ECEN 765

Machine Learning with Networks

Fall 2017

TR 8:00-9:15

Office Hour: F 10:00am - noon

Aug 31 R

Lecture I Machine Learning with Networks

Sept 5 T

Useful link

PMTK

Office Hour

Friday, 10:00am - noon

Lecture 1 Machine Learning with Networks

ML $\begin{cases} \textit{Know your question} \\ \textit{Know your data} \end{cases}$

PAC learnability theory

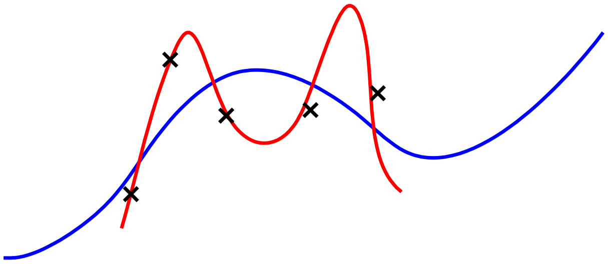
(Leslie Valiant)

Probably Approximately Correct

VC-dimension

Sparse learning: feature selection

Example



① reduce the number of parameters

② reduce the number of features

	Feature I	Feature II	…
Param I			
Param II			
⋮			

only use Feature I & II, not all features

Feature selection is different from dimension reduction

Kernel methods:

$$\langle \vec{x}_1, \vec{x}_2 \rangle \;\rightarrow\; \langle f(\vec{x}_1), f(\vec{x}_2) \rangle$$

Ensemble learning (e.g., Adaboost)

Deep learning

Bayesian learning

Difference between ML & Statistical

Inference: ML does not know the model, SI knows the model (by hypothesis) and does the hypothesis test

ML is not accurate (with uncertainty)

DL is accurate but does not know the confidence

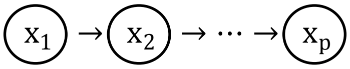
BL is accurate and also knows the confidence

Lecture 2 Probability and Statistics for Machine Learning

Union bound

$$A_i, \forall i \qquad \Pr\left(\bigcup_i A_i\right) \leq \sum_i \Pr(A_i)$$

$$\begin{aligned} \underset{2^p - 1\ params}{p(x_1, x_2, \ldots, x_p)} &= p(x_1)p(x_2, \ldots, x_p | x_1) \\ &= p(x_1)p(x_2|x_1)p(x_3, \ldots, x_p | x_1, x_2) \\ &= \prod_{t=2}^{p} p(x_t|x_{t-1})\, p(x_1) \qquad \left[\begin{matrix} \textit{if } x_t \textit{ is ONLY dependent on } x_{t-1}, \\ \textit{and independent on others} \end{matrix}\right] \end{aligned}$$



1st order Markov chain

important in probability graphical model

PGM’s

① Statistical inference / decision making

② Parameter learning

③Structure learning

Example. (Laplace's sunrise problem: What is the probability that the sun will rise tomorrow?)

$\mathcal{D}$: $\hat{y} = \hat{f}(x)$

[With the input value x of the data and the model we estimated (learned), we have inference $\hat{y} = \hat{f}(x)$]

Given x, $\Pr(\hat{y} = y) = \theta$

[For each sample's input x^i, we have an estimate $\hat{y}^i$ based on our model, comparing it with the sample's true label/value y^i, it can be accurate or inaccurate. Suppose the probability of the estimation being accurate for each sample is (the same) θ.]

$\mathcal{T} = \{x^1, x^2, \cdots x^n; y^1, y^2, \cdots y^n\}$

[The existing data of inputs and outputs.]

$\mathcal{TD} = \{n \text{ testing data points}, \text{all accurate}\}$

[The event that all previous estimates are accurate.]

$\Pr\left(\hat{y}^{n+1} = y^{n+1} \middle| \mathcal{TD}\right) = ?$

[Suppose all previous *n* testing data are accurate, what is the probability that the next estimation will be accurate?]

[If we already know the sun rose in the last *n* days, what is the probability that the sun will also rise tomorrow?]

① MLE (maximum likelihood estimation)

② MAP (maximum a posteriori estimation)

③ Bayesian learning

Machine learning only cares about the final accuracy, i.e., how many samples in the data have accurate estimate based on our model. Machine learning does not care about the randomness / probability distribution of the accuracy, i.e., it does not view the probability of being accurate as a random variable. Thus, ML can be accurate, but does not know the confidence of the accuracy. Bayesian learning take the further step of studying the randomness / probability distribution of the accuracy (i.e., the probability distribution of the probability of being accurate), and by assuming some prior distribution of the accuracy, it calculates the posterior distribution of the accuracy based on the existing data. Thus, BL can be accurate and also know the confidence of the accuracy.

As an example, suppose we have *n* testing data points, and among them *m* points are accurate. Denote this event as $\mathcal{TD}_m = \{n \text{ testing data points}, m \text{ accurate}\}$. Suppose the accuracy is $\Pr(\hat{y} = y) = \theta$ for each sample, and the prior distribution of θ is $\Pr(\theta)$. Based on the conditional probability [*n* i.i.d. events with *m* truth, Binomial distribution $m \sim B(n, \theta)$]

$$\Pr\left(\mathcal{TD}_m \middle| \theta\right) = \binom{n}{m} \theta^m (1-\theta)^{n-m}$$

the posterior distribution of the accuracy can be calculated as

$$\Pr(\theta|\mathcal{TD}_m) = \frac{\Pr(\mathcal{TD}_m, \theta)}{\Pr(\mathcal{TD}_m)}$$
$$= \frac{\Pr(\mathcal{TD}_m|\theta)\Pr(\theta)}{\int \Pr(\mathcal{TD}_m|\theta)\Pr(\theta)d\theta}$$

If we select $\theta \sim \mathrm{Beta}(a, b)$, i.e., assuming

$$\Pr(\theta) = \frac{1}{\mathrm{Beta}(a, b)}\theta^{a-1}(1-\theta)^{b-1}$$

the posterior distribution will has the same form as the prior distribution, although with different parameter $(\theta \sim \mathrm{Beta}(a + m, b + n - m)|\mathcal{TD}_m)$. Thus, Beta distribution and Binomial distribution are called conjugate priors.

