Regularized Learning: When Data Are Not Enough

FuzzyMAD 2017

Carlos María Alaíz Gudín

Escuela Politécnica Superior
Universidad Autónoma de Madrid

Tuesday 19th December, 2017

Contents

Contents: Introduction

Supervised Learning These Days

- Nowadays, there is an increasing amount of data available.
- These data can be a powerful source to extract automatically information.
- The huge amount of data implies a lot of spurious information that can lead to erroneous conclusions.

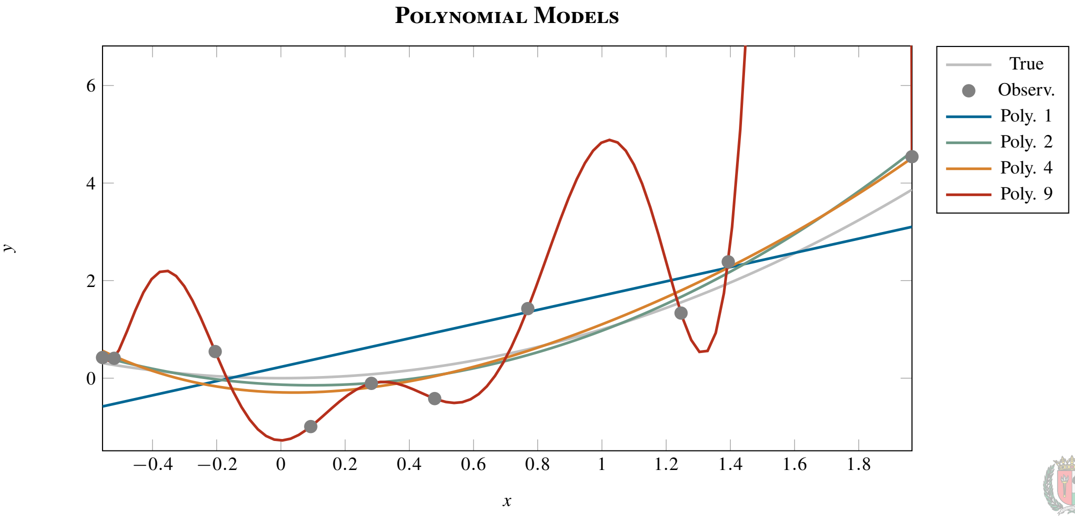
Definition (Supervised Learning)

The machine learning task of inferring a function from labelled training data.

- The problem is usually defined by a training set $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{p}$, where $x_i \in \mathbb{R}^d$ and $y_i \in \mathbb{R}$ (regression) or $y_i \in \{c_1, \dots, c_l\}$ (classification).
- The objective is to approximate an unknown function f such that $f(x_i) \approx y_i$ through a certain model.
  - This is usually stated as an optimization problem.
  - The model is defined by some parameters.
  - The parameters are selected to minimize a certain criterion.

- **Is it enough to design the model just minimizing the error with respect to the targets?**

Motivational Examples (I)



Motivational Examples (II)

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.

Bias–Variance and Regularization

UAM

Bias–Variance Trade-Off

Bias Difference between the expected prediction of the model and the correct value to be predicted.

Variance Variability of a model prediction for a given data point.

Definition (Regularization)

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)

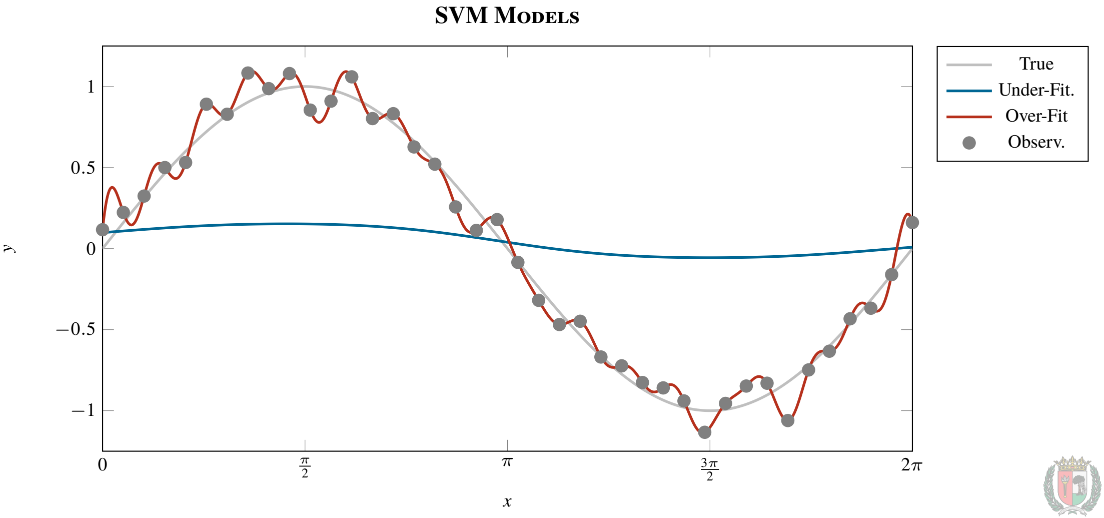
Over-Fitting

- 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.

Under-Fitting

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

Over-Fitting and Under-Fitting (II): Example



Regularized Learning

- Regularized learning consists in models trained by optimizing an objective function $\mathcal{F}$ of the form:

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

- 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.

