CS 480/680

1/29/26

Lecture 7

Gradient Desant

Decision Tree

HW3 = out

Tomorrow 9:30, 10:30
Tutorial : matplotlib

Lecture II: Linear regression and classification. Loss functions

Marina Meilă
mmp@uwaterloo.ca

With Thanks to Pascal Poupart & Gautam Kamath
Cheriton School of Computer Science
University of Waterloo

January 12, 2026

Predictors

- K-Nearest-Neighbor
- Linear – for regression
  – for classification
- logistic regression
- Perceptron, LDA
- Decision Trees (CART)

Algorithms

LS Regression

Logistic Regression by
Gradient Ascent/Descent

Concepts

- Decision Region, Dec. Boundary

Training error, Test error,
Expected error

Variance, Bias

Loss functions – training/
test/expected loss

Max Likelihood

Linear predictors generalities ✓

Loss functions ✓

Least squares linear regression ✓

- Linear regresssion as minimizing L_{LS}
- Linear regresssion as maximizing likelihood
- Linear Discriminant Analysis (LDA)
- QDA (Quadratic Discriminant Analysis)
- Logistic Regression ←
- The PERCEPTRON algorithm

Decision trees

Reading HTF Ch.: 2.1–5,2.9, 7.1–4 bias-variance tradeoff, Murphy Ch.: 1., 8.6[1], Bach Ch.:

[1] Neither textbook is close to these notes except in a few places; take them as alternative perspectives or related reading

Logistic Regression

Fitting a linear predictor for classification, another approach.
Let $f(x) = \beta^T x$ model the **log odds** of class 1

$$f(X) \;=\; \frac{P(Y=1|X)}{P(Y=-1|X)} \tag{31}$$

Then

- $\hat{y} = 1$ iff $P(Y=1|X) > P(Y=-1|X)$
  - just like in the previous case! so what's the difference?
  - Answer: We don't assume each class is Gaussian, so we are in a more general situation than LDA
- What is $p(x) = P(Y=1|X=x)$ under our linear model?

$$\ln\frac{p}{1-p} \;=\; f, \quad \frac{p}{1-p} \;=\; e^f, \quad p \;=\; \frac{e^f}{1+e^f} \quad 1-p \;=\; \frac{1}{1+e^f} \tag{32}$$

An alternative "symmetric" expression for $p, 1-p$ is

$$p \;=\; \frac{e^{f/2}}{e^{f/2}+e^{-f/2}}, \quad 1-p \;=\; \frac{e^{-f/2}}{e^{f/2}+e^{-f/2}}. \tag{33}$$

Estimating the parameters by Max Likelihood

- Denote $y_* = (1-y)/2 \in \{0,1\}$
- The likelihood of a data point is $P_{Y|X}(y|x) = \frac{e^{y_* f(x)}}{1+e^{f(x)}}$
- The log-likelihood is $l(\beta; x) = y_* f(x) - \ln(1+e^{f(x)})$
- $\frac{\partial l}{\partial f} = y_* - \frac{e^f}{1+e^f} = y_* - \frac{1}{1+e^{-f}}$
  This is a scalar, and $\mathrm{sgn}\frac{\partial l}{\partial f} = y$
- We have also $\frac{\partial f(x)}{\partial \beta} = x$
- Now, the gradient of l w.r.t the parameter vector β is

$$\frac{\partial l}{\partial \beta} = \frac{\partial l}{\partial f}\frac{\partial f}{\partial \beta} = \left(y_* - \frac{1}{1+e^{-f(x)}}\right)x \tag{34}$$

  Interpretation: The infinitezimal change of β to increase log-likelihood for a single data point is along the direction of x, with the sign of y
- Log-likelihood of the data set $\mathcal{D}$

$$l(\beta; \mathcal{D}) = \frac{1}{N}\sum_{i=1}^{d} l(\beta; (x^i, y^i)) \tag{35}$$

- The optimal β maximizes $l(\beta; \mathcal{D})$ and therefore

$$\frac{\partial l(\beta; \mathcal{D})}{\partial \beta} = \frac{1}{N}\sum_{i=1}^{d}\left(y_*^i - \frac{1}{1+e^{-f(x^i)}}\right)x^i = 0 \tag{36}$$

- Unfortunately, (36) does not have a closed form solution!
  We maximize the (log)likelihood by iterative methods (e.g. gradient ascent) to obtain the β of the classifier.