Other recommended books on Bayesian statistics from internet:

1. Richard McElreath. *Statistical Rethinking*.
2. John Kruschke. *Doing Bayesian Data Analysis: A Tutorial with R, JAGS and Stan*.
3. Andrew Gelman and John B. Carlin. *Bayesian Data Analysis*.
4. Devinderjit Sivia and John Skilling. *Data Analysis: A Bayesian Tutorial*.
5. Cameron Davidson-Pilon. *Probabilistic Programming and Bayesian Methods for Hackers*.
6. Jean-Michel Marin and Christian Robert. *Bayesian Core: A Practical Approach to Computational Bayesian Statistics*.
7. William M. Bolstad and James M. Curran. *Introduction to Bayesian Statistics*.
8. E. T. Jaynes and G. Larry Bretthorst. *Probability Theory: The Logic of Science*.

Sept 7 R

---

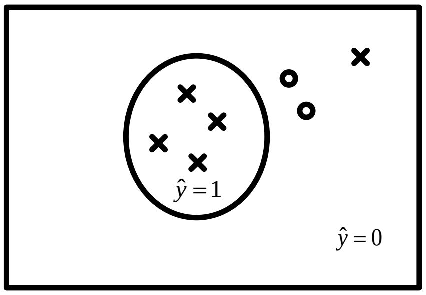
machine learning (EE)

① System Identification
② Feature Representation
③ Prediction
④ Decision Theory

} under uncertainty

↓

Bayesian Learning



M: $\hat{f}$ ← learned prediction

$\hat{y} = \hat{f}(x)$

$$\min_{\hat{f}} \Pr\left(\hat{y} = \hat{f}(x) \neq y\right) \text{ generalization error}$$

Empirical Risk Minimization

$$\min_{\hat{f}} \sum_{i=1}^{5} \mathbb{1}\left(\hat{y}^{i} \neq y^{i}\right) \leftarrow \text{training set}$$

Structural Risk Minimization

Example.

Testing data → $\mathcal{D} = \{(x^1, y^1), \dots, (x^5, y^5)\}$

parameter → $\theta = \Pr(\hat{y} \neq y)$

Question: $\hat{\theta} = ?$

Random Variable: $0 \leq \theta \leq 1$

data likelihood: $\Pr(\mathcal{D}|\theta) = \theta^5$

n testing points, *m* accurate

$$\text{Prediction} = \binom{n}{m} \theta^{m} (1-\theta)^{n-m}$$

MLE:

$$\hat{\theta} = \operatorname*{argmax}_{\theta} \Pr(\mathcal{D}|\theta)$$

$$= \operatorname*{argmax}_{\theta} \binom{n}{m} \theta^{m} (1-\theta)^{n-m}$$

m and *n* are from the test data set $\mathcal{D}$

necessary condition to get the extreme point is

$$\frac{\partial \Pr(\mathcal{D}|\theta)}{\partial \theta} = 0$$

$$\Rightarrow m(1-\theta) = (n-m)\theta$$

$$\Rightarrow m = n\theta$$

$$\Rightarrow \hat{\theta} = \frac{m}{n}$$

MAP:

$$\hat{\theta} = \operatorname*{argmax}_{\theta} \Pr(\theta|\mathcal{D})$$

prior probability: Pr⬚(θ)

if $\theta \sim \mathrm{Beta}(a, b)$,

$$\Pr(\theta) = \frac{1}{\mathrm{Beta}(a,b)} \theta^{a-1} (1-\theta)^{b-1}, \quad 0 \leq \theta \leq 1$$

$$\Pr(\theta|\mathcal{D}) = \frac{\Pr(\mathcal{D}, \theta)}{\Pr(\mathcal{D})}$$

$$= \frac{\Pr(\mathcal{D}|\theta)\Pr(\theta)}{\int \Pr(\mathcal{D}|\theta)\Pr(\theta)d\theta}$$

$$= \frac{\binom{n}{m}\theta^{m}(1-\theta)^{n-m}\cdot\frac{1}{\mathrm{Beta}(a,b)}\theta^{a-1}(1-\theta)^{b-1}}{\int_0^1\binom{n}{m}\theta^{m}(1-\theta)^{n-m}\cdot\frac{1}{\mathrm{Beta}(a,b)}\theta^{a-1}(1-\theta)^{b-1}d\theta}$$

$$= \frac{\theta^{m+a-1}(1-\theta)^{n-m+b-1}}{\int_0^1\theta^{m+a-1}(1-\theta)^{n-m+b-1}d\theta}$$

$$= \frac{1}{\mathrm{Beta}(m+a,n-m+b)}\theta^{m+a-1}(1-\theta)^{n-m+b-1}$$

$$(\theta \sim \mathrm{Beta}(m+a,n-m+b)|\mathcal{D})$$

$$\frac{\partial \Pr(\theta|\mathcal{D})}{\partial\theta} = 0$$

$$\Rightarrow \hat{\theta} = \frac{m+a-1}{n+a+b-2}$$

Decision:

$$\Pr(y^{n+1}|\mathcal{D})$$

Bayesian Learning:

$$\Pr(y^{n+1}|\mathcal{D}) = \int_0^1 \Pr(y^{n+1},\theta|\mathcal{D})d\theta$$

$$= \int_0^1 \Pr(y^{n+1}|\theta,\mathcal{D})\Pr(\theta|\mathcal{D})d\theta$$

$$= \int_0^1 \Pr(y^{n+1}|\theta)\Pr(\theta|\mathcal{D})d\theta \quad [\text{conditionally independent}]$$

$$= \int_0^1 \Pr(\hat{y}^{n+1} = y^{n+1}|\theta)\Pr(\theta|\mathcal{D})d\theta$$

$$= \int_0^1 \theta\frac{1}{\mathrm{Beta}(m+a,n-m+b)}\theta^{m+a-1}(1-\theta)^{n-m+b-1}d\theta$$

$$= \frac{\mathrm{Beta}(m+a+1,n-m+b)}{\mathrm{Beta}(m+a,n-m+b)}$$

$$= \frac{\Gamma(m+a+1)}{\Gamma(m+a)\cdot(n+a+b)}$$

$$= \frac{m+a}{n+a+b}$$

If θ ~ Uniform, then a=b=1; furthermore, if m=n (meaning all previous results are correct), then the posterior probability is $\frac{n+1}{n+2}$, different from 1! [Bayesian argument of Laplace’s sunrise problem]

