Regularized Optimization Algorithms for High Dimensional Missing Data Problems

Yanling Li
The Pennsylvania State University
Department of Human Development and Family Studies

December 5th, 2019

EM algorithm in the high-dimensional setting

X^{obs}: observed data
X^{mis}: missing data
$\log f(X^{obs}, X^{mis}|\theta)$: the log-likelihood of complete data

- Standard EM algorithm
  - E-step: $Q(\theta|\theta^{(t)}) = E_{\theta^{(t)}}\{\log f(X^{obs}, X^{mis}|\theta)|X^{obs}, \theta^{(t)}\}$
  - M-step: $\theta^{(t+1)} = \operatorname{argmax} Q(\theta|\theta^{(t)})$

- In the high-dimensional setting
  - $Q(\theta|\theta^{(t)})$ can be complicated or even intractable
  - the M-step may not be well-defined
    - E.g., the M-step involves inverting a matrix that is not full rank

Regularized EM algorithm

Basic idea: regularizing the M-step in which the regularization parameter $\lambda^{(t)}$ is updated at each iteration to match the target estimation error.

Algorithm 1 Regularized EM Algorithm

Input: data, regularizer $\mathcal{R}$, starting values $\theta^{(0)}$, initial regularization parameter $\lambda^{(0)}$, estimated statistical error Δ, contractive factor κ.

In each iteration,

1. E-step: compute $Q(\theta|\theta^{(t)})$
2. Regularization parameter update: $\lambda^{(t)} = \kappa\lambda^{(t-1)} + \Delta$
3. Regularized M-step: $\theta^{(t+1)} = \mathrm{argmax}[Q(\theta|\theta^{(t)}) - \lambda^{(t)}\mathcal{R}(\theta)]$

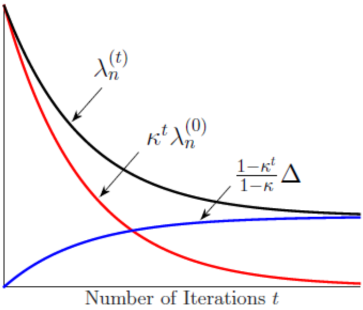
Regularization parameter update

$$\lambda^{(t)} = \kappa\lambda^{(t-1)} + \Delta$$
$$\lambda^{(t)} = \kappa^t\lambda^{(0)} + \frac{1-\kappa^t}{1-\kappa}\Delta$$

$\Delta = O(\sqrt{\log p/n})$ characterizes the final estimation error $\|\theta^{(T)} - \theta^*\|_2$.

$\kappa\lambda^{(t-1)}$ characterizes the optimization error $\|\theta^{(t)} - \theta^{(T)}\|_2$.

In each iteration, $\lambda^{(t)}$ is suggested to be proportional to the target estimation error $\|\theta^{(t)} - \theta^*\|_2$.



Imputation-regularized optimization (IRO) algorithm

Basic idea: replacing the E-step with an imputation step when the E-step is intractable as well as regularizing the M-step.

Algorithm 2 IRO Algorithm

Input: data, regularizer $\mathcal{R}$, starting values $\theta^{(0)}$, regularization parameter λ.

In each iteration,
1. I-step: draw $\tilde{X}^{mis}$ from $h(x^{mis}|X^{obs}, \theta^{(t)})$ so that the pseudo-complete data is $\tilde{X} = (X^{obs}, \tilde{X}^{mis})$.
2. RO-step: $\theta^{(t+1)} = \mathrm{argmax}[E_{\theta^{(t)}}\{\log f(\tilde{X}|\theta)\} - \lambda\mathcal{R}(\theta)]$

Simulation

Toy example: high-dimensional missing covariate regression

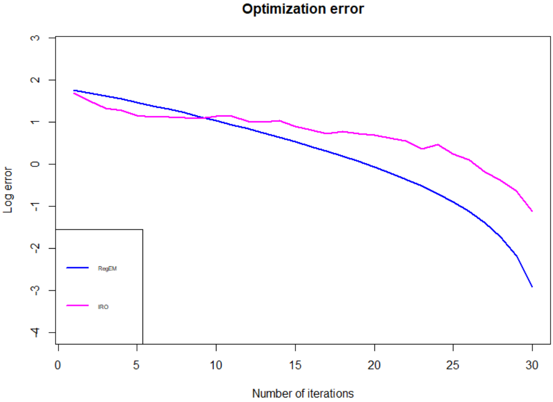
- Model: $Y = X\beta^* + W$ where $X \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_p)$, $W \sim N(0, \sigma^2)$, and β^* is a sparse vector containing only 5 non-zero elements.
- Missing data generation

$$x_{ij} = \begin{cases} x_{ij}, & \text{with probability} 1-\rho \\ missing, & \text{with probability} \rho \end{cases}$$

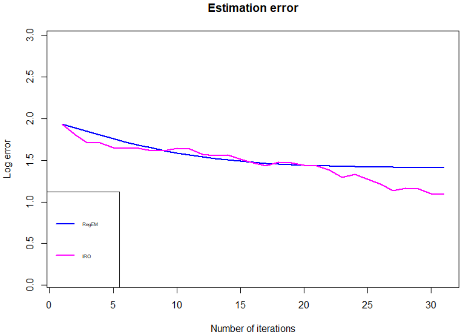
- Goal
  - Monitor the optimization error $\|\beta^{(t)} - \beta^{(T)}\|_2$ and the estimation error $\|\beta^{(t)} - \beta^*\|_2$
  - Compare the computational cost between the two algorithms

Preliminary results

$n = 100, \ p = 200, \ \rho = 0.3, \ T = 30$



Preliminary results



- Wall time
  - Regularized EM: 636.47 seconds
  - IRO: 7.00 seconds

Discussion

- The regularized EM algorithm converges faster than IRO.
- For this toy example, the regularized EM algorithm has higher computational cost because it involves optimizing a complex surrogate function.
- If it is easy to implement parameter estimation when there are no missing data, then IRO is recommended; otherwise, the regularized EM algorithm is recommended given its higher convergence rate.

More algorithms

- Monte Carlo EM (Wei & Tanner, 1990)
  - applied when the E-step is analytically intractable
- misgLasso (Stadler & Bühlmann, 2012)
  - specifically designed for Gaussian graphical models
- misPALasso (Städler et al., 2014)
  - specifically designed for multivariate Gaussian data
- matrix completion algorithm (Cai et al., 2010)
  - designed for large incomplete matrices