Review of Linear Models

Máster Universitario en Ciencia de Datos - Métodos Avanzados en Aprendizaje Automático

Carlos María Alaíz Gudín

Escuela Politécnica Superior
Universidad Autónoma de Madrid

Academic Year 2020/21

Universidad Autónoma de Madrid

Contents

Introduction to Regression

Supervised Learning - Regression (I)

Definition (Supervised Learning)

Supervised learning is the machine learning task of learning a function that maps an input to an output based on example input-output pairs.

Definition (Regression Problem)

A **regression problem** is a supervised learning problem where the outputs are continuous.

Examples (Regression Problems)

- Predicting the wind energy production at a certain hour using Numerical Weather Predictions.
- Predicting the weight of a person based on the height, age, gender, etc.
- Predicting the future price of a stock based on its current value, the value of related stocks, the current trends, etc.

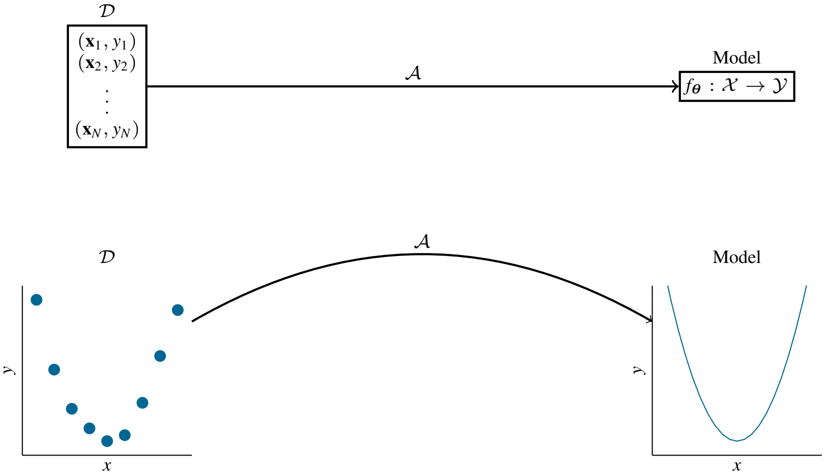
Supervised Learning - Regression (II)

Elements of a Supervised Learning Problem

Data	Set of input-output pairs, $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$.
Features	Vector of attributes (independent/input variables, covariates...), $\mathbf{x}_i \in \mathcal{X}$.
Target	Label (dependent variables, outcome...), $y_i \in \mathcal{Y}$.
Model	Mapping from the input to the output space, $f_{\boldsymbol{\theta}} : \mathcal{X} \to \mathcal{Y}$, with $\boldsymbol{\theta}$ the model parameters.
Learning Algorithm	Procedure to obtain a model based on the data, $\mathcal{A} : \mathcal{D} \to f_{\boldsymbol{\theta}}(\cdot)$.

- In a regression setting usually $\mathcal{Y} = \mathbb{R}$.
- In many situations, specially after preprocessing the data, $\mathcal{X} = \mathbb{R}^d$.

Illustration



Linear Models

- A simple model consists in defining the output as a linear combination of the inputs (**linear models**).

Advantages

1. Simple.
2. Robust (small variance).
3. Interpretable.
4. Easy to train.
5. Easy to predict.

Disadvantages

1. Limited flexibility.
2. Under-fitting (large bias).

Multiple Linear Regression

Linear Model

- For simplicity, $\mathcal{X} = \mathbb{R}^d$.
- The data becomes $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$, with $\mathbf{x}_i = (x_{i,1}, x_{i,2}, \cdots, x_{i,d}) \in \mathbb{R}^d$ and $y_i \in \mathbb{R}$.

- The corresponding linear model is a hyperplane, with parameters $\boldsymbol{\theta} = \{b, \mathbf{w}\}$.
  - $b \in \mathbb{R}$ is the intercept or bias term.
  - $\mathbf{w} = (w_1, w_2, \cdots, w_d) \in \mathbb{R}^d$ is the normal vector of the hyperplane.
  - The model is defined as:

$$f_{\boldsymbol{\theta}}(\mathbf{x}) = b + \mathbf{w}^\intercal \mathbf{x} = b + \sum_{i=1}^{d} w_i x_i.$$

- The **learning algorithm** will determine b and $\mathbf{w}$ using $\mathcal{D}$.

Linear Model - Exercise

Exercise

Given a 2-dimensional linear model with parameters $\boldsymbol{\theta} = \{b, \mathbf{w}\}$, with $b = 1$ and $\mathbf{w} = (1, 2)^\intercal$.

1. Compute the output of the model for $\mathbf{x} = (1, 1)^\intercal$.
2. Compute the output of the model for $\mathbf{x} = (-1, 0)^\intercal$.

Solution

1. $f_{\boldsymbol{\theta}}((1, 1)^\intercal) = 4$.
2. $f_{\boldsymbol{\theta}}((-1, 0)^\intercal) = 0$.

Notebook

Multiple Linear Regression: First Example

Linear Equations (I)

- A procedure is needed to determine the bias b and the vector $\mathbf{w}$.
- A first approach is to try to match all input-output pairs $(\mathbf{x}_i, y_i)$, $i = 1, \dots, N$. Specifically:

$$\begin{cases} b + \mathbf{w}^\intercal \mathbf{x}_1 = y_1 \\ b + \mathbf{w}^\intercal \mathbf{x}_2 = y_2 \\ \dots \\ b + \mathbf{w}^\intercal \mathbf{x}_N = y_N \end{cases} \equiv \begin{cases} b + w_1 x_{1,1} + w_2 x_{1,2} + \dots + w_d x_{1,d} = y_1 \\ b + w_1 x_{2,1} + w_2 x_{2,2} + \dots + w_d x_{2,d} = y_2 \\ \dots \\ b + w_1 x_{N,1} + w_2 x_{N,2} + \dots + w_d x_{N,d} = y_N \end{cases} .$$

- The following matrix notation can simplify the equations:

$$\mathbf{X} = \begin{pmatrix} x_{1,1} & x_{1,2} & \dots & x_{1,d} \\ x_{2,1} & x_{2,2} & \dots & x_{2,d} \\ \vdots & \vdots & \ddots & \vdots \\ x_{N,1} & x_{N,2} & \dots & x_{N,d} \end{pmatrix}; \quad \tilde{\mathbf{X}} = \begin{pmatrix} 1 & x_{1,1} & \dots & x_{1,d} \\ 1 & x_{2,1} & \dots & x_{2,d} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{N,1} & \dots & x_{N,d} \end{pmatrix}; \quad \mathbf{y} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{pmatrix}; \quad \tilde{\mathbf{w}} = \begin{pmatrix} b \\ w_1 \\ \vdots \\ w_d \end{pmatrix},$$

where $\mathbf{X} \in \mathbb{R}^{N \times d}$ is the data matrix, $\tilde{\mathbf{X}} \in \mathbb{R}^{N \times (d+1)}$ is the data matrix with a constant term, $\mathbf{y} \in \mathbb{R}^N$ is the vector of targets and $\tilde{\mathbf{w}} \in \mathbb{R}^{d+1}$ is the weight vector with intercept.

Linear Equations (II)

- The system of equations becomes:

$$\tilde{\mathbf{X}}\tilde{\mathbf{w}} = \mathbf{y}.$$

- Since $\tilde{\mathbf{X}} \in \mathbb{R}^{N \times (d+1)}$ and $\mathbf{y} \in \mathbb{R}^N$:
  - N equations.
  - $d + 1$ unknowns.
- Usually, $N \gg d + 1$ and the system is **overdetermined**.
- The inverse of $\tilde{\mathbf{X}}$ is not defined.

- The Moore-Penrose pseudo-inverse can be used instead, $\tilde{\mathbf{X}}^\dagger = \left(\tilde{\mathbf{X}}^\intercal \tilde{\mathbf{X}}\right)^{-1} \tilde{\mathbf{X}}^\intercal$.
- A **different approach** also justifies this method.

Quality of the Model

- A procedure is needed to determine the bias b and the vector $\mathbf{w}$.
- The solution is to optimize the **quality** of the model, probably not fitting exactly the training data.
- The quality of the model has to be defined. Usually from two points of view:

  Error An error term $\mathcal{E}_{\mathcal{D}}(\boldsymbol{\theta})$ measures how well the model fits the training data.

  Complexity A regularization term $\mathcal{R}(\boldsymbol{\theta})$ penalizes the complexity of the model.

Error Term for a Linear Model

Residual For the i-th pattern, $r_i = y_i - f_{\boldsymbol{\theta}}(\mathbf{x}_i) = y_i - (b + \mathbf{w}^\intercal \mathbf{x}_i)$.

