PARIS - CACHAN . TROYES . MONTPELLIER . SAINT-NAZAIRE . DAKAR

Predictive modeling and unsupervised clustering

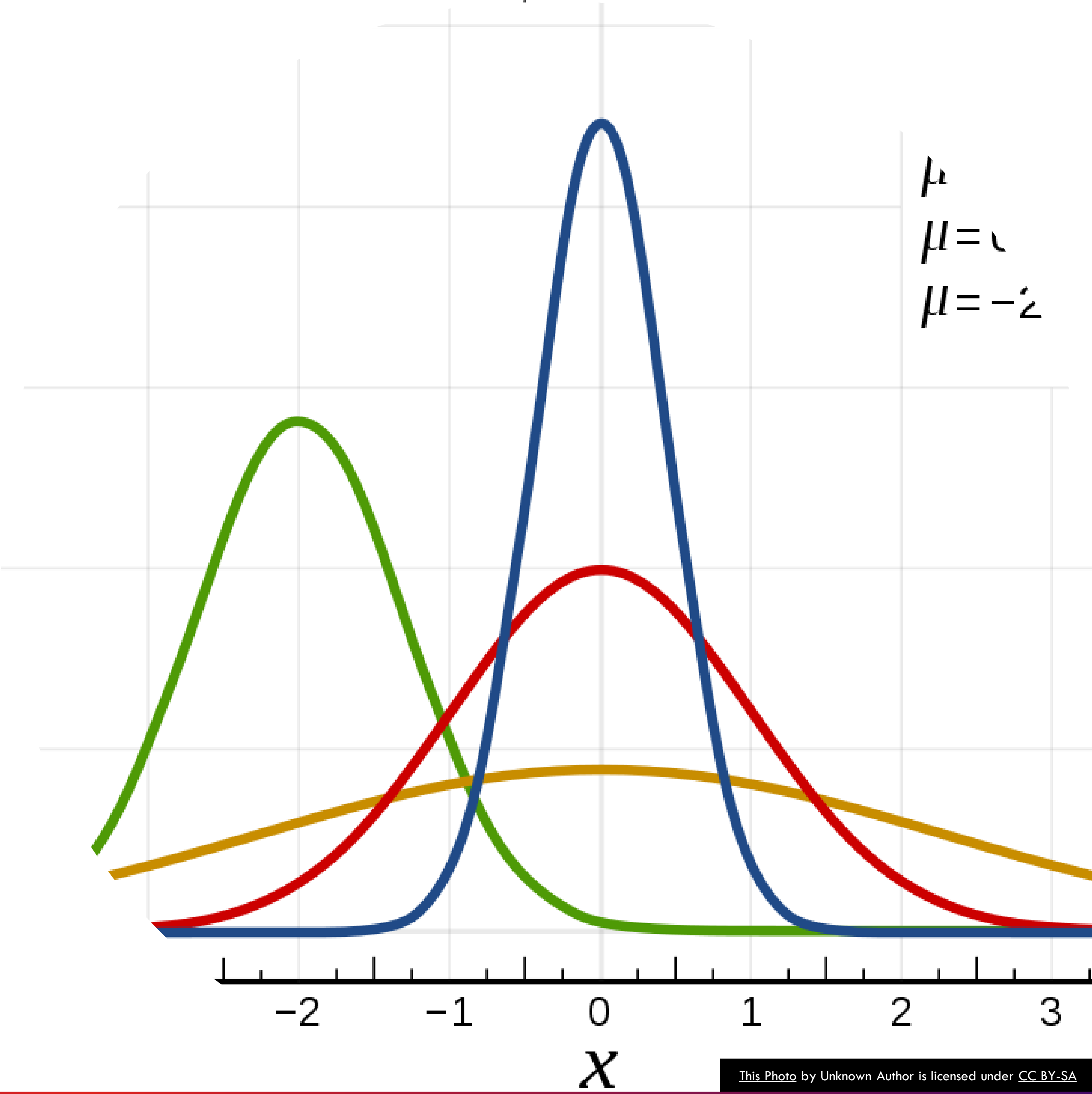
Thalita Drumond

ÉCOLE D'INGÉNIEUR·E·S
Creating the future together

Extra Exercises

Learning predictive models

Gradient descent for linear regression



GD update with MSE cost function

Ordinary least squares

Minimize the residual sum of squares (RSS):

$$\mathrm{J}(\theta) = \frac{1}{2}\sum_{i=1}^{n}\left(\varepsilon^{(i)}\right)^2$$