2. Likelihood

TRAINING

Model $f(x) = \beta^T x$

Predict $\hat{y}(x) = 1$ iff $f(x) \geq 0$

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

$$L(\beta) = \prod_{i=1}^{n} P[y_*^i \mid x^i]$$

$$\ell(\beta) = \sum_{i=1}^{n} \ln P[y_*^i \mid x^i] = \sum_{i=1}^{n} \left[y_*^i f(x^i) - \ln\left(1 + e^{f(x^i)}\right) \right]$$

STAT

Calculus + Opt.

$$\underset{\beta}{\arg\max}\, L(\beta) = \hat{\beta}$$

$\beta \in \mathbb{R}^d$

$x \in \mathbb{R}^d$

3. $\nabla \ell \equiv \partial \ell / \partial \beta$

$\nabla(a^T z) = a$

$n = 1 \qquad (x, y) : \ \ell = \boxed{y_* f} - \boxed{\ln(1 + e^f)}$

$$\mathbb{R}^d \ni \frac{\partial \ell}{\partial \beta} = \underbrace{\frac{\partial \ell}{\partial f}}_{\mathbb{R}} \cdot \underbrace{\frac{\partial f}{\partial \beta}}_{\mathbb{R}^d}$$

$$\frac{\partial \ell}{\partial f} = y_* - \frac{e^f}{\underbrace{1 + e^f}_{p = P[y = 1 \mid x]}}$$

$$\Rightarrow \frac{\partial \ell}{\partial \beta} = \nabla_\beta \ell = \underbrace{\left(y_* - \frac{e^f}{1 + e^f} \right)}_{w \in \mathbb{R}} x$$

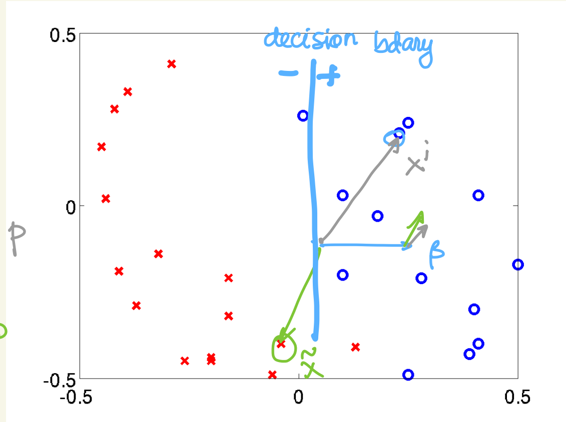
$y_* = 1 \Rightarrow w = 1 - p$

$y_* = 0 \Rightarrow w = -p$

$$\frac{\partial f}{\partial \beta} \equiv \nabla_\beta f = \nabla_\beta (x^T \beta) = x$$

$$n > 1 \qquad \nabla_\beta \ell = \sum_{i=1}^{n} \underbrace{\left(y_*^i - \frac{e^{f(x^i)}}{1 + e^{f(x^i)}} \right)}_{w_i} x^i$$

wanted $= 0$



Training by Gradient Descent

$$\ell(\beta)=\frac{1}{n}\sum_{i=1}^{n}\left[y_*^i f(x^i)-\ln\left(1+e^{f(x^i)}\right)\right]$$ concave ∩

$$\nabla\ell(\beta)=\frac{1}{n}\sum_{i=1}^{n}\Bigg[\underbrace{y_*^i-\frac{e^{f(x^i)}}{1+e^{f(x^i)}}}_{\pm w_i}\Bigg]x^i \leftarrow \max_\beta$$

Loss

$$L^{train}_{logit}(\beta)=-\ell(\beta) \leftarrow \min_\beta$$ convex ∪ by Grad. Descent

$$\nabla L^{train}_{logit}=-\nabla\ell(\beta)$$

Grad. Descent Algo

In $\mathcal{D}, \eta, tol$

Init $\beta^0=0$

Do for $t=1,2,\dots T$

- calculate $L(\beta^{t-1}), \nabla L(\beta^{t-1})$
- $\beta^t \leftarrow \beta^{t-1}-\boxed{\eta}\nabla L(\beta^{t-1})$