Mean Squared Error $\mathrm{MSE}(b, \mathbf{w}) = \mathbb{E}\left[R^2\right] \approx \frac{1}{N} \sum_{i=1}^{N} (y_i - (b + \mathbf{w}^\intercal \mathbf{x}_i))^2$.

Mean Absolute Error $\mathrm{MAE}(b, \mathbf{w}) = \mathbb{E}[|R|] \approx \frac{1}{N} \sum_{i=1}^{N} |y_i - (b + \mathbf{w}\mathbf{x}_i)|$.

Quality of the Multidimensional Model - Exercise

Exercise

Given a 2-dimensional linear model with parameters $\boldsymbol{\theta} = \{b, \mathbf{w}\}$, with $b = 1$ and $\mathbf{w} = (1, 2)^\intercal$, and for the following data:

$x_{i,1}$	$x_{i,2}$	y_i
1	1	4
-1	0	2

1. Compute the Mean Absolute Error.
2. Compute the Mean Squared Error.

Solution

1. $\text{MAE}(b, \mathbf{w}) = 1$.
2. $\text{MSE}(b, \mathbf{w}) = 2$.

Training a Linear Model

- The most common choice for the error function is the MSE.
  - It is **differentiable**.
  - It corresponds to the **distance** between the vector of predictions and the vector of targets.
  - It is a natural choice when the observation noise is assumed to be **Gaussian**.

- The learning algorithm for training the linear model consists in solving the problem:

$$\min_{\substack{b\in\mathbb{R}\\ \mathbf{w}\in\mathbb{R}^d}} \{\mathrm{MSE}(b,\mathbf{w})\} = \min_{\substack{b\in\mathbb{R}\\ \mathbf{w}\in\mathbb{R}^d}} \left\{ \frac{1}{N}\sum_{i=1}^{N}\left(y_i - (b + \mathbf{w}^\intercal \mathbf{x}_i)\right)^2 \right\}.$$

- How is this problem solved?
  - It is **differentiable**: the optimum is characterized by the zeros of the gradient.
  - It is **convex**: there are no local minima.

Training a Linear Model - Optimization (I)

$$\min_{\substack{b\in\mathbb{R}\\ \mathbf{w}\in\mathbb{R}^d}} \{\mathrm{MSE}(b,\mathbf{w})\} = \min_{\substack{b\in\mathbb{R}\\ \mathbf{w}\in\mathbb{R}^d}} \left\{ \frac{1}{N}\sum_{i=1}^{N} (y_i - (b + \mathbf{w}\mathbf{x}_i))^2 \right\} \equiv \min_{\tilde{\mathbf{w}}\in\mathbb{R}^{d+1}} \left\{ \left(\mathbf{y} - \tilde{\mathbf{X}}\tilde{\mathbf{w}}\right)^\intercal \left(\mathbf{y} - \tilde{\mathbf{X}}\tilde{\mathbf{w}}\right) \right\}.$$

$$\begin{aligned}
\nabla_{\tilde{\mathbf{w}}} \mathrm{MSE}(\tilde{\mathbf{w}})\big|_{\tilde{\mathbf{w}}=\tilde{\mathbf{w}}^\star} = \mathbf{0} &\implies 2\tilde{\mathbf{X}}^\intercal \left(\mathbf{y} - \tilde{\mathbf{X}}\tilde{\mathbf{w}}^\star\right) = \mathbf{0} \\
&\implies \tilde{\mathbf{X}}^\intercal \mathbf{y} - \tilde{\mathbf{X}}^\intercal \tilde{\mathbf{X}}\tilde{\mathbf{w}}^\star = \mathbf{0} \\
&\implies \tilde{\mathbf{X}}^\intercal \tilde{\mathbf{X}}\tilde{\mathbf{w}}^\star = \tilde{\mathbf{X}}^\intercal \mathbf{y} \\
&\implies \boxed{\tilde{\mathbf{w}}^\star = \left(\tilde{\mathbf{X}}^\intercal \tilde{\mathbf{X}}\right)^{-1} \tilde{\mathbf{X}}^\intercal \mathbf{y} = \tilde{\mathbf{X}}^\dagger \mathbf{y}}.
\end{aligned}$$

Training a Linear Model - Optimization (II)

- In summary, the Least Squares Linear Model is the solution of the following problem:

$$\min_{\substack{b \in \mathbb{R} \\ \mathbf{w} \in \mathbb{R}^d}} \left\{ \frac{1}{N} \sum_{i=1}^{N} \left(y_i - (b + \mathbf{w}^\intercal \mathbf{x}_i)\right)^2 \right\}.$$

Least Squares Linear Model

$$\begin{pmatrix} b^\star \\ \mathbf{w}^\star \end{pmatrix} = \tilde{\mathbf{w}}^\star = \tilde{\mathbf{X}}^\dagger \mathbf{y} = \begin{bmatrix} \mathbf{1} & \mathbf{X} \end{bmatrix}^\dagger \mathbf{y}.$$

Notebook

Multiple Linear Regression: Optimization

Introduction to Classification

Supervised Learning - Classification (I)

Definition (Classification Problem)

A **classification problem** is a supervised learning problem where the outputs are discrete.

Examples (Classification Problems)

- Predicting if a patient has a certain disease or not depending on medical data.
- Predicting the type of object that appears in a picture.
- Distinguishing the type of fish captured using the data provided by several sensors.

Supervised Learning - Classification (II)

Elements of a Supervised Learning Problem

Data	Set of input-output pairs, $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$.
Features	Vector of attributes (independent/input variables, covariates...), $\mathbf{x}_i \in \mathcal{X}$.
Label	Target (dependent variables, outcome...), $y_i \in \mathcal{Y}$.
Model	Mapping from the input to the output space, $f_{\boldsymbol{\theta}} : \mathcal{X} \to \mathcal{Y}$, with $\boldsymbol{\theta}$ the model parameters.
Learning Algorithm	Procedure to obtain a model based on the data, $\mathcal{A} : \mathcal{D} \to f_{\boldsymbol{\theta}}(\cdot)$.

- In a classification setting $\mathcal{Y} = \{\mathcal{C}_1, \mathcal{C}_2, \cdots, \mathcal{C}_K\}$.
- In many situations, specially after preprocessing the data, $\mathcal{X} = \mathbb{R}^d$.
- The resultant model assigns to each input a certain class, $f_{\boldsymbol{\theta}} : \mathcal{X} \to \{\mathcal{C}_1, \mathcal{C}_2, \cdots, \mathcal{C}_K\}$.

Binary Classification and Linear Models

- Probably the most important case is $K = 2$ (**binary classification**).
  - If $K > 2$, there are encoding techniques to transform the problem into several binary subproblems.
- The classes are usually denoted as $\mathcal{C}_0$ and $\mathcal{C}_1$, and they are represented with a $0/1$ (or $-1/1$) encoding.
  - The labels are transformed to:

$$t_i = \begin{cases} 0 & \text{if } y_i = \mathcal{C}_0, \\ 1 & \text{if } y_i = \mathcal{C}_1. \end{cases}$$

- A simple model consists in defining the output as a linear combination of the inputs (**linear models**) plus a **transformation**.
  - Simple. Robust (small variance). Interpretable. Easy to train. Easy to predict.
  - Limited flexibility. Under-fitting (large bias).

Binary Linear Classification

Binary Linear Model

- For simplicity, $\mathcal{X} = \mathbb{R}^d$.
- The data becomes $\mathcal{D} = \{(\mathbf{x}_i, t_i)\}_{i=1}^N$, with $\mathbf{x}_i = (x_{i,1}, x_{i,2}, \cdots, x_{i,d}) \in \mathbb{R}^d$ and $t_i \in \{0, 1\}$.

- The corresponding linear model is a hyperplane, with parameters $\boldsymbol{\theta} = \{b, \mathbf{w}\}$.
  - $b \in \mathbb{R}$ is the intercept or bias term.
  - $\mathbf{w} = (w_1, w_2, \cdots, w_d) \in \mathbb{R}^d$ is the normal vector of the hyperplane.
  - The model is defined as:

$$f_{\boldsymbol{\theta}}(\mathbf{x}) = \begin{cases} 0 & \text{if } b + \mathbf{w}^\intercal \mathbf{x} < 0, \\ 1 & \text{if } b + \mathbf{w}^\intercal \mathbf{x} \geq 0. \end{cases}$$

  - The hyperplane divides the space into two halves, one for class $\mathcal{C}_0$ and the other for class $\mathcal{C}_1$.

- The **learning algorithm** will determine b and $\mathbf{w}$ using $\mathcal{D}$.

Binary Linear Model - Exercise

Exercise

Given a 2-dimensional binary linear classification model with parameters $\boldsymbol{\theta} = \{b, \mathbf{w}\}$, with $b = 1$ and $\mathbf{w} = (1, 2)^\intercal$.

