MLCC 2022
Regularization and Linear Models

Lorenzo Rosasco
MaLGa, DIBRIS, UNIGE;
CBMM, MIT;
IIT

About this class

- We introduce a class of learning algorithms based on *Tikhonov regularization*
- We study computational aspects of these algorithms .

Empirical Risk Minimization (ERM)

- Empirical Risk Minimization (ERM): probably the most popular approach to design learning algorithms.
- **General idea:** considering the empirical error

$$\hat{\mathcal{E}}(f) = \frac{1}{n}\sum_{i=1}^{n} \ell(y_i, f(x_i)),$$

as a proxy for the expected error

$$\mathcal{E}(f) = \mathbb{E}[\ell(y, f(x))] = \int dxdyp(x,y)\ell(y, f(x)).$$

The Expected Risk is Not Computable

Recall that

- ℓ measures the price we pay predicting $f(x)$ when the true label is y
- $\mathcal{E}(f)$ cannot be directly computed, since $p(x, y)$ is unknown

From Theory to Algorithms: The Hypothesis Space

To turn the above idea into an actual algorithm, we:

- Fix a suitable hypothesis space H
- Minimize $\hat{\mathcal{E}}$ over H

H should allow feasible computations and be *rich*, since the complexity of the problem is not known a priori.

Example: Space of Linear Functions

The simplest example of H is the space of linear functions:

$$H = \{f : \mathbb{R}^d \to \mathbb{R} \ : \ \exists w \in \ \mathbb{R}^d \text{ such that } f(x) = x^T w, \ \forall x \in \mathbb{R}^d\}.$$

- Each function f is defined by a vector w
- $f_w(x) = x^T w$.

Rich Hypothesis spaces May Require Regularization

- If H is rich enough, solving ERM may cause overfitting (solutions highly dependent on the data)
- Regularization induces a bias in the search of a solution ensuring stability and generalization

Tikhonov Regularization

Consider the *Tikhonov* regularization scheme,

$$\min_{w \in \mathbb{R}^d} \hat{\mathcal{E}}(f_w) + \lambda \|w\|^2 \tag{1}$$

It describes a large class of methods sometimes called Regularization Networks.

The Regularizer

- $\|w\|^2$ is called *regularizer*
- It controls the stability of the solution and prevents overfitting
- λ balances the error term and the regularizer

Minimal norm solution/interpolant

If $\lambda \mapsto 0$ we are considering

$$\min_{w \in M} \|w\|$$

where

$$M = \operatorname*{argmin}_{w \in \mathbb{R}^d} \hat{\mathcal{E}}(f_w)$$

Loss Functions

- Different loss functions ℓ induce different classes of methods
- We will see common aspects and differences in considering different loss functions
- There exists no general computational scheme to solve Tikhonov Regularization
- The solution depends on the considered loss function

The Regularized Least Squares Algorithm

Regularized Least Squares: *Tikhonov* regularization

$$\min_{w \in \mathbb{R}^D} \hat{\mathcal{E}}(f_w) + \lambda \|w\|^2, \quad \hat{\mathcal{E}}(f_w) = \frac{1}{n} \sum_{i=1}^{n} \ell(y_i, f_w(x_i)) \tag{2}$$

Square loss function:

$$\ell(y, f_w(x)) = (y - f_w(x))^2$$

We then obtain the RLS optimization problem (linear model):

$$\min_{w \in \mathbb{R}^D} \frac{1}{n} \sum_{i=1}^{n} (y_i - w^T x_i)^2 + \lambda w^T w, \quad \lambda \geq 0. \tag{3}$$

Matrix Notation

- The $n \times d$ matrix X_n, whose rows are the input points
- The $n \times 1$ vector Y_n, whose entries are the corresponding outputs.