For ML: posterior probability is 1 (θ=1 based on previous data)

For BL: posterior truth probability is $\frac{n+1}{n+2}$, not 1

$\mathrm{Pr}\left(\hat{y}^{\mathrm{future}} \,|\mathcal{D}\right)$: posterior predictive distribution

effective characteristic

Bayesian Decision Theory

Cost: $\mathrm{L}\left(\hat{y}, y^{\mathrm{desired}}\right) = \left(\hat{y} - y^{\mathrm{desired}}\right)^2 \rightarrow$ squared loss function

$$\min_{\hat{x}} \int_{Y} \mathrm{L}\left(\hat{y}, y^{\mathrm{desired}}\right) \mathrm{Pr}(\hat{y}|\mathcal{D}) d\hat{y}$$

Dirichlet-Multinormial Conjugate

$$\mathrm{Pr}(\mathcal{D}|\theta) = \binom{50}{10 \quad 30 \quad 10 \quad 0 \quad 0} \theta_A^{10} \theta_B^{30} \theta_C^{10} \theta_D^{0} \theta_F^{0}$$

$$\theta_A + \theta_B + \theta_C + \theta_D + \theta_F = 1, 0 \leq \theta_i \leq 1$$

$$\mathrm{Pr}(\theta) = \mathrm{Dirichlet}(\vec{a})$$

$$= \frac{1}{Z(\vec{a})} \prod_{k=1}^{K} \theta_k^{a_k - 1} \mathbb{1}\left(\sum_k \theta_k = 1\right)$$

latent Dirichlet allocation model

Gaussian-Inverse Wishart

$\hat{y} \sim N(\vec{\mu}, \Sigma)$

Sept 12 T

Bayesian Learning

Example 1.

Question: What is the prediction accuracy? [can be called coin flipping probability]

Data: Tested the algorithm on n testing points with all accurate prediction.

The mathematical question is:

model: $\theta \in [0, 1]$

① prior: $\theta \sim \mathrm{Beta}(a, b)$

a, b: hyperparameters

hierarchical Bayesian model

everything is parametric, e.g., θ is parametric by a and b

② likelihood: $p(\mathcal{D}|\theta) = \binom{n}{n} \theta^n (1-\theta)^{n-n} = \theta^n$

<u>Computation</u>:

③ posterior: $p(\theta|\mathcal{D}) \sim \mathrm{Beta}(n + a, b)$

a, b are pseudo-counts

④ Inference (prediction & decision making)

$p(\hat{y}^{\mathrm{next}} = y^{\mathrm{next}} \,|\mathcal{D})$

$$p(\theta|\mathcal{D}) = \frac{p(\mathcal{D}|\theta) \cdot p(\theta)}{\int p(\mathcal{D}|\theta) \cdot p(\theta) d\theta}$$

Bayesian statistics calculation: calculate the integral (like by MCMC)

If you are making decision based on threshold, you need to calculate the integral to get the absolute value of the posterior probability. If you only need to know the relative value to compare different posterior probability for different result and then make decision, you might not need to calculate the integral.

Frequency learning

$$p(\mathcal{D}|\theta) = p(\{(x^1, y^1), (x^2, y^2), \dots, (x^n, y^n)\}|\theta) \quad (\text{i.i.d. given } \theta)$$

$$= \prod_{i=1}^{n} p\left((x^i, y^i)\middle|\theta\right)$$

$\rightarrow$ sampling distribution

$$\begin{cases} \mathit{Emperical\ Risk\ Minimization} \rightarrow \text{FL} \\ \mathit{Structural\ Risk\ Minimization} \rightarrow \text{BL} \end{cases}$$

In FL, you are imposing structure model

Example:

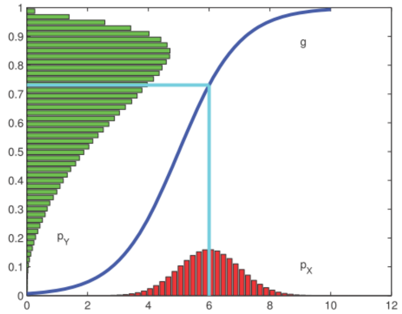


Figure 5.2 Example of the transformation of a density under a nonlinear transform. Note how the mode of the transformed distribution is not the transform of the original mode. Based on Exercise 1.4 of (Bishop 2006b). Figure generated by `bayesChangeOfVar`.

Figure from K. Murphy, *Machine Learning: A Probabilistic Perspective*

If y is has a nonlinear relationship with x. As the probability of unit area on both X and Y spaces should be the same for them to integral to unity over the whole space at the same time, that is, they should have the same unit on probability (but maybe different unit on probability density):

$$\int |p(x)dx| = \int |\tilde{p}(y)dy|$$

$$\Rightarrow |p(x)dx| = |\tilde{p}(y)dy|$$

where $p(x)$ and $\tilde{p}(y)$ are the probability density of x and y respectively. Thus,

$$\tilde{p}(y) = \left|\frac{dy}{dx}\right| \cdot p(x)$$

because the probability density function is defined to be positive semi-definite. Now if there is a nonlinear relationship between them such as

$$y = f(x) = \frac{1}{1 + e^{-x}}$$

Then the derivative in the RHS of the previous equation might influence the mode of the probability density, meaning that if the mode of $p(x)$ locates at some x^*, the mode of $\tilde{p}(y)$ might be located at some position y^* different from $f(x^*)$, which is undesired.