1. Compute the output of the model for $\mathbf{x}_1 = (1, 1)^\intercal$.
2. Compute the output of the model for $\mathbf{x}_2 = (1, -2)^\intercal$.
3. Compute the output of the model for $\mathbf{x}_3 = (0, 0)^\intercal$.

Solution

1. $b + \mathbf{w}^\intercal \mathbf{x}_1 = 4 \implies f_{\boldsymbol{\theta}}(\mathbf{x}_1) = 1 \implies \mathcal{C}_1$.
2. $b + \mathbf{w}^\intercal \mathbf{x}_2 = -2 \implies f_{\boldsymbol{\theta}}(\mathbf{x}_2) = 0 \implies \mathcal{C}_0$.
3. $b + \mathbf{w}^\intercal \mathbf{x}_3 = 1 \implies f_{\boldsymbol{\theta}}(\mathbf{x}_3) = 1 \implies \mathcal{C}_1$.

Notebook

Binary Linear Classification: First Example

Quality of the Model

- A procedure is needed to determine the bias b and the hyperplane $\mathbf{w}$.
- The solution is to optimize the **quality** of the model.
- The quality of the model has to be defined. Usually from two points of view:

Fitness: A fitness term $\mathcal{F}_{\mathcal{D}}(\boldsymbol{\theta})$ measures how well the model fits the training data.

Complexity: A regularization term $\mathcal{R}(\boldsymbol{\theta})$ penalizes the complexity of the model.

Fitness Term for a Classification Linear Model

Correct Prediction: For the i-th pattern,

$$c_i = \begin{cases} 0 & \text{if } t_i \neq f_{\boldsymbol{\theta}}(\mathbf{x}_i) \\ 1 & \text{if } t_i = f_{\boldsymbol{\theta}}(\mathbf{x}_i) \end{cases} = \begin{cases} 0 & \text{if } (t_i = 0, b + \mathbf{w}^\intercal \mathbf{x} \geq 0) \text{ or } (t_i = 1, b + \mathbf{w}^\intercal \mathbf{x} < 0), \\ 1 & \text{if } (t_i = 0, b + \mathbf{w}^\intercal \mathbf{x} < 0) \text{ or } (t_i = 1, b + \mathbf{w}^\intercal \mathbf{x} \geq 0). \end{cases}$$

Accuracy: $\operatorname{Acc}(b, \mathbf{w}) = \mathbb{E}[C] \approx \frac{1}{N} \sum_{i=1}^{N} c_i.$

Quality of the Model - Exercise

Exercise

Given a 2-dimensional binary linear classification model with parameters $\boldsymbol{\theta} = \{b, \mathbf{w}\}$, with $b = 1$ and $\mathbf{w} = (1, 2)^\intercal$, and for the following data:

$x_{i,1}$	$x_{i,2}$	t_i
1	1	1
1	-2	0
0	0	0

1. Compute the Accuracy.

Solution

1. $\text{Acc}(b, \mathbf{w}) = \frac{2}{3} \approx 66.66\,\%$.

Notebook

Binary Linear Classification: Quality of the Model

Training a Linear Model: Using the Regression Framework

- The most common choice for the evaluating the model the Accuracy.
  - It is a sensible and intuitive measure.
  - It is **non-convex**.
  - It is **non-differentiable**.
  - It is **discontinuous**.

- Optimizing the accuracy is a problem that cannot (in general) be tackled directly.

- An alternative idea could be to train a linear regression model.
  - Labels $-1/1$.
  - The label is predicted taking the sign.

Notebook

Binary Linear Classification: Training a Regression Linear Model

Training a Linear Model: Logistic Regression (I)

- A different quality measure is needed.
  - It should be simpler to optimize than the Accuracy.
  - It should not penalize points far from the decision boundary (on the correct side).

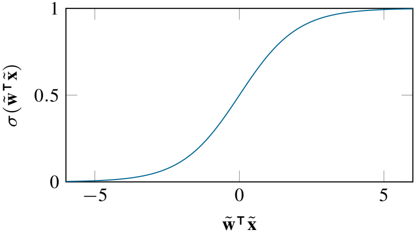
- A probabilistic approach can be helpful.
- In particular, the main framework is the **Logistic Regression**.
  - The linear model is used to estimate the posterior probability of one class.
  - A sigmoid transformation is used.

Training a Linear Model: Logistic Regression (II)

- Denoting by $\tilde{\mathbf{x}} = [1, \mathbf{x}]$ and by $\tilde{\mathbf{w}} = [b, \mathbf{w}]$, the posterior probabilities are defined as:

$$\mathrm{p}(\mathcal{C}_1 | \tilde{\mathbf{x}}; \tilde{\mathbf{w}}) = \sigma(\tilde{\mathbf{w}}^\intercal \tilde{\mathbf{x}}) = \frac{1}{1 + e^{-\tilde{\mathbf{w}}^\intercal \tilde{\mathbf{x}}}},$$

$$\mathrm{p}(\mathcal{C}_0 | \tilde{\mathbf{x}}; \tilde{\mathbf{w}}) = 1 - \mathrm{p}(\mathcal{C}_1 | \tilde{\mathbf{x}}; \tilde{\mathbf{w}}) = 1 - \frac{1}{1 + e^{-\tilde{\mathbf{w}}^\intercal \tilde{\mathbf{x}}}} = \frac{e^{-\tilde{\mathbf{w}}^\intercal \tilde{\mathbf{x}}}}{1 + e^{-\tilde{\mathbf{w}}^\intercal \tilde{\mathbf{x}}}} = \frac{1}{1 + e^{\tilde{\mathbf{w}}^\intercal \tilde{\mathbf{x}}}} = \sigma(-\tilde{\mathbf{w}}^\intercal \tilde{\mathbf{x}}).$$



- $\tilde{\mathbf{w}}^\intercal \tilde{\mathbf{x}} < 0 \implies \mathrm{p}(\mathcal{C}_1 | \tilde{\mathbf{x}}; \tilde{\mathbf{w}}) < 0.5$: Class $\mathcal{C}_0$ is predicted.
- $\tilde{\mathbf{w}}^\intercal \tilde{\mathbf{x}} \geq 0 \implies \mathrm{p}(\mathcal{C}_1 | \tilde{\mathbf{x}}; \tilde{\mathbf{w}}) \geq 0.5$: Class $\mathcal{C}_1$ is predicted.

Training a Linear Model: Logistic Regression - Exercise

Exercise

Given a 2-dimensional binary linear classification model with parameters $\boldsymbol{\theta} = \{b, \mathbf{w}\}$, with $b = 1$ and $\mathbf{w} = (1, 2)^\intercal$.

1. Compute the probability of $\mathbf{x}_1$ belonging to class $\mathcal{C}_1$ for $\mathbf{x}_1 = (1, 1)^\intercal$.
2. Compute the probability of $\mathbf{x}_2$ belonging to class $\mathcal{C}_1$ for $\mathbf{x}_2 = (1, -2)^\intercal$.
3. Compute the probability of $\mathbf{x}_3$ belonging to class $\mathcal{C}_1$ for $\mathbf{x}_3 = (0, 0)^\intercal$.

Solution

1. $b + \mathbf{w}^\intercal \mathbf{x}_1 = 4 \implies \mathrm{p}(\mathcal{C}_1 | \tilde{\mathbf{x}}_1; \tilde{\mathbf{w}}) \approx 98.2\,\%$.
2. $b + \mathbf{w}^\intercal \mathbf{x}_2 = -2 \implies \mathrm{p}(\mathcal{C}_1 | \tilde{\mathbf{x}}_2; \tilde{\mathbf{w}}) \approx 11.9\,\%$.
3. $b + \mathbf{w}^\intercal \mathbf{x}_3 = 1 \implies \mathrm{p}(\mathcal{C}_1 | \tilde{\mathbf{x}}_3; \tilde{\mathbf{w}}) \approx 73.1\,\%$.

