Chapter 11
Unimodal Models

supplementary slides to
Machine Learning Fundamentals

published by Cambridge University Press

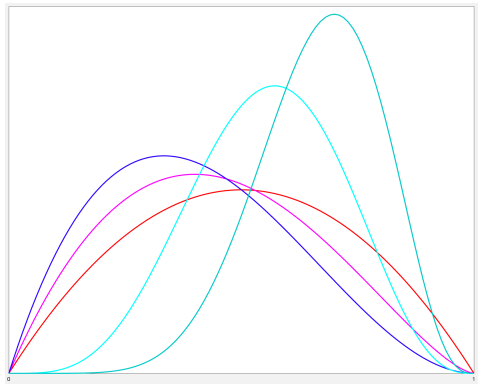
August 2020

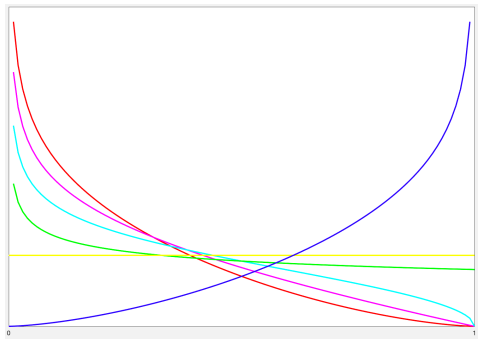
CAMBRIDGE
UNIVERSITY PRESS

Outline

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

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

- the log-likelihood function:

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

Multivariate Gaussian Models (II)

$$\frac{\partial l(\boldsymbol{\mu}, \boldsymbol{\Sigma})}{\partial \boldsymbol{\mu}} = 0 \implies \sum_{i=1}^{N} \boldsymbol{\Sigma}^{-1}(\mathbf{x}_i - \boldsymbol{\mu}) = 0 \implies \boldsymbol{\mu}_{\text{MLE}} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{x}_i$$

$$\frac{\partial l(\boldsymbol{\mu}, \boldsymbol{\Sigma})}{\partial \boldsymbol{\Sigma}} = 0 \implies -\frac{N}{2}(\boldsymbol{\Sigma}^{\mathsf{T}})^{-1} + \frac{1}{2}(\boldsymbol{\Sigma}^{\mathsf{T}})^{-1}\Big[\sum_{i=1}^{N}(\mathbf{x}_i - \boldsymbol{\mu})(\mathbf{x}_i - \boldsymbol{\mu})^{\mathsf{T}}\Big](\boldsymbol{\Sigma}^{\mathsf{T}})^{-1}$$

$$\implies \boldsymbol{\Sigma}_{\text{MLE}} = \frac{1}{N}\sum_{i=1}^{N}(\mathbf{x}_i - \boldsymbol{\mu}_{\text{MLE}})(\mathbf{x}_i - \boldsymbol{\mu}_{\text{MLE}})^{\mathsf{T}}$$

Note that $\frac{\partial}{\partial A}\Big(\mathbf{x}^{\mathsf{T}} A^{-1}\mathbf{y}\Big) = -(A^{\mathsf{T}})^{-1}\mathbf{x}\mathbf{y}^{\mathsf{T}}(A^{\mathsf{T}})^{-1}$ (square A)

$\frac{\partial}{\partial A}\Big(\ln|A|\Big) = (A^{-1})^{\mathsf{T}} = (A^{\mathsf{T}})^{-1}$ (square A)

Gaussian Models for Classification

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

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

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

Separation Boundary of Gaussian Models

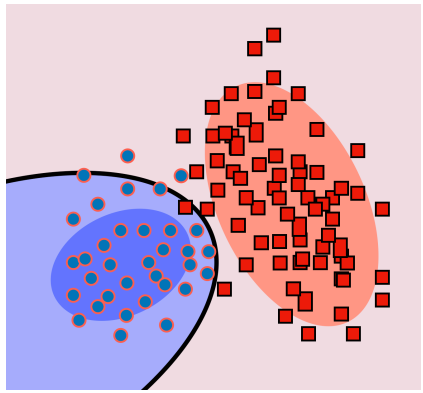
- quadratic discriminant analysis (QDA)
  - Gaussian models lead to quadratic classifiers
  - separation boundaries between classes are a parabola-like quadratic surface
- linear discriminant analysis (LDA)
  - use a common covariance matrix

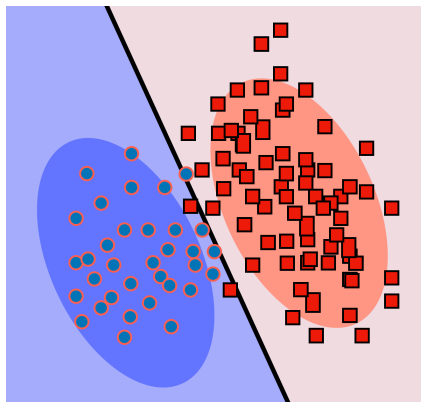
$$\{\mathcal{D}_1, \mathcal{D}_2, \cdots, \mathcal{D}_K\} \longrightarrow \boldsymbol{\Sigma}_{\text{MLE}}$$

$$\mathcal{D}_k \longrightarrow \boldsymbol{\mu}_{\text{MLE}}^{(k)} \quad (k = 1, \cdots, K)$$