With $\varepsilon^{(i)}$ being the error between the hypothesis h_θ for input $\boldsymbol{x}^{(i)}$ and the ground truth values $y^{(i)}$:

$$\varepsilon^{(\boldsymbol{i})} := h_\theta\left(\boldsymbol{x}^{(\boldsymbol{i})}\right) - y^{(i)} = \boldsymbol{\theta}^{\boldsymbol{T}}\underbrace{\boldsymbol{x}^{(i)} - y^{(i)}}$$

$\left(\boldsymbol{x}^{(\boldsymbol{i})}, y^{(i)}\right)$ is the i-th sample in the dataset $\mathcal{D}$

Exercise 6

Suppose we will minimize the least-squares cost function using gradient descent:

$$\theta_j[t+1] = \theta_j[t] - \eta\, \Delta\theta_j[t]$$

where η is the learning rate (or step size) and $\Delta\theta_j = \frac{\partial J}{\partial \theta_j}$.

Show that the parameter update is given by $\Delta\theta_j = x_j^{(i)} \varepsilon^{(i)}$

Binary classification with MLE

Binary logistic regression

Binary classification with logistic regression

The hypothesis function is:

$$h_\theta(\boldsymbol{x}) = \sigma\left(\sum_{j=0}^{d} \theta_j x^j\right) = \sigma(\boldsymbol{\theta}^T \boldsymbol{x})$$

$$\boldsymbol{\theta}^T : [\theta_1 \cdots \theta_j \ldots \theta_d]$$

Where $\sigma(a)$ is the logistic function:

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

$$\boldsymbol{x} : \begin{bmatrix} x^1 \\ \vdots \\ x^j \\ \vdots \\ x^d \end{bmatrix}$$

This hypothesis predicts values of

$$h_\theta(\boldsymbol{x}) \in [0,1]$$

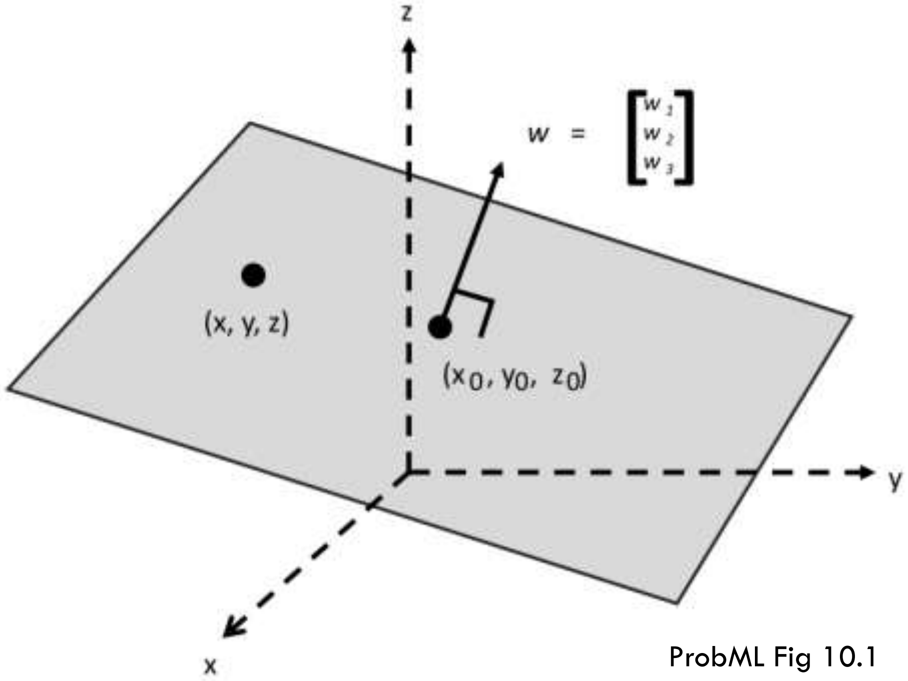
Binary classification with logistic regression

