Pattern Recognition and Machine Learning!

Meichen Lu (ml574@cam.ac.uk)

April 19, 2018

Contents

1 Overview

Tom M. Mitchell provided a widely quoted, more formal definition: "A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P if its performance at tasks in T, as measured by P, improves with experience E."

1.1 Different classes of learning

1.1.1 Supervised learning

- Classification
- Regression

1.1.2 Unsupervised learning

- Clustering
- Factor analysis, PCA, ICA

1.1.3 Reinforcement learning

1.2 Pipeline of machine learning

1.2.1 Preprocessing

- Dimension reduction

1.2.2 Feature extraction

1.2.3 Training/learning

1.2.4 Validation

- S-fold cross validation: when $S = N$ it is called leave-one-out
- Information criteria

1.3 Red flags

- Overfitting
- Curse of dimensionality

2 Preliminaries

2.1 Probability Theory

Frequentist/Classical Probabilities as the frequencies of random, repeatable events

Bayesian Probabilities as quantification of uncertainty.

2.1.1 Rules of probability

Sum rule

$$p(X) = \sum_{Y} p(X, Y) \qquad / \qquad p(x) = \int p(x, y) dy$$

$p(X)$ is also called marginal probability

Product rule

$$p(X, Y) = p(Y|X)p(X) \qquad / \qquad p(x, y) = p(y|x)p(x)$$

Bayes' theorem Rewritten of product rule

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

where $\mathcal{D}$ represent data and $\boldsymbol{w}$ represent model.
$p(\mathcal{D}|\boldsymbol{w})$ is called the likelihood function.

$$\text{posterior} \propto \text{likelihood} \times \text{prior}$$

2.1.2 Binomial distribution

Bernoulli distribution

$$\text{Bern}(x|\mu) = \mu^x(1-\mu)^{1-x}$$

$$\mathbb{E}[x] = \mu \qquad \text{var}[x] = \mu(1-\mu)$$

Maximum likelihood from sample is $\mu_{ML} = \frac{1}{N}\sum_{n=1}^{N} x_n$

Binomial distribution

$$\text{Bin}(m|N,\mu) = \binom{N}{m}\mu^m(1-\mu)^{N-m}$$

Maximum likelihood from sample is $\mu_{ML} = \frac{m}{N}$

Beta distribution

$$\text{Beta}(\mu|a,b) = \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)}\mu^{a-1}(1-\mu)^{b-1}$$

2.1.3 Polynomial distribution

$\boldsymbol{x}$ is a vector for K categories, e.g. $K = 6$ and $x_3 = 1$, $\boldsymbol{x} = (0,0,1,0,0,0)^T$

$$p(\boldsymbol{x}|\boldsymbol{\mu}) = \prod_{k=1}^{K}\mu_k^{x_k}$$

$$\mathbb{E}[\boldsymbol{x}|\boldsymbol{\mu}] = (\mu_1,...,\mu_M)^T = \boldsymbol{\mu}$$

Maximum likelihood from sample is $\mu_k^{ML} = \frac{m_k}{N}$

Multinomial distribution

$$\text{Mult}(m_1,m_2,...,m_K|N,\boldsymbol{\mu}) = \binom{N}{m_1 m_2 ... m_K}\prod_{k=1}^{K}\mu_k^{m_k}$$

2.1.4 Gaussian distribution

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

For a D-dimensional vector $\boldsymbol{x}$ of continuous variables

$$\mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu},\boldsymbol{\Sigma}) = \frac{1}{(\sqrt{2\pi})^D}\frac{1}{|\boldsymbol{\Sigma}|^{1/2}}\exp\left\{\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right\}$$

Maximum likelihood solution for a N *independent and identically distributed* (i.i.d.) numbers from a normal distribution

$$\mu_{ML} = \frac{1}{N}\sum_{n=1}^{N} x_n, \quad \sigma^2_{ML} = \frac{1}{N}\sum_{n=1}^{N}(x_n - \mu_{ML})^2$$

However, the unbaised estimator for the variance is (see why divide by $N-1$)

$$\tilde{\sigma} = \frac{1}{N-1}\sum_{n=1}^{N}(x_n - \mu_{ML})^2$$

2.2 Decision theory

Example: classification: posterior is $p(\mathcal{C}_k|x)$
Basic scenario: To minimise p(mistake)/maximise the p(correct) $\equiv$ maximise posterior probability.
Different cost for different types of mistake: use a loss matrix L_{kj} and minimise

$$\sum_k L_{kj} p(\mathcal{C}_k|x)$$

Rejection option to avoid making decisions on the difficult cases

Inference and decision Three types of approaches with decreasing order of complexity

