Matrix Competition for Missing Value Imputation

Ji Liu
Department of Computer Science, University of Rochester

April 20, 2017

1 Low Rank Matrix Completion

Given n training samples $\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n$ to form a data matrix

$$X = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n] \in \mathbb{R}^{d \times n},$$

it usually contains missing values. We use the set Ω to indicate indices of observed elements in matrix X, for example, $\Omega = (1,3), (2,5)$. To apply most algorithms such as SVM, logistic regression (if we also have labels), we have to fill missing values in the data first. The naive way is to use the mean value to fill the missing elements. However, it does not make sense in many case. For example, given 5 users whose "monthly salaries" are $[10k, 12k, 12k, 13k, ?]$ with the last one missing, the probably best estimation for the last one is $12k$ (both the mean and the median). What if you are given one more feature "yearly income" for these 5 users $[12k0, 144k, 144k, 156k, 240k]$? You will probably change your mind to estimate the missing monthly salary for the last guy to be $20k$. This toy example indicates that when more features available and there exist dependence among features, it is possible to make a better guess for missing values. If the linear (or approximately linear) dependence exist among features, the low rank matrix completion (LRMC) algorithm is a good approach to capture this property.

It basically estimates missing values by utilizing the low rank property of the complete data matrix. Of course, the real data matrix is not low rank in general, LRMC performs quite well as long as the data matrix is approximately with low rank.

$$\min_{M} \quad \sum_{(i,j)\in\Omega} (M_{ij} - X_{ij})^2 \tag{1}$$

$$\text{s.t.} \quad \text{Rank}(M) \leq r \tag{2}$$

where r is a predefined rank parameter. M is the estimated low rank data matrix. In most cases, we just use elements of M in $\bar{\Omega}$ – the complementary set to Ω – to fill the original data matrix X, that is, $X_{\bar{\Omega}} = M_{\bar{\Omega}}$.

To solve this problem, we can rewrite it into an equivalent form

$$\min_{M, X_{\bar{\Omega}}} \quad \sum_{(i,j)} (M_{ij} - X_{ij})^2 = \min_{M} \left(\sum_{(i,j)\in\Omega} (M_{ij} - X_{ij})^2 + \underbrace{\min_{X_{\bar{\Omega}}} \sum_{(i,j)\in\Omega} (M_{ij} - X_{ij})^2}_{=0} \right) \tag{3}$$

$$\text{s.t.} \quad \text{Rank}(M) \leq r. \tag{4}$$

Next we can apply the alternative optimization method by **repeating** the following two steps until M does not change too much:

Fix $X_{\bar{\Omega}}$ to optimize M: It essentially solves the following problem

$$\min_{M} \sum_{(i,j)\in\Omega} (M_{ij} - X_{ij})^2 \quad \text{s.t.} \quad \text{Rank}(M) \leq r$$

The closed form can be computed from the SVD of X by $X = U\Sigma V^\top$:

$$M = U_{:,1:r}\Sigma_{1:r,1:r}V^\top_{:,1:r}$$

Fix M to optimize $X_{\bar{\Omega}}$: It essentially solves the following problem

$$\min_{X_{\bar{\Omega}}} \sum_{(i,j)\in\bar{\Omega}} (M_{ij} - X_{ij})^2$$

which leads to the following simple solution $X_{\bar{\Omega}} = M_{\bar{\Omega}}$.