Chapter 11
Unimodal Models

supplementary slides to
Machine Learning Fundamentals

published by Cambridge University Press

August 2020

CAMBRIDGE
UNIVERSITY PRESS

Outline

1 Gaussian Models

2 Multinomial Models

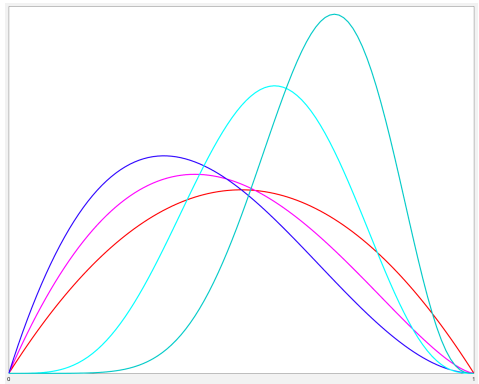
3 Markov Chain Models

4 Generalized Linear Models

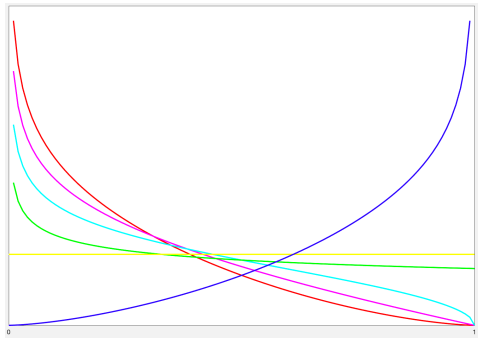
Unimodal Models

- unimodal models:

  simple generative models with a single peak
  extended to include all bounded monotonic functions
  unimodality for multivariate models: all marginal distributions are unimodal



- unimodal models include almost all common probability distributions, Markov chains, generalized linear models, etc.
- parameter estimation is straightforward
- suitable for simple data distributions where the probability mass is concentrated only in a single region of the space



Multivariate Gaussian Models (I)

- given a training set: $D = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$ $(\forall \mathbf{x}_i \in \mathbb{R}^d)$
- choose a multivariate Gaussian distribution to model D:

$$p_{\boldsymbol{\mu},\boldsymbol{\Sigma}}(\mathbf{x}) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu},\boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{d/2}|\boldsymbol{\Sigma}|^{1/2}} e^{-\frac{(\mathbf{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})}{2}}$$

- the log-likelihood function:

$$\begin{aligned} l(\boldsymbol{\mu},\boldsymbol{\Sigma}) &= \sum_{i=1}^{N} \ln p_{\boldsymbol{\mu},\boldsymbol{\Sigma}}(\mathbf{x}_i) \\ &= C - \frac{N}{2}\ln|\boldsymbol{\Sigma}| - \frac{1}{2}\sum_{i=1}^{N}(\mathbf{x}_i-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}_i-\boldsymbol{\mu}) \end{aligned}$$

Multivariate Gaussian Models (II)

$$\frac{@l(\ ;\ )}{@} = 0 \ =) \ \sum_{i=1}^{N} \ ^{-1}(x_i \quad) = 0 \ =) \ _{MLE} = \frac{1}{N} \sum_{i=1}^{N} x_i$$

$$\frac{@l(\ ;\ )}{@} = 0 \ =) \ \frac{N}{2}(\ \ |)^{-1} + \frac{1}{2}(\ \ |)^{-1} \left[\sum_{i=1}^{N} (x_i \quad)(x_i \quad)| \right] (\ \ |)^{-1}$$

$$=) \ _{MLE} = \frac{1}{N} \sum_{i=1}^{N} (x_i \quad _{MLE})(x_i \quad _{MLE})|$$

Note that $\frac{@}{@A} \ x|A^{-1}y \ = \ (A|)^{-1}xy|(A|)^{-1}$ (square A)

$\frac{@}{@A} \ \ln|jAj \ = (A^{-1})| = (A|)^{-1}$ (square A)