1. Generative model
   - Solve the inference problem for $p(x|\mathcal{C}_k)$
   - Infer the prior class probabilities $p(\mathcal{C}_k)$
   - We can model joint probability $p(x, \mathcal{C}_k)$ directly
   - It is called generative because we can generate synthetic data points in the input space.
2. Discriminative model: only solve for $p(x|\mathcal{C}_k)$
3. Opaque model without knowing the probabilities

Benefits of knowing the posterior probability

- Minimizing risk (wrt loss matrix)
- Rejection option
- Compensating for class priors: e.g. when different classes are disproportionate, can skew the original distribution.
- Combining models

2.3 Information theory

Entropy of a random variable x

$$H[x] = -\sum_x p(x)\log_2 p(x)$$

or

$$H[x] = -\int p(x)\ln p(x)dx$$

- The noiseless coding theorem (Shannon, 1948) states that the entropy is a lower bound on the number of bits needed to transmit the state of a random variable.
- The continuous expression is called the differential entropy (in 'nats' instead of bits)
- Given the mean of a distribution is μ and the variance is σ^2, the distribution that maximizes the differential entropy is the Gaussian !!! The differential entropy is $H[x] = \frac{1}{2}\{1+\ln(2\pi\sigma^2)\}$. This entropy can be negative, i.e. for $\sigma^2 < 1/(2\pi e)$. See also *central limit theorem*.

Conditional entropy $H[y|x]$ satiefies $H[x,y] = H[y|x] + H[x]$

Relative entropy and mutual information

$$\begin{aligned}\mathrm{KL}(p||q) &= -\int p(x)\ln q(x)dx - \left(-\int p(x)\ln p(x)dx\right) \\ &= -\int p(x)\ln\left\{\frac{q(x)}{p(x)}\right\}dx\end{aligned} \tag{1}$$

- relative entropy or Kullback-Leibler divergence, or KL divergence
- Not symmetrical $\mathrm{KL}(p||q) \neq \mathrm{KL}(p||q)$

Mutual information

$$\begin{aligned}I[x,y] \equiv & \quad \mathrm{KL}(p(x,y)||p(x)p(y)) \\ = & \quad -\iint \ln\left(\frac{p(x)p(y)}{p(x,y)}\right)dxdy\end{aligned}$$

We can view the mutual information as the reduction in the uncertainty about x by virtue of being told the value of y $I[x,y] = H[x] - H[x|y] = H[y] - H[y|x]$

3 Advanced Probability

3.1 Conjugate prior

Beta function is the conjugate prior of binomial distribution. It describes the distribution of μ based on the hyperpamaters a, b. The posterior distribution given m 'heads' and l 'tails' is

$$p(\mu|m,l,a,b) = \mathrm{Beta}(\mu|a,b)p(\mu|m,l) = \mathrm{Beta}(\mu|a,b)\mathrm{Bin}(m|\mu,l) = \frac{\Gamma(m+a+l+b)}{\Gamma(m+a)\Gamma(l+b)}\mu^{m+a-1}(1-\mu)^{l+b-1}$$

- We can interpret a and b as effective number of observations of $x = 1$ and $x = 0$
- For a finite dataset, μ lies between the prior mean and the MLE of binomial distribution

Dirichlet distribution The conjugate prior for the parameters μ_k of polynomial distribution

$$p(\boldsymbol{\mu}|\boldsymbol{\alpha}) \propto \prod_{k=1}^{K} \mu_k^{\alpha_k - 1}$$

where the normalised form is

$$p(\boldsymbol{\mu}|\boldsymbol{\alpha}) = \frac{\Gamma(\alpha_0)}{\Gamma(\alpha_1) \cdots \Gamma(\alpha_K)} \prod_{k=1}^{K} \mu_k^{\alpha_k - 1}$$

where $\alpha_0 = \sum_{k=1}^{K} \alpha_k$

4 Density estimation

Density estimation walks the line between unsupervised learning, feature engineering, and data modeling.

Density estimation: general framework For a small region $\mathcal{R}$, assume that the probability density is constant $p(x)$. The probability mass associated with the region is

$$P = \int_{\mathcal{R}} p(x) dx$$

If we collected a data set containing N observations drawn from $p(x)$, the probability follows the binomial distribution and the mean of the number of points that lie inside $\mathcal{R}$ is $\mathbb{E}[K] = NP$. The probability density is

$$p(x) \approx \frac{P}{V} = \frac{K}{NV}$$

- If we fix K and determine V from the data, we have the K-nearest-neighbours (§4.2) density estimator.
- If we fix V and determine K from the data, we have the Kernel density estimators (KDE).