With this notation,

$$\frac{1}{n}\sum_{i=1}^{n}(y_i - w^T x_i)^2 = \frac{1}{n}\|Y_n - X_n w\|^2.$$

Gradients of the ER and of the Regularizer

By direct computation,

- Gradient of the empirical risk w. r. t. w

$$-\frac{2}{n} X_n^T (Y_n - X_n w)$$

- Gradient of the regularizer w. r. t. w

$$2w$$

The RLS Solution

By setting the gradient to zero, the solution of RLS solves the linear system

$$(X_n^T X_n + \lambda n I)w = X_n^T Y_n.$$

λ controls the *invertibility* of $(X_n^T X_n + \lambda n I)$

Choosing the Cholesky Solver

- Several methods can be used to solve the above linear system
- Cholesky decomposition is the method of choice, since

$$X_n^T X_n + \lambda I$$

  is symmetric and positive definite.

Time Complexity

Time complexity of the method :

- Training: $O(nd^2)$ (assuming $n >> d$)
- Testing: $O(d)$

Dealing with an Offset

For linear models, especially in low dimensional spaces, it is useful to consider an *offset*:

$$w^T x + b$$

How to estimate b from data?

Idea: Augmenting the Dimension of the Input Space

- Simple idea: augment the dimension of the input space, considering $\tilde{x} = (x, 1)$ and $\tilde{w} = (w, b)$.

- This is fine if we do not regularize, but if we do then this method tends to prefer linear functions passing through the origin (zero offset), since the regularizer becomes:

$$\|\tilde{w}\|^2 = \|w\|^2 + b^2.$$

Avoiding to Penalize the Solutions with Offset

We want to regularize considering only $\|w\|^2$, without penalizing the offset.

The modified regularized problem becomes:

$$\min_{(w,b)\in\mathbb{R}^{D+1}} \frac{1}{n}\sum_{i=1}^{n}(y_i - w^T x_i - b)^2 + \lambda\|w\|^2.$$

Solution with Offset: Centering the Data

It can be proved that a solution w^*, b^* of the above problem is given by

$$b^* = \bar{y} - \bar{x}^T w^*$$

where

$$\bar{y} = \frac{1}{n} \sum_{i=1}^{n} y_i$$

$$\bar{x} = \frac{1}{n} \sum_{i=1}^{n} x_i$$

Solution with Offset: Centering the Data

w^* solves

$$\min_{w \in \mathbb{R}^D} \frac{1}{n} \sum_{i=1}^{n} (y_i^c - w^T x_i^c)^2 + \lambda \|w\|^2.$$

where $y_i^c = y - \bar{y}$ and $x_i^c = x - \bar{x}$ for all $i = 1, \ldots, n$.

Note: This corresponds to centering the data and then applying the standard RLS algorithm.

Introducing: Regularized Logistic Regression

Regularized logistic regression: *Tikhonov* regularization

$$\min_{w \in \mathbb{R}^d} \hat{\mathcal{E}}(f_w) + \lambda \|w\|^2, \quad \hat{\mathcal{E}}(f_w) = \frac{1}{n} \sum_{i=1}^{n} \ell(y_i, f_w(x_i)) \tag{4}$$

With the *logistic loss function*:

$$\ell(y, f_w(x)) = \log(1 + e^{-y f_w(x)})$$

The Logistic Loss Function

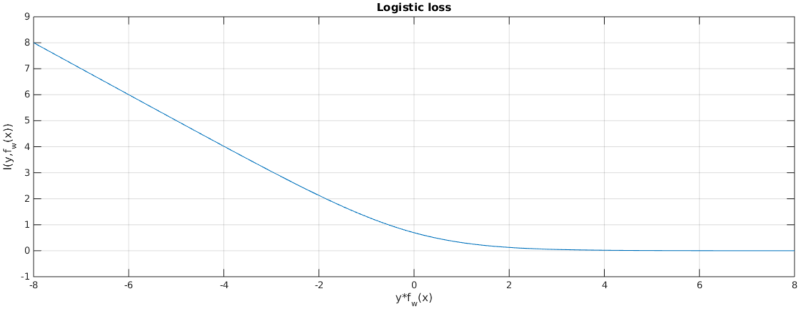


Figure: Plot of the logistic regression loss function

Minimization Through Gradient Descent

- The logistic loss function is differentiable
- The candidate to compute a minimizer is the *gradient descent (GD)* algorithm

Regularized Logistic Regression (RLR)