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

$$g(\mathbf{x}) = \arg\max_k \mathcal{N}(\mathbf{x} \,|\, \boldsymbol{\mu}_{\text{MLE}}^{(k)}, \boldsymbol{\Sigma}_{\text{MLE}})$$





Multinomial Models

- discrete data consists of some *independent* observations, e.g. $\mathbf{X} = \{x_1, x_2, \cdots, x_T\}$, 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 $i = 1, 2, \cdots, M$
- the sum-to-one constraint: $\sum_{i=1}^{M} p_i = 1$
- use a multinomial model for $\mathbf{X}$:

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

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

Maximum Likelihood Estimation of Multinomial Models

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

$$\arg \max_{p_1, p_2, \cdots p_M} \ln \Pr(\mathbf{X} \mid p_1, p_2, \cdots p_M)$$

  subject to $\sum_{i=1}^{M} p_i = 1$
- construct the Lagrangian function:

$$\mathcal{L}(p_1, p_2, \cdots, p_M, \lambda) = C + \sum_{i=1}^{M} r_i \cdot \ln p_i - \lambda \cdot \Big(\sum_{i=1}^{M} p_i - 1 \Big)$$

$$\frac{\partial}{\partial p_i} \mathcal{L}(p_1, p_2, \cdots, p_M, \lambda) = 0 \implies p_i = \frac{r_i}{\lambda}$$

- maximum likelihood estimation:

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

Example: Multinomal Models for DNA Sequences

$\mathbf{X} =$ GAATTCTTCAAAGAGTTCCAGATATCCACAGGCAGATTCTACAAAAG
GCACACATCTCAATGAAGTTCCTGAGAAAGCTTCTGTCTAGTTTTTATGCTCT
GAAAATATTTCCTTTTCCATCATGGGCCTCAAAGCGCTCAAAATGAACAAAAC
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

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

- 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?

$$\mathbf{X} = \left\{ x_1 \; x_2 \; x_3 \cdots x_{t-1} \; x_t \; x_{t+1} \cdots x_T \right\}$$

- consider the product rule:

$$\begin{aligned} \Pr(\mathbf{X}) \quad = \quad & p\big(x_1\big)p\big(x_2\big|x_1\big)p\big(x_3\big|x_1x_2\big)\cdots \\ & p\big(x_t\big|x_1\cdots x_{t-1}\big)\cdots p\big(x_T\big|x_1\cdots x_{T-1}\big) \end{aligned}$$

- 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:

$$p\big(x_t \,\big|\, x_1 \cdots x_{t-1}\big) = p\big(x_t \,\big|\, x_{t-1}\big)$$

- 2nd-order Markov assumption:

$$p\big(x_t \,\big|\, x_1 \cdots x_{t-1}\big) = p\big(x_t \,\big|\, x_{t-2}\, x_{t-1}\big)$$

- Markov chain models:

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

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

Markov Chain Models: Two More Assumptions

- **stationary assumption**:
  - conditional distributions do not change over time

$$p\left(x_t \,\middle|\, x_{t-1}\right) = p\left(x_{t'} \,\middle|\, x_{t'-1}\right)$$

  - 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 finite set of M distinct symbols, i.e. $\{\omega_1, \omega_2, \cdots, \omega_M\}$
  - represent the conditional distribution as a transition matrix:

$$\mathbf{A} = \left[\quad a_{ij} \quad \right]_{M \times M}$$

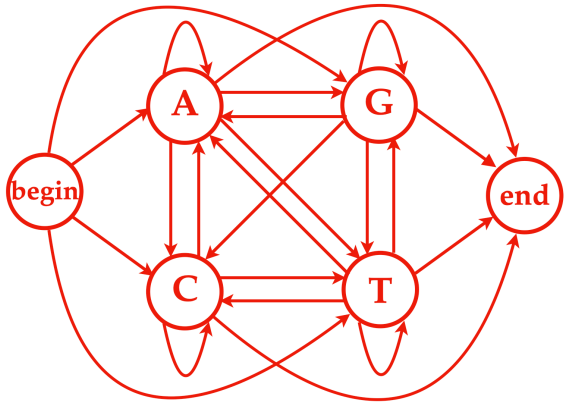
with $a_{ij} = \Pr\left(x_t = \omega_j \,\middle|\, x_{t-1} = \omega_i\right)$