The fundamental reason of this problem is that Frequency learning cares about the local probability, but not taking a global view. On the contrary, Bayesian learning takes into account all points by taking integral over the parameter on the whole space. This way, the influence of the derivative is removed, which can be clearly seen from the following identity:

$$|p(x)dx| = |\tilde{p}(y)dy|$$

This is important when calculating probability, expectation, etc. for Bayesian learning.

Standard Statistical Inference

Always put things not interesting as Null Hypothesis

Confusion Matrix

	Accept H_0	Reject H_0
H_0	TN	FP
H_1	FN	TP

Positive means that our conclusion is to reject the null hypothesis.

FP is called the type 1 error, meaning that the real result is negative (H_0), but our conclusion is positive (to reject H_0).

FN is called the type 2 error, meaning that the real result is positive (H_1), but our conclusion is negative (to accept H_0).

negative events: accept H_0

positive events: reject H_0

$$\begin{cases} precision = \dfrac{|\mathrm{TP}|}{|\mathrm{TP}| + |\mathrm{FP}|} \\ recall/sensitivity = \dfrac{|\mathrm{TP}|}{|\mathrm{FN}| + |\mathrm{TP}|} \\ specificity = \dfrac{|\mathrm{TN}|}{|\mathrm{TN}| + |\mathrm{FP}|} \end{cases}$$

Precision describes the accuracy (proportion of correct results) of our conclusion among all the results that ***we conclude to be positive***. The higher the precision, the more proportion of all our positive conclusions is really positive.

Recall/sensitivity describes the accuracy (proportion of correct results) of our conclusion among all the ***real positive*** results. The higher the sensitivity, the more proportion of all the real positive results will be detected by us.

Specificity describes the accuracy (proportion of correct results) of our conclusion among all the the ***real negative*** results. The higher the specificity, the less proportion of real negative results mistakenly alarmed to be positive by us.

ROC curve

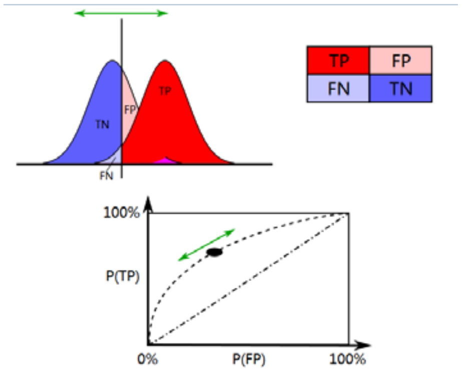


Figure from Internet

The horizontal axis is FPR (False Positive Ratio), which equals to 1-Specificity; the vertical axis is Sensitivity.

PR curve (how your accurate precision are actually true)

The horizontal axis is Precision; the vertical axis is Recall (Sensitivity).

More powerful means more sensitivity.

$$\alpha = 1 - \text{Specificity}$$

$$\text{Power} = 1 - \beta = \text{Sensitivity}$$

$$\beta = \text{P}(\text{Accept } H_0 | H_1) = \frac{|\text{FN}|}{|\text{FN}| + |\text{TP}|} = 1 - \text{Sensitivity}$$

We want low α and high $1 - \beta$.

Hypothesis Testing:

model: θ: random variable / model parameter of interest

(Bayesian) (Frequentist)

data: X: toss the coins, e.g. 10 times, 6 heads, and X is the number of heads from 10 tosses, i.e., X=6 in this case.

Rejection Region based on X:

$$\begin{matrix}\text{testing} \\ \text{statistics}\end{matrix} \begin{cases} \text{T(X)} \in \text{R: Reject } H_0 \\ \text{otherwise: Accept } H_0 \end{cases}$$

$\text{R} = \{\text{T(X)} = \text{X} > 8\}$

$$\begin{aligned} &\text{P}(\text{T(X)} > 8 | \theta = 0.5) \quad \text{Type I error} \\ &= \alpha \quad \text{significance level} \end{aligned}$$

Power of type II error:

$$\text{power} = 1 - \text{P}(\text{T(X)} \leq 8 | \theta \neq 0.5)$$

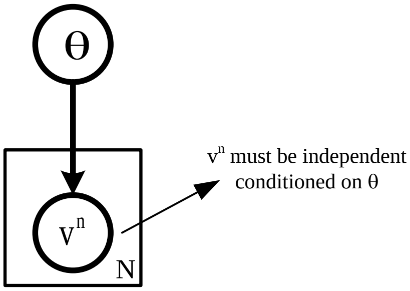
In this example, we are conducting experiment by tossing the coin, and we want to examine if the coin is fair ($\theta \overset{?}{=} 0.5$). Our testing T(X) is the number of heads from the tosses ($\text{T(X)} = \text{X}$). The null hypothesis H_0 (things not interesting) is $\theta = 0.5$. We can make our rejection region R to be $\{\mathrm{T}(\mathrm{X}) = \mathrm{X} > 8\}$, and thus Type I error (FP) will be $\{\text{Reject } H_0 | H_0\} = \{\mathrm{X} > 8 | \theta = 0.5\}$, and the probability of Type I error is α, which is also called significance level in statistics. Type II error is $\{\text{Accept } H_0 | H_1\}$, which will be $\{\mathrm{X} \leq 8 | \theta \neq 0.5\}$, and the probability of Type II error is the sensitivity.

$$1 - \alpha = \text{specificity}$$
$$1 - \beta = \text{power} = \text{sensitivity}$$

subjective & difficult to do the calculation & Hypothesis Test

Lecture 3 From Bayesian Statistics, to Learning, and to Model Selection

Plate Representation of PGMs



Naïve Bayes Classification (NBC)

Example. (Frequency Learning)

Question: predict C (label outcome) given feature (X)

