Foundations of Machine Learning
AI2000 and AI5000

FoML-16
Least Squares for Regression

Dr. Konda Reddy Mopuri
Department of AI, IIT Hyderabad
July-Nov 2025

భారతీయ సాంకేతిక విజ్ఞాన సంస్థ హైదరాబాద్
भारतीय प्रौद्योगिकी संस्थान हैदराबाद
Indian Institute of Technology Hyderabad

DiL
Data-driven Intelligence
& Learning Lab

So far in FoML

- Intro to ML and Probability refresher
- MLE, MAP, and fully Bayesian treatment
- Supervised learning
    a. Linear Regression with basis functions (regularization, model selection)
    b. Bias-Variance Decomposition (Bayesian Regression)
    c. Decision Theory - three broad classification strategies
        - Probabilistic Generative Models - Continuous & discrete data
        - Discriminant Functions

Least Squares for Classification

Least Squares for Classification

- Consider K classes
- Each class 'k' has its own linear model $y_k(\mathbf{x}) = w_k^T \mathbf{x} + w_{k0}$

Least Squares for Classification

targets one-hot encoded

$$T = \begin{bmatrix} - t_1^T - \\ \vdots \\ - t_N^T - \end{bmatrix}_{N \times K}$$

$$t_n = (0\,0 \cdots 1\,0\,0)^T_{1 \times K}$$

- Shorter notation $y(\mathbf{x}) = \widetilde{\mathbf{W}}^T \tilde{\mathbf{x}}$

$$\widetilde{\mathbf{W}} = \left(\begin{bmatrix} w_{10} - w_1 - \\ \vdots \\ w_{K0} - w_K - \end{bmatrix}_{K \times M}\right)^T$$

Assign x to C_k, where

$$\tilde{\mathbf{x}} = (1, \underline{x})^T_{1 \times m} \qquad X = \begin{bmatrix} \tilde{x}_1 \\ \vdots \\ \tilde{x}_N \end{bmatrix}_{N \times M}$$

$$k = \underset{j}{\operatorname{argmax}}\ y_j(\underline{x})$$

$$y(\mathbf{x}) = \begin{pmatrix} y_1(\underline{x}) \\ \vdots \\ y_K(\underline{x}) \end{pmatrix}_{K \times 1}$$

on single sample

$$E(\underline{x}_n) = \| y(\underline{x}_n) - t_n \|_2^2$$

Least Squares for Classification

- Data matrix $X_{N \times M}$ [row is a sample]
- Target matrix $T_{N \times K}$

parameter matrix $\widetilde{W}_{M \times K}$ [column is per class discriminant]

Use regression (sum of squares) error function

$$E_D(\widetilde{\mathbf{W}}) = \frac{1}{2} \sum_{n=1}^{N} \sum_{k=1}^{K} \sum_{m=1}^{M} \left(t_{nk} - x_{nm} \widetilde{w}_{mk} \right)^2$$

Least Squares for Classification

The error function can be conveniently written as

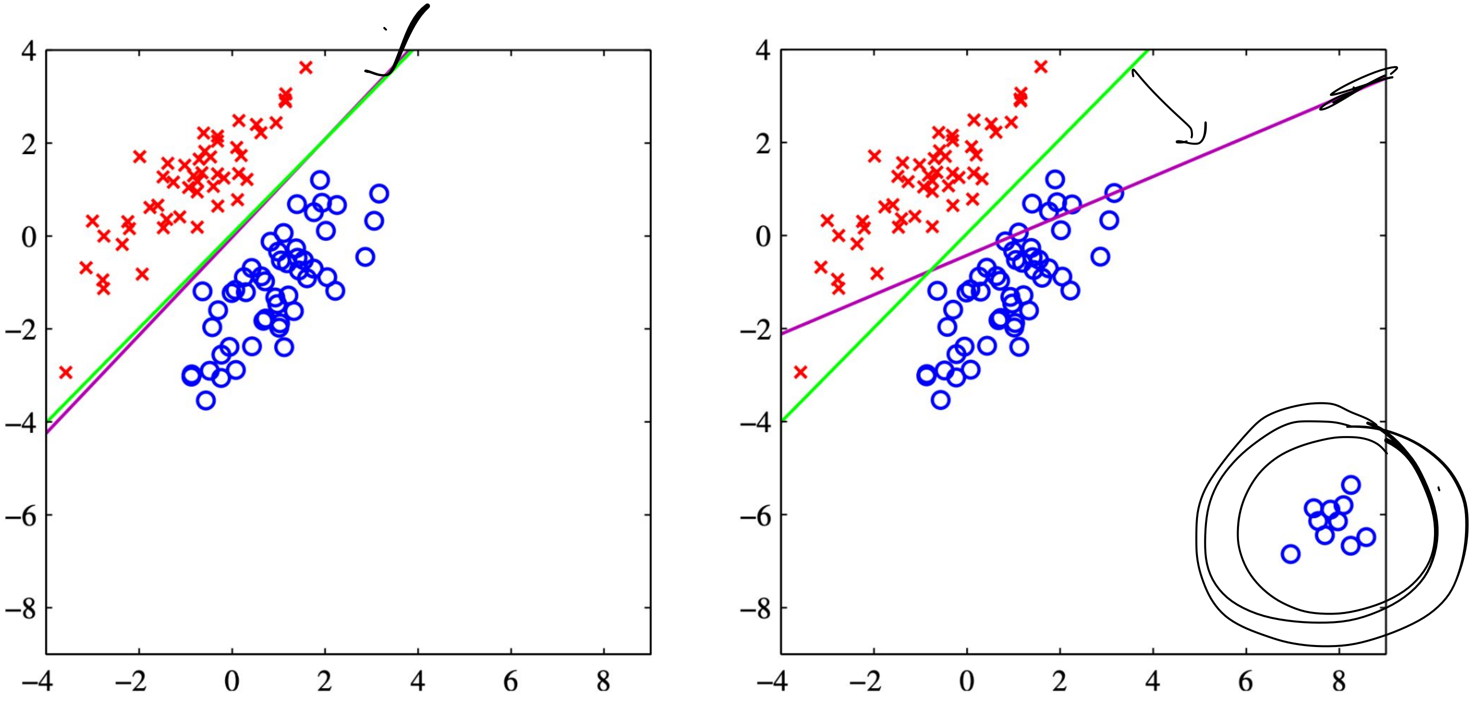
$$E_D(\widetilde{\mathbf{W}}) = \frac{1}{2}\mathrm{Tr}\left\{(\widetilde{\mathbf{X}}\widetilde{\mathbf{W}} - \mathbf{T})^{\mathrm{T}}(\widetilde{\mathbf{X}}\widetilde{\mathbf{W}} - \mathbf{T})\right\}$$

