AOS1 — Fall 2019

Bayesian regularization — Project

The project consists in implementing one of the two classification methods seen in the courses. You have the choice between quadratic discriminant analysis and binary logistic regression. You must implement the regularized version of the chosen method, and then apply it to the datasets provided.

You will have to provide a report (LaTeX recommended, 10 pages maximum) which *briefly* recalls the main equations behind each classification method ; which describes your implementation ; and which provides a presentation and some comments on the results obtained.

You will also provide the Python code with your functions and the script with your training/test procedure.

The projects are due for Dec. 20th, 2019.

1 Quadratic Discriminant Analysis (QDA)

Prior choice Recall that the Gaussian-inverse-Wishart prior is

$$\mu_k \sim \mathcal{N}(\mu_{kp}, \Sigma_k/\kappa_{kp}), \quad \Sigma_k \sim IW(\nu_{kp}, \Lambda_{kp});$$

the meta-parameters of these distributions should satisfy $\kappa_{kp} > 0$ and $\nu_{kp} > d-1$ (with d standing for the dimension of the space). The covariance matrix for μ_k will be finite provided $\nu_{kp} > d+1$.

The priors can be interpreted as information provided by "virtual data" ; they should make it possible to make model estimation more robust without involving any information regarding the actual distribution of the data (remember that while with simulated data the actual distribution is known, this will not be the case for real datasets).

You have to imagine a way to provide meaningful estimates using only the observed training data at your disposal. You can test various values for the shrinkage parameter and degrees of freedom and analyze the influence on the results.

Code The function for training QDA should take the following inputs :
- the $n \times d$ matrix Xapp of instances $\boldsymbol{x}_i \in \mathbb{R}^p$ (instances are represented row-wise),
- the $n \times K$ matrix zapp of (row-wise) binary class indicators ;
- the $K \times p$ matrix mprior of expectations μ_{kp} for the Gaussian prior on μ_k,
- the shrinkage parameter df_exp of the Gaussian prior (κ_{kp}),
- the $p \times p \times K$ matrix Sprior of covariance matrices Λ_{kp}^{-1} for the inverse-Wishart prior on Σ_k,
- the number df_cov of degrees of freedom of the inverse-Wishart prior ($\nu_{kp} \geq d-1$).

It should provide :
- the $1 \times K$ vector pi of class proportions,
- the $K \times p$ matrix mu of estimated expectations,
- the $p \times p \times K$ matrix Sig of estimated covariance matrices.

The function for evaluating test data should take the following inputs :
- the $n \times p$ matrix Xtst of test instances $\boldsymbol{x}_i \in \mathbb{R}^p$ (instances are represented row-wise),
- the $1 \times K$ vector pi of class proportions, the $K \times p$ matrix mu of estimated expectations, and the $p \times p \times K$ matrix Sig of estimated covariance matrices ;

and it should provide
- the $n \times K$ matrix prob of estimated class posterior probabilities,
- the $n \times 1$ vector pred of corresponding decisions.

Tests Process the data by varying the amount of available training data. Compare the results obtained without and with priors. You will consider the following datasets : Optdigits, Pageblocks, Satimage, Segment.

2 Logistic regression

Prior choice You will test both priors on the vector parameter $\boldsymbol{\beta}$ mentioned during the course : the Gaussian prior and the Laplacean prior.

Code The logistic regression code provided by the function logfit.m takes the following inputs :
- the $n \times d$ matrix X of instances $\boldsymbol{x}_i \in \mathbb{R}^d$ (instances are represented row-wise),
- the $n \times K$ matrix y of binary class indicators z_{ik} (row-wise, possibly imprecise : whenever the constraint $\sum_k z_{ik} = 1$ is relaxed, they provide the sets of possible classes for the instances) ;
- a flag parameter intercept indicating whether an intercept should be used in the model, or not ;
- the scalar lambda used for regularization,
- the precision scalar precision used to check for convergence,
- the initial parameter betaini standing for the initial vector of parameters $\boldsymbol{\beta}^{(0)}$.

It should provide :
- the $p \times 1$ (no intercept) or $(p+1) \times 1$ (intercept) vector beta which corresponds to the maximum-likelihood estimate $\widehat{\boldsymbol{\beta}}$ for $\boldsymbol{\beta}$ (or maximum a posteriori if $\lambda > 0$),
- the log-likelihood logL corresponding to $\log L(\widehat{\boldsymbol{\beta}}; \cdots)$ (if $\lambda = 0$) or $\pi_\beta(\widehat{\boldsymbol{\beta}}; \cdots)$ (if $\lambda > 0$).

The test function should take as input
- the $n \times p$ matrix Xtst of test instances $\boldsymbol{x}_i \in \mathbb{R}^p$ (instances are represented row-wise),
- the $p \times 1$ vector (or $(p+1) \times 1$ vector if intercept) beta of estimated parameters ;

and it should provide
- the $n \times K$ matrix prob of estimated class posterior probabilities,
- the $n \times 1$ vector pred of corresponding decisions.

2.1 Tests

Process the data by varying the amount of available training data. Compare the results obtained without and with priors. You will consider the following datasets : Breastcancer, Ionosphere, Sonar, Spambase.