Training a Linear Model - Maximum Likelihood (I)

- The probabilistic interpretation can help to define a quality measure.
- The **likelihood** of the data is commonly the choice:

$$\mathcal{L}(\mathcal{D}; \tilde{\mathbf{w}}) = \prod_{i=1}^{N} \mathrm{p}(t_i | \tilde{\mathbf{x}}_i; \tilde{\mathbf{w}}) = \prod_{i=1}^{N} \underbrace{\mathrm{p}(\mathcal{C}_0 | \tilde{\mathbf{x}}_i; \tilde{\mathbf{w}})^{1-t_i} \, \mathrm{p}(\mathcal{C}_1 | \tilde{\mathbf{x}}_i; \tilde{\mathbf{w}})^{t_i}}_{\begin{cases} \mathrm{p}(\mathcal{C}_0 | \tilde{\mathbf{x}}_i; \tilde{\mathbf{w}}) & \text{if } t_i = 0, \\ \mathrm{p}(\mathcal{C}_1 | \tilde{\mathbf{x}}_i; \tilde{\mathbf{w}}) & \text{if } t_i = 1. \end{cases}} .$$

- The **cross-entropy** error is defined as the minus log-likelihood:

$$\begin{aligned} \mathrm{CE}(\tilde{\mathbf{w}}) &= -\log \mathcal{L}(\mathcal{D}; \tilde{\mathbf{w}}) \\ &= \sum_{i=1}^{N} \left(-(1 - t_i) \log(\mathrm{p}(\mathcal{C}_0 | \tilde{\mathbf{x}}_i; \tilde{\mathbf{w}})) - t_i \log(\mathrm{p}(\mathcal{C}_1 | \tilde{\mathbf{x}}_i; \tilde{\mathbf{w}})) \right) \\ &= \sum_{i=1}^{N} \left(-(1 - t_i) \log(1 - \sigma(\tilde{\mathbf{w}}^\intercal \tilde{\mathbf{x}}_i)) - t_i \log(\sigma(\tilde{\mathbf{w}}^\intercal \tilde{\mathbf{x}}_i)) \right). \end{aligned}$$

Training a Linear Model - Maximum Likelihood - Exercise

Exercise

Given a 2-dimensional binary linear classification model with parameters $\boldsymbol{\theta} = \{b, \mathbf{w}\}$, with $b = 1$ and $\mathbf{w} = (1, 2)^\intercal$, and for the following data:

$x_{i,1}$	$x_{i,2}$	t_i
1	1	1
1	-2	0
0	0	0

1. Compute the likelihood of this model.

Solution

1. $\mathcal{L}(\mathcal{D}; \tilde{\mathbf{w}}) = \mathrm{p}(\mathcal{C}_1 | \tilde{\mathbf{x}}_1; \tilde{\mathbf{w}}) \underbrace{\mathrm{p}(\mathcal{C}_0 | \tilde{\mathbf{x}}_2; \tilde{\mathbf{w}})}_{1-\mathrm{p}(\mathcal{C}_1 | \tilde{\mathbf{x}}_2; \tilde{\mathbf{w}})} \underbrace{\mathrm{p}(\mathcal{C}_0 | \tilde{\mathbf{x}}_3; \tilde{\mathbf{w}})}_{1-\mathrm{p}(\mathcal{C}_1 | \tilde{\mathbf{x}}_3; \tilde{\mathbf{w}})} \approx 23.3\,\%.$

Training a Linear Model - Maximum Likelihood (II)

- The minimizer of $\mathrm{CE}(\tilde{\mathbf{w}})$ is the maximizer of $\mathcal{L}(\mathcal{D}; \tilde{\mathbf{w}})$.
- The learning algorithm for training a Linear Logistic Regression model consists in solving the problem:

$$\min_{\tilde{\mathbf{w}} \in \mathbb{R}^{d+1}} \{\mathrm{CE}(\tilde{\mathbf{w}})\} = \min_{\tilde{\mathbf{w}} \in \mathbb{R}^{d+1}} \left\{ \sum_{i=1}^{N} (-(1 - t_i) \log(1 - \sigma(\tilde{\mathbf{w}}^\intercal \tilde{\mathbf{x}}_i)) - t_i \log(\sigma(\tilde{\mathbf{w}}^\intercal \tilde{\mathbf{x}}_i))) \right\}.$$

- How is this problem solved?
  - It is **convex**: there are no local minima.
  - It is **differentiable**: the optimum is characterized by the zeros of the gradient.

Training a Linear Model - Optimization (I)

$$\min_{\tilde{\mathbf{w}} \in \mathbb{R}^{d+1}} \{\mathrm{CE}(\tilde{\mathbf{w}})\} = \min_{\tilde{\mathbf{w}} \in \mathbb{R}^{d+1}} \left\{ \sum_{i=1}^{N} (-(1 - t_i) \log(1 - \sigma(\tilde{\mathbf{w}}^\intercal \tilde{\mathbf{x}}_i)) - t_i \log(\sigma(\tilde{\mathbf{w}}^\intercal \tilde{\mathbf{x}}_i))) \right\}.$$

$$\begin{aligned}
\nabla_{\tilde{\mathbf{w}}} \mathrm{CE}(\tilde{\mathbf{w}}) &= \sum_{i=1}^{N} (-(1 - t_i) \nabla_{\tilde{\mathbf{w}}} \log(1 - \sigma(\tilde{\mathbf{w}}^\intercal \tilde{\mathbf{x}}_i)) - t_i \nabla_{\tilde{\mathbf{w}}} \log(\sigma(\tilde{\mathbf{w}}^\intercal \tilde{\mathbf{x}}_i))) \\
&= \sum_{i=1}^{N} ((1 - t_i) \sigma(\tilde{\mathbf{w}}^\intercal \tilde{\mathbf{x}}_i) \tilde{\mathbf{x}}_i - t_i (1 - \sigma(\tilde{\mathbf{w}}^\intercal \tilde{\mathbf{x}}_i)) \tilde{\mathbf{x}}_i) \\
&= \sum_{i=1}^{N} \sigma(\tilde{\mathbf{w}}^\intercal \tilde{\mathbf{x}}_i) \tilde{\mathbf{x}}_i - t_i \sigma(\tilde{\mathbf{w}}^\intercal \tilde{\mathbf{x}}_i) \tilde{\mathbf{x}}_i - t_i \tilde{\mathbf{x}}_i + t_i \sigma(\tilde{\mathbf{w}}^\intercal \tilde{\mathbf{x}}_i) \tilde{\mathbf{x}}_i \\
&= \sum_{i=1}^{N} (\sigma(\tilde{\mathbf{w}}^\intercal \tilde{\mathbf{x}}_i) - t_i) \tilde{\mathbf{x}}_i.
\end{aligned}$$

Training a Linear Model - Optimization (II)

- In summary, the Linear Logistic Regression Model is the solution of the following problem:

$$\min_{\tilde{\mathbf{w}} \in \mathbb{R}^{d+1}} \left\{ \sum_{i=1}^{N} \left(-(1 - t_i) \log(1 - \sigma(\tilde{\mathbf{w}}^\intercal \tilde{\mathbf{x}}_i)) - t_i \log(\sigma(\tilde{\mathbf{w}}^\intercal \tilde{\mathbf{x}}_i)) \right) \right\}.$$

- There is not closed-form solution to the resultant equation for the stationary points:

$$\nabla_{\tilde{\mathbf{w}}} \operatorname{CE}(\tilde{\mathbf{w}}) = \sum_{i=1}^{N} (\sigma(\tilde{\mathbf{w}}^\intercal \tilde{\mathbf{x}}_i) - t_i) \tilde{\mathbf{x}}_i = 0.$$

- An iterative algorithm, such as **gradient descent**, should be used.

Training a Linear Model - Optimization (III)

- The minus gradient is a descent direction:

$$f(\mathbf{x}+\boldsymbol{\epsilon}) \approx f(\mathbf{x}) + \nabla_{\mathbf{x}} f(\mathbf{x})^{\intercal} \boldsymbol{\epsilon}$$
$$\Longrightarrow f(\mathbf{x} - \eta \nabla_{\mathbf{x}} f(\mathbf{x})) \approx f(\mathbf{x}) - \eta \|\nabla_{\mathbf{x}} f(\mathbf{x})\|_2^2 \leq f(\mathbf{x}).$$

- Updating the current estimation in the direction of the minus gradient seems a sensible idea.

Linear Logistic Regression Model

- The model can be trained iteratively by updating the weights as:

$$\tilde{\mathbf{w}}^{(k+1)} = \tilde{\mathbf{w}}^{(k)} - \eta^{(k)} \sum_{i=1}^{N} \left(\sigma\left(\left(\tilde{\mathbf{w}}^{(k)} \right)^{\intercal} \tilde{\mathbf{x}}_i \right) - t_i \right) \tilde{\mathbf{x}}_i .$$

Notebook

Binary Linear Classification: Optimization

Introduction to Regularized Learning

Bias–Variance and Regularization

Bias–Variance Trade-off

- Error due to **Bias**: Difference between the expected prediction of the model and the correct value to be predicted.
- Error due to **Variance**: Variability of a model prediction for a given data point.

Definition (Regularization)

- **Regularization** usually denotes the set of techniques that attempt to improve the estimates by biasing them away from their sample-based values towards values that are deemed to be more "physically plausible".
- The variance of the model is reduced to the expense of a potentially higher bias.