Gaussian Models for Classification

- assuming K classes $\{\omega_1; \quad ; \omega_K\}$, we collect a training set D_k for each class ω_k $(k = 1; 2; \quad ; K)$
- if each feature vector is continuous $(\mathbf{x} \in \mathbb{R}^d)$ and follows a unimodal distribution, we may choose a multivariate Gaussian for each class, i.e. $N(\mathbf{x}|\boldsymbol{\mu}^{(k)}; \boldsymbol{\Sigma}^{(k)})$ $(k = 1; 2; \quad ; K)$
- maximum likelihood estimation: $D_k \Rightarrow \boldsymbol{\mu}^{(k)}_{\text{MLE}}; \boldsymbol{\Sigma}^{(k)}_{\text{MLE}}$
- classify any new $\mathbf{x}$ using the plug-in MAP decision rule:

$$g(\mathbf{x}) = \arg\max_k \Pr(\omega_k) p(\mathbf{x}|\omega_k) = \arg\max_k N(\mathbf{x}|\boldsymbol{\mu}^{(k)}_{\text{MLE}}; \boldsymbol{\Sigma}^{(k)}_{\text{MLE}})$$

assuming all priors $\Pr(\omega_k)$ are equiprobable.

Separation Boundary of Gaussian Models

- quadratic discriminant analysis (QDA)

  Gaussian models lead to quadratic classifiers separation boundaries between classes are a parabola-like quadratic surface

- linear discriminant analysis (LDA)

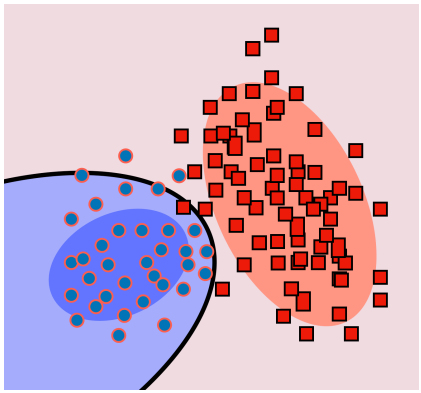
  use a common covariance matrix

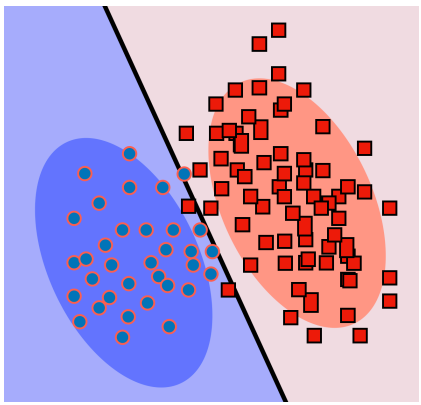
$$D_1; D_2; \quad ; D_K \quad ! \quad \text{MLE}$$

$$D_k \quad ! \quad \boldsymbol{\mu}_{\text{MLE}}^{(k)} \quad (k = 1; \quad ; K)$$

  the plug-in MAP rule leads to a linear separation boundary:

$$g(\mathbf{x}) = \arg\max_k \; N(\mathbf{x} j \; {}^{(k)}_{\text{MLE}}; \quad {}_{\text{MLE}})$$





Multinomial Models

- discrete data consists of some independent observations, e.g. $X = \{x_1, x_2, \ldots, x_T\}$, each of which is a distinct symbol
- assume there are M distinct symbols in total
- denote p_i as the probability of observing the i-th symbol for all $i = 1, 2, \ldots, M$
- the sum-to-one constraint: $\sum_{i=1}^{M} p_i = 1$
- use a multinomial model for X:

$$\Pr(X \mid p_1, p_2, \ldots, p_M) = \frac{(r_1 + r_2 + \cdots + r_M)!}{r_1! r_2! \cdots r_M!} p_1^{r_1} p_2^{r_2} \cdots p_M^{r_M}$$

where r_i $(i = 1, 2, \ldots, M)$ denotes the frequency of the i-th symbol appearing in X