- The regularized ERM problem associated with the logistic loss is called *regularized logistic regression*
- Its solution can be computed via gradient descent
- Note:

$$\nabla \hat{\mathcal{E}}(f_w) = \frac{1}{n}\sum_{i=1}^{n} x_i \frac{-y_i e^{-y_i x_i^T w_{t-1}}}{1+e^{-y_i x_i^T w_{t-1}}} = \frac{1}{n}\sum_{i=1}^{n} x_i \frac{-y_i}{1+e^{y_i x_i^T w_{t-1}}}$$

RLR: Gradient Descent Iteration

For $w_0 = 0$, the GD iteration applied to

$$\min_{w \in \mathbb{R}^d} \hat{\mathcal{E}}(f_w) + \lambda \|w\|^2$$

is

$$w_t = w_{t-1} - \gamma \underbrace{\left(\frac{1}{n} \sum_{i=1}^{n} x_i \frac{-y_i}{1 + e^{y_i x_i^T w_{t-1}}} + 2\lambda w_{t-1} \right)}_{a}$$

for $t = 1, \ldots T$, where

$$a = \nabla(\hat{\mathcal{E}}(f_w) + \lambda \|w\|^2)$$

Logistic Regression and Confidence Estimation

- The solution of logistic regression has a probabilistic interpretation
- It can be derived from the following model

$$p(1|x) = \underbrace{\frac{e^{x^T w}}{1 + e^{x^T w}}}_{h(w)}$$

  where h is called *logistic function*.
- This can be used to compute a *confidence* for each prediction

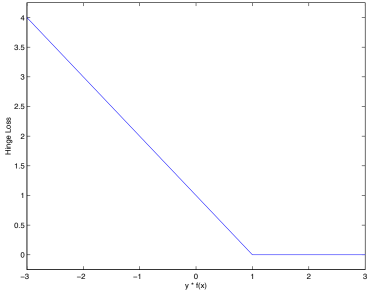
Support Vector Machines

Formulation in terms of Tikhonov regularization:

$$\min_{w \in \mathbb{R}^d} \hat{\mathcal{E}}(f_w) + \lambda \|w\|^2, \quad \hat{\mathcal{E}}(f_w) = \frac{1}{n} \sum_{i=1}^{n} \ell(y_i, f_w(x_i)) \tag{5}$$

With the *Hinge loss function*:

$$\ell(y, f_w(x)) = |1 - y f_w(x)|_+$$



A more classical formulation (linear case)

$$w^* = \min_{w \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^{n} |1 - y_i w^\top x_i|_+ + \lambda \|w\|^2$$

with $\lambda = \frac{1}{C}$

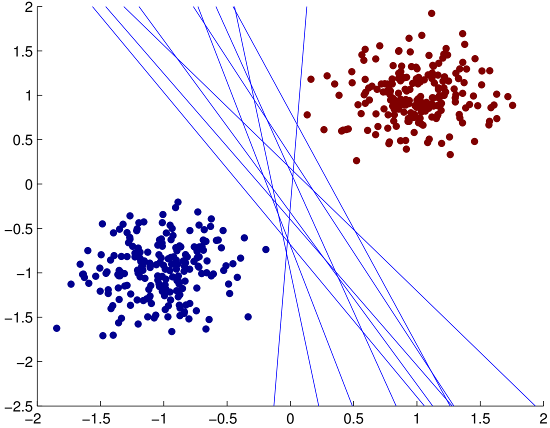
A more classical formulation (linear case)

$$w^* = \min_{w \in \mathbb{R}^d, \xi_i \geq 0} \|w\|^2 + \frac{C}{n} \sum_{i=1}^{n} \xi_i \quad \text{subject to}$$

$$y_i w^\top x_i \geq 1 - \xi_i \quad \forall i \in \{1 \dots n\}$$

A geometric intuition - classification

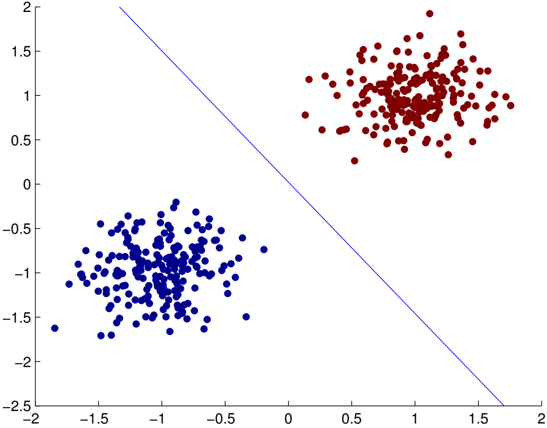
In general do you have many solutions



What do you select?