$$p = \text{logistic}(a) = \boldsymbol{\sigma}(a) \triangleq \frac{1}{1 + e^{-a}} = \frac{e^a}{1 + e^a}$$

$$a = \text{logit}(p) = \boldsymbol{\sigma}^{-1}(p) \triangleq \log\left(\frac{p}{1 - p}\right)$$

$$p(y|\boldsymbol{x};\boldsymbol{\theta}) = \text{Ber}(y|\boldsymbol{\sigma}(\underbrace{w^T x + b}_{a = \boldsymbol{w}^{\mathsf{T}}\boldsymbol{x} + b}))$$

$$p(y = 1|\boldsymbol{x};\boldsymbol{\theta}) = \boldsymbol{\sigma}(a) = \frac{1}{1 + e^{-a}}$$



ProbML Fig 10.1

Exercise 7

Based on the NLL loss function derived in exercise 4,

$$\mathrm{NLL}(\boldsymbol{\theta}) = -\sum_{i=1}^{N} y_i \log h_\theta(\boldsymbol{x}_i) + (1 - y_i) \log\big(1 - h_\theta(\boldsymbol{x}_i)\big)$$

use the logistic hypothesis

$$p(y = 1 | \boldsymbol{x}; \boldsymbol{\theta}) = \boldsymbol{\sigma}(a) = \frac{1}{1 + e^{-a}}$$

to write the cost function for logistic regression.

Binary classification with MLE
GD update for logistic regression

Binary classification with MLE
Gradient descent update

NLL cost function:

$$J(\theta) = NLL(\theta) = -\log \mathcal{L}(\theta)$$

$$= -\sum_{i=1}^{N} \boldsymbol{y_i} \log h_\theta(\boldsymbol{x_i}) + (1 - y_i)\log\left(1 - h_\theta(\boldsymbol{x_i})\right)$$

$$\Delta\theta_j \triangleq \frac{\partial}{\partial\theta_j} J(\theta)?$$

Exercise 8

Suppose we will minimize the binary classification NLL cost function using gradient descent:

$$\theta_j[t+1] = \theta_j[t] - \eta\, \Delta\theta_j[t]$$

where η is the learning rate (or step size) and $\Delta\theta_j = \frac{\partial J}{\partial \theta_j}$.