Contents: Common Regularization Functions

Introduction

- An approach to regularize the models is needed.
- In the case of **Regularized Learning**, it is expressed as some function of the model.
  - The model is defined by its parameters.
  - A first idea is just to impose "simplicity" through the parameters.
  - Indeed, in many models the information flows multiplied by the parameters.
    - Linear Models.
    - Neural Networks.
- Controlling the norm of the parameters looks like a sensible approach.
  - **Which norm should be used?**

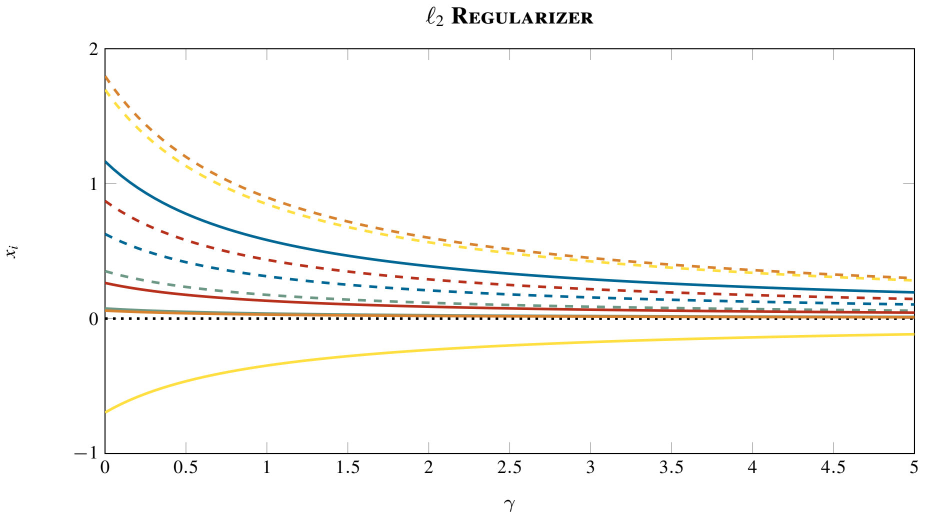
ℓ_2 Norm (I)

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

$$\mathcal{R}(x) = \|x\|_2^2 = \sum_{i=1}^{d} x_i^2 \quad .$$

- 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)



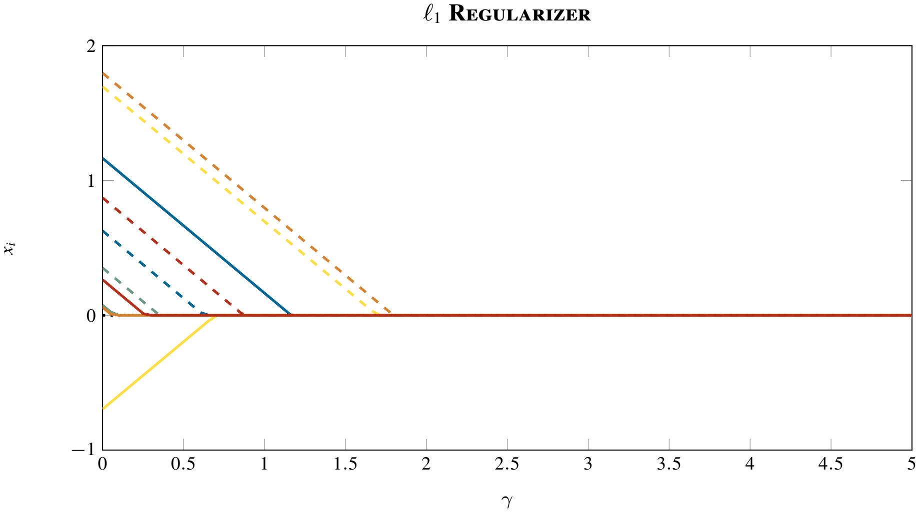
ℓ_1 Norm (I)

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

$$\mathcal{R}(x) = \|x\|_1 = \sum_{i=1}^{d} |x_i| \quad .$$

- 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)



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

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

$$x = \left(x_{1,1}, \ldots, x_{1,d_f}, \ldots, x_{d_g,1}, \ldots, x_{d_g,d_f}\right)^\top ,$$

where $x_{g,f}$ is the f-th entry of the g-th group.

  - This framework can be easily extended to groups of different sizes.
- The variable x can be seen also as a matrix with d_f rows and d_g columns.
- The regularizers should respect this structure.

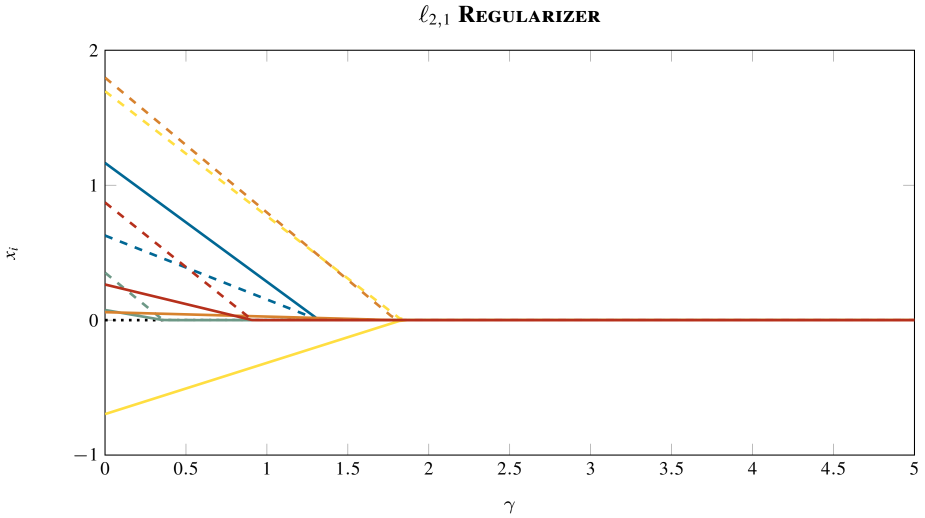
$\ell_{2,1}$ Norm (II)

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

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

  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 (non-squared) is non-differentiable around zero, and 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 (III)