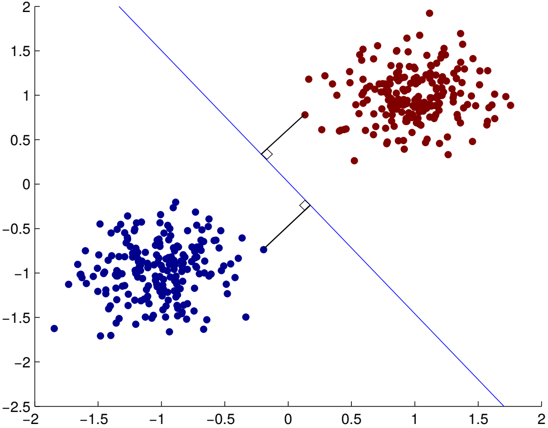
A geometric intuition - classification

Intuitively I would choose an "equidistant" line



A geometric intuition - classification

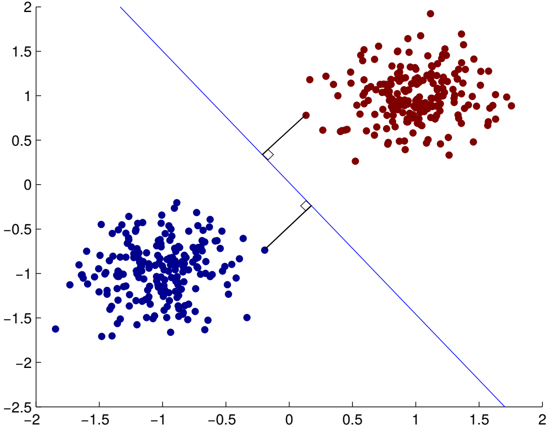
Intuitively I would choose an "equidistant" line



Maximum margin classifier

I want the classifier that

- classifies perfectly the dataset
- maximize the distance from its closest examples

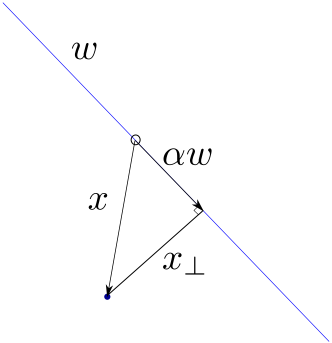


Point-Hyperplane distance

How to do it mathematically? Let w our separating hyperplane. We have

$$x = \alpha w + x_\perp$$

with $\alpha = \frac{x^\top w}{\|w\|}$ and $x_\perp = x - \alpha w$.



Point-Hyperplane distance: $d(x, w) = \|x_\perp\|$

Margin

An hyperplane w well classifies an example (x_i, y_i) if

- $y_i = 1$ and $w^\top x_i > 0$ or
- $y_i = -1$ and $w^\top x_i < 0$

therefore x_i is well classified iff $y_i w^\top x_i > 0$
Margin: $m_i = y_i w^\top x_i$
Note that $x_\perp = x - \frac{y_i m_i}{\|w\|} w$

Maximum margin classifier definition

I want the classifier that

- classifies perfectly the dataset
- maximize the distance from its closest examples

$$w^* = \max_{w \in \mathbb{R}^d} \min_{1 \le i \le n} d(x_i, w)^2 \quad \text{subject to}$$

$$m_i > 0 \quad \forall i \in \{1 \dots n\}$$

Let call μ the smallest m_i thus we have

$$w^* = \max_{w \in \mathbb{R}^d} \min_{1 \le i \le n, \mu \ge 0} \|x_i\| - \frac{(x_i^\top w)^2}{\|w\|^2} \quad \text{subject to}$$

$$y_i w^\top x_i \ge \mu \quad \forall i \in \{1 \dots n\}$$

that is

Computation of w^*

$$w^* = \max_{w \in \mathbb{R}^d} \min_{\mu \geq 0} - \frac{\mu^2}{\|w\|^2} \quad \text{subject to}$$