4.1 Kernel density estimators

kernel density estimation is a non-parametric way to estimate the probability density function of a random variable. Kernel density estimation is a fundamental data smoothing problem where inferences about the population are made, based on a finite data sample.

Top-hat kernel function We can choose any kernel function subject to the conditions

$$k(\boldsymbol{u}) \geq 0, \tag{2}$$

$$\int k(\boldsymbol{u}) d\boldsymbol{u} = 1 \tag{3}$$

A tophat function describes is a hypercube with sides 1.

$$k(\boldsymbol{u}) = \begin{cases} 1, & |u_i| \leq 1/2, \qquad i = 1, ..., D, \\ 0, & \text{otherwise} \end{cases}$$

The total number of data lying inside the hypercube is

$$K = \sum_{n=1}^{N} k\left(\frac{\boldsymbol{x} - \boldsymbol{x}_n}{h}\right)$$

Thus the estimated density at $\boldsymbol{x}$ is thus

$$p(\boldsymbol{x}) = \frac{K}{NV} = \frac{1}{N}\frac{1}{h^D}\sum_{n=1}^{N} k\left(\frac{\boldsymbol{x} - \boldsymbol{x}_n}{h}\right)$$

Gaussian kernel function

$$p(\boldsymbol{x}) = \frac{1}{N}\sum_{n=1}^{N} \frac{1}{(2\pi h^2)^{1/2}} \exp\left\{-\frac{||\boldsymbol{x} - \boldsymbol{x}_n||^2}{2h^2}\right\}$$

- Our density model is obtained by placing a Gaussian over each data point and then adding up the contributions over the whole data set
- h plays the role of a smoothing parameter.

4.2 K-nearest-neighbour

4.2.1 K-nearest-neighbour density estimation

The parameter h could be dependent on location within the data space the strategy is to fix K and search for K neighbours within a radius R, thereby the volume V can be estimated.

4.2.2 K-nearest-neighbour classifier

Likelihood function:

$$p(\boldsymbol{x}|\mathcal{C}) = \frac{K_k}{N_k V}$$

Unconditional density is (uniform)

$$p(\boldsymbol{x}) = \frac{K}{NV}$$

Class prior

$$p(\mathcal{C}_k) = \frac{N_k}{N}$$

Posterior probability:

$$p(\mathcal{C}_k|\boldsymbol{x}) = \frac{p(\boldsymbol{x}|\mathcal{C})p(\mathcal{C}_k)}{p(\boldsymbol{x})} = \frac{K_k}{K}$$

5 Unsupervised learning

5.1 Mixture of Gaussians and Expectation maximization

5.1.1 Mixture of Gaussians

Gaussian mixture model (GMM) as a simple linear superposition of Gaussian components. We now turn to a formulation of Gaussian mixtures in terms of discrete **latent variables**.
Latent variables, as opposed to 'observed variables', are variables that are not directly observed but are rather inferred (through a mathematical model) from other variables that are observed (directly measured). When it is corresponding to physical variable, is also called a **hidden variable** but it can also refer to an abstract concept.

Components

- There are K states, i.e., K Gaussian populations
- Latent variable $\boldsymbol{z}$ is a K-dimensional binary vector with a particular element z_k equal to 1, which means that the k^{th} state is on.
- Observed variable $\boldsymbol{x}$
- GMM likelihood function (a linear superposition of Gaussians)

$$p(\boldsymbol{x}) = \sum_{z} p(\boldsymbol{z}) p(\boldsymbol{x}|\boldsymbol{z}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \tag{4}$$

  - Possibility that the k^{th} state is on: π_k
  - Gaussian parameters of the k^{th} state/population: $\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k$

5.1.2 GMM for the i.i.d. data set

Given N observations:

- matrix $\mathbf{X} \in \mathbb{R}^{N \times m}$ in which the n^{th} row is given by $x^{(n)^T}$
- matrix $\mathbf{Z} \in \mathbb{R}^{N \times k}$ in which the n^{th} row is given by $z^{(n)^T}$
- **i.i.d.** means that the data points are drawn **independently** from the distribution.