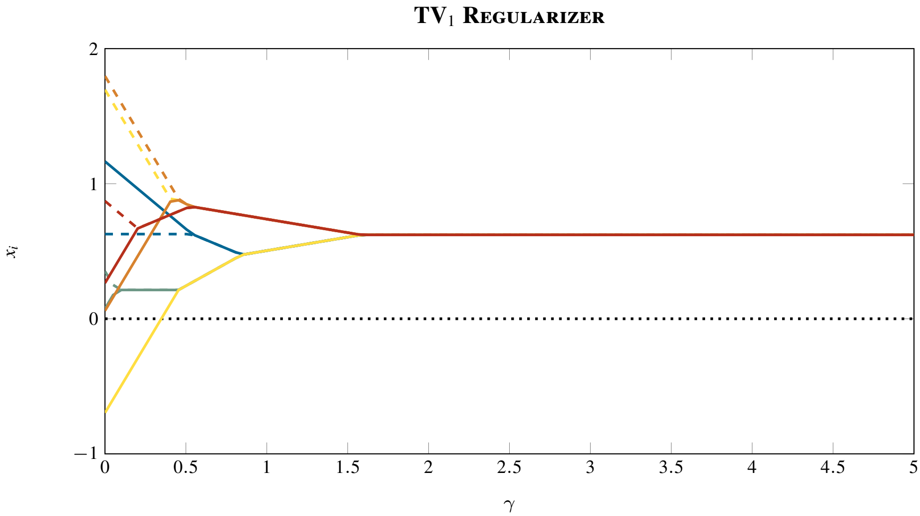
Transformed Norms: Total Variation (I)

- The Total Variation is a special family of regularizers that penalize the differences between adjacent entries.
  - It assumes some spatial location.
- It is based on transforming the variable through a differentiating matrix D, with $D_{i,i} = -1$, $D_{i,i+1} = 1$ and $D_{i,j} = 0$ elsewhere.
- The TV_1 regularizer penalizes the ℓ_1 norm of the differences:

$$\mathcal{R}(x) = \mathrm{TV}_1(x) = \|Dx\|_1 = \sum_{i=2}^{d} |x_i - x_{i-1}| \ .$$

  - The ℓ_1 norm enforces sparsity.
  - Some of the terms $x_i - x_{i-1}$ are zero, and hence $x_i = x_{i-1}$.
  - The vector x is piece-wise constant.

Transformed Norms: Total Variation (II)



Transformed Norms: Others

- There are several other approaches based on the norm of a transformed vector, $\|Mx\|$.

Graph-Based Total Variation

- An extension of the Total Variation regularizer.
- The difference 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.

$\ell_1 + \ell_2$

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

$\ell_1 + \mathrm{TV}_1$

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

Contents: Regularized Linear Models

Linear Models

- There exists an increasing interest in problems with a big amount of data (**big data**), in terms of:
  - A high dimensionality.
  - A large number of patterns (samples).
- This has resulted in the revival of the linear models.

- Notation:

  w Parameters (weights) of the model, $w \in \mathbb{R}^d$.

  X Matrix of inputs, $X \in \mathbb{R}^{p \times d}$; x_i is the input vector of the i-th pattern.

  y Vector of real outputs, $y \in \mathbb{R}^p$.

  $\tilde{y}$ Vector of predicted outputs, $\tilde{y} \in \mathbb{R}^p$.
- For an input vector $x \in \mathbb{R}^d$, the predicted output is $\tilde{y}_x = x^\top w$.

Linear Models: Mean Squared Error

- For regression problems, the most common choice for $\mathcal{E}_{\mathcal{D}}$ is the Mean Squared Error (MSE):

$$\text{MSE} = \frac{1}{2p} \|\tilde{y} - y\|_2^2 = \frac{1}{2p} \sum_{i=1}^{p} (\tilde{y}_i - y_i)^2 \ .$$

- In the case of a linear model with weights w:

$$\mathcal{E}_{\mathcal{D}}(w) = \text{MSE}(w) = \frac{1}{2p} \|Xw - y\|_2^2 \ .$$

- This term is differentiable, with Lipschitz gradient:

$$\nabla \mathcal{E}_{\mathcal{D}}(w) = \frac{1}{p} \left(X^\top X w - X y \right) \ .$$

Ridge Regression (I)

- This linear model uses the Tikhonov regularization:

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

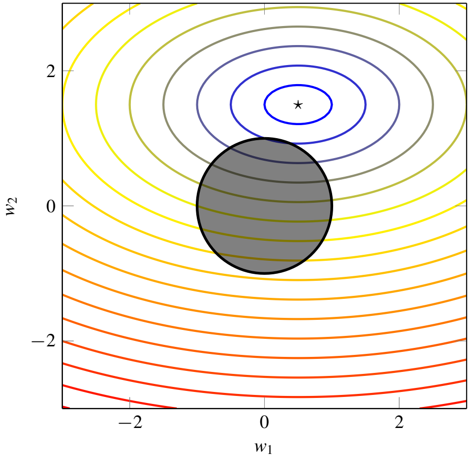
- The objective function is:

$$\mathcal{F}(w) = \mathrm{MSE}(w) + \frac{\gamma}{2d}\|w\|_2^2 \quad .$$

- The complexity of the model is controlled, but no structure is imposed.
- The resultant model typically depends on all the variables.

Ridge Regression (II)

Example Ridge Regression



Lasso (I)

- This linear model uses as regularizer the ℓ_1 norm:

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

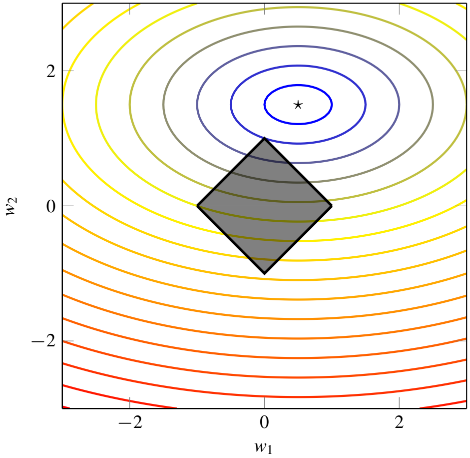
- The objective function is:

$$\mathcal{F}(w) = \mathrm{MSE}(w) + \frac{\gamma}{d}\|w\|_1 \quad .$$

- This regularizer enforces some of the coefficient 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.

