Machine Learning (Lab support)

Regularization and feature selection

Abdelkrime Aries

Laboratoire de la Communication dans les Systèmes Informatiques (LCSI)
École nationale Supérieure d'Informatique (ESI, ex. INI), Algiers, Algeria

Academic year: 2024-2025

Machine Learning (Lab support)

Regularization and feature selection: Introduction

- You saw in the lecture ...
  - the overfitting problem;
  - regularization by penalty (L2 as an example);
  - the different approaches to feature selection.
- In this lab support, we will present ...
  - some regularization approaches;
  - three penalty regularization techniques;
  - a reminder about feature selection approaches
  - detailing the techniques of each approach (with example from scikit-learn)

Machine Learning (Lab support)

Regularization and feature selection: Plan

1. **Regularization**
   - L2 Loss
   - L1 Loss
   - ElasticNet
2. **Feature selection**
   - Filter
   - Embedded
   - Wrapper

Section 1

Regularization

Regularization and feature selection

Regularization

- used to reduce overfitting
- can cause faster convergence
- **By augmentation**: add more data
  - automatic data generation
  - Ex. Generate new images by rotation
- **Early stopping**: use validation in the stopping decision
  - train on one dataset and use another for validation in each iteration
  - when the validation error increases, stop
- **By penalty**: add a penalty to the objective function
  - reduce model complexity
  - by adding another constraint on the parameters (penalty)

Regularization and feature selection

Regularization: By penalty

$$\begin{cases} \min J_{\text{cost}}(\theta) \\ \wedge \\ \min J_{\text{complexity}}(\theta) \end{cases} \Rightarrow \min J_{\text{cost}}(\theta) + J_{\text{complexity}}(\theta)$$

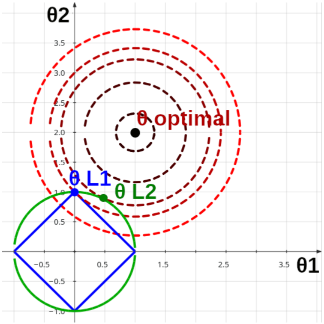
- minimize the complexity by increasing the bias: θ_0 has no constraint, so the model gives it more importance during training
- the complexity penalty uses a hyper-parameter λ
    - if λ is too big, the model will be too simple (not dependent on attributes). So, ***underfitting***.
    - if λ is too small, the model will be too complex (too dependent on attributes). So, ***overfitting***.

Regularization and feature selection: Regularization

L2 Loss: Tikhonov regularization, L2 loss

$$J_{L2} = \frac{\lambda}{2M} \sum_{j=1}^{N} \theta_j^2$$

- θ_0 is not affected by regularization
- θ_j converges to 0, but does not equal 0
- $\lambda \to \infty \Rightarrow \theta_j \to 0$
- $L2$ tries to pull the values of θ inside a sphere
- the size of the sphere is inversely proportional to λ

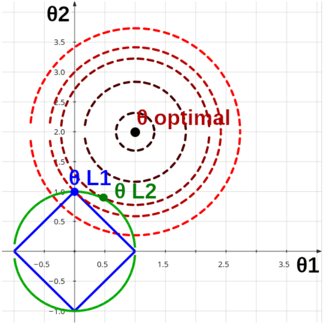


Regularization and feature selection: Regularization

L1 Loss: L1 loss

$$J_{L1} = \frac{\lambda}{M} \sum_{j=1}^{N} |\theta_j|$$

- least absolute shrinkage and selection operator
- θ_0 is not affected by regularization
- θ_j is canceled after a few iterations (it will last depending on the importance of its attribute)
- $it \rightarrow \infty \Rightarrow \theta_j = 0$
- $L1$ tries to pull the values of θ inside a cube
- the size of the cube is inversely proportional to λ



Regularization and feature selection: Regularization

L1 Loss: Optimization

- L1 is not differential in 0
- Several techniques have been proposed to find the solution path $L1$
  - Subgradient methods; ex. least-angle regression (LARS)
  - Coordinate descent
  - Proximal gradient methods (by operator ***soft thresholding***)

$$S_{\frac{\lambda}{M}}(\theta_j) = \begin{cases} \theta - \frac{\lambda}{M} & \theta_j > \frac{\lambda}{M} \\ 0 & \theta_j \in [-\frac{\lambda}{M}, \frac{\lambda}{M}] \\ \theta + \frac{\lambda}{M} & \theta_j < -\frac{\lambda}{M} \end{cases}$$