Maximum Likelihood Estimation of Multinomial Models

- estimate all parameters of a multinomial model, i.e. $\{p_1; p_2; \ldots; p_M\}$, from some observation $\mathcal{X}$
- maximize the log-likelihood function:

$$\arg \max_{p_1; p_2; \ldots; p_M} \ln \Pr(\mathcal{X} \mid p_1; p_2; \ldots; p_M)$$

subject to $\sum_{i=1}^{M} p_i = 1$

- construct the Lagrangian function:

$$L(p_1; p_2; \ldots; p_M; \lambda) = C + \sum_{i=1}^{M} r_i \ln p_i - \lambda \left(\sum_{i=1}^{M} p_i - 1 \right)$$

$$\frac{\partial}{\partial p} L(p_1; p_2; \ldots; p_M; \lambda) = 0 \Rightarrow p_i = \frac{r_i}{\lambda}$$

- maximum likelihood estimation:

$$p_i^{(\mathrm{MLE})} = \frac{r_i}{\sum_{i=1}^{M} r_i} \quad (i = 1; 2; \ldots; M)$$

Example: Multinomal Models for DNA Sequences

X = GAATTCTTCAAAGAGTTCCAGATATCCACAGGCAGATTCTA
GCACACATCTCAATGAAGTTCCTGAGAAAGCTTCTGTCTAGTTTT
GAAAATATTTCCTTTTCCATCATGGGCCTCAAAGCGCTCAAAATG
TTGCAGATACTAGAGAAAGACTGTTT

- assume all nucleotides in a sequence are independent
- p_1 denotes the probability of observing G at any location, p_2 for A, p_3 for C, p_4 for T

$$\Pr(X \mid p_1, p_2, p_3, p_4) = \frac{(r_1 + r_2 + r_3 + r_4)!}{r_1! r_2! r_3! r_4!} \prod_{i=1}^{4} p_i^{r_i}$$

- maximum likelihood estimation:

$$p_i^{(\text{MLE})} = \frac{r_i}{\sum_{i=1}^{4} r_i} \quad (i = 1, 2, 3, 4)$$

Markov Chain Models

- multinomial models treat sequence as a bag of symbols by ignoring the order information
- how to model the sequential information of sequences?

$$X = \{ x_1 x_2 x_3 \quad x_{t-1} x_t x_{t+1} \quad x_T \}$$

- consider the product rule:

$$\Pr(X) = p(x_1)\, p(x_2 \mid x_1)\, p(x_3 \mid x_1 x_2) \quad p(x_t \mid x_1 \quad x_{t-1}) \quad p(x_T \mid x_1 \quad x_{T-1})$$

- uncontrollable complexity: as a sequence gets longer and longer, it requires some conditional distributions involving more and more parameters.

Markov Chain Models: Markov Assumption

- Markov assumption : every random variable in a sequence only depends on its most recent history, independent from the others given the most recent history
- 1st-order Markov assumption:

$$p(x_t \mid x_1, \ldots, x_{t-1}) = p(x_t \mid x_{t-1})$$

- 2nd-order Markov assumption:

$$p(x_t \mid x_1, \ldots, x_{t-1}) = p(x_t \mid x_{t-2}, x_{t-1})$$

- Markov chain models:

$$Pr(X) = p(x_1) \prod_{t=2}^{T} p(x_t \mid x_{t-1})$$

$$Pr(X) = p(x_1)\, p(x_2 \mid x_1) \prod_{t=3}^{T} p(x_t \mid x_{t-2}, x_{t-1})$$