Data: $\vec{\mathrm{X}} = (\mathrm{x}_1, \mathrm{x}_2, \ldots, \mathrm{x}_5)$; $\mathrm{C} = \{\mathrm{S}, \mathrm{E}\}$

model:

$$\mathrm{P}(\vec{\mathrm{X}}, \mathrm{C}) = \mathrm{p}(\vec{\mathrm{X}} | \mathrm{C}) \mathrm{p}(\mathrm{C})$$
$$= \prod_{i=1}^{5} \mathrm{p}(\mathrm{X}_i | \mathrm{C})\, \mathrm{p}(\mathrm{C})$$

where $\mathrm{p}(\mathrm{C})$ is percentage of people in the whole country, and X_i are conditionally independent. As all features are binary character, $\mathrm{p}(\mathrm{C})$ need only one parameter θ, and $\mathrm{p}(\vec{\mathrm{X}} | \mathrm{C})$ need $(2^5 - 1) \times 2$ parameters. But if we employ Naïve Bayes method, since $\mathrm{p}(\mathrm{X}_i | \mathrm{C})$ only needs 2 parameters ($\mathrm{p}(\mathrm{X}_i = 1 | \mathrm{C} = 1)$ and $\mathrm{p}(\mathrm{X}_i = 1 | \mathrm{C} = 0)$), the total number of parameters will be $2 \times 5 + 1 = 11$.

Now instead of one θ, we have 11 θ's, and we will assume they are independent. Section 10.2 have Bayesian learning example.

Sept 14 R

Naïve Bayes Classifier

Question: $\Pr(c^* = 1|\vec{x}^*; \mathcal{D})$

Data: $\mathcal{D}$=$\{(\vec{x}^1, c^1), (\vec{x}^2, c^2), \ldots, (\vec{x}^n, c^n)\}$

model: $\theta = \{P(c = 1); P(\vec{x}|c = 1); P(\vec{x}|c = 0)\}$

Assumption:

$$p(\vec{x}|c = 1/0) = \prod_{i=1}^{D} p\left(x^i\middle|c = 1/0\right) \qquad \text{totally D different features}$$

$$p\left(x^i\middle|c = 1/0\right) = \theta_{x^i|c}$$

① $\begin{cases} p\left(x^i = 1\middle|c = 0\right) \\ p\left(x^i = 1\middle|c = 1\right) \end{cases}$

② $x^i|c = 1 \sim \mathcal{N}\left(\mu_1^i, {\sigma_1^i}^2\right)$

e.g. prior: $\mu_1^i \sim \mathcal{N}\left(\mu^0, {\sigma^0}^2\right)$, ${\sigma_1^i}^2 \sim \mathrm{InvGamma}(\alpha, \nu)$

Standard NBC: $p(c^*|x^*;\ \theta^*)$

Frequency NBC: point estimate

calculate the probability

Bayesian NBC (Bayesian Decision Theory):

nothing to do with the point estimate of θ, looking at the whole distribution of θ

Loss function:

$$\ell(\hat{c}, c) = \begin{cases} 1;\ \hat{c} \neq c \\ 0;\ \hat{c} = c \end{cases}$$

(zero-one loss)

mean-square loss: $\ell(\hat{y}, y) = (\hat{y} - y)^2$

Minimize expected loss:

$$L(\hat{c}) = E_{c|\mathcal{D}}\left[\ell(\hat{c}, c)\right] = \int \ell(\hat{c}, c)p(c|\mathcal{D})dc$$

$$\hat{c} = \hat{c}(\vec{x}) \Rightarrow L(\hat{c}) = \int \ell(\hat{c}(\vec{x}), c)p(c|\vec{x}; \mathcal{D})dc$$

we need to get $\min L(\hat{c})$, or

$$L(\hat{c}) = \sum_{c\in\{0,1\}} \ell(\hat{c}, c)\, p(c|\vec{x}; \mathcal{D})$$

$$= \begin{cases} p(c = 1|\vec{x}; \mathcal{D}), & \hat{c} = 0 \\ p(c = 0|\vec{x}; \mathcal{D}), & \hat{c} = 1 \end{cases} \qquad \rightarrow \text{Bayes error}$$

To get

$$\min_{\hat{c}} L(\hat{c})$$

we have:

if $p(c = 1|\vec{x}; \mathcal{D}) < p(c = 0|\vec{x}; \mathcal{D})$, $\hat{c} = 0$

if $p(c = 1|\vec{x}; \mathcal{D}) > p(c = 0|\vec{x}; \mathcal{D})$, $\hat{c} = 1$

$$
\begin{aligned}
& p(c^* = 1|\vec{x}^*; \mathcal{D}) \qquad \leftarrow \text{predictive posterior} \\
= & \int_\Theta p(c^* = 1, \theta|\vec{x}^*; \mathcal{D}) d\theta \\
= & \int_\Theta p(c^* = 1|\theta;\ \vec{x}^*; \mathcal{D}) p(\theta|\vec{x}^*; \mathcal{D}) d\theta \\
= & \int_\Theta p(c^* = 1|\vec{x}^*; \theta) p(\theta|\mathcal{D}) d\theta
\end{aligned}
$$

$p(\theta|\mathcal{D})$ is posterior

1) θ does not depend on new coming input data, so $p(\theta|\vec{x}^*; \mathcal{D}) = p(\theta|\mathcal{D})$;

2) new label c^* does not depend on original data $\mathcal{D}$ because all data are independent, so $p(c^* = 1|\theta;\ \vec{x}^*; \mathcal{D}) = p(c^* = 1|\vec{x}^*; \theta)$.