- In case of the proximal gradient, the parameters are updated as follows:
- $\theta = S_{\frac{\lambda}{M}}(\theta - \alpha \frac{\partial}{\partial \theta} J_{\text{cost}}(\theta))$

Regularization and feature selection: Regularization
ElasticNet

$$J_{EN} = \frac{\lambda}{M} \sum_{j=1}^{N} \left(r|\theta_j| + \frac{(r-1)}{2} \theta_j^2 \right)$$

- θ_0 is not affected by regularization
- $r \in [0, 1]$ controls the percentage of $L1$ regularization

Section 2

Feature selection

Regularization and feature selection

Feature selection

- can reduce overfitting: the model can overfit over noise features
- can improve system accuracy
- reduce training time
- **Filter**: independently of the estimator, the best features are selected based on univariate statistical tests
- **Embedded**: the best features are selected during training
- **Wrapper**: the best features are selected for a given estimator before training

Regularization and feature selection: Feature selection

Filter: Score

	Input	
Output	Numerical	Categorical
Numerical (Regression)	Pearson	Mutual information
Categorical (Classification)	ANOVA, Mutual information	Chi-2, Mutual information

- **Pearson**: sklearn.feature_selection.f_regression
- **ANOVA**: sklearn.feature_selection.f_classif
- **Chi2**: sklearn.feature_selection.chi2
- **Mutual information**: feature_selection.mutual_info_classif
- **Mutual information**: feature_selection.mutual_info_regression

Regularization and feature selection: Feature selection

Filter: Selection

- Based on previous scores, the most important features can be selected in several ways
- **Number**: number of features to select
  - sklearn.feature_selection.SelectKBest
- **Percentile**: percentile of highest scores.
  - sklearn.feature_selection.SelectPercentile
- **P-value**: max threshold of accepted p-values
  - sklearn.feature_selection.SelectFpr

Regularization and feature selection: Feature selection

Filter: One-Way ANOVA (logic)

- Given a feature A (M samples) with numerical values
- the values are split on sets A_j where j is a class among N classes. We call them ***Treatments***
- an attribute is representative (well correlated) of the output classes, if ...
  - values of the same class have less variance (intra-class variance)
  - class values have more variance with the rest (inter-class variance)
  - **SO**, the ratio (inter-class variance)/(intra-class variance) must be large
  - We call this ratio: ***F-value*** of ANOVA

Regularization and feature selection: Feature selection

Filter: One-Way ANOVA (math)

- Correlation factor: $CF = \frac{(\sum_{ij} A_{ij})^2}{M}$
- Total Sum of Squares: $TotalSS = \sum_{ij} A_{ij}^2 - CF$
- Treatment Sum of Squares: $TreatmentSS = \sum_j \frac{(\sum_i A_{ij})^2}{|A_j|} - CF$
- Error Sum of Squares: $ErrorSS = TotalSS - TreatmentSS$
- Mean of Squares Between: $MSB = \frac{TreatmentSS}{(N-1)}$
- Mean of Squares Within: $MSW = \frac{ErrorSS}{(M-N)}$
- $Fvalue = \frac{MSB}{MSW}$

Regularization and feature selection: Feature selection
Embedded

- **Decision trees**
  - In each node, the best split feature is chosen.
  - Certain features will not be considered.
  - Finally, the tree will only use features with high separation capacity.
- **L1 regularization**
  - It forces parameters to have small values.
  - In L1, certain parameters are set to 0 after convergence.
  - In this case, during the estimation the model does not take into account features with parameters of zero.

Regularization and feature selection: Feature selection
Wrapper

- use an estimator and try to find feature combination giving more performance sklearn.feature_selection.SequentialFeatureSelector
- **Ascending (Forward selection)**
  - the initial set of features is empty
  - start by selecting a single feature that maximizes cross-validation
  - add more features in the same way until reaching the desired number
  - SequentialFeatureSelector(direction="forward")
- **Backward elimination**
  - the initial set of features equals to N
  - start by eliminating a single feature which minimizes cross-validation
  - eliminate more features in the same way until reaching the desired number
  - SequentialFeatureSelector(direction="backward")

The end of the ~~world~~ presentation

Stop scrolling

...

It is the end!