Over-Fitting and Under-Fitting (I)

Over-Fitting

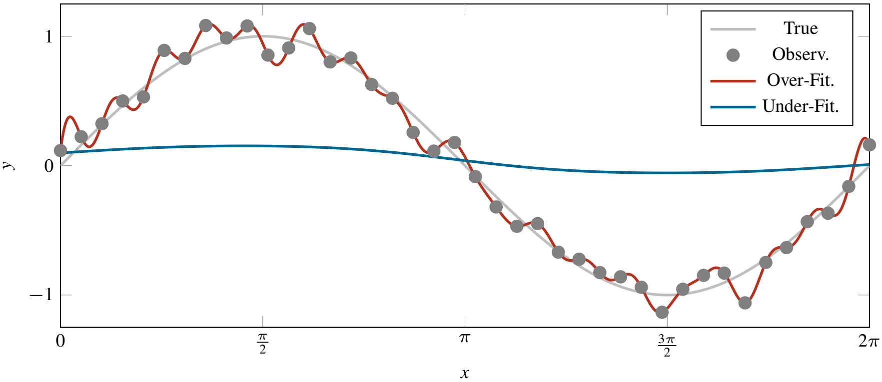
- The resultant model is overly complex to describe the data under study.
  - Limited number of training data.
  - Learning machine too complex (many free parameters).
- Large variance, small bias.

Under-Fitting

- The resultant model is overly simple to describe the data under study.
  - Learning machine too simple.
- Large bias, small variance.

Over-Fitting and Under-Fitting (II)

Under-Fitting and Over-Fitting



Need of Regularization - Example

Example ("Ill-Posed" Problem)

- Regression dataset `E2006-log1p` of the LIBSVM repository.
  - 16 087 patterns for training, 3308 patterns for testing.
  - 4 272 227 features.
- Even the simplest models (linear) will have 220 free parameters per pattern.
- The complexity of the model has to be controlled.
- Probably not all the features will be relevant.
  - A model based on a subset of the features seems a sensible option.

Need of Regularization - Exercise

Exercise

Given a 3-dimensional problem with the following data:

$x_{i,1}$	$x_{i,2}$	$x_{3,2}$	y_i
1	0	1	2
1	1	1	3

1. Define a linear model $\{b, w_1, w_2, w_3\}$ with the smaller possible MSE. Is it possible to get a perfect training prediction?
2. Are there more than one model that can solve perfectly the problem above? Is there anyway to determine which one should be preferred?

Solution

1. The model $\{b = 2, w_1 = 0, w_2 = 1, w_3 = 0\}$ fits the data perfectly.
2. For example, $\{b = 0, w_1 = 1, w_2 = 1, w_3 = 1\}$. There is no information to prefer one or the other.

Why Is Regularization Necessary?

1. There are more variables than observations ($d \gg N$).
2. The optimum estimator is not unique.
3. Numerical instabilities (e.g. if $\mathbf{X}^\intercal \mathbf{X}$ is close to singular): small changes in the data lead to large changes in the model.
4. Over-fitting avoidance: obtain more robust models that generalize well.
5. Parsimony and interpretability: simpler model than can help to understand the relation between inputs and outputs.

Notebook

The Need of Regularization

Regularized Learning

- Regularized learning consists in models trained by optimizing objective functions of the form:

$$\mathcal{S} = \mathcal{E}_{\mathcal{D}} + \gamma \mathcal{R}.$$

- The main term of the objective function is an error term $\mathcal{E}_{\mathcal{D}}$.
  - It represents how well the model fits the training data $\mathcal{D}$.
  - Examples: mean squared error (regression) and minus (log)likelihood (classification).
- The additional term is a regularization term $\mathcal{R}$. It penalizes the complexity of the model, with several purposes:
  - Avoid over-fitting.
  - Introduce prior knowledge.
  - Enforce certain desirable properties.
- γ is a regularization parameter.
  - It is responsible for the balance between accuracy and complexity.

Regularization Functions

Regularization Functions

- There are different regularization functions $\mathcal{R}(\boldsymbol{\theta})$ that assigns to each set of parameters $\boldsymbol{\theta}$ a measure of its complexity.
- Depending on the chosen function, the effect over $\boldsymbol{\theta}$ will change.
- The influence of the regularization functions is particularly clear on linear models.
  - Each coefficient of $\mathbf{w}$ corresponds to an input feature.
  - If $w_i = 0$, then the i-th feature is ignored.
  - If $w_i = w_j$, then the i-th feature is somehow similar to the j-th feature.

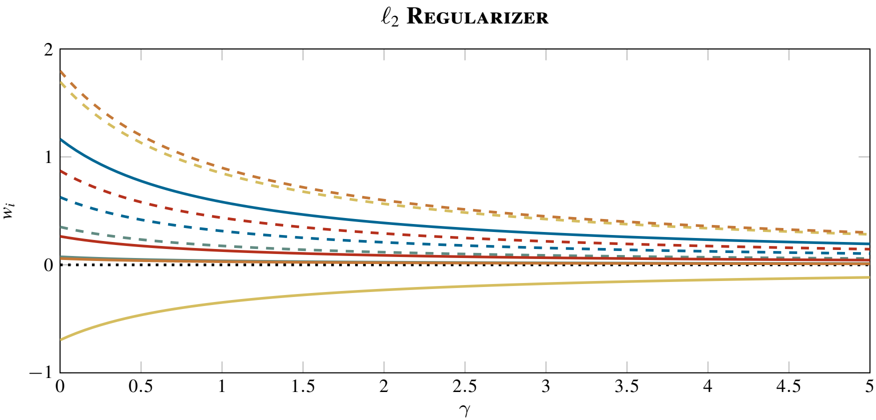
ℓ_2 Norm (I)

- Classical term, known as Tikhonov regularization, it corresponds to the sum of the squares of the entries:

$$\mathcal{R}(\mathbf{w}) = \|\mathbf{w}\|_2^2 = \sum_{i=1}^{d} w_i^2 .$$

- It controls the complexity of the model.
- It is differentiable, and hence easy to optimize.
- It pushes the entries towards zero.

ℓ_2 Norm (II)



ℓ_2 Norm - Exercise

Exercise

Given the following 3-dimensional linear models, compute their squared ℓ_2 norm to check which one is simpler according to this criterion:

1. $\{w_1 = 1, w_2 = 1, w_3 = 1\}$.
2. $\{w_1 = 3, w_2 = 0, w_3 = 0\}$.
3. $\{w_1 = 2, w_2 = 2, w_3 = 0\}$.

Solution

1. $\|\mathbf{w}\|_2^2 = 3$.
2. $\|\mathbf{w}\|_2^2 = 9$.
3. $\|\mathbf{w}\|_2^2 = 8$.

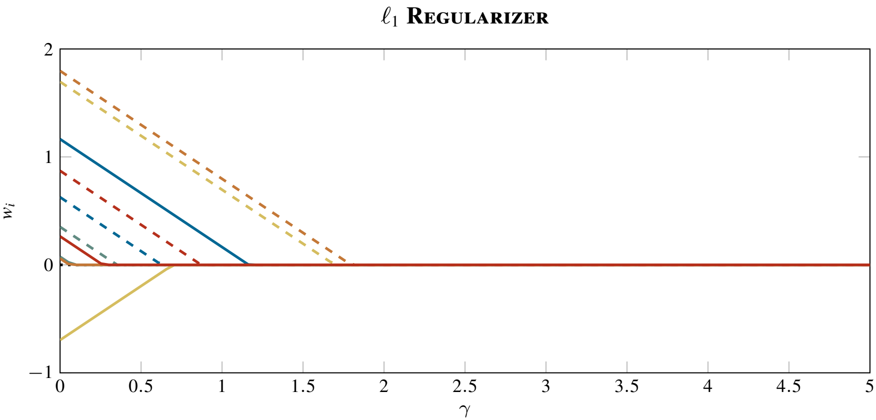
ℓ_1 Norm (I)

- It corresponds to the sum of the absolute values of the entries:

$$\mathcal{R}(\mathbf{w}) = \|\mathbf{w}\|_1 = \sum_{i=1}^{d} |w_i|.$$

- It controls the complexity of the model.
- The absolute value is non-differentiable around zero, and hence this term is more involved to optimize.
- It pushes the entries towards zero enforcing some of them to be identically zero.
    - It enforces sparsity.

ℓ_1 Norm (II)



ℓ_1 Norm - Exercise

Exercise

Given the following 3-dimensional linear models, compute their ℓ_1 norm to check which one is simpler according to this criterion:

1. $\{w_1 = 1, w_2 = 1, w_3 = 1\}$.
2. $\{w_1 = 3, w_2 = 0, w_3 = 0\}$.
3. $\{w_1 = 2, w_2 = 2, w_3 = 0\}$.

Solution

1. $\|\mathbf{w}\|_1 = 3$.
2. $\|\mathbf{w}\|_1 = 3$.
3. $\|\mathbf{w}\|_1 = 4$.

