn  $\theta$  from training data  $x, w$ . e.g. MAP:

$$\begin{aligned}\hat{\theta} &= \underset{\theta}{\operatorname{argmax}} P(\theta | w_{1...I}, x_{1...I}) \\ &= \underset{\theta}{\operatorname{argmax}} P(w_{1...I} | x_{1...I}, \theta) P(\theta) \\ &= \underset{\theta}{\operatorname{argmax}} \prod_{i=1}^I P(w_i | x_i, \theta) P(\theta)\end{aligned}$$

4. Inference algorithm: Just evaluate  $P(w|x)$  for new data  $x$ .

#### Model $P(x, w)$ - Generative

1.  $P(x, w | \theta) = \text{Norm}_{x,w}[\mu, \Sigma], \theta = \{\mu, \Sigma\}$
2. How to model: (a) Concatenate  $x$  and  $w$  to make  $z = [x^\top, w^\top]^\top$ . (b) Model the pdf of  $z$  as normal distribution. (c) pdf makes parameters  $\mu$  and  $\Sigma$  that define its shape.
3. Learning algorithm: Learn parameters  $\theta$  from training data  $x, w$ .
4. Inference algorithm: Compute  $P(w|x)$  using Bayes' rule  $P(w|x) = \frac{P(x,w)}{P(x)} = \frac{P(x,w)}{\int P(x,w) dw}$ .

#### Model $P(x|w)$ - Generative

1.  $P(x|w, \theta) = \text{Norm}_x[\phi_0 + \phi_1 w, \sigma^2], \theta = \{\phi_0, \phi_1, \sigma^2\}$
2. How to model: (a) Choose normal distribution over  $x$ . (b) Make mean  $\mu$  linear function of  $w$  (variance constant). (c) Parameters are  $\phi_0, \phi_1, \sigma^2$ .

3. Learning algorithm: Learn  $\theta$  from training data  $x, w$ . e.g. MAP
4. Inference algorithm: Compute  $P(w|x)$  using Bayes' rule  $P(w|x) = \frac{P(x,w)}{P(x)} = \frac{P(x,w)}{\int P(x,w)dw}$ .

### 3.4.4 Example: Classification

Consider a simple case:

1. We make a univariate continuous measurement  $x$ .
2. Use this to predict a discrete binary world  $w \in \{0, 1\}$ .

#### Model $P(w|x)$ - Discriminative

1.  $P(w|x, \theta) = \text{Bern}_w[\sigma(\phi_0 + \phi_1 x)], \theta = \phi_0, \phi_1$
2. How to model: (a) Choose Bernoulli distribution for  $P(w)$ . (b) Make parameters a sigmoid-activated function of  $x$ . (c) Function takes parameters  $\phi_0$  and  $\phi_1$ . This model is called *logistic regression*.
3. Learning algorithm: Learning by standard methods, e.g. ML, MAP, Bayesian Approach.
4. Inference algorithm: Just evaluate  $P(w|x)$ .

#### Model $P(x, w)$ - Generative

Can't build this mode very easily:

1. Concatenate continuous vector  $x$  and discrete  $w$  to make  $z$ .
2. No obvious probability distribution to model joint probability of discrete and continuous.

#### Model $P(x|w)$ - Generative

1.  $P(x|w, \theta) = \text{Norm}_x[\mu_w, \sigma_w^2], \theta = \{\mu_0, \mu_1, \sigma_0^2, \sigma_1^2\}$
2. How to model: (a) Choose a Normal distribution for  $P(x)$ . (b) Make parameters a function of discrete binary  $w$ . (c) Function takes parameters  $\mu_0, \mu_1, \sigma_0^2, \sigma_1^2$  that define its shape.
3. Learning algorithm: Learning by standard methods, e.g. ML, MAP, Bayesian Approach.
4. Define prior  $P(w)$  and then compute  $P(w|x)$  using Bayes' rule  $P(w|x) = \frac{P(x|w)P(w)}{\int P(x|w)P(w)dw}$ .

# Chapter 4

## Regression

### 4.1 Linear Regression

### 4.2 Non-linear Regression

### 4.3 Logistic Regression

# Chapter 5

## Support Vector Machines

# Chapter 6

## EM Algorithm

# Chapter 7

## Boosting



# Chapter 8

## Decision Tree & Random Forest

# Chapter 9

## Graphical Models & Markov Network

# Bibliography

# Appendix A

## Statistical Assessment

### A.1 Hypothesis Testing

### A.2 Confidence Intervals