Markov Chain Models: Two More Assumptions

- stationary assumption :

  conditional distributions do not change over time

$$p\ x_t\ \ x_{t\ 1}\ = p\ x_{t0}\ \ x_{t0\ 1}$$

  allow to use one conditional distribution for all time instances

- discrete observation assumption: all observations in a sequence are the same random variable out of a nite set of distinct symbols, i.e.f ! $_1$; ! $_2$; ; ! $_M$ g

  represent the conditional distribution as a transition matrix:

$$A = \begin{bmatrix} & \\ a_{ij} & \\ & \end{bmatrix}_{M \quad M}$$

  with a_{ij} = Pr x_t = ! $_j$ $x_{t\ 1}$ = ! $_i$

  represent Markov chain models as directed graphs

Example 1: Markov Chain Models for DNA Sequences

- 1st-order Markov chain model for DNA sequence

- MLE formula: $a_{ij}^{(MLE)} = \frac{r(!_i !_j)}{r(!_i)}$ for all (1 i; j M)
- for any new sequence:

$$Pr(GAATC) = p(Gjbegin)p(AjG)p(AjA)p(TjA)p(CjT)p(endjC)$$

$$= 0:25 \quad 0:16 \quad 0:18 \quad 0:12 \quad 0:35 \quad 0:01$$

Example 2: N-gram Language Models

- n-gram language models: use Markov chain models for languages
  - each word only depends on its previous word(s)
  - a set of word conditional probabilities
- given any sentence:

  S = I would like to fly from Toronto to Chicago this Friday
- 1st-order Markov chain (bi-gram model): N^2 parameters

$$\Pr(S) = p(\text{I}|\text{begin})\, p(\text{would}|\text{I})\, p(\text{like}|\text{would}) \cdots p(\text{end}|\text{Friday})$$

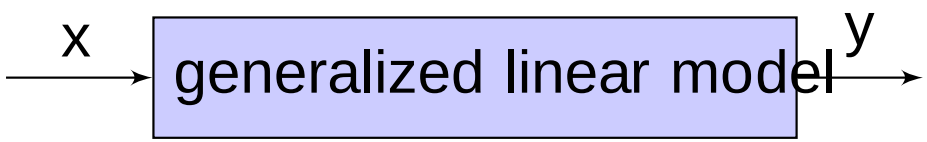
- 2nd-order Markov chain (tri-gram model): N^3 parameters

$$\Pr(S) = p(\text{I}|\text{begin})\, p(\text{would}|\text{begin; I})\, p(\text{like}|\text{I; would}) \cdots p(\text{end}|\text{this; Friday})$$

- zero-order Markov chain (uni-gram model): N parameters

$$\Pr(S) = p(\text{I})\, p(\text{would})\, p(\text{like}) \cdots p(\text{Friday})$$

Generalized Linear Models



generalized linear models (GLMs): extend linear regression to deal with non-Gaussian distributions

1. select a unimodal probability distribution for output, i.e. $p(y)$
   e.g. binomial, multinomial, Poisson

2. choose a link function $g(\cdot)$ to associate the mean of y to a linear predictor of input x:
   $$\mathrm{E}\, y = g(w^{\mathrm{T}} x)$$
   e.g. exp( ), sigmoid, softmax
   the range of the link function must match the domain of mean

Generalized Linear Models: Some Examples

GLM	y	distribution	g( )
linear regression	R	Gaussian	identity
logistic regression	binary	binomial	sigmoid
probit regression	binary	binomial	probit
Poisson regression	count	Poisson	exp( )
log-linear model	categorical	multinomial	softmax

Table: Some popular generalized linear models (GLMs)

GLM Example (1): Logistic & Probit Regression

- GLMs for binary classification: $y \in \{0, 1\}$
- choose binomial distribution for $p(y)$:

$$y \sim B(y \mid N = 1, p) = p^{y}(1 - p)^{1-y}$$

where $E[y] = p \in (0, 1)$

- logistic regression uses the sigmoid function as the link function: $p = \sigma(w^{T} x)$

$$\hat{p}_{w}(y|x) = \sigma(w^{T} x)^{y} \left[1 - \sigma(w^{T} x)\right]^{1-y}$$