Notebook

Regularization Functions: The ℓ_p Norm

Combinations

- The previous regularizers can be combined to enforce several structures at the same time.

ℓ_1 and ℓ_2

- Advantages of the ℓ_1 and ℓ_2 approaches combined.
- The ℓ_2 term controls the overall complexity.
- The ℓ_1 term imposes sparsity.

Notebook

Regularization Functions: Combination of the ℓ_1 Norm and the ℓ_2 Norm

Regularized Linear Models

The Optimization Problem of a Regularized Model

- The optimization problem to train a regularized model can be formulated as:

$$\min_{\boldsymbol{\theta}} \; \{\mathcal{E}_{\mathcal{D}}(\boldsymbol{\theta}) + \gamma \mathcal{R}(\boldsymbol{\theta})\}.$$

- There exists an equivalence between this unconstrained model and the following constrained formulation:

$$\min_{\boldsymbol{\theta}} \; \{\mathcal{E}_{\mathcal{D}}(\boldsymbol{\theta})\} \;\; \text{s.t.} \;\; \mathcal{R}(\boldsymbol{\theta}) \le c.$$

- In the case of a regression linear model:

$$\min_{\mathbf{w} \in \mathbb{R}^d} \; \left\{ \|\mathbf{y} - \mathbf{X}\mathbf{w}\|_2^2 + \gamma \mathcal{R}(\mathbf{w}) \right\} \equiv \min_{\mathbf{w} \in \mathbb{R}^d} \; \left\{ \|\mathbf{y} - \mathbf{X}\mathbf{w}\|_2^2 \right\} \;\; \text{s.t.} \;\; \mathcal{R}(\mathbf{w}) \le c.$$

- In the case of a classification linear model:

$$\min_{\mathbf{w} \in \mathbb{R}^d} \; \{\mathrm{CE}(\mathbf{w}) + \gamma \mathcal{R}(\mathbf{w})\} \equiv \min_{\mathbf{w} \in \mathbb{R}^d} \; \{\mathrm{CE}(\mathbf{w})\} \;\; \text{s.t.} \;\; \mathcal{R}(\mathbf{w}) \le c.$$

Notebook

Linear Models and the ℓ_p Norm

Ridge Regression

- This linear model uses the Tikhonov regularization:

$$\mathcal{R}(\mathbf{w}) = \frac{1}{2}\|\mathbf{w}\|_2^2 = \frac{1}{2}\sum_{i=1}^{d} \mathbf{w}_i^2 .$$

- The objective function is:

$$\mathcal{S}(\mathbf{w}) = \mathrm{MSE}(\mathbf{w}) + \frac{\gamma}{2}\|\mathbf{w}\|_2^2 .$$

- The complexity of the model is controlled.
  - In the presence of noise:

$$\mathbf{w}^\intercal(\mathbf{x} + \epsilon) = \mathbf{w}^\intercal\mathbf{x} + \mathbf{w}^\intercal\epsilon \le \mathbf{w}^\intercal\mathbf{x} + \|\mathbf{w}\|_2\|\epsilon\|_2 \overset{?}{\approx} \mathbf{w}^\intercal\mathbf{x}.$$

- No structure is imposed.
  - The resultant model typically depends on all the variables.

- The problem is convex and differentiable.

Ridge Regression: Optimization

$$\min_{\mathbf{w}\in\mathbb{R}^d} \left\{ \frac{1}{2}\|\mathbf{y}-\mathbf{X}\mathbf{w}\|_2^2 + \frac{\gamma}{2}\|\mathbf{w}\|_2^2 \right\}.$$

$$\begin{aligned}
\nabla_{\mathbf{w}}\mathcal{S}(\mathbf{w})\big|_{\mathbf{w}=\mathbf{w}^\star} = \mathbf{0} &\implies -\mathbf{X}^\intercal(\mathbf{y}-\mathbf{X}\mathbf{w}^\star) + \gamma\mathbf{w}^\star = \mathbf{0} \\
&\implies -\mathbf{X}^\intercal\mathbf{y} + \mathbf{X}^\intercal\mathbf{X}\mathbf{w}^\star + \gamma\mathbf{w}^\star = \mathbf{0} \\
&\implies (\mathbf{X}^\intercal\mathbf{X} + \gamma\mathbf{I})\mathbf{w}^\star = \mathbf{X}^\intercal\mathbf{y} \\
&\implies \boxed{\mathbf{w}^\star = (\mathbf{X}^\intercal\mathbf{X} + \gamma\mathbf{I})^{-1}\mathbf{X}^\intercal\mathbf{y}}.
\end{aligned}$$

Notebook

Ridge Regression

Lasso

- This linear model uses as regularizer the ℓ_1 norm:

$$\mathcal{R}(\mathbf{w}) = \|\mathbf{w}\|_1 = \sum_{i=1}^{d} |w_i|.$$

- The objective function is:

$$\mathcal{S}(\mathbf{w}) = \mathrm{MSE}(\mathbf{w}) + \gamma \|\mathbf{w}\|_1.$$

- This regularizer enforces some of the coefficients to be identically zero.
  - The model performs an implicit feature selection, the features with coefficient equal to zero can be discarded.
  - It also avoids the over-fitting.

- The problem is convex but non-differentiable.

Notebook

Lasso

Elastic–Net

- This linear model combines the advantages of the ℓ_1 norm with those of the ℓ_2 norm.
- It is more stable than Lasso regarding feature selection.
- The regularizer is therefore a combination of both:

$$\mathcal{R}(\mathbf{w}) = \|\mathbf{w}\|_1 + \frac{\gamma_2'}{2} \|\mathbf{w}\|_2^2.$$

- Thus the objective function is:

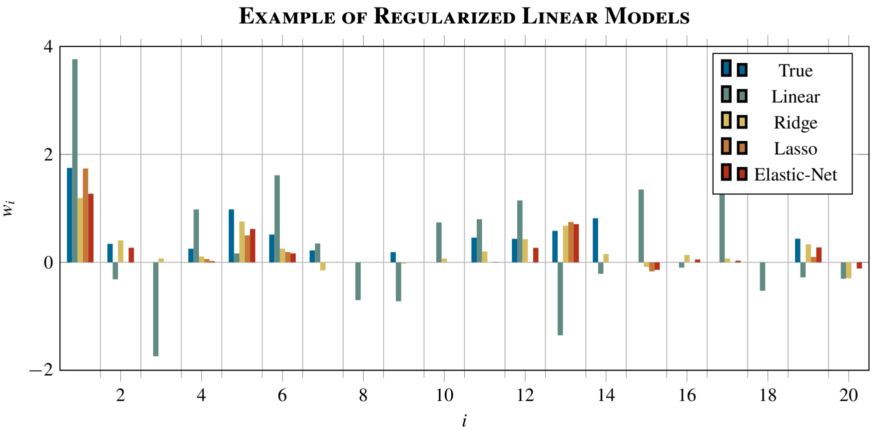
$$\mathcal{S}(\mathbf{w}) = \mathrm{MSE}(\mathbf{w}) + \gamma_1 \|\mathbf{w}\|_1 + \frac{\gamma_2}{2} \|\mathbf{w}\|_2^2.$$

- The problem is convex but non-differentiable.

Notebook

Elastic–Net

Illustration



Review of Linear Models

Carlos María Alaíz Gudín

Additional Material - Linear Regression Models

Training a Linear Model - Example

Example (Perfect Case)

- In the perfectly linear case, $y_i = \mathbf{w}^\intercal \mathbf{x}_i + b$.
- In matrix notation, $\mathbf{y} = \tilde{\mathbf{X}}\tilde{\mathbf{w}}$.
- Therefore, the linear model becomes:

$$
\begin{aligned}
\tilde{\mathbf{w}}^\star &= \tilde{\mathbf{X}}^\dagger \mathbf{y} \\
&= \left(\tilde{\mathbf{X}}^\intercal \tilde{\mathbf{X}}\right)^{-1} \tilde{\mathbf{X}}^\intercal \mathbf{y} \\
&= \left(\tilde{\mathbf{X}}^\intercal \tilde{\mathbf{X}}\right)^{-1} \tilde{\mathbf{X}}^\intercal \left(\tilde{\mathbf{X}}\tilde{\mathbf{w}}\right) \\
&= \left(\tilde{\mathbf{X}}^\intercal \tilde{\mathbf{X}}\right)^{-1} \left(\tilde{\mathbf{X}}^\intercal \tilde{\mathbf{X}}\right) \tilde{\mathbf{w}} \\
&= \tilde{\mathbf{w}}.
\end{aligned}
$$

Training a Linear Model - Bayesian Perspective (I)

- There is an additional justification for using the MSE in a linear model.
- The output is assumed to be a linear transformation of the input corrupted with Gaussian noise:

$$y_i = \mathbf{w}^\intercal \mathbf{x}_i + \epsilon_i,$$

where $\epsilon_i \sim \mathcal{N}(0, \sigma)$.

