Week 12 - Recommendation Systems part 2

James Quacinella

11/10/2015

Exercise 9.4.2

If we wish to start out, as in Fig. 9.10, with all U and V entries set to the same value, what value minimizes the RMSE for the matrix M of our running example?

Answer

We have a $5x5M$ matrix which we want to decompose into $d = 2$ dimensional U and V matricies.

"If we have chosen d as the lengths of the short sides of U and V , and a is the average nonblank element of M , then the elements of U and V should be sqrt(a/d)."

```
d <- 2;
M <- matrix(c(5,2,4,4,3,3,1,2,4,1,2,NA,3,1,4,2,5,4,3,5,4,4,5,4,NA), ncol=5, byrow = TRUE);
sqrt(mean(M, na.rm = TRUE) / 2); # 1.276885
```

```
## [1] 1.276885
```

Exercise 9.4.3

```r
# Starting off
U <- matrix(c(2.6, 1, 1.178, 1, 1, 1, 1, 1, 1, 1), ncol=2);
V <- matrix(c(1.617,1,1,1,1,1,1,1,1,1), ncol=5);

M <- matrix(c(5.204, 2.617, 2.905, 2.617, 2.617, 3.6, 2, 2.178, 2, 2, 3.6, 2, 2.178, 2, 2, 3.6, 2, 2.178

# Calculate P
P <- U %*% V;

# Show inititial error
sqrt(sum((M - P)**2)); # 0.0002650962
```

```
## [1] 0.0002650962
```

Answer

Starting with the U and V matrices in Fig. 9.16, do the following in order:

(a) Reconsider the value of u_{11} . Find its new best value, given the changes that have been made so far.

```r
max_j <- 5;
r <- 1;
s <- 1;

# Find the updated value by alculating the numerator and denominator
numerator <- sum(sapply(1:max_j, function(j) { V[s, j] * (M[r, j] - sum(U[r, -c(s)] * V[-c(s), j])); })
denominator <- sum(sapply(1:max_j, function(j) { V[s, j]**2; }))
numerator
denominator

# Update and print U
U[r, s] <- numerator / denominator
U

# Show sum of squared error
sqrt(sum((M - U %*% V)**2)); # 0.0002333765

# Update P
P <- U %*% V;
```

```
## [1] 17.19787
## [1] 6.614689
##          [,1] [,2]
## [1,] 2.599951    1
## [2,] 1.000000    1
## [3,] 1.178000    1
## [4,] 1.000000    1
## [5,] 1.000000    1
## [1] 0.0002333765
```

(b) Then choose the best value for u_{52}

```r
max_j <- 5;
r <- 5;
s <- 2;

# Find the updated value by alculating the numerator and denominator
numerator <- sum(sapply(1:max_j, function(j) { V[s, j] * (M[r, j] - sum(U[r, -c(s)] * V[-c(s), j])); }))
denominator <- sum(sapply(1:max_j, function(j) { V[s, j]**2; }))
numerator
denominator

# Update and print U
U[r, s] <- numerator / denominator
U

# Show sum of squared error
sqrt(sum((M - U %*% V)**2));

# Update P
P <- U %*% V;
```

```
## [1] 5
## [1] 5
##          [,1] [,2]
## [1,] 2.599951    1
## [2,] 1.000000    1
## [3,] 1.178000    1
## [4,] 1.000000    1
## [5,] 1.000000    1
## [1] 0.0002333765
```

(c) Then choose the best value for v_{22}.

```r
max_i <- 5;
r <- 2;
s <- 2;

# Find the updated value by alculating the numerator and denominator
numerator <- sum(sapply(1:max_i, function(i) { U[i, r] * (M[i, s] - sum(U[s, -c(r)] * U[-c(r), s])); }))
denominator <- sum(sapply(1:max_i, function(i) { U[i, r]**2; }))
numerator
```

```
## [1] -8.222
```

```r
denominator
```

```
## [1] 5
```

```r
# Update and print U
V[r, s] <- numerator / denominator
V
```

```
##        [,1]    [,2] [,3] [,4] [,5]
## [1,] 1.617  1.0000    1    1    1
## [2,] 1.000 -1.6444    1    1    1
```

```r
# Show sum of squared error
sqrt(sum((M - U %*% V)**2));
```

```
## [1] 5.91308
```

```r
# Update P
P <- U %*% V;
```