Minimize $E_D(\widetilde{\mathbf{W}})$ as a function of $\widetilde{\mathbf{w}}$: $(\tilde{X}^T\tilde{X})^{-1}\tilde{X}^T \cdot T$ $\quad p(C_k|x)$

$$\frac{\partial}{\partial \tilde{w}}(\cdot) = 0$$

the familiar pseudo inverse formulation

Least Squares Issues - Outliers

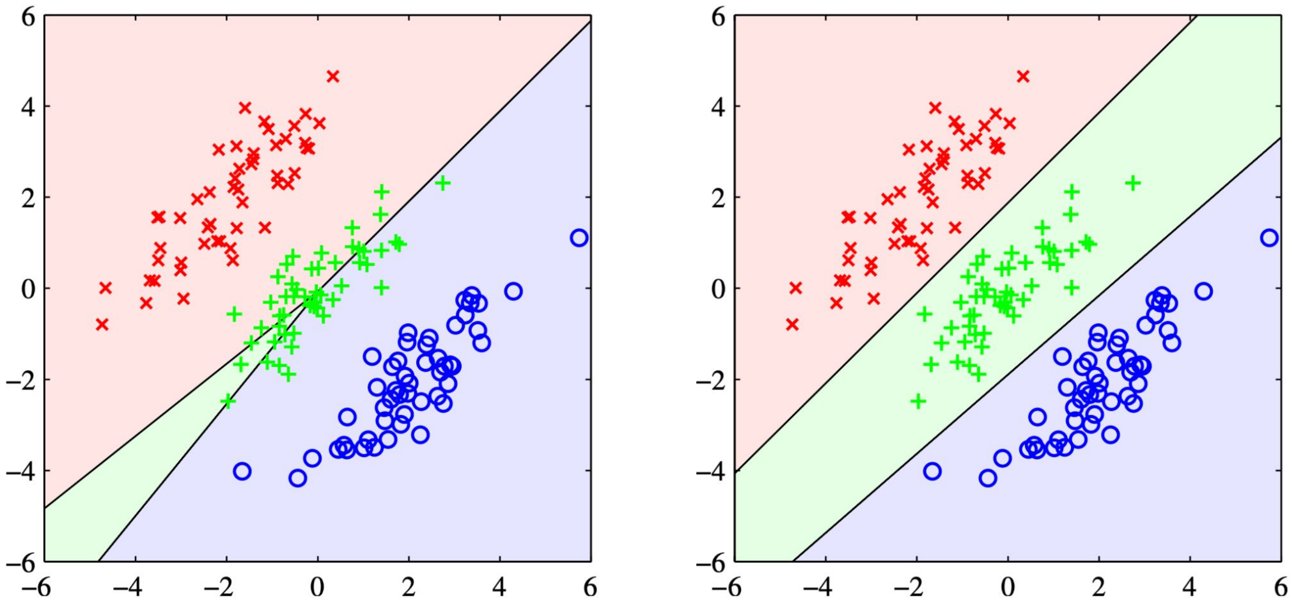


Magenta - LS classifier
Green - Logistic Regression classifier

Least Squares Issues - Masking

$[\cdot \quad - \quad \frac{1}{\uparrow} \quad \cdot \quad -]$

prediction for the correct class is expected to be close to 1



$\frac{y(x)}{\|w\|}$

Left - LS classifier
Right - Logistic Regression classifier

Least Squares Issues - Predictions ≠ Probabilities

$\mathbf{y}_{LS}(\mathbf{x})$ are not probabilities

If $a^T \underline{t_n} + b = 0$

$\Rightarrow a^T y(\underline{x_n}) + b = 0$

with

$\underline{a} = [1\ 1\ 1\ 1\ 1]$

$b = -1$

we know that

$\underline{a}^T \underline{t_n} + b = 0$

holds $\forall n$

this implies

$$\sum_{i=1}^{K} y(\underline{x_n})_i = 1 \quad \forall n$$

Rough

Next

The Perceptron