Lasso (II)

Example Lasso



Elastic Net (I)

- 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:

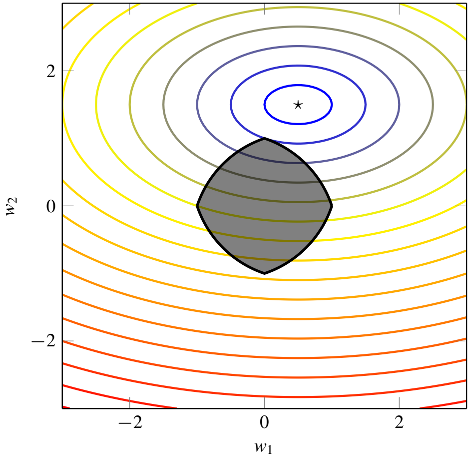
$$\mathcal{R}(w) = \frac{1}{d}\|w\|_1 + \frac{\gamma_2'}{2d}\|w\|_2^2 \ .$$

- Thus the objective function is:

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

Elastic Net (II)

Example Elastic Net



Group Variants

- In certain circumstances, some features are grouped as corresponding to the same source.
  - For example, 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 versions of the previous models can be formulated.

Group Lasso Model

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

$$\mathcal{F}(w) = \mathrm{MSE}(w) + \frac{\gamma}{d}\|w\|_{2,1} \quad .$$

Group Elastic Net Model

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

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

Fused Lasso

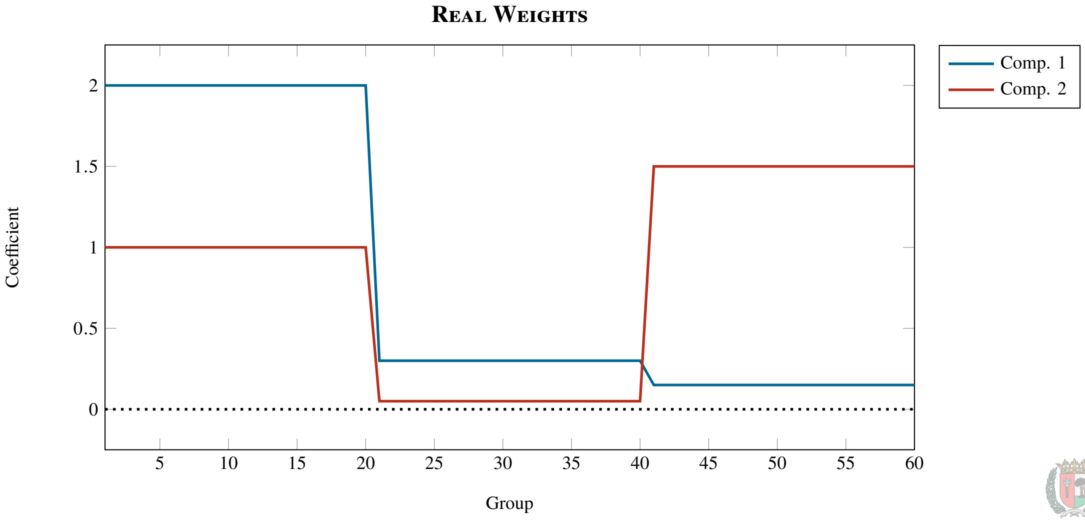
- This linear model uses as regularizer the ℓ_1 norm and the TV_1 regularizer:

$$\mathcal{R}(w) = \frac{1}{d}\|w\|_1 + \frac{\gamma_{2'}}{d}\mathrm{TV}_1(w) \ .$$

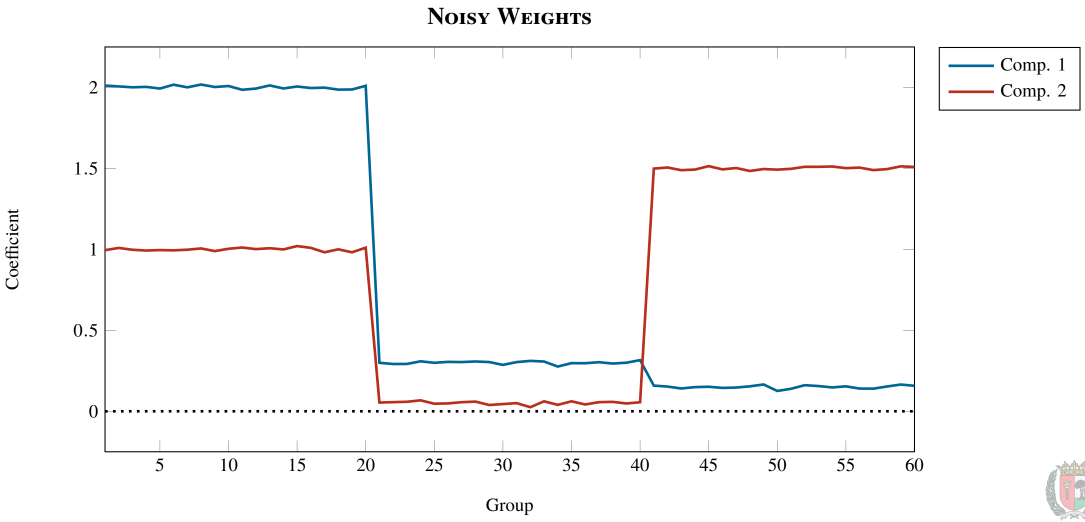
- It assumes that the features have some spatial location, and that they are ordered according to it.
  - A sensible model should assign similar coefficients to adjacent features.
- The coefficients tend to be sparse and piece-wise constant.
- The objective function is:

$$\mathcal{F}(w) = \mathrm{MSE}(w) + \frac{\gamma_1}{d}\|w\|_1 + \frac{\gamma_2}{d}\mathrm{TV}_1(w) \ .$$