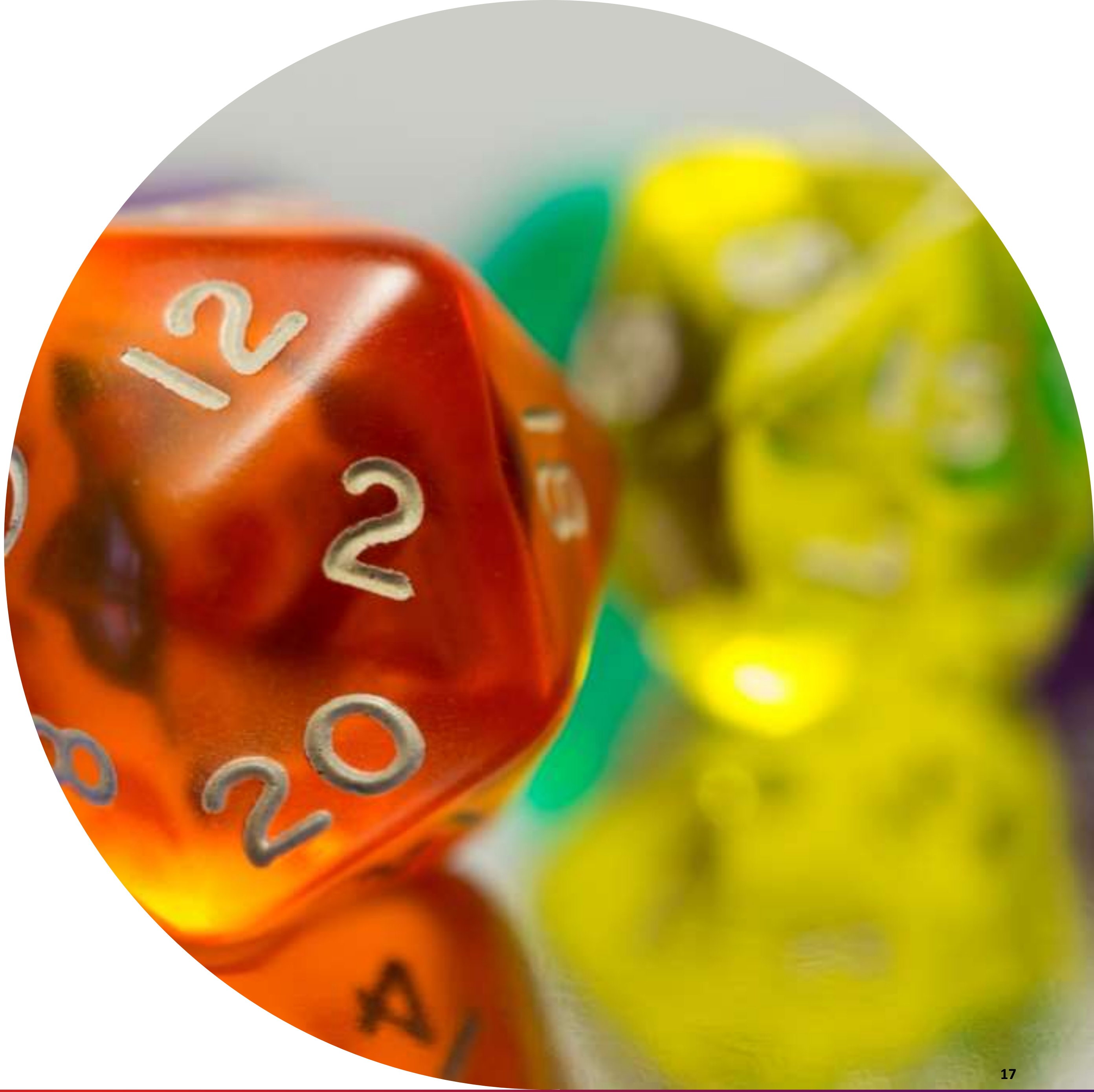
Given the NLL cost function

$$J(\boldsymbol{\theta}) = NLL(\boldsymbol{\theta}) = -\sum_{i=1}^{N} y_i \log h_\theta(\boldsymbol{x}_i) + (1 - y_i) \log\big(1 - h_\theta(\boldsymbol{x}_i)\big)$$

Show that the parameter update is given by $\Delta\theta_j = x_{ij}\varepsilon_i$

Multiclass classification with MLE

Multinomial logistic regression



Multinomial logistic regression

Problem: learn a conditional probability distribution for each class l

$$p(y = l|\boldsymbol{x};\boldsymbol{\theta}) = f_l(\boldsymbol{x};\boldsymbol{\theta})$$

Multinomial logistic regression:

$$p(y|\boldsymbol{x};\boldsymbol{\theta}) = \mathrm{Cat}(y|\boldsymbol{f}(\boldsymbol{x};\boldsymbol{\theta}))$$

with

$$\boldsymbol{f}(\boldsymbol{x};\boldsymbol{\theta}) = \boldsymbol{\mathcal{S}}(\mathbf{W}\boldsymbol{x} + \boldsymbol{b})$$

That is

$$p(y = l|\boldsymbol{x};\boldsymbol{\theta}) = f_l = \mu_l(\mathbf{W}\boldsymbol{x} + \boldsymbol{b})$$

with $\mathbf{W} \subset \mathbb{R}^C \times \mathbb{R}^D$ and $\boldsymbol{\theta} = [\boldsymbol{W};\boldsymbol{b}]$

