Generalized Low Rank Model for Feature Selection

Zhihao Huang

University of Michigan

December, 2018

Outline

Outline

Motivation

Principal Component Analysis (PCA)

- Identifies a low rank matrix that minimizes the error based on least-squares

Principal Component Analysis

Suppose $A \in \mathbb{R}^{n\times p}$, $X \in \mathbb{R}^{n\times r}$ and $Y \in \mathbb{R}^{p\times r}$, the principal component analysis optimize the following object function

$$\text{argmin}_{X,Y} \|A - XY^T\|_F^2$$

Generalized Low Rank Model (GLRM)

- Extension of PCA
- Add regularization on low-dimensional factors
- Change loss function on approximation error

Motivation

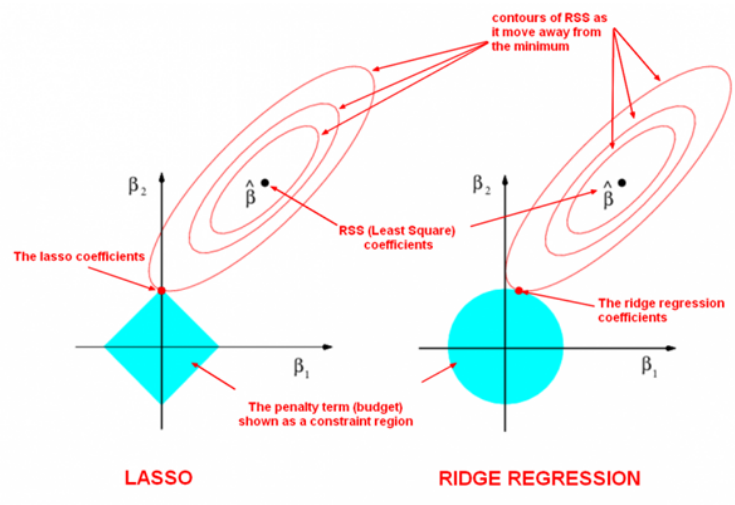
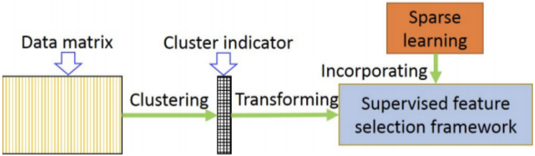
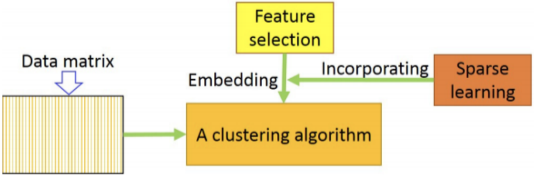


Figure 1: Sparse Learning

Motivation: Unsupervised Feature Selection



(a) Existing sparse learning based unsupervised feature selection method



(b) Embedded unsupervised feature selection

Figure 2: Difference between existing sparse learning based and embedded unsupervised feature selection

Goals and Challenges

Goals

- Dimensional reduction with concentration on feature selection
- Build a C++ program based on a generalized low rank matrix decomposition algorithm along with user-defined generalizations

Challenges

- Optimization algorithm for sparse learning
- Fast performance
- Generalized code framework

Outline

Models

Basis framework:

Generalized Low Rank Model

Suppose $A \in \mathbb{R}^{n\times p}$, $X \in \mathbb{R}^{n\times r}$ and $Y \in \mathbb{R}^{p\times r}$, the glrm optimize the following object function

$$\operatorname{argmin}_{X,Y} \operatorname{loss}(A, X, Y) + \lambda_1 \operatorname{reg}_1(X) + \lambda_2 \operatorname{reg}_2(Y)$$

Models

Quadratic regularized PCA (QRPcaClass)

$$\text{argmin}_{X,Y} \|A - XY^T\|_F^2 + \lambda_1 \|X\|_F^2 + \lambda_2 \|Y\|_F^2$$

Quadratic loss feature selection (QSPcaClass)

$$\text{argmin}_{X,Y} \|A - XY^T\|_F^2 + \lambda \|Y\|_{2,1}$$

$$s.t. \quad X^T X = I$$

Robust feature selection (RSPcaClass)

$$\text{argmin}_{X,Y} \|A - XY^T\|_{2,1} + \lambda \|Y\|_{2,1}$$

$$s.t. \quad X^T X = I$$

Notations: $\|A\|_F^2 = \sum_{i=1}^n \sum_{j=1}^p A_{ij}^2$, $\|Y\|_{2,1} = \sum_{i=1}^p \sqrt{\sum_{j=1}^r Y_{ij}^2}$