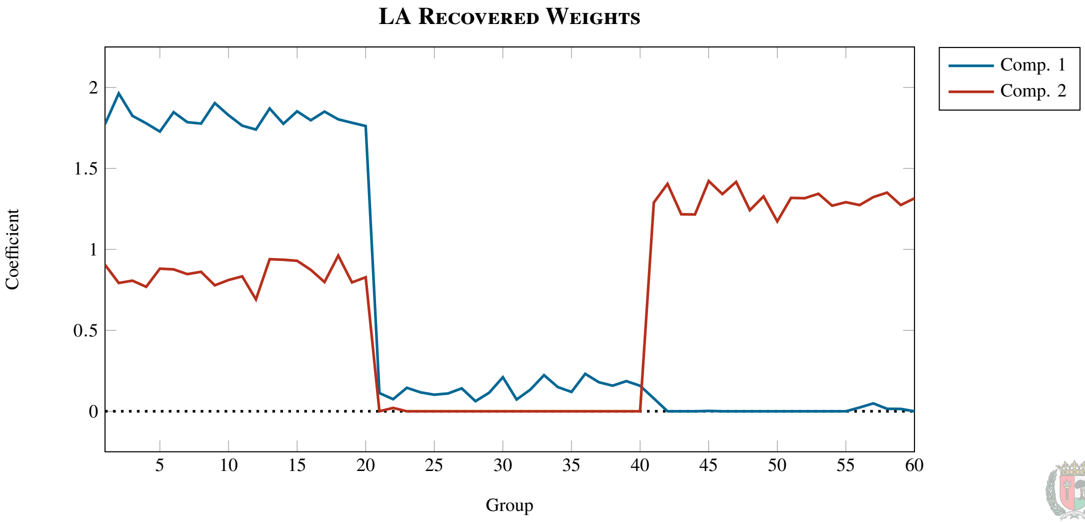
Illustration



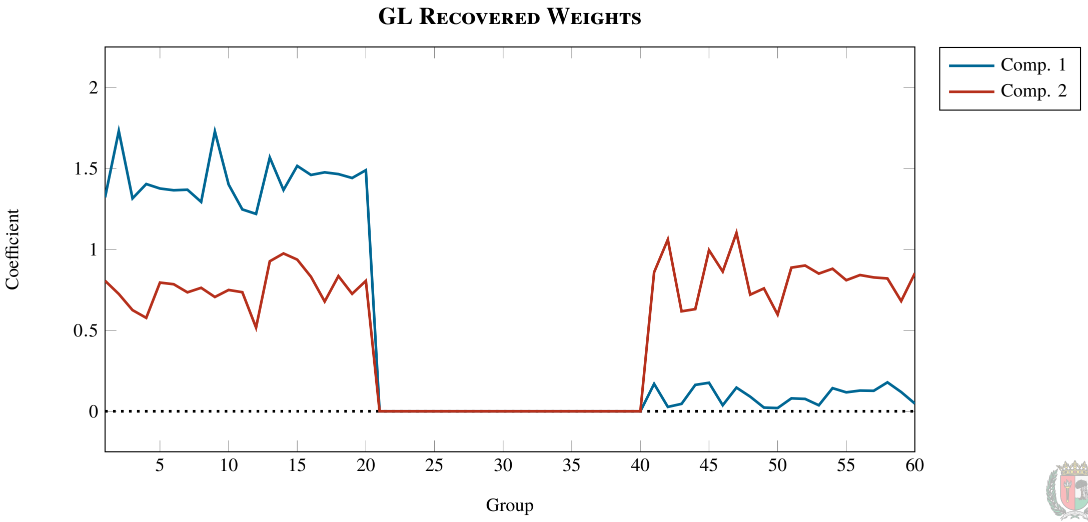
Illustration



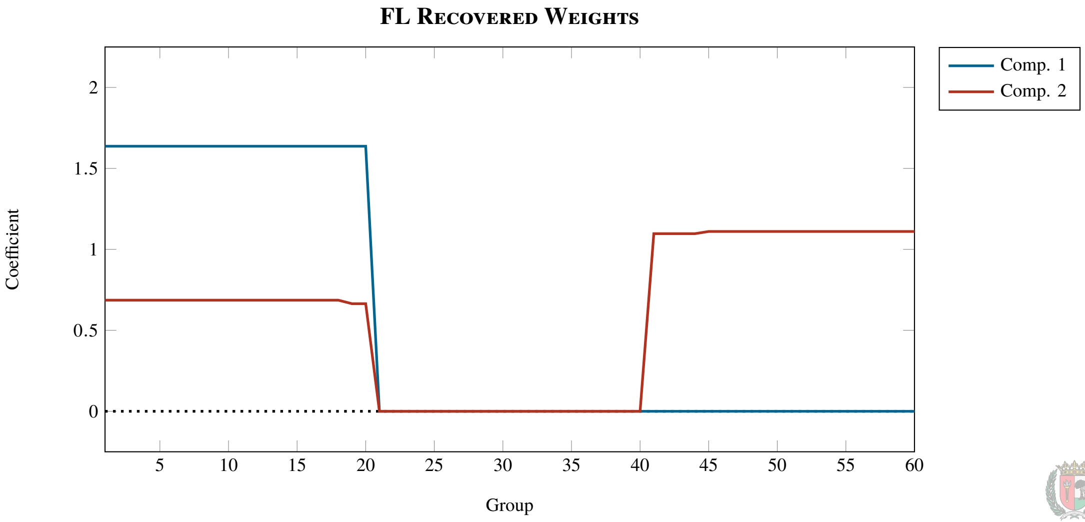
Illustration



Illustration



Illustration



Contents: Other Regularization Models and Approaches

Application to Other Models (I)

- The regularization terms defined above can be used for many other models.
  - For example, for classification linear models by changing only the error term, to get regularized logistic regression.
- They are specially well suited when the parameters can be interpreted as weights.
  - The correspondence between the inputs and the parameters is more clear.
- Depending on the optimization framework, adding these terms can be complex.
  - **Proximal Methods** provide a useful modular approach.

