Recommender Systems - using Linear Algebra

Giri Iyengar

Introductions

- Chief Architect at AOL Platforms. Machine Learning startup recently acquired by AOL/Verizon
- Over 2 decades of building models and teaching computers to do interesting things
- Electrical Engineer, transitioned to Pattern Recognition/Machine Learning during my doctoral studies
- IIT Mumbai, MIT, IBM Research, 2 start-ups
- Designed & Teaching this course for the past 4 terms

Course Outline

Linear Algebra

- Matrices and Vectors
- Some standard Linear Algebra techniques
- Applications to Data Analytics & Machine Learning

Probability and Statistics

- Probability Theory, Independence, Conditional Independence, Bayes Rule, etc
- Statistics, Hypothesis testing, Statistical Learning
- Frequentist vs Bayesian points of view

Calculus and Numerical Techniques

- Quick review of Calculus and Numerical techniques
- Gradient Descent, L-BFGS for machine learning

Assignments and Grading

- 1 Assignment every week
- 2 Problem sets. First is quiz-style and the second is a bit more involved – will ask you to code
- 1 Final exam. Take home style. You'll have a weekend to work on it
- Encourage you to share, discuss, and build a community

Recommender Systems

Let's set up a toy recommendation problem. There are 4 users, reading 5 different blogs. Here is a data frame that contains information about how many times each person read these 5 blogs.

`http://www.ibm.com/developerworks/library/os-recommender1/index.html`

Sample dataset

Blogs	Marc	Megan	Elise	Jill
Linux	13	3	11	-
Open Source	10	-	-	3
Cloud Computing	6	1	9	-
Java	-	6	-	9
Agile	-	7	1	8

Reading habits of Users

```
Marc  <- c(13,10,6,0,0)
Megan <- c(3,0,1,6,7)
Elise <- c(11,0,9,0,1)
Jill  <- c(0,3,0,9,8)
u <- data.frame(Marc,Megan,Elise,Jill)
rownames(u) <- c('Linux','Open source','Cloud computing','Java
u
```

```
##                  Marc Megan Elise Jill
## Linux              13     3    11    0
## Open source        10     0     0    3
## Cloud computing     6     1     9    0
## Java                0     6     0    9
## Agile               0     7     1    8
```