until $\dfrac{|L(\beta^{t-1})-L(\beta^t)|}{L(\beta^{t-1})} < tol$

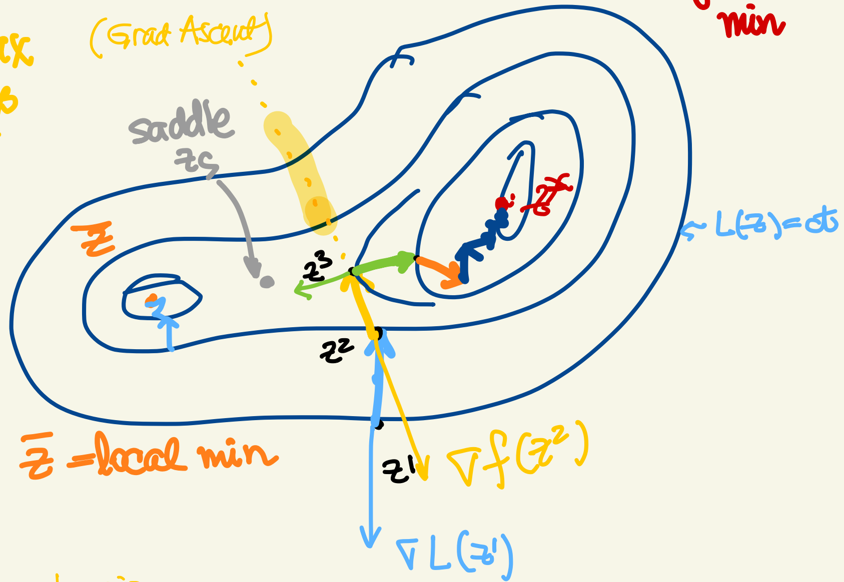
$tol = 10^{-3}, \dots 10^{-8}$

η = step size

no decrease in L

want $\operatorname{argmin} L(z)$

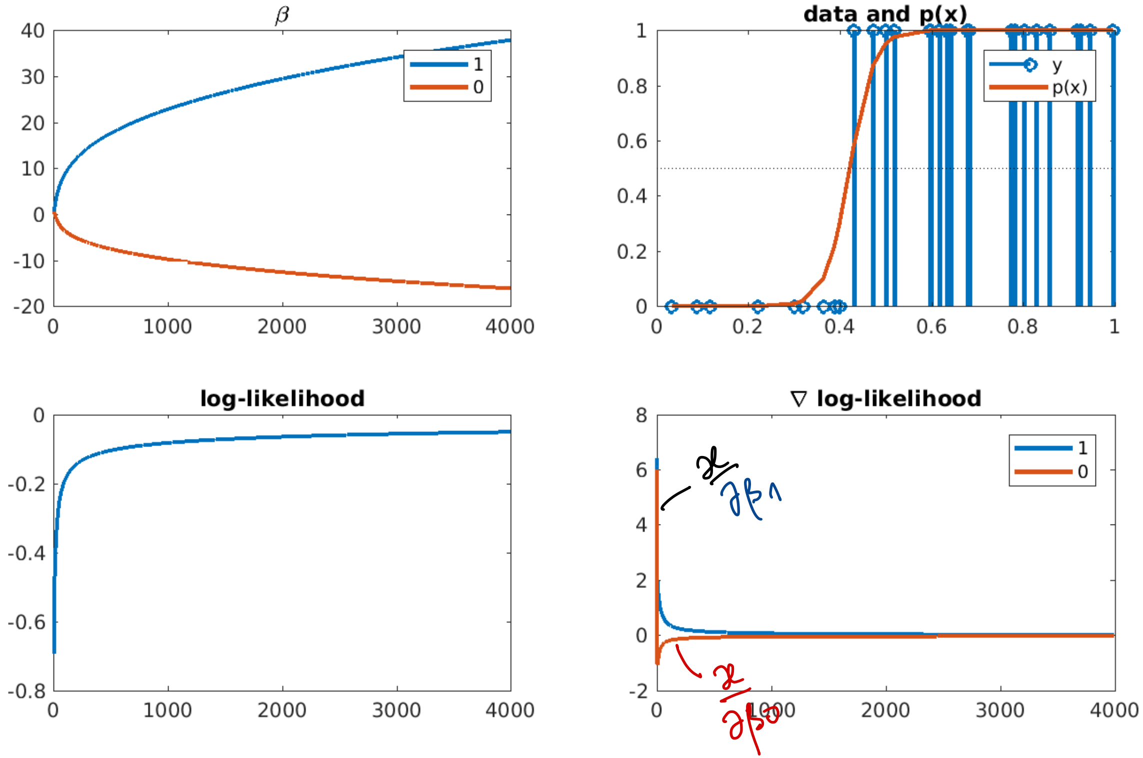
z^* = global min



$\bar{z}$ = local min

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

$$f(x) = \beta_1 x + \beta_0$$



Lecture III: Classification and Decision Trees (CART)

Marina Meilă
mmp@uwaterloo.ca

With Thanks to Pascal Poupart & Gautam Kamath
Cheriton School of Computer Science
University of Waterloo