Application to Other Models (II): Regularized Multilayer Perceptron

- The MLP can be too complex if the number of hidden units and/or layers is large.
  - It will tend to over-fit the data.
  - Some form of regularization is needed.
- The previously defined regularization functions can be used.
- When the ℓ_2 norm is used, it becomes the classical **weight decay term**.
  - This term pushes the weights towards zero at each gradient-descent step.
- When the ℓ_1 norm is used some of the weights go to zero.
  - The network gets a structure based on the data.

Support Vector Machines

- The SVMs are regularized by their own definition.
  - Maximizing the margin corresponds to minimizing $\|w\|_2^2$.
  - Similar to the Tikhonov regularization.
- In the classification case, the error term is encoded in the constraints.
  - For hard-margin SVMs, no errors are allowed.
  - For soft-margin SVMs, the errors are minimized in the objective function.
- The regularization parameter γ is substituted by C.

$$\min_{w,b} \left\{ \frac{1}{2}\|w\|_2^2 + C\sum_{i=1}^{p} \xi_i \right\} \quad \text{s.t. } y_i\left(w^\top x_i + b\right) \geq 1 - \xi_i,\ \xi_i \geq 0 \ .$$

Generation of Patterns

Generation of Patterns

- The over-fitting problem often arises when there are no enough training data.
- A possible solution is to generate new samples.
  - Naive approach: repeat the same samples corrupting them with some noise.
  - Advanced approaches: try to fit the distribution of the data, or use some expert knowledge about the possible corruptions (such as rotations, dilations...).

Relation with Regularization

- The larger number of patterns reduces the variance of the model.
- This can be considered as a form of regularization.

Contents: Overview of Optimization

Optimization

Definition (Optimize)

To make the best or most effective use of (a situation or resource).

- An optimization problem consists in finding the best element of a certain space $\mathcal{S}$ with respect to some criteria given by an objective function $\mathcal{F}$:

$$\min_{x \in \mathcal{S}} \left\{ \mathcal{F}(x) \right\} .$$

- Many **learning machines** are trained by solving an optimization problem.
  - The minimization is done over the parameters that define the model.
  - The "best" model according to some criteria and data is obtained:

$$\min_{p \text{ pars}} \left\{ \mathcal{F}(p) \right\} .$$

  - Examples: Linear Models, SVMs, Multilayer Perceptrons...

Examples (I)

Example (Ridge Regression)

- Parameters $w \in \mathbb{R}^d$, data $\mathcal{D} = \left\{X \in \mathbb{R}^{p \times d}, y \in \mathbb{R}^p\right\}$.
- $\mathcal{E}_\mathcal{D}(w) = \frac{1}{2p}\|Xw - y\|_2^2 \quad ; \quad \mathcal{R}(w) = \frac{1}{2d}\|w\|_2^2 = \frac{1}{2d}\sum_{i=1}^d w_i^2$.

Example (Multilayer Perceptron with Weight Decay)

- Parameters $w \in \mathbb{R}^M$, data $\mathcal{D} = \left\{X \in \mathbb{R}^{p \times d}, y \in \mathbb{R}^p\right\}$.
- $\mathcal{E}_\mathcal{D}(w) = \|f_{\text{MLP}}(X, w) - y\|_2^2 \quad ; \quad \mathcal{R}(w) = \frac{1}{2d}\|w\|_2^2 = \frac{1}{2}\sum_{i=1}^d w_i^2$.

Example (Lasso)

- Parameters $w \in \mathbb{R}^d$, data $\mathcal{D} = \left\{X \in \mathbb{R}^{p \times d}, y \in \mathbb{R}^p\right\}$.
- $\mathcal{E}_\mathcal{D}(w) = \frac{1}{2p}\|Xw - y\|_2^2 \quad ; \quad \mathcal{R}(w) = \frac{1}{d}\|w\|_1 = \frac{1}{d}\sum_{i=1}^d |w_i|$.

Examples (II)

Example (Ridge Regression)

- Closed-form solution: $w^* = \left(X^\top X + \frac{\gamma p}{d} I\right)^{-1} X^\top y$.

Example (Multilayer Perceptron with Weight Decay)

- Iterative solution: $w^{(k+1)} = w^{(k)} - \lambda^{(k)} \nabla \mathcal{E}_{\mathcal{D}}\left(w^{(k)}\right) - \lambda^{(k)} \frac{\gamma}{d} w^{(k)}$.
- The current solution is updated with a gradient-descent step.

Example (Lasso)

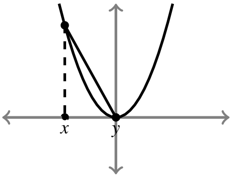
- $\mathcal{R}(w)$ is not differentiable, its gradient is not defined at every point.
- An alternative to gradient-descent is needed:
  - Proximity operator $\operatorname{prox}_{\mathcal{R}}$.
    - Between the gradient-descent step and the projection.
  - Iterative solution: $w^{(k+1)} = \operatorname{prox}_{\lambda^{(k)} \gamma \mathcal{R}} \left(w^{(k)} - \lambda^{(k)} \nabla \mathcal{E}_{\mathcal{D}}\left(w^{(k)}\right)\right)$.

Convex Optimization

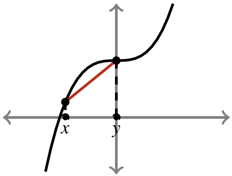
Definition (Convexity)

An extended real function f is called convex if $\operatorname{dom} f$ is a convex set, and $\forall x, y \in \mathbb{E}$ and $\forall t \in [0, 1]$

$$f(tx + (1-t)y) \leq tf(x) + (1-t)f(y) \quad .$$



Convex function.



Non-convex function.

- The convexity of a problem guarantees the uniqueness of the minimum.
  - Regularized learning is often based on convex formulations.

Gradient-Based Optimization

- If the objective function $\mathcal{F}$ is convex and differentiable, a minimum x^* is characterized by the zeros of the gradient:
$$\nabla \mathcal{F}(x^*) = 0 \ .$$
  - In some cases, this equation has a closed-form solution.
  - The classical gradient-descent step can also be applied:
$$x^{(k+1)} = x^{(k)} - \lambda^{(k)} \nabla \mathcal{F}\left(x^{(k)}\right) \ .$$
  - There are other methods that use higher order information (e.g. Newton).
  - There are projected methods to deal with constraints.