- probit regression uses the probit function as the link function: $p = \Phi(w^{T} x)$

$$\hat{p}_{w}(y|x) = \Phi(w^{T} x)^{y} \left[1 - \Phi(w^{T} x)\right]^{1-y}$$

- sigmoid

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

- probit

$$\Phi(x) = \frac{1}{2}\left[1 + \mathrm{erf}(x)\right]$$

GLM Example (2): Poisson Regression

- A GLM for count data: $y = 0, 1, 2, 3, \ldots$
- choose Poisson distribution for $p(y)$:

$$y \quad p(y|\ ) = \frac{e^{\quad y}}{y!} \quad \forall y = 0, 1, 2, 3, \ldots$$

where $E\ y\ =\ >\ 0$

- Poisson regression uses the exponential function as the link function: $\quad = \exp(w^{|} x)$

$$\hat{p}_w(y|x) = \frac{1}{y!} \exp \quad \exp(w^{|} x) \ \exp\ y w^{|} x \qquad y = 0, 1, 2, 3, \ldots$$

- derive the maximum likelihood estimation $\hat{w}_{MLE}$

GLM Example (3): Log-linear Model

- A GLM for K-class pattern classification, i.e. $y \in \{\omega_1, \omega_2, \ldots, \omega_K\}$
- use the 1-of-K representation to encode y as a K-dimension one-hot vector $\mathbf{y} = [y_1\ y_2\ \ldots\ y_K]^T$
- choose multinomial distribution for $p(\mathbf{y})$:

$$\mathbf{y} \sim \text{Mult}(\mathbf{y}; N = 1, p_1, \ldots, p_K) = \prod_{k=1}^{K} p_k^{y_k}$$

where $E[\mathbf{y}] = [p_1\ p_2\ \ldots\ p_K]^T$

- choose the softmax function as the link function:

$$E[\mathbf{y}] = \text{softmax}(\mathbf{x}) = \left[\frac{e^{\mathbf{w}_1^T \mathbf{x}}}{\sum_{k=1}^{K} e^{\mathbf{w}_k^T \mathbf{x}}} \quad \frac{e^{\mathbf{w}_2^T \mathbf{x}}}{\sum_{k=1}^{K} e^{\mathbf{w}_k^T \mathbf{x}}} \quad \cdots \quad \frac{e^{\mathbf{w}_K^T \mathbf{x}}}{\sum_{k=1}^{K} e^{\mathbf{w}_k^T \mathbf{x}}} \right]^T$$

- log-linear model:

$$\hat{p}_{\mathbf{w}_1, \ldots, \mathbf{w}_K}(\mathbf{y} \mid \mathbf{x}) = \prod_{k=1}^{K} \left(\frac{e^{\mathbf{w}_k^T \mathbf{x}}}{\sum_{k=1}^{K} e^{\mathbf{w}_k^T \mathbf{x}}} \right)^{y_k}$$

Log-linear Models for Text Categorization

- text categorization: automatically classify text documents into different categories
- feature engineering: use some pre-defined rules to extract a fixed-size feature vector $\mathbf{x}$ to represent each text document
- use log-linear models as the classifier
- given a training set as $D = \{(\mathbf{x}^{(i)}, y^{(i)}) \mid i = 1, 2, \ldots, N\}$
- MLE: to maximize log-likelihood function

$$I(\mathbf{w}_1, \ldots, \mathbf{w}_K) = \sum_{i=1}^{N} \sum_{k=1}^{K} y_k^{(i)} \ln \left(\frac{e^{\mathbf{w}_k^T \mathbf{x}^{(i)}}}{\sum_{k=1}^{K} e^{\mathbf{w}_k^T \mathbf{x}^{(i)}}} \right)$$

- classify any new document $\mathbf{x}$:

$$\hat{k} = \arg\max_k \Pr(\omega_k | \mathbf{x}) = \arg\max_k \frac{e^{(\mathbf{w}_k^{(\text{MLE})})^T \mathbf{x}}}{\sum_{k=1}^{K} e^{(\mathbf{w}_k^{(\text{MLE})})^T \mathbf{x}}}$$

$$= \arg\max_k \mathbf{x}^T \mathbf{w}_k^{(\text{MLE})}$$