January 27, 2026

Classification and regression tree(s) (CART)

Learnin a CART

Predicting with a CART

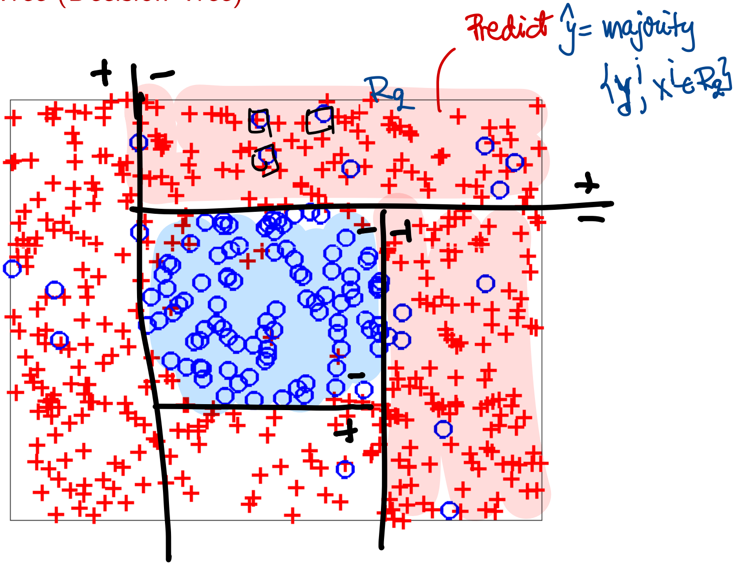
Some issues with CART

Reading HTF Ch.: 9.2 CART, Murphy Ch.: 16.2.1–4 CART, Bach Ch.:

Classification and regression trees (CART)

- A **classification tree** or (**decision tree**) is built recursively by splitting the data with hyperplanes parallel to the coordinate axes.
  - At each split, try to separate $+$ examples from $-$ examples as well as possible.
  - Eventually, all the regions will be "pure", i.e. will contain examples from one class only.
- Classification trees can be used in multiway classification as well (there we try to create a pure region on at least one side of the split)
- A **regression tree** uses the same principle for regression
  here we try to obtain regions where the outputs are nearly the same

Classification Tree (Decision Tree)

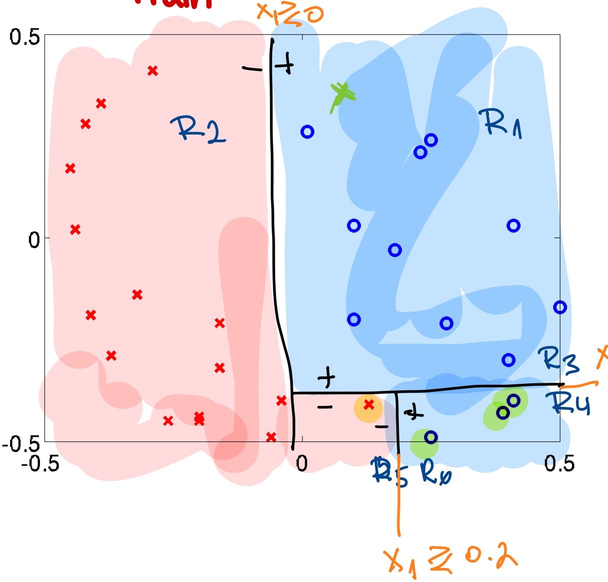


- Can approximate any decision regions
- Can classify correctly any $\mathcal{D}$ $\Rightarrow$ can overfit (variance)

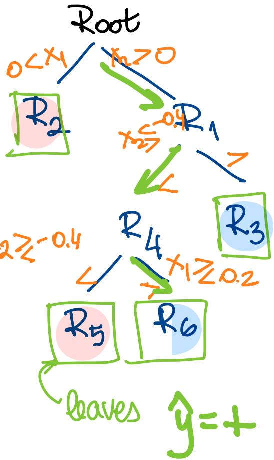
Classification Tree (Decision Tree)