- The likelihood of the data becomes:

$$\mathrm{p}(\mathcal{D}|\mathbf{w}) \propto \prod_{i=1}^{N} \exp\left(-\frac{\epsilon_i^2}{2\sigma^2}\right) = \prod_{i=1}^{N} \exp\left(-\frac{(y_i - \mathbf{w}^\intercal \mathbf{x}_i)^2}{2\sigma^2}\right).$$

- $\mathbf{w}^\star \in \mathbb{R}^d$ is selected as the maximizer of the likelihood:

$$\max_{\mathbf{w}\in\mathbb{R}^d} \left\{\prod_{i=1}^{N} \mathrm{p}(\mathcal{D}|\mathbf{w})\right\} = \max_{\mathbf{w}\in\mathbb{R}^d} \left\{\prod_{i=1}^{N} \exp\left(-\frac{(y_i - \mathbf{w}^\intercal \mathbf{x}_i)^2}{2\sigma^2}\right)\right\}.$$

Training a Linear Model - Bayesian Perspective (II)

- Equivalently, instead of maximizing the likelihood, the minus log-likelihood is minimized:

$$\min_{\mathbf{w}\in\mathbb{R}^d} \left\{ \sum_{i=1}^{N} (y_i - \mathbf{w}^\intercal \mathbf{x}_i)^2 \right\},$$

  which coincides with the least squares problem for a linear model.

- Bayesian linear regression is more than this.
- The **prior** can be used to impose structure, use prior knowledge, etc.

Additional Material - Linear Classification Models

Expressions for the Gradient of the Sigmoid Transformation

- The linear model with sigmoid transformation satisfies the following equations:

$$\begin{aligned}
\nabla_{\tilde{\mathbf{w}}} \sigma(\tilde{\mathbf{w}}^\intercal \tilde{\mathbf{x}}) &= \nabla_{\tilde{\mathbf{w}}} \frac{1}{1 + e^{-\tilde{\mathbf{w}}^\intercal \tilde{\mathbf{x}}}} = \frac{1}{(1 + e^{-\tilde{\mathbf{w}}^\intercal \tilde{\mathbf{x}}})^2} e^{-\tilde{\mathbf{w}}^\intercal \tilde{\mathbf{x}}} \tilde{\mathbf{x}} = \frac{1}{1 + e^{-\tilde{\mathbf{w}}^\intercal \tilde{\mathbf{x}}}} \frac{e^{-\tilde{\mathbf{w}}^\intercal \tilde{\mathbf{x}}}}{1 + e^{-\tilde{\mathbf{w}}^\intercal \tilde{\mathbf{x}}}} \tilde{\mathbf{x}} \\
&= \sigma(\tilde{\mathbf{w}}^\intercal \tilde{\mathbf{x}})(1 - \sigma(\tilde{\mathbf{w}}^\intercal \tilde{\mathbf{x}}))\tilde{\mathbf{x}}; \\
\nabla_{\tilde{\mathbf{w}}} \log(\sigma(\tilde{\mathbf{w}}^\intercal \tilde{\mathbf{x}})) &= \frac{1}{\sigma(\tilde{\mathbf{w}}^\intercal \tilde{\mathbf{x}})} \nabla_{\tilde{\mathbf{w}}} \sigma(\tilde{\mathbf{w}}^\intercal \tilde{\mathbf{x}}) = (1 - \sigma(\tilde{\mathbf{w}}^\intercal \tilde{\mathbf{x}}))\tilde{\mathbf{x}}; \\
\nabla_{\tilde{\mathbf{w}}} \log(1 - \sigma(\tilde{\mathbf{w}}^\intercal \tilde{\mathbf{x}})) &= \nabla_{\tilde{\mathbf{w}}} \log(\sigma(-\tilde{\mathbf{w}}^\intercal \tilde{\mathbf{x}})) = -(1 - \sigma(-\tilde{\mathbf{w}}^\intercal \tilde{\mathbf{x}}))\tilde{\mathbf{x}} = -\sigma(\tilde{\mathbf{w}}^\intercal \tilde{\mathbf{x}})\tilde{\mathbf{x}}.
\end{aligned}$$

- These properties are one of the reasons why this function is so commonly used

Additional Material - More Regularization Functions

$\ell_{2,1}$ Norm: Framework

- Each $\mathbf{w}$ is composed by d_g groups of $d_f = \frac{d}{d_g}$ features each group:

$$\mathbf{w} = \begin{pmatrix} w_{1,1} \\ \vdots \\ w_{1,d_f} \\ \vdots \\ w_{d_g,1} \\ \vdots \\ w_{d_g,d_f} \end{pmatrix},$$

  where $w_{g,f}$ is the f-th entry of the g-th group.
  - This framework can be easily extended to groups of different sizes.
- The variable $\mathbf{w}$ can be seen also as a matrix with d_f rows and d_g columns.
- The regularizers should respect this structure.

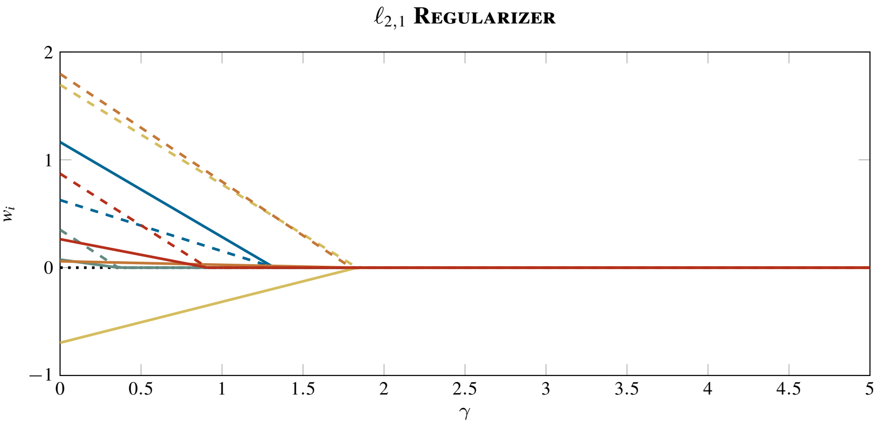
$\ell_{2,1}$ Norm (I)

- The regularizer is the $\ell_{2,1}$ norm:

$$\mathcal{R}(\mathbf{w}) = \|\mathbf{w}\|_{2,1} = \sum_{g=1}^{d_g} \sqrt{\sum_{f=1}^{d_f} w_{g,f}^2},$$

  which is just the ℓ_1 norm of the ℓ_2 norm of the different groups.
- It controls the complexity of the model.
- The ℓ_2 norm (not squared) is non-differentiable around zero, hence this term is more involved to optimize.
- It pushes the groups towards zero enforcing some of them to be identically zero.
  - It enforces sparsity at group level.

$\ell_{2,1}$ Norm (II)



Transformed Norms

- The regularization is applied over a linear transformation $\mathbf{Tw}$.
- The transformation allows for more involved structures.

Generalized ℓ_2 Norm

- The regularizer is $\mathcal{R}(\mathbf{w}) = \|\mathbf{Tw}\|_2^2$.
- It pushes the transformed vector towards zero.

Generalized Lasso

- The regularizer is $\mathcal{R}(\mathbf{w}) = \|\mathbf{Tw}\|_1$.
- It pushes the transformed vector towards zero enforcing some of the elements to be identically zero.
  - It enforces sparsity over the transformed vector.

Transformed Norms: Total Variation (I)

- The Total Variation is a family of regularizers that penalize the differences between adjacent entries.
  - It assumes some spatial location.
- It transforms the variable through a differentiating matrix:

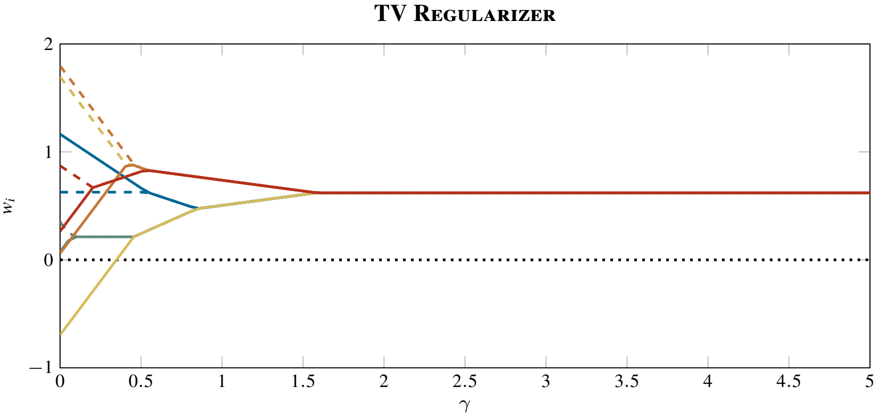
$$\mathbf{D} = \begin{pmatrix} -1 & 1 & 0 & \dots & 0 & 0 \\ 0 & -1 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & -1 & 1 \end{pmatrix}.$$

- The TV regularizer penalizes the ℓ_1 norm of the differences:

$$\mathcal{R}(\mathbf{w}) = \|\mathbf{D}\mathbf{w}\|_1 = \sum_{i=2}^{d} |w_i - w_{i-1}|.$$

  - The ℓ_1 norm enforces sparsity.
  - Some of the terms $w_i - w_{i-1}$ are zero, and hence $w_i = w_{i-1}$.
  - The vector $\mathbf{w}$ is piece-wise constant.