- This only makes sense for differentiable functions.
  - It is limited to simple cases.

Ad Hoc Methods

- There are specific algorithms for many regularized models.

SVMs

- The dual problem is usually solved.
- The most popular approach is SMO, a coordinate descent method that optimizes over two dual variables at each iteration.

Lasso

- The problem is non-differentiable.
- When only one variable is considered, there exists a closed-form solution.
  - Coordinate descent methods are often the choice.

- A general framework can make the regularized model design easier.
  - Proximal Methods.

Proximity Operator (I)

Definition (Proximity Operator)

For a function $f \in \Gamma_0(\mathbb{E})$, its proximity operator, prox_f, is the function defined as the solution, at each point $x \in \mathbb{E}$, of the problem:

$$\operatorname{prox}_f(x) = \arg\min_{\hat{x} \in \mathbb{E}} \left\{ \frac{1}{2} \|\hat{x} - x\|^2 + f(\hat{x}) \right\} .$$

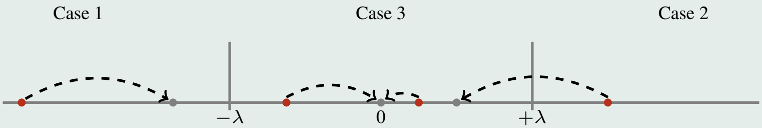
- It is also the resolvent of the subdifferential.
- It can be interpreted as a generalization of the gradient-descent step, and of the projection operator.
- The fixed points of the proximity operator are the minima of the objective function.

Proximity Operator (II): Some Examples

Example (Proximity Operator of the ℓ_1 Norm.)

- Absolute value function, $f : \mathbb{R} \to \mathbb{R}$, $\lambda f(x) = \lambda |x|$.

$$\operatorname{prox}_{\lambda f}(x) = \operatorname{soft}_\lambda(x) = \operatorname{sign}(x)\left(|x| - \lambda\right)^+ = \begin{cases} x + \lambda & \text{if } x \leq -\lambda \ , \\ 0 & \text{if } -\lambda \leq x \leq +\lambda \ , \\ x - \lambda & \text{if } x \geq +\lambda \ . \end{cases}$$

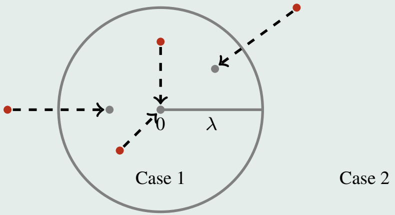


Proximity Operator (III): Some Examples

Example (Proximity Operator of the ℓ_2 Norm.)

- Euclidean (ℓ_2) norm function, $f : \mathbb{R}^d \to \mathbb{R}$, $\lambda f(x) = \lambda \|x\|_2$.

$$\operatorname{prox}_{\lambda f}(x) = x\left(1 - \frac{\lambda}{\|x\|_2}\right)^{+} = \begin{cases} 0 & \text{if } \|x\|_2 \leq \lambda \ , \\ x\left(1 - \frac{\lambda}{\|x\|_2}\right) & \text{if } \|x\|_2 \geq \lambda \ . \end{cases}$$



Proximity Point Algorithm

- A first approach to minimize a certain function $f \in \Gamma_0(\mathbb{E})$ is to iterate the proximity operator.

Proximal Point

Input: $f \in \Gamma_0(\mathbb{E})$;
Output: $x^{(k)} \simeq x^* = \arg\min_{x \in \mathbb{E}} \{f(x)\}$;
Initialization: $x^{(0)} \in \mathbb{E}$;
  set $\lambda^{(k)} \in \left(\lambda^{\min}, \lambda^{\max}\right)$;
  for $k = 0, 1, \ldots$ **do**
    $x^{(k+1)} \leftarrow \operatorname{prox}_{\lambda^{(k)} f}\left(x^{(k)}\right)$;
  end for

- This method requires the computation of the proximity operator.
  - It is an optimization problem itself.
- Alternative: Proximal Methods.
  - Exploit the structure of the problem.
  - **Proximal Methods** are based precisely on a convenient splitting of the objective function.

ISTA

- The Iterative Shrinking–Thresholding Algorithm (ISTA) is a method to minimize the sum of a **smooth** and a **non-smooth** functions.

ISTA

Input: $f_1 \in \Gamma_0(\mathbb{E})$; f_2 convex with ∇f_2 β-Lipschitz;
Output: $x^{(k)} \simeq x^* = \arg\min_{x\in\mathbb{E}} \{f_1(x) + f_2(x)\}$;
Initialization: $x^{(0)} \in \mathbb{E}$;
for $k = 0, 1, \ldots$ **do**
$x^{(k+1)} \leftarrow \operatorname{prox}_{\frac{1}{\beta} f_1}\left(x^{(k)} - \frac{1}{\beta}\nabla f_2\left(x^{(k)}\right)\right)$;
end for

- There are approaches to estimate β automatically.

FISTA

- The Fast ISTA (FISTA) is a modification that accelerates the convergence.
- It is based on the improved gradient method of Nesterov.

FISTA

Input: $f_1 \in \Gamma_0(\mathbb{E})$; f_2 convex with ∇f_2 β-Lipschitz;
Output: $x^{(k)} \simeq x^* = \arg\min_{x \in \mathbb{E}} \{f_1(x) + f_2(x)\}$;
Initialization: $x^{(0)} \in \mathbb{E}$;
$y^{(1)} \leftarrow x^{(0)}$; $t^{(0)} \leftarrow 1$;
for $k = 0, 1, \ldots$ **do**

$$x^{(k)} \leftarrow \operatorname{prox}_{\frac{1}{\beta} f_1}\left(y^{(k)} - \frac{1}{\beta} \nabla f_2\left(y^{(k)}\right)\right);$$

$$t^{(k+1)} \leftarrow \frac{1}{2}\left(1 + \sqrt{1 + 4\left(t^{(k)}\right)^2}\right);$$

$$y^{(k+1)} \leftarrow x^{(k)} + \frac{t^{(k)} - 1}{t^{(k+1)}}\left(x^{(k)} - x^{(k-1)}\right);$$

end for

- There are approaches to estimate β automatically.
- The current interest on Nesterov accelerations has produced several recent improvements.