Recursive
Greedy

Train



Prediction

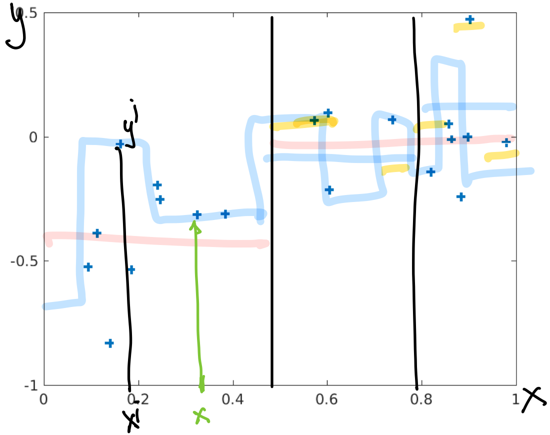


Regression Tree

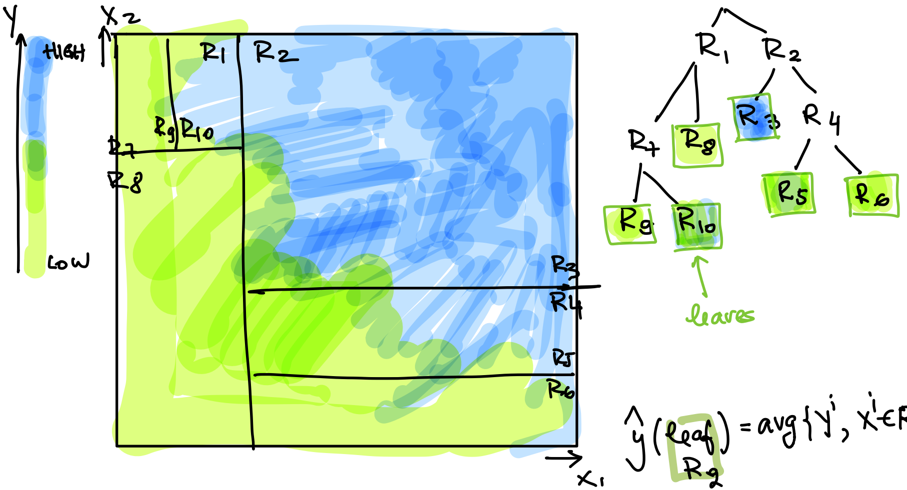
Predict

$\hat{y} = \text{avg}\{y^i, x^i \in R_2\}$

leaf $\ni x$



Regression Tree



$$\hat{y}(\underbrace{\boxed{\text{leaf}}}_{R_2}) = \text{avg}\{y^i, x^i \in R_2\}$$

Hierarchical partitions

- a **hierarchical partition** $\mathcal{T}$ of $\mathbb{R}^d$ is a set of regions $\{R_q\}$, so that $\mathbb{R}^d = \bigcup_q R_q$ and between any two $R_q, R_{q'}$ we have either

$$R_q \cap R_{q'} = \emptyset, \text{ or } R_q \subset R_{q'} \text{ or } R_{q'} \subset R_q. \tag{1}$$

  (we include the boundariy between 2 regions $R_q, R_{q'}$ arbitrarily in a single one of them)
- In a CART, the partitions are usually chosen to be axis-aligned, i.e. $R_q = \{x \,|\, \pm x_{j_1} " > " \tau_1, \pm x_{j_2} " > " \tau_2, \ldots \pm x_{j_l} " > " \tau_l\}$ where $" > "$ stands for one of $>$ or $\geq$, so that the union of all regions covers $\mathbb{R}^d$.
- The number of inequalities l defining the region is called the *level* of the region.
- R_q is a **leaf** of $\mathcal{T}$ iff there is no other $R_{q'}$ included in it.

Example (**A hierarchical partition with 3 levels over $\mathbb{R}^2$**)

Level 1: $R_1 = \{x \,|\, x_2 > 3\}$,
$R_2 = \{x \,|\, x_2 \leq 3\}$

Level 2: $R_3 = \{x \,|\, x_2 > 3,\ x_1 \geq -2\}$,
$R_4 = \{x \,|\, x_2 > 3,\ x_1 < -2\}$,
$R_5 = \{x \,|\, x_2 \leq 3, x_1 > 0\}$,
$R_6 = \{x \,|\, x_2 \leq 3, x_1 \leq 0\}$

Level 3: $R_7 = \{x \,|\, x_2 > 3,\ x_1 \geq -2, x_1 < 4\}$,
$R_8 = \{x \,|\, x_2 > 3,\ x_1 \geq 4\}$,
$R_9 = \{x \,|\, x_2 < 3, x_1 \geq 1\}$
$R_{10} = \{x \,|\, x_2 \leq 3, x_1 \leq 0, x_2 > -1\}$,
$R_{11} = \{x \,|\, x_2 \leq -1, x_1 \leq 0\}$,
$R_{12} = \{x \,|\, x_2 < 3, x_1 > 0, x_1 < 1\}$

The leaves are $R_4, R_7, \ldots R_{12}$. Not all leaves are at the same level; for example R_4 is at level 2.