[Thanks to Ruiting Chang for explanation of these two identities.]

$$
= \int_\Theta \frac{p(\vec{x}^*|c^* = 1;\ \theta) p(c^* = 1|\theta)}{p(\vec{x}^*|c^* = 1;\ \theta) p(c^* = 1|\theta) + p(\vec{x}^*|c^* = 0;\ \theta) p(c^* = 0|\theta)} \cdot p(\theta|\mathcal{D}) d\theta
$$

Section 10.2:

$$p(c^* = 1|\vec{x}^*; \mathcal{D}) \propto p(\vec{x}^*|c^* = 1; \mathcal{D}) p(c^* = 1|\mathcal{D})$$

$$p(c^* = 1|\mathcal{D}) = \int_{\theta_{c^*=1}} p(c^* = 1|\theta_{c^*=1}) p(\theta_{c^*=1}|\mathcal{D})\, d\theta_{c^*=1}$$

Optimal Bayes Classification (Lori Dalton)

Bayesian Model Selection/Averaging

Don't want to bias to any model, so typically all models have the same probability ($p(M_i) = p(M_j)$)

Bayesian Computation

① Inference: $p(c^*|\theta)$

② Decision Making: $E[\ell(\hat{c}, c)]$

③ Model Selection: $p(M|\mathcal{D})$ or $p(\mathcal{D}|M)$

$$\text{Bayes factor} = \frac{p(\mathcal{D}|M_i)}{p(\mathcal{D}|M_j)}$$

typically larger than some value, like 10, to prefer one of them (if only larger than 1, it would be not safe)

$$
\begin{aligned}
p(\mathcal{D}|M_i) &= \int_{\Theta_i} p(\mathcal{D}|\theta_i) p(\theta_i|M_i) d\theta_i \\
&= \int_{\Theta_i} e^{-f(\theta_i)}\, d\theta_i
\end{aligned}
$$

$$\approx \int e^{-\theta_i{}^T A_i \theta_i + b_i{}^T \theta_i}\, d\theta_i$$

Laplace's method

find the mode and use quadratic form near the mode

$$f(\theta) = f(\theta^m) + \nabla f(\theta^m) \cdot (\theta - \theta^m) + \frac{1}{2}(\theta - \theta^m)^T \nabla^2 f(\theta^m)(\theta - \theta^m) + O(|\theta - \theta^m|^2)$$

θ^m is the mode, so $\nabla f(\theta^m) = \vec{0}$

$$f(\theta) \approx f(\theta^m) + \frac{1}{2}(\theta - \theta^m)^T H(\theta - \theta^m)$$

$$p(\mathcal{D}|M_i) \approx \int_{\Theta_i} e^{-\left[f(\theta^m) + \frac{1}{2}(\theta - \theta^m)^T H(\theta - \theta^m)\right]} d\,\theta = e^{-f(\theta^m)} \det(2\pi H^{-1})^{\frac{1}{2}}$$

$$\log p(\mathcal{D}|M_i) \approx \log p\left(\mathcal{D}\middle|\theta^m, M_i\right) + \log p\left(\theta^m\middle|M_i\right) + \frac{K}{2}\log 2\pi - \frac{K}{2}\log N$$

Sept 19 T

---

① prior $p(\theta)$

② likelihood

⇒③ posterior $p(\theta|\mathcal{D})$ → MLE, MAP → $\hat{\theta}$

model is important

If Bayesian Learning:

④ $p(\hat{y}|\mathcal{D})$ predictive posterior → model is not important

⑤ Bayesian Decision Theory

cost/risk:

loss function: $\ell(\tilde{y}, \hat{y})$

y is true value, $\hat{y}$ is estimate

$$\min_{\tilde{y}}\left\{E\left[\ell(\tilde{y}, \hat{y})\right]\right\}$$

$$= \int_{\hat{Y}} \ell(\tilde{y}, \hat{y}) p(\hat{y}|\mathcal{D}) d\hat{y}$$

utility: $\mathcal{U}(\tilde{y}, \hat{y})$

$$\max_{\tilde{y}} E[\mathcal{U}(\tilde{y}, \hat{y})]$$

Bayesian Model Selection/Averaging

Bayesian factor: $\frac{p(\mathcal{D}|M_1)}{p(\mathcal{D}|M_2)}$

$$p(\hat{y}|\mathcal{D}) = \sum_{i=1}^{m} p(\hat{y}|M_i; \mathcal{D}) p(M_i|\mathcal{D})$$

$$p(M_i|\mathcal{D}) = \frac{p(\mathcal{D}|M_i)p(M_i)}{p(\mathcal{D})}$$

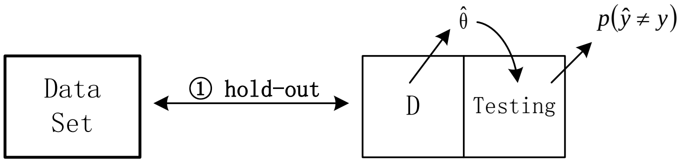
As data has already been observed, model is not independent on data, so cannot assume $p(M_i|\mathcal{D}) = p(M_i)$.

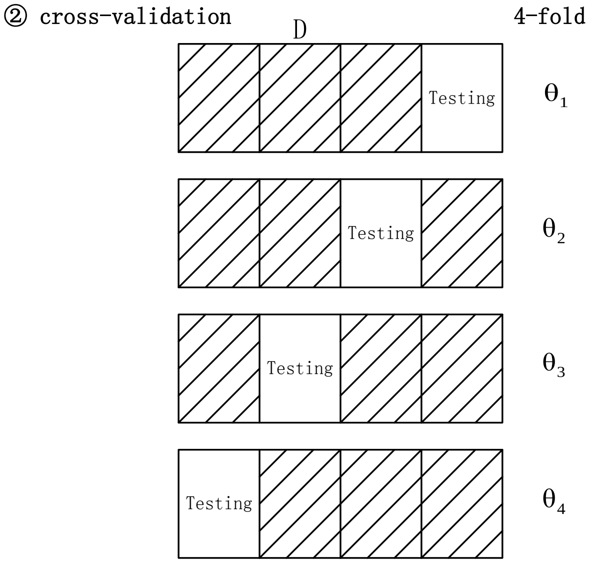
$$\begin{cases} Random\ Forest \\ Weighted\ kNN \end{cases}$$

① bootstrap ← cross-validation

② bagging $\leftarrow \begin{cases} \text{Random Forest} \\ Bootstrap\ Aggregation \end{cases}$