Other Proximal Methods

- There are many more Proximal Methods.

Douglas–Rachford

- Method to minimize the sum of two non-smooth functions.
- It is based on the iteration of a fixed equation.

Dykstra

- Method to minimize the sum of two non-smooth functions, f_1 and f_2, plus a deviation term that represent the distance to a reference point, $\frac{1}{2}\|\cdot - r\|^2$:

$$\min_{x \in \mathbb{E}} \left\{ \frac{1}{2}\|x - r\|^2 + f_1(x) + f_2(x) \right\} .$$

- The unique solution is precisely $\mathrm{prox}_{f_1+f_2}$, which can be hard to compute directly.
- It therefore allows to compute complex proximity operator decomposing them into “easier” ones.
- There are extensions to compute the proximity operator of the sum of an arbitrary number of functions.

FISTA for Regularized Linear Models (I): Gradients and Proximity Operators

MSE

$$f_2(w) = \mathrm{MSE}(w) \implies \nabla f_2(w) = \frac{1}{p}\left(X^\top X w - Xy\right) .$$

MSE + ℓ_2 Norm

$$f_2(w) = \mathrm{MSE}(w) + \frac{\gamma_2}{2d}\|w\|_2^2 \implies \nabla f_2(w) = \frac{1}{p}\left(X^\top X w - Xy\right) + \frac{\gamma_2}{d} w .$$

ℓ_1 Norm

$$f_1(w) = \frac{\gamma}{d}\|w\|_1 \implies \left(\mathrm{prox}_{\lambda f_1}(w)\right)_i = \mathrm{sign}\left(w_i\right)\left(|w_i| - \lambda\frac{\gamma}{d}\right)^+ .$$

$\ell_{2,1}$ Norm

$$f_1(w) = \frac{1}{d}\|w\|_{2,1} \implies \left(\mathrm{prox}_{\lambda f_1}(w)\right)_{g,f} = \mathrm{sign}\left(w_{g,f}\right)\left(|w_{g,f}| - \lambda\frac{\gamma}{d\|w_g\|_2}\right)^+ .$$

FISTA for Regularized Linear Models (II)

Model	$\mathcal{E}_{\mathcal{D}}(\mathbf{w})$	$\gamma\mathcal{R}(\mathbf{w})$	**Solution**
RR	$\frac{1}{2p}\|Xw-y\|_2^2$	$\frac{\gamma}{2d}\|w\|_2^2$	$w^* = \left(X^\top X + \frac{\gamma p}{d} I\right)^{-1} X^\top y$
LA	$\frac{1}{2p}\|Xw-y\|_2^2$	$\frac{\gamma}{d}\|w\|_1$	FISTA
ENet	$\frac{1}{2p}\|Xw-y\|_2^2$	$\frac{\gamma_1}{d}\|w\|_1 + \frac{\gamma_2}{2d}\|w\|_2^2$	FISTA
GL	$\frac{1}{2p}\|Xw-y\|_2^2$	$\frac{\gamma}{d}\|w\|_{2,1}$	FISTA
GENet	$\frac{1}{2p}\|Xw-y\|_2^2$	$\frac{\gamma_1}{d}\|w\|_{2,1} + \frac{\gamma_2}{2d}\|w\|_2^2$	FISTA
FL	$\frac{1}{2p}\|Xw-y\|_2^2$	$\frac{\gamma_1}{d}\|w\|_1 + \frac{\gamma_2}{d}\mathrm{TV}_1(w)$	FISTA

Model	$\mathbf{f_1}$	$\left(\mathrm{prox}_{\lambda \mathbf{f_1}}(\mathbf{w})\right)_{\mathbf{g},\mathbf{f}}$
LA	$\frac{\gamma}{d}\|w\|_1$	$\mathrm{sign}(w_{g,f})\left(\lvert w_{g,f}\rvert - \lambda\frac{\gamma}{d}\right)^+$
ENet	$\frac{\gamma_1}{d}\|w\|_1$	$\mathrm{sign}(w_{g,f})\left(\lvert w_{g,f}\rvert - \lambda\frac{\gamma_1}{d}\right)^+$
GL	$\frac{\gamma}{d}\|w\|_{2,1}$	$\mathrm{sign}(w_{g,f})\left(\lvert w_{g,f}\rvert - \lambda\frac{\gamma}{d\|w_g\|_2}\right)^+$
GENet	$\frac{\gamma_1}{d}\|w\|_{2,1}$	$\mathrm{sign}(w_{g,f})\left(\lvert w_{g,f}\rvert - \lambda\frac{\gamma_1}{d\|w_g\|_2}\right)^+$
FL	$\frac{\gamma_1}{d}\|w\|_1 + \frac{\gamma_2}{d}\mathrm{TV}_1(w)$	Dual problem + Soft-thresholding

Contents: Conclusions

Conclusions

- **Regularized Learning** permits to adapt models avoiding over-fitting, inducing a certain structure and/or using prior knowledge.
- Regularized models minimize two terms: an **error** term and a **regularization** term.
- Several **regularization functions**:
  - ℓ_2 Norm.
  - ℓ_1 Norm.
  - $\ell_{2,1}$ Norm.
  - Transformations, combinations...
- The regularizers allow to define a family of **regularized linear models**, but it can also be extended to other types of learning machines.
- There are other approaches: SVMs, pattern generation, ensembles...
- A general framework is required to solve non-differentiable convex problems: **proximal methods**.
- The gradient descent step is substituted by the **proximity operator**.
- Several algorithms allow to take advantage of the **problem structure**.

Regularized Learning: When Data Are Not Enough

Carlos María Alaíz Gudín

Thank you for your attention.