$$y_i w^\top x_i \geq \mu \quad \forall i \in \{1 \dots n\}$$

Computation of w^*

$$w^* = \max_{w \in \mathbb{R}^d,\, \mu \geq 0} \frac{\mu^2}{\|w\|^2} \quad \text{subject to}$$

$$y_i w^\top x_i \geq \mu \quad \forall i \in \{1 \dots n\}$$

Note that if $y_i w^\top x_i \geq \mu$, then $y_i(\alpha w)^\top x_i \geq \alpha\mu$ and $\frac{\mu^2}{\|w\|^2} = \frac{(\alpha\mu)^2}{\|\alpha w\|^2}$ for any $\alpha \geq 0$. Therefore we have to fix the scale parameter, in particular we choose $\mu = 1$.

Computation of w^*

$$w^* = \max_{w \in \mathbb{R}^d} \frac{1}{\|w\|^2} \quad \text{subject to}$$

$$y_i w^\top x_i \geq 1 \quad \forall i \in \{1 \dots n\}$$

Computation of w^*

$$w^* = \min_{w \in \mathbb{R}^d} \|w\|^2 \quad \text{subject to}$$

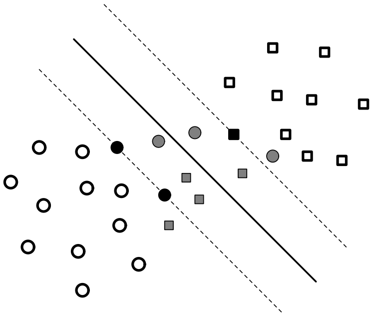
$$y_i w^\top x_i \geq 1 \quad \forall i \in \{1 \dots n\}$$

What if the problem is not separable?

We relax the constraints and penalize the relaxation

$$w^* = \min_{w \in \mathbb{R}^d} \|w\|^2 \quad \text{subject to}$$

$$y_i w^\top x_i \geq 1 \quad \forall i \in \{1 \dots n\}$$



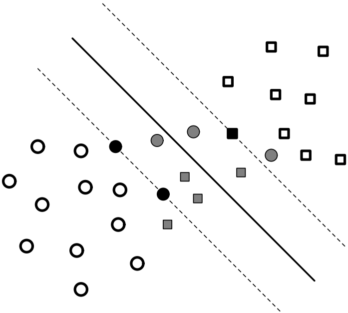
What if the problem is not separable?

We relax the constraints and penalize the relaxation

$$w^* = \min_{w \in \mathbb{R}^d, \xi_i \geq 0} \|w\|^2 + \frac{C}{n} \sum_{i=1}^{n} \xi_i \quad \text{subject to}$$

$$y_i w^\top x_i \geq 1 - \xi_i \quad \forall i \in \{1 \dots n\}$$

where C is a penalization parameter for the average error $\frac{1}{n}\sum_{i=1}^{n} \xi_i$.



Dual formulation

It can be shown that the solution of the SVM problem is of the form

$$w = \sum_{i=1}^{n} \alpha_i y_i x_i$$

where α_i are given by the solution of the following quadratic programming problem:

$$\max_{\alpha \in \mathbb{R}^n} \quad \textstyle\sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j=1}^n y_i y_j \alpha_i \alpha_j x_i^T x_j \quad i = 1, \dots, n$$

$$\text{subj to} \qquad \alpha_i \geq 0$$

- The solution requires the estimate of n rather than D coefficients
- α_i are often sparse. The input points associated with non-zero coefficients are called *support vectors*

Wrapping up

Regularized Empirical Risk Minimization

$$w^* = \min_{w \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^{n} \ell(y_i, w^\top x_i) \, + \, \lambda \|w\|^2$$

Examples of Regularization Networks

- $\ell(y,t) = (y-t)^2$ (Square loss) leads to Least Squares
- $\ell(y,t) = log(1 + e^{-yt})$ (Logistic loss) leads to Logistic Regression
- $\ell(y,t) = |1 - yt|_+$ (Hinge loss) leads to Maximum Margin Classifier

Next class

... beyond linear models!