The **softmax function** is defined as

$$\boldsymbol{\mathcal{S}}: \mathbb{R}^C \to [0,1]^C$$

$$\boldsymbol{\mathcal{S}}(\boldsymbol{a}) \triangleq [\mu_1(\boldsymbol{a}) \quad \cdots \quad \mu_j(\boldsymbol{a}) \quad \cdots \quad \mu_C(\boldsymbol{a})]$$

where

$$\mu_j: \mathbb{R}^C \to [0,1]$$

$$\mu_j(\boldsymbol{a}) = \frac{\exp(-a_j)}{\sum_{l=1}^{C} \exp(-a_l)}$$

$\boldsymbol{a}$ values are called **logits**

Exercise 9

Based on the NLL loss function derived in exercise 5,

$$NLL(\theta) = -\frac{1}{N}\sum_{i=1}^{N}\sum_{l=1}^{C} y_{il}\log f_{il}$$

use the softmax hypothesis

$$\boldsymbol{f}(\boldsymbol{x};\boldsymbol{\theta}) = \boldsymbol{\mathcal{S}}(\mathbf{W}\boldsymbol{x} + \boldsymbol{b})$$

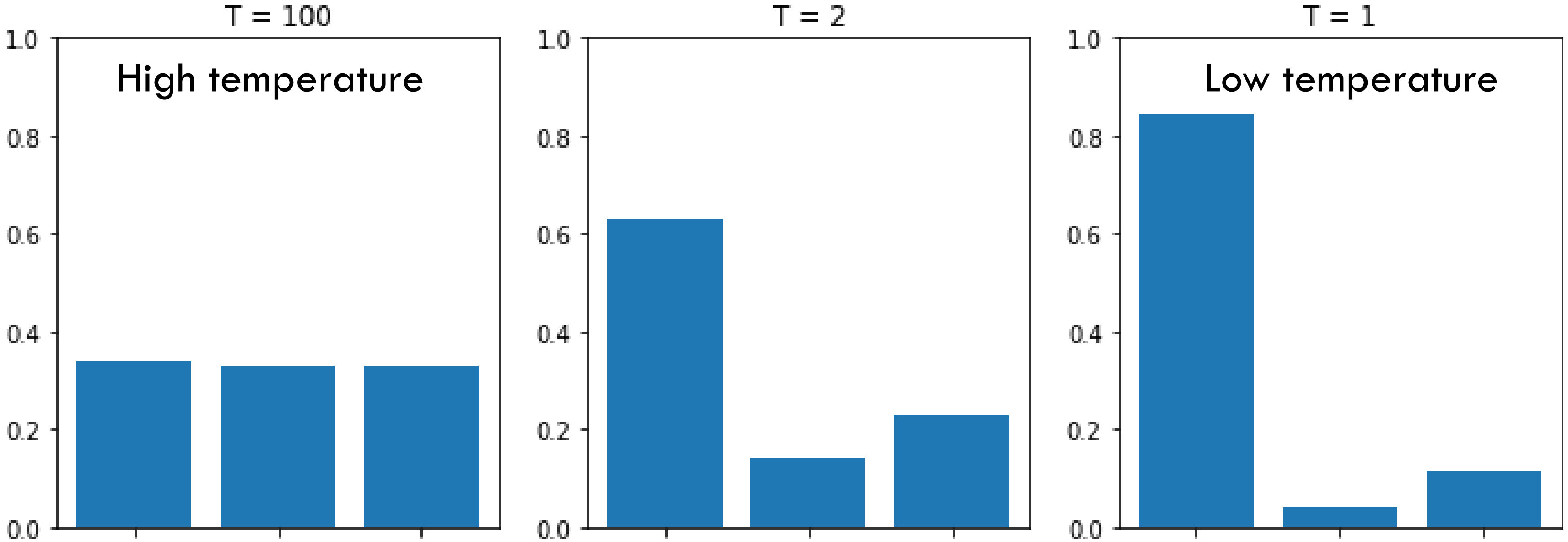
to write the cost function for multinomial logistic regression.