  - represent Markov chain models as directed graphs

Example 1: Markov Chain Models for DNA Sequences

- 1st-order Markov chain model for DNA sequence

	begin	A	C	G	T	*end*
begin	0	0.28	0.24	0.25	0.23	0
A	0	0.18	0.27	0.42	0.12	0.01
C	0	0.17	0.36	0.27	0.19	0.01
G	0	0.16	0.34	0.37	0.12	0.01
T	0	0.08	0.35	0.38	0.18	0.01
end	0	0	0	0	0	1



- MLE formula: $a_{ij}^{(\text{MLE})} = \frac{r(\omega_i\omega_j)}{r(\omega_i)}$ for all $(1 \leq i, j \leq M)$
- for any new sequence:

$$\Pr(\text{GAATC}) = p(\text{G}|begin)p(\text{A}|\text{G})p(\text{A}|\text{A})p(\text{T}|\text{A})p(\text{C}|\text{T})p(end|\text{C})$$

$$= 0.25 \times 0.16 \times 0.18 \times 0.12 \times 0.35 \times 0.01$$

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:

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

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

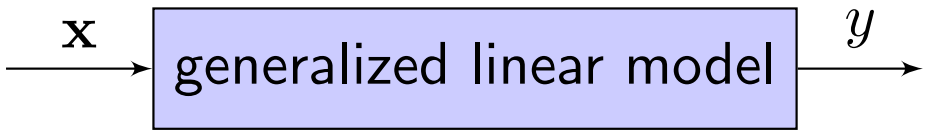
- 2nd-order Markov chain (tri-gram model): N^3 parameters

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

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

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

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 $\mathbf{x}$:

$$\mathbb{E}\big[y\big] = g(\mathbf{w}^{\mathsf{T}}\mathbf{x})$$

   - e.g. $\exp(\cdot)$, sigmoid, softmax
   - the range of the link function must match the domain of mean

Generalized Linear Models: Some Examples

GLM	y	distribution	$g(\cdot)$
linear regression	$\mathbb{R}$	Gaussian	identity
logistic regression	binary	binomial	sigmoid
probit regression	binary	binomial	probit
Poisson regression	count	Poisson	$\exp(\cdot)$
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 $p(y)$:

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

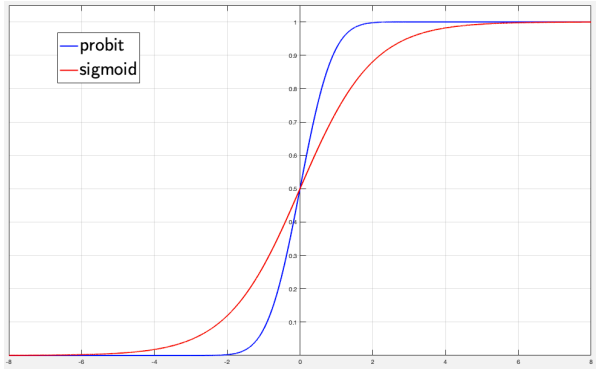
where $\mathbb{E}[y] = p \in (0, 1)$

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

$$\hat{p}_{\mathbf{w}}(y|\mathbf{x}) = \big(l(\mathbf{w}^{\mathsf{T}}\mathbf{x})\big)^y \big(1 - l(\mathbf{w}^{\mathsf{T}}\mathbf{x})\big)^{1-y}$$

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

$$\hat{p}_{\mathbf{w}}(y|\mathbf{x}) = \Big(\Phi\big(\mathbf{w}^{\mathsf{T}}\mathbf{x}\big)\Big)^y \Big(1 - \Phi\big(\mathbf{w}^{\mathsf{T}}\mathbf{x}\big)\Big)^{1-y}$$



- sigmoid

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

- probit

$$\Phi(x) = \frac{1}{2}\Big(1 + \mathsf{erf}(x)\Big)$$

GLM Example (2): Poisson Regression

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

$$y \sim p(y \,|\, \lambda) = \frac{e^{-\lambda} \cdot \lambda^y}{y!} \quad \forall y = 0, 1, 2, 3, \cdots$$

where $\mathbb{E}\big[y\big] = \lambda > 0$

- **Poisson regression** uses the exponential function as the link function: $\lambda = \exp(\mathbf{w}^{\mathsf{T}}\mathbf{x})$

$$\hat{p}_{\mathbf{w}}(y|\mathbf{x}) = \frac{1}{y!} \exp\left(-\exp(\mathbf{w}^{\mathsf{T}}\mathbf{x})\right) \exp\left(y\mathbf{w}^{\mathsf{T}}\mathbf{x}\right) \quad y = 0, 1, 2, 3, \cdots$$

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

GLM Example (3): Log-linear Model

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

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

  where $\mathbb{E}[\mathbf{y}] = [p_1 \; p_2 \; \cdots \; p_K]^{\mathsf{T}}$
- choose the softmax function as the link function:

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

- **log-linear model**:

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

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 $\mathcal{D} = \left\{(\mathbf{x}^{(i)}, \mathbf{y}^{(i)}) \mid i = 1, 2, \cdots N\right\}$
- MLE: to maximize log-likelihood function

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

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

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