Transformed Norms: Total Variation (II)



Transformed Norms: Others

Graph-Based Total Variation

- An extension of the Total Variation regularizer.
- The differences between any pair of entries connected according to a graph are penalized.
- The classical Total Variation is recovered when the graph is a chain.
- When the graph is a lattice, it becomes a two-dimensional Total Variation.

Trend Filtering

- Similar idea than Total Variation but for higher degrees.
- Instead of penalizing the first differences, higher orders are penalized.

Combinations

- The previous regularizers can be combined to enforce several structures at the same time.

ℓ_1 and $\ell_{2,1}$

- Sparsity both at group level and at coefficient level.

ℓ_1 and Total Variation

- Some of the entries are identically zero.
- The remaining entries tend to be piece-wise constant.

Additional Material - More Regularized Linear Models

Group Variants: Framework

- In certain circumstances, some features are grouped as corresponding to the same source.
  - E.g., different meteorological variables (wind speed, temperature) corresponding to the same geographical point.
- A grouping effect in the features is thus desirable.
  - All the features of a group should be active, or inactive, at the same time.
  - But they are different features, and they can have different coefficients.
- In this way, relevant groups can be detected.

Group Lasso and Group Elastic–Net

Group Lasso Model

- This linear model uses as regularizer the $\ell_{2,1}$ norm, $\mathcal{R}(\mathbf{w}) = \|\mathbf{w}\|_{2,1}$.
- The objective function is:

$$\mathcal{S}(\mathbf{w}) = \mathrm{MSE}(\mathbf{w}) + \gamma \|\mathbf{w}\|_{2,1}.$$

Group Elastic–Net Model

- The regularizer is a combination of the $\ell_{2,1}$ norm and the ℓ_2 norm.
- The objective function is:

$$\mathcal{S}(\mathbf{w}) = \mathrm{MSE}(\mathbf{w}) + \gamma_1 \|\mathbf{w}\|_{2,1} + \frac{\gamma_2}{2} \|\mathbf{w}\|_2^2.$$

Fused Lasso

- This linear model uses as regularizer the ℓ_1 norm and the TV regularizer:

$$\mathcal{R}(\mathbf{w}) = \|\mathbf{w}\|_1 + \gamma_2' \, \mathrm{TV}(\mathbf{w}).$$

- It assumes that the features have some spatial location, and that they are ordered according to it.
  - A sensible model should assign similar coefficients to adjacent features.
- There are, therefore, sparse and piece-wise constant coefficients.
- The objective function is:

$$\mathcal{S}(\mathbf{w}) = \mathrm{MSE}(\mathbf{w}) + \gamma_1 \|\mathbf{w}\|_1 + \gamma_2 \, \mathrm{TV}(\mathbf{w}).$$

Illustration (I)

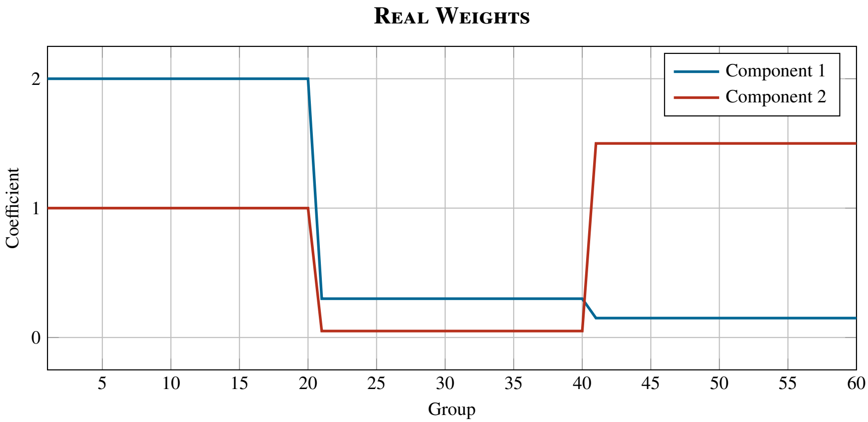


Illustration (II)

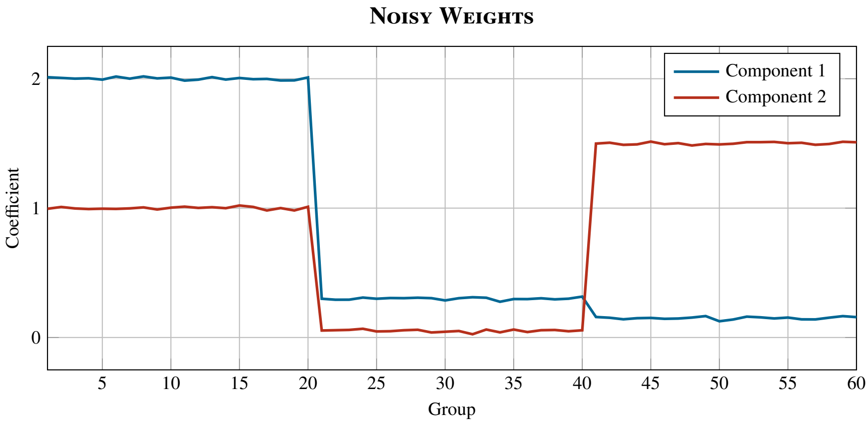


Illustration (III)

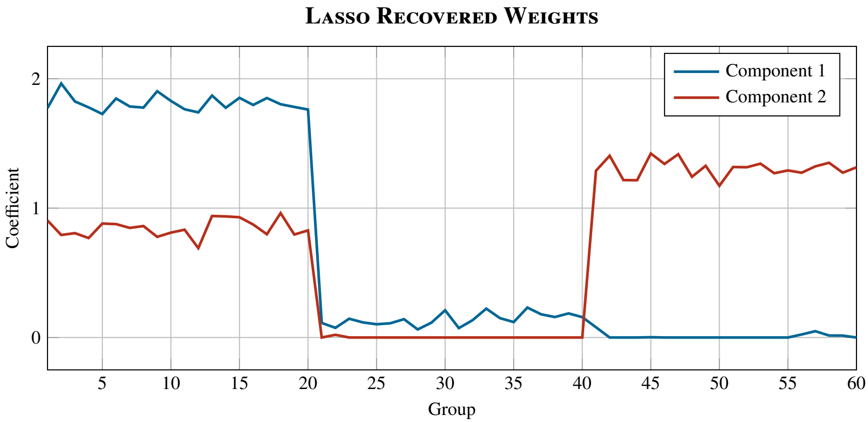


Illustration (IV)

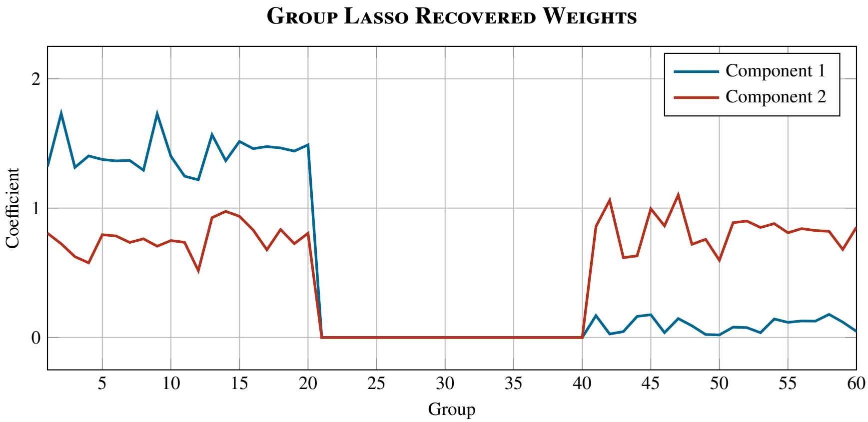


Illustration (V)