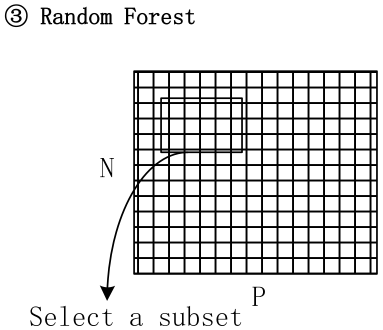
③ boosting





$$\hat{\theta} = \frac{1}{4}\sum_{i=1}^{4}\theta_i$$

Model is the same for the four



Select a subset

Model is different because features selected are different kNN: k=1 most complicated (most number of classes)

k=∞ simplest (all things are the same)

To compute $p\left(o_a, o_b \middle| H_{indep}\right)$:

$$\begin{aligned}
p\left(o_a, o_b \middle| H_{indep}\right) &= \int_{A,B} p\left(o_a, o_b, \alpha, \beta \middle| H_{indep}\right) d\alpha d\beta \\
&= \int_{A,B} p\left(o_a, o_b \middle| \alpha, \beta, H_{indep}\right) p\left(\alpha, \beta \middle| H_{indep}\right) d\alpha d\beta \\
&= \int_{A,B} p\left(o_a, o_b \middle| \alpha, \beta, H_{indep}\right) p\left(\alpha \middle| H_{indep}\right) p\left(\beta \middle| H_{indep}\right) d\alpha d\beta \\
&\qquad\qquad\qquad \leftarrow \alpha, \beta \text{ are independent} \\
&= \int_{A,B} p\left(o_a \middle| \alpha, H_{indep}\right) p\left(o_b \middle| \beta, H_{indep}\right) p\left(\alpha \middle| H_{indep}\right) p\left(\beta \middle| H_{indep}\right) d\alpha d\beta
\end{aligned}$$

$p\left(o_a \middle| \alpha, H_{indep}\right)$ & $p\left(o_b \middle| \beta, H_{indep}\right)$ are likelihood

$p\left(\alpha \middle| H_{indep}\right)$ & $p\left(\beta \middle| H_{indep}\right)$ are prior probability

Example 2.

two machine learning algorithms (Deep Learning vs SVM)

N fold cross-validation

$$\left.\begin{aligned}
\hat{\theta}_{DL} &= \frac{1}{N}\sum_{i=1}^{N} \theta_{DL}^{i} \\
\hat{\theta}_{SVM} &= \frac{1}{N}\sum_{i=1}^{N} \theta_{SVM}^{i}
\end{aligned}\right\} \begin{array}{c}\text{observed}\\ \text{accuracy}\\ \text{(empirical)}\end{array}$$

DL: θ_{DL}, SVM: θ_{SVM} → true prediction accuracy

H_0: $\theta_{DL} - \theta_{SVM} \leq 0$

H_1: $\theta_{DL} - \theta_{SVM} > 0$

want to reject the null ⇒ DL model is better than SVM

$T\left(\theta_{DL}^{i}, \theta_{SVM}^{i}\right)$ (t test)

$= \hat{\theta}_{DL} - \hat{\theta}_{SVM} > c$

$T(X) > c$ is the rejection region

$p(T(X)|\theta_{DL}, \theta_{SVM})$

testing data:

$$(x^1, y^1) \xrightarrow{\text{DL algorithm } (\theta_{DL})} \theta_{DL}{}^{\mathbb{1}(y^1=\hat{y}^1)} \cdot (1-\theta_{DL})^{\mathbb{1}(y^1\neq\hat{y}^1)}$$

$\vdots$

$$(x^n, y^n) \xrightarrow{\text{DL algorithm } (\theta_{DL})} \theta_{DL}{}^{\mathbb{1}(y^n=\hat{y}^n)} \cdot (1-\theta_{DL})^{\mathbb{1}(y^n\neq\hat{y}^n)}$$

$\theta_{DL}^{i} \sim \binom{n}{m} \theta_{DL}{}^{m} (1-\theta_{DL})^{n-m}$

m is the number of good prediction, and n-m is the number of bad prediction

$\bar{\theta}_{\mathrm{DL}} \sim \mathcal{N}(\theta_{\mathrm{DL}}, \sigma^2)$ averaging over a large number of testing

$\hat{\theta}_{\mathrm{DL}} \sim \mathcal{N}(\theta_{\mathrm{DL}}, \sigma^2)$, $\hat{\theta}_{\mathrm{SVM}} \sim \mathcal{N}(\theta_{\mathrm{SVM}}, \sigma^2)$

(assume they have the same σ; not true in reality)

$\mu = \theta_{\mathrm{DL}} - \theta_{\mathrm{SVM}} < 0$ (null hypothesis)

model parameter

$\mathrm{T(X)} = \hat{\theta}_{\mathrm{DL}} - \hat{\theta}_{\mathrm{SVM}} \rightarrow$ testing statistics

$\beta = \Pr(\mathrm{T(X)} > c | \theta_{\mathrm{DL}} - \theta_{\mathrm{SVM}} > 0)$

$\mathrm{T(X)} = \hat{\theta}_{\mathrm{DL}} - \hat{\theta}_{\mathrm{SVM}} \sim \mathcal{N}(\mu, 2\sigma^2)$

$$\mathrm{T(X)} > c \Leftrightarrow \frac{\mathrm{T(X)} - \mu}{\sqrt{2}\sigma} > \frac{c - \mu}{\sqrt{2}\sigma}$$

$$\beta = 1 - \Phi\left(\frac{c-\mu}{\sqrt{2}\sigma}\right)$$

size of test $\mathrm{T(X)} > c$

$$\alpha = \max_{\mu \le 0}\left(1 - \Phi\left(\frac{c-\mu}{\sqrt{2}\sigma}\right)\right)$$

$$= 1 - \Phi\left(\frac{c}{\sqrt{2}\sigma}\right)$$

α is the size (significant level, confidence) of the test

$\mu \le 0$ is the null hypothesis H_0

$\alpha \Rightarrow \mathrm{c} = \sqrt{2}\sigma\Phi^{-1}(1-\alpha)$

$$\hat{\theta}_{\mathrm{DL}} = \frac{1}{\mathrm{N}}\sum_{\mathrm{i}=1}^{\mathrm{N}} \theta_{\mathrm{DL}}^{\mathrm{i}}, \theta_{\mathrm{DL}}^{\mathrm{i}} \sim \mathcal{N}(\theta_{\mathrm{DL}}, \sigma^2)$$

$$\hat{\theta}_{\mathrm{DL}} \sim \mathcal{N}\left(\theta_{\mathrm{DL}}, \frac{\sigma^2}{\mathrm{N}}\right)$$

$$\Rightarrow \mathrm{c} = \frac{1}{\sqrt{\mathrm{N}}}\sqrt{2}\sigma\Phi^{-1}(1-\alpha)$$

test more to get smaller threshold to reject (more confidence to reject due to smaller α with the same c)