Log-likelihood function for the data set

$$ln(p(\mathbf{X}|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{n=1}^{N} \ln \left\{ \sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{x}^{(n)}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right\} \tag{5}$$

Derivative of log-likelihood function w.r.t. $\boldsymbol{\mu}_k$

$$-\sum_{n=1}^{N} \frac{\pi_k \mathcal{N}(\boldsymbol{x}^{(n)}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_j \pi_j \mathcal{N}(\boldsymbol{x}^{(n)}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)} \boldsymbol{\Sigma}(\boldsymbol{x}^{(n)} - \boldsymbol{\sigma}_k) = 0 \tag{6}$$

The direct gradient descent algorithm is too complicated, and amazing mathematicians have invented expectation maximization!

5.1.3 EM for GMM

In Eq. 6, we see that $\frac{\pi_k \mathcal{N}(\boldsymbol{x}^{(n)}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_j \pi_j \mathcal{N}(\boldsymbol{x}^{(n)}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}$ is the posterior probability, we define it as $\gamma(z_k^{(n)})$.

Posterior probability/Responsibility $\gamma(z_k^{(n)})$ can be viewed as the responsibility that component k takes for explaining the observation$x$.

E step We guess the parameters of the model and compute the posterior probability of the latent variables

M step Using the posterior probability, we **update** the parameters that maximizes the probability of the observed variables.

EM algorithm

1. Initialise the parameter: the means $\boldsymbol{\mu}_k$, covariances $\boldsymbol{\Sigma}_k$ and the mixing coefficients π_k; and evaluate the initial log likelihood
2. **E step** Evaluate the posterior probabilities using the current parameter values

$$\gamma(z_k^{(n)}) = \frac{\pi_k \mathcal{N}(\boldsymbol{x}^{(n)}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\boldsymbol{x}^{(n)}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)} \tag{7}$$

3. **M step** Re-estimate the parameters using the current posterior probabilities

$$\boldsymbol{\mu}_k^{new} = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_k^{(n)}) \boldsymbol{x}^{(n)} \tag{8}$$

$$\boldsymbol{\Sigma}_k^{new} = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_k^{(n)}) (\boldsymbol{x}^{(n)} - \boldsymbol{\mu}_k^{new})(\boldsymbol{x}^{(n)} - \boldsymbol{\mu}_k^{new})^T \tag{9}$$

$$\pi_k^{new} = \frac{N_k}{N} \tag{10}$$

where

$$N_k = \sum_{n=1}^{N} \gamma(z_k^{(n)}) \tag{11}$$

We can interpret N_k as the effective number of points assigned to cluster k.

4. Evaluate the log likelihood and check the convergence. If not, return to step 2

5.2 EM Revisited

Instead of calculating the likelihood of the observed variable $\boldsymbol{x}$, we can treat the vector $[\boldsymbol{x}, \boldsymbol{z}]$ as the *complete* data set. Although we do not know $\boldsymbol{z}$, we can represent it as the posterior distribution, i.e. the*expected value* and it is found in the **E step**. The expectation, denoted $\mathcal{Q}(\boldsymbol{\theta}, \boldsymbol{\theta}^{old})$

$$\mathcal{Q}(\boldsymbol{\theta}, \boldsymbol{\theta}^{old}) = \sum_{\mathbf{Z}} p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta}^{old}) \ln p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta}) \tag{12}$$

In the **M step**, the parameters are revised

$$\boldsymbol{\theta}^{new} = \arg\max_{\theta} \mathcal{Q}(\boldsymbol{\theta}, \boldsymbol{\theta}^{old}) \tag{13}$$

5.2.1 Mixture of Gaussians Revisited

Under this framework, the likelihood function for the complete data set $\boldsymbol{X}, \boldsymbol{Z}$ is thus

$$(p(\mathbf{X},\ \mathbf{Z}|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \prod_{n=1}^{N} \prod_{k=1}^{K} \pi_k^{z_k^{(n)}} \mathcal{N}(\boldsymbol{x}^{(n)}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)^{z_k^{(n)}} \tag{14}$$

Giving the log-likelihood function of

$$ln(p(\mathbf{X},\ \mathbf{Z}|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{n=1}^{N} \sum_{k=1}^{K} z_k^{(n)} \{\ln \pi_k + \ln \mathcal{N}(\boldsymbol{x}^{(n)}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)\} \tag{15}$$

Note that this form is more easily optimised than Eq. 5

5.3 K-means clustering

K-means algorithm is a particular limit of EM for Gaussian mixtures.
Consider a Gaussian mixture model in which the covariance matrices of the mixture components are given by $\epsilon\mathbf{I}$, where ϵ is a variance parameter that is shared by all the components.
For $\epsilon \to 0$, it can be shown that responsibilities $\gamma(z_k^{(n)})$ for the data point $\boldsymbol{x}^{(n)}$ all go to zero except for the term j, where the mean $\boldsymbol{\mu}_j$ is closest to the data, which goes to unity. Thus, in this limit, we obtain a **hard assignment** of data points to clusters

5.4 Factor analysis

Factor analysis is based on a probabilistic model, and parameter estimation used the iterative EM algorithm.