Softmax function with temperature

High temperature → uniform distribution
Low temperature → true max selection

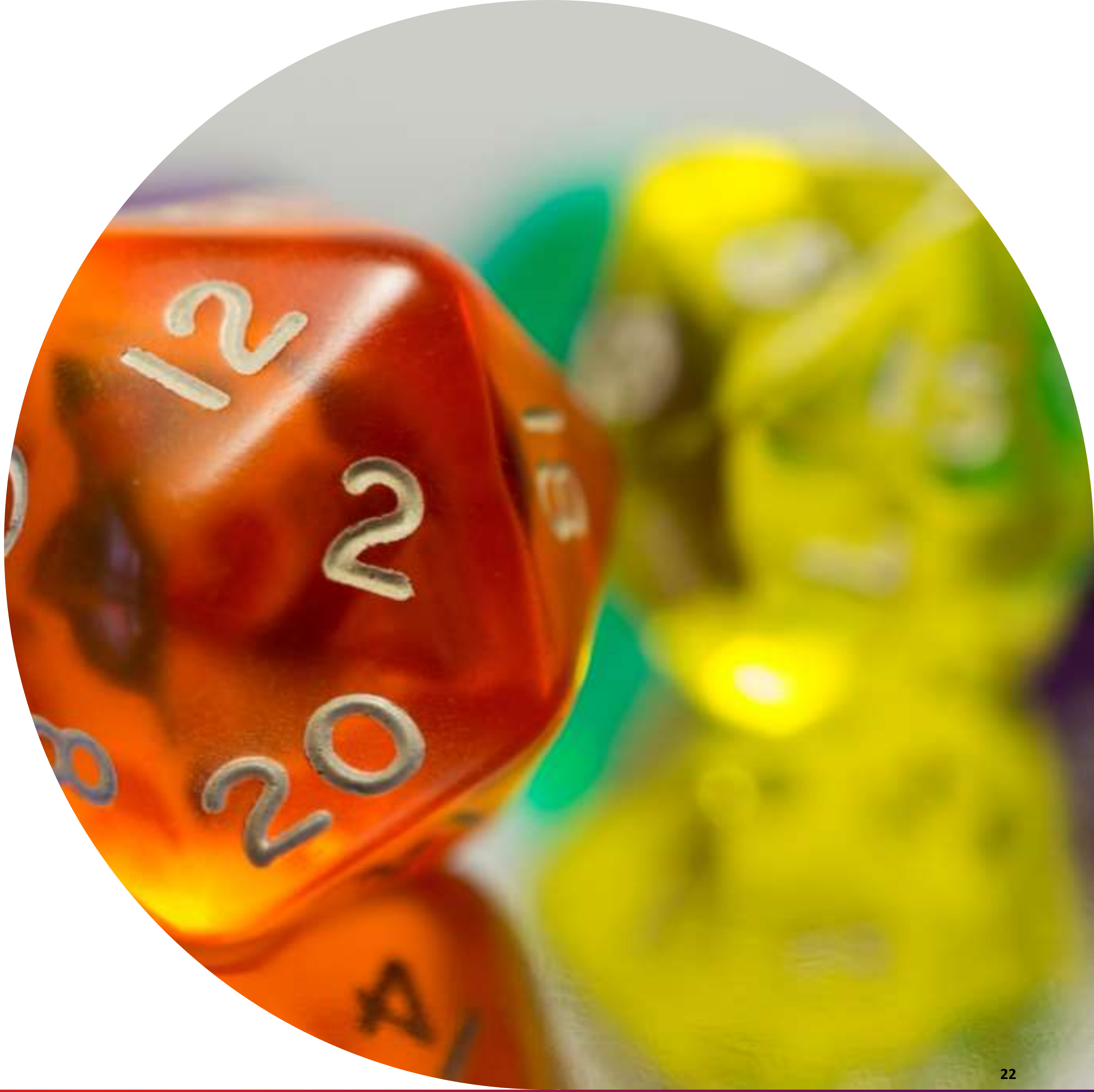
As $T \to 0$

$$\mathcal{S}(\boldsymbol{a}/T)_c = \begin{cases} 1.0 & \text{if } c = \operatorname{argmax}_{c'} a_{c'} \\ 0.0 & \text{otherwise} \end{cases}$$