Bayesian Hypothesis Test: $\mathrm{p}(\mu > 0 | \mathcal{D})$

Sept 21 R

Multiple Hypothesis Testing

α = Probability of FP (false positive)

M (4000 genes) tests

Probability of at least one false positive $\le 1 - (1-\alpha)^{\mathrm{M}} \le \alpha\mathrm{M}$

compound rate

Bonferonni Correction

instead of using α, use $\frac{\alpha}{M}$ in the beginning

related to overfitting

if the desired p-value is p_0, then the actual p-value used for the testing have threshold $\frac{p_0}{M}$

if the number of model parameter change with the number of data, it is non-parameteric learning

semi-parametric model

Bayes error is the minimum that can be achieved

$$e_{Bayes} \leq e_{nn} \leq 2e_{Bayes} + c$$

c is related to c-Lipshitz function

(For nearest-neighbor algorithm)

For kNN: $e_{Bayes} \leq e_{nn} \leq \left(1 + \sqrt{\frac{8}{k}}\right) e_{Bayes} + c$

Basic concepts:

c-Lipshitz function: $|f(x) - f(x')| \leq c|x - x'|$

$d(x, x') = \langle x - x^{'}, x - x' \rangle \leftarrow$ inner product

$d^{'(x,x^{'})} = \langle \psi(x) - \psi(x^{'}), \psi(x) - \psi(x^{'}) \rangle$

$\quad = K(x, x') \leftarrow$ Kernel function

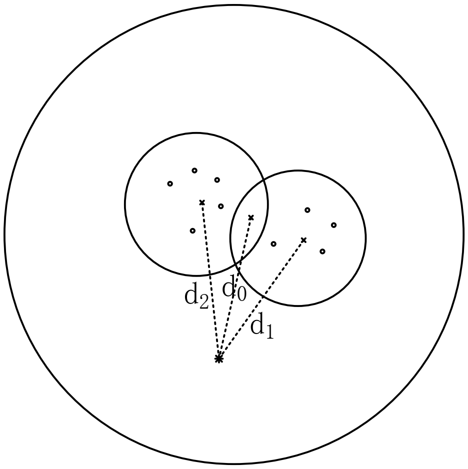
$d = (x - x')^T \Sigma^{-1} (x - x')$

Σ is symmetric, so $\Sigma = B^T B$

$d = (Bx - Bx')^T (Bx - Bx')$

using B, we can remove noise

Procrustes distance



if $d_0 > d_1$ & $d_0 > d_2$, does not need to compute lower level distance (like tree)

missing values:

① imputation

② matrix completion ← compressive sensing

ADMM

model selection: hold the testing data, and do cross-validation on the training data and select the optimal k in the projection, need to explain the testing error.

$$(*) \quad e_{\text{Bayes}} \le e_{\text{nn}} \le 2e_{\text{Bayes}} + c \text{ (Binary classification)}$$

Proof:

$$\begin{aligned} e_{\text{nn}} &= E_D[y(x) \neq y_{\text{nn}}(x)|x] \\ &= E_D[\Pr(y(x) \neq y(x_{\text{nn}})|x)] \end{aligned}$$

Let $\eta(x) = \Pr(y(x) = 1|x)$ is c-Lipshitz.

x can be only 0 or 1

$$\begin{aligned} \Pr(y(x) \neq y(x_{\text{nn}})|x) &= \eta(x)\big(1-\eta(x_{\text{nn}})\big) + \eta(x_{\text{nn}})\big(1-\eta(x)\big) \\ & \eta(x): y(x) = 1, \eta(x_{\text{nn}}): y(x_{\text{nn}}) = 1 \\ &= \eta(x) + \eta(x_{\text{nn}}) - 2\eta(x)\eta(x_{\text{nn}}) \\ &= 2\eta(x)\big(1-\eta(x)\big) + \big(\eta(x) - \eta(x_{\text{nn}})\big)(2\eta(x) - 1) \end{aligned}$$

Let $\eta = \eta(x), \eta^{'} = \eta(x_{\text{nn}})$,

$$\begin{aligned} & E_D[y(x) \neq y_{\text{nn}}(x)|x] \\ &= E_D[2\eta(1-\eta) + (\eta - \eta^{'})(2\eta - 1)] \\ &= E_D[2\eta(1-\eta)] + E_D[(\eta - \eta^{'})(2\eta - 1)] \\ &\le E_D[2\eta(1-\eta)] + 2E_D[\eta - \eta^{'}] \\ &\le E_D[2\eta(1-\eta)] + 2cE_D[|x - x_{\text{nn}}|] \\ &\equiv E_D[2\eta(1-\eta)] + 2c' \end{aligned}$$

$\Rightarrow (*) \Leftrightarrow E_D[\eta(1-\eta)] \le e_{\text{Bayes}} = E_D[\min(\eta, 1-\eta)]$

$E_D[\eta(1-\eta)] = E_D[\max * \min] \le E_D[\min]$

The left side is trivial, because

$e_{\text{nn}} = E_D[\eta(1-\eta') + \eta'(1-\eta)] \ge E_D[\eta * \min + \min * (1-\eta)] = e_{\text{Bayes}}$

Q.E.D.

$$\psi(x)\begin{cases} PCA \\ manifold\ embedding \\ kernel\ space \end{cases} \rightarrow \text{like spiral data model (use Laplacian embedding)}$$

Sept 26 T