Optimization

Alternating Direction Method of Multiplier (ADMM)

Applied to solve the following optimization problem

$$\min_x f(x) + g(x)$$

$$\Leftrightarrow \min_{x,y} f(x) + g(y), \quad s.t. \quad c := (x - y) = 0$$

- Attack the problem with constrained optimization
- Separate the problem into smaller pieces

Augmented Lagrangian

$$\Leftrightarrow \min_{x,y} f(x) + g(y) + \frac{\mu}{2} c^2 + \lambda_l \cdot c$$

- Penalty coefficient updates: $\mu^{(k+1)} := \rho \cdot \mu^{(k)} \quad (\rho > 1)$
- Lagrangian multiplier updates: $\lambda_l^{(k+1)} := \lambda_l^{(k)} + \mu^{(k)} * c$

Example: algorithm for robust feature selection

Quadratic loss feature selection

$$\text{argmin}_{X,Y} \|A - XY^T\|_{2,1} + \lambda \|Y\|_{2,1}$$
$$s.t. \quad X^T X = I$$

Equivalent optimization problem:

$$\text{argmin}_{E,X,Y} \|E\|_{2,1} + \lambda \|Y\|_{2,1}$$
$$s.t. \quad X^T X = I, E = A - XY^T$$

Augmented Lagrangian (object function to optimize):

$$\text{argmin}_{E,X,Y} \|E\|_{2,1} + \lambda \|Y\|_{2,1} + < L, A - E - XY^T > + \frac{\mu}{2} \|A - E - XY^T\|_F^2$$
$$s.t. \quad X^T X = I$$

where L is the lagrangian multiplier, mu is the penalty coefficient and $< \cdot, \cdot >$ is the matrix inner product. Specifically, $< A, B >= \text{tr}(A^T B)$.

Example: algorithm for robust feature selection

Data: A, λ, r
Random initialization X and Y;
while *not converge* **do**
    while *not converge* **do**
        $X = \mathrm{argmin}_{\mathrm{X}} < L, A - E - XY^T > + \frac{\mu}{2}\|A - E - XY^T\|_F^2$;
        $Y = \mathrm{argmin}_{\mathrm{Y}}\ \lambda\|Y\|_{2,1} + < L, A - E - XY^T >$
        $+ \frac{\mu}{2}\|A - E - XY^T\|_F^2$;
        $E = \mathrm{argmin}_{\mathrm{E}}\ \|E\|_{2,1} + < L, A - E - XY^T > + \frac{\mu}{2}\|A - E - XY^T\|_F^2$;
    end
    $\mu = \rho\mu$, $L = L + \mu\|A - E - XY^T\|_F^2$;
end
Result: X, Y

Convergence condition:

- Object function decreases slowly
- Reaches largest iteration times

Outline

Implementation

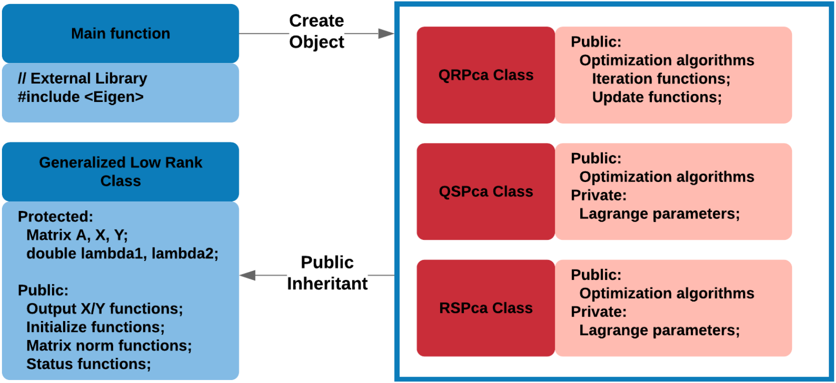


Figure 3: Cpp code structure

Implementation

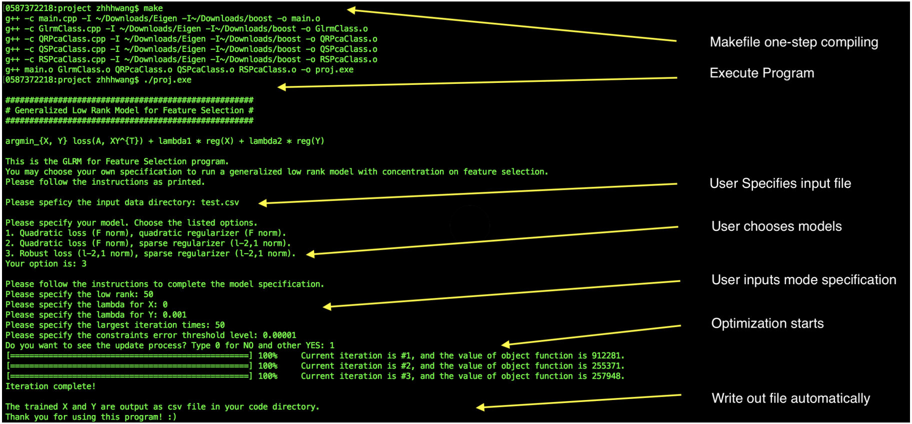


Figure 4: Program Demo

Outline

Experiment

Figure 5: Original Data

warpAR10P Human Face Data Set

- Unstructured data (0 - 255)
- Class size: 10
- Sample size: 130
- Feature size: 2400 (60*40 pixels)
- More details in the report ...

Experiment

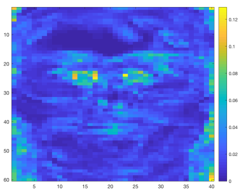
- Used the same tuning and optimization parameters
- Fitted the models
- Ranked the feature importance
- Chose top-ranked features for evaluations
- Averaged performance on different choices of #features

Metric	**R: Sparse PCA**	**QRPcaClass**	**QSPcaClass**	**RSPcaClass**
Acc (KNN)	0.6606	0.6457	0.7138	0.7211
NMI	0.3008	0.3317	0.4223	0.4216
Acc (Clustering)	0.3178	0.3054	0.4068	0.4070

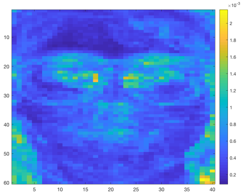
Table 1: Feature Selection Evaluation

Experiment

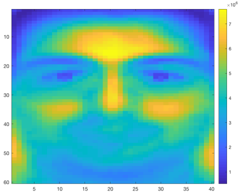
Figure 6: Feature Selection Result



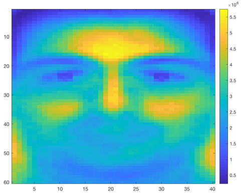
(a) R: Sparse PCA



(b) QRPcaClass

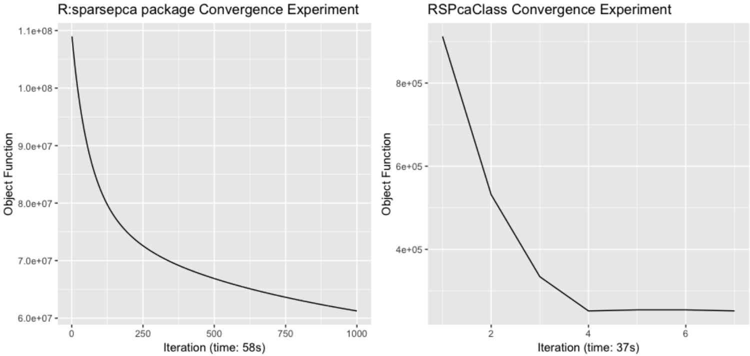


(c) QSPcaClass



(d) RSPcaClass

Figure 7: Convergence Experiment



(a) R: Sparse PCA

(b) RSPcaClass

Outline

Conclusion

Achievements

- Completed a C++ program. Users can ...
  - Specify the models
  - Add user-defined classes (other glrm models)
  - Obtain outputs of low rank representation and feature importance
- Conducted experiment shows effectiveness in...
  - The optimization algorithm
  - The feature selection functionality
- Typed up a comprehensive program instruction

Potential Improvements in ...

- Algorithm speed: parallel computing (CUDA)
- Include more generalized models