ProbML Section 2.5, Figure 2.12

Multiclass classification with MLE
GD update



Multiclass classification with MLE
GD update

NLL cost function:

$$J(\theta) = \frac{1}{N} NLL(\theta) = -\frac{1}{N} \sum_{n=1}^{N} \sum_{c=1}^{C} y_{nc} \log \mu_{nc}$$

with $\mu_{nc} = \mu_{nc}(\boldsymbol{a}) = \frac{\exp(-a_{nc})}{\sum_{\mathrm{j}=1}^{C} \exp(-a_{nj})}$ and $\boldsymbol{a} = \boldsymbol{W}\boldsymbol{x}$ (assume the $\boldsymbol{b}$ term to be included in W)

Gradient descent update:

$$\Delta\theta_j \triangleq \frac{\partial}{\partial \theta_j} J(\theta)?$$

Exercise 10

We are going to show that the parameter update $\Delta\theta_j = \frac{\partial J}{\partial \theta_j}$ is AGAIN given by $\Delta\theta_j = x_{ij}\varepsilon_i$.

We will assume the weight matrix $\boldsymbol{D} \times \boldsymbol{C}$ is flattened into a $\boldsymbol{CD}$ vector.

1. First, use the chain rule to decompose the derivative as follows

$$\nabla_{\boldsymbol{w}_j} NLL_n = \sum_c \frac{\partial NLL_n}{\partial \mu_{nc}} \frac{\partial \mu_{nc}}{\partial a_{nj}} \frac{\partial a_{nj}}{\partial \boldsymbol{w}_j}$$

   where $\boldsymbol{w}_j$ denotes the vector of weights associated with class j

2. Then compute the partial derivatives.
   a) (optional) It can be derived that for any sample, $\frac{\partial \mu_c}{\partial a_j} = \mu_c\left(\delta_{cj} = \mu_j\right)$, where $\delta_{cj} = \mathbb{I}(c = j\ )$.
   b) Show that $\frac{\partial NLL_n}{\partial \mu_{nc}} = -\frac{y_{nc}}{\mu_{nc}}$
   c) Show that $\frac{\partial a_{nj}}{\partial \boldsymbol{w}_j} = \boldsymbol{x}_n$

3. Use these three partial derivatives to show that

$$\nabla_{\boldsymbol{w}_j} NLL_n = \left(\mu_{nj} - y_{nj}\right)\boldsymbol{x}_n$$

Empirical risk minimization

Empirical risk minimization cost function

Average **loss** of the predictive model on the training set

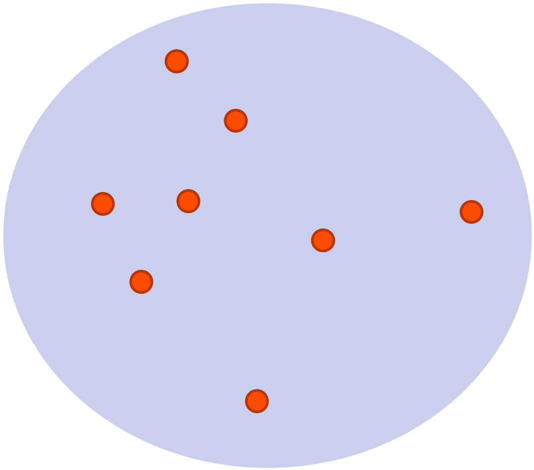
$$\mathcal{L}(\theta) \stackrel{\text{def}}{=} \frac{1}{N} \sum_{n=1}^{N} \ell(\boldsymbol{y}_n, \boldsymbol{\theta}; \boldsymbol{x}_n)$$

Empirical distribution of the dataset $\mathcal{D}$ with samples $\boldsymbol{y}_n \sim p(\boldsymbol{Y})$:

$$p_{\mathcal{D}}(\boldsymbol{y}_n) \stackrel{\text{def}}{=} \frac{1}{N} \sum_{n=1}^{N} \delta(\boldsymbol{y} - \boldsymbol{y}_n)$$

i. e., δ-functions centered at each sample $\boldsymbol{y}_n$:

Data distribution $p(X, Y)$



Dataset samples $(\boldsymbol{x}_n, \boldsymbol{y}_n) \in \mathcal{D}$

Samples are drawn from data distribution $\boldsymbol{x}_n, \boldsymbol{y}_n \sim p(\mathcal{X}, \mathcal{Y})$

Empirical risk minimization

Example using least-squares error

Cost = Mean squared error

$$\mathcal{L}(\theta) \stackrel{\text{def}}{=} \frac{1}{N}\sum_{n=1}^{N}(\boldsymbol{y}_n - f(\boldsymbol{x}_n;\theta))^2$$

Here the loss is the squared error:

$$\ell(\boldsymbol{y}_n, \boldsymbol{\theta}; \boldsymbol{x}_n) = (\boldsymbol{y}_n - f(\boldsymbol{x}_n;\theta))^2$$

Empirical risk minimization
Example using misclassification rate

Cost = Misclassification rate

$$\mathcal{L}(\theta) \stackrel{\text{def}}{=} \frac{1}{N}\sum_{n=1}^{N} \underbrace{\mathbb{I}(\boldsymbol{y}_n \neq f(\boldsymbol{x}_n; \theta))}$$

Counts the number of misclassified samples

$$\mathbb{I}(e) = \begin{cases} 1 & \text{if } e \text{ is true} \\ 0 & \text{if } e \text{ is false} \end{cases}$$

Here the loss is the Indicator function

$$\ell(\boldsymbol{y}_n, \boldsymbol{\theta}; \boldsymbol{x}_n) = \mathbb{I}(\boldsymbol{y}_n \neq f(\boldsymbol{x}_n; \theta))$$

Exercise 11

MLE is equivalent to empirical risk minimization (under a particular loss)

Looking back to the negative log likelihood cost function derived in exercise 1, compare it to the empirical risk cost function and identify the loss $\ell(\boldsymbol{y}_n, \boldsymbol{\theta}; \boldsymbol{x}_n)$ that makes both cost functions equivalent.

Summary

	Hypothesis $p(y\|\boldsymbol{x};\boldsymbol{\theta}) = p(y\|h_\theta(\boldsymbol{x}))$	Error derivative $\varepsilon_i = h_\theta(\boldsymbol{x_i}) - y_i$ $\frac{\partial}{\partial\theta_j}\varepsilon_i = \frac{\partial}{\partial\theta_j}h_\theta(\boldsymbol{x_i})$	Loss	Parameter update for a single sample $\boldsymbol{x_i}$ $\boldsymbol{\nabla}_{\theta_j}\mathrm{NLL}(\boldsymbol{\theta})$
Linear regression	$\mathcal{N}(y\|\boldsymbol{\theta}^T\boldsymbol{x}, \sigma^2)$	x_{ij}	MSE	$\Delta\theta_j = \varepsilon_i x_{ij}$
Logistic regression	$\mathrm{Ber}\big(\mathrm{y}\|\sigma(\boldsymbol{\theta}^T\boldsymbol{x})\big)$ $\sigma(z) = \frac{1}{1+\mathrm{e}^{-z}}$	$\sigma'(x_{ij})x_{ij}$	Cross-entropy	$\Delta\theta_j = \varepsilon_i x_{ij}$
Multinomial logistic regression	$\mathrm{Cat}\big(y\|\boldsymbol{\mathcal{S}}(\boldsymbol{\theta}^T\boldsymbol{x})\big)$ $p(y=k\|x;\theta) = [\boldsymbol{\mathcal{S}}(\boldsymbol{\theta}^T\boldsymbol{x})]_k$ $[\boldsymbol{\mathcal{S}}(\boldsymbol{a})]_j = \mu_j(\boldsymbol{a}) = \frac{\exp(-\mathrm{a}_j)}{\sum_{\mathrm{k}=1}^{C}\exp(-a_k)}$	$\mu_l\big(\delta_{lj} - \mu_j\big)x_{ij}$ $= \begin{cases} 0 & if\ j = l \\ -\mu_l\mu_j x_{ij} & if\ j \neq l \end{cases}$	Cross-entropy	$\Delta\boldsymbol{\theta}_j = \varepsilon_i\boldsymbol{x_i}$