6 Learning theory

Courtesy to CS229 by Andrew Ng.

6.1 Bias-variance trade-off

Intuition:

- Bias: Low model complexity → model unable to fit data well (high training and testing error)
  - "Informally, we define the bias of a model to be the expected generalization error even if we were to fit it to a very (say, infinitely) large training set."
- Variance: High model complexity → model unable to generalize well. (low training error and high testing error)
  - The variance could mean that the model fits the variation in the training data.

More quantitative analysis:

The **mean squared error (MSE)** is:

$$\mathbb{E}\left((y - \hat{f}(x))^2\right)$$

If $y = f(x) + \epsilon$ The above equation can be rewritten as

$$\mathbb{E}\left((\epsilon + f(x) - \hat{f}(x))^2\right)$$

Rearrange the terms

$$\text{TestMSE} = \sigma^2 + \left(\mathbb{E}(f(x) - \hat{f}(x))\right)^2 + Var\left(f(x) - \hat{f}(x)\right)$$

The three terms are noise variance, Bias $\hat{f}(x)$ and Variance of the model.

6.2 Error analysis

For a machine learning pipeline

Groud-truth plugin By plugging-in the ground-truth for each component, see how accuracy changes

Ablative analysis By removing a component, see how accuracy changes

Analysing the mistake See whether there is trend.

6.3 Learning theory theorems

6.3.1 Preliminaries

lemma: The union bound The probability of any one of k events happening is at most the sums of the probabilities of the k different events.

lemma: Hoeffding inequality For m i.i.d. random variables drawn from a Bernoulli(ϕ) distribution. Let $\hat{\phi}$ be the mean of the sample. Then

$$P(|\phi - \hat{\phi}| > \gamma) \leq 2\exp(-\gamma^2 m)$$

Some definitions

- **PAC** stands for probably approximately correct which is a **framework** and set of assumptions under which numerous results on learning theory were proved. Of these, the assumption of training and testing on the **same distribution**, and the assumption of the independently drawn training examples, were the most important.
- **training error** (also called the empirical risk or empirical error in learning theory)
- **empirical risk minimization (ERM)** The process of choosing the the hypothesis that minimises the empirical risk (i.e. training error)
- **Hypothesis class** $\mathcal{H}$ used by a learning algorithm is the set of all classifiers considered by it.

6.3.2 Finite hypothesis class

Theorem Let $|\mathcal{H}| = k$, and let any m, δ be fixed. Then with probability at least 1δ, we have that

$$\epsilon(\hat{h}) \leq \left(\min_{h \in \mathcal{H}} \epsilon(h)\right) + 2\sqrt{\frac{1}{2m}\log\frac{2k}{\delta}}$$

- With more complex model ($\mathcal{H} \subseteq \mathcal{H}'$, k is higher), the training error ($\epsilon(h)$) will be lower, the second term is higher (variance)
- We can calculate the size m for a given error $\epsilon(\hat{h})$ for fixed probability.

6.3.3 Infinite hypothesis class

Shatter Given a set (S) of points, we say that $\mathcal{H}$ shatters S if $\mathcal{H}$ can realize any labeling on S.

Vapnik-Chervonenkis dimension Given a hypothesis class $\mathcal{H}$, we then define its Vapnik-Chervonenkis dimension, written VC($\mathcal{H}$), to be the size of the largest set that is shattered by v. (If $\mathcal{H}$ can shatter arbitrarily large sets, then VC($\mathcal{H}$) $= \infty$.)

Theorem Let $\mathcal{H}$ be given, and let $d = \text{VC}(\mathcal{H})$. Then with probability at least 1δ, we have that

$$\epsilon(\hat{h}) \leq \left(\min_{h \in \mathcal{H}} \epsilon(h)\right) + O\left(\sqrt{\frac{d}{m}\log\frac{m}{d} + \frac{1}{m}\log\frac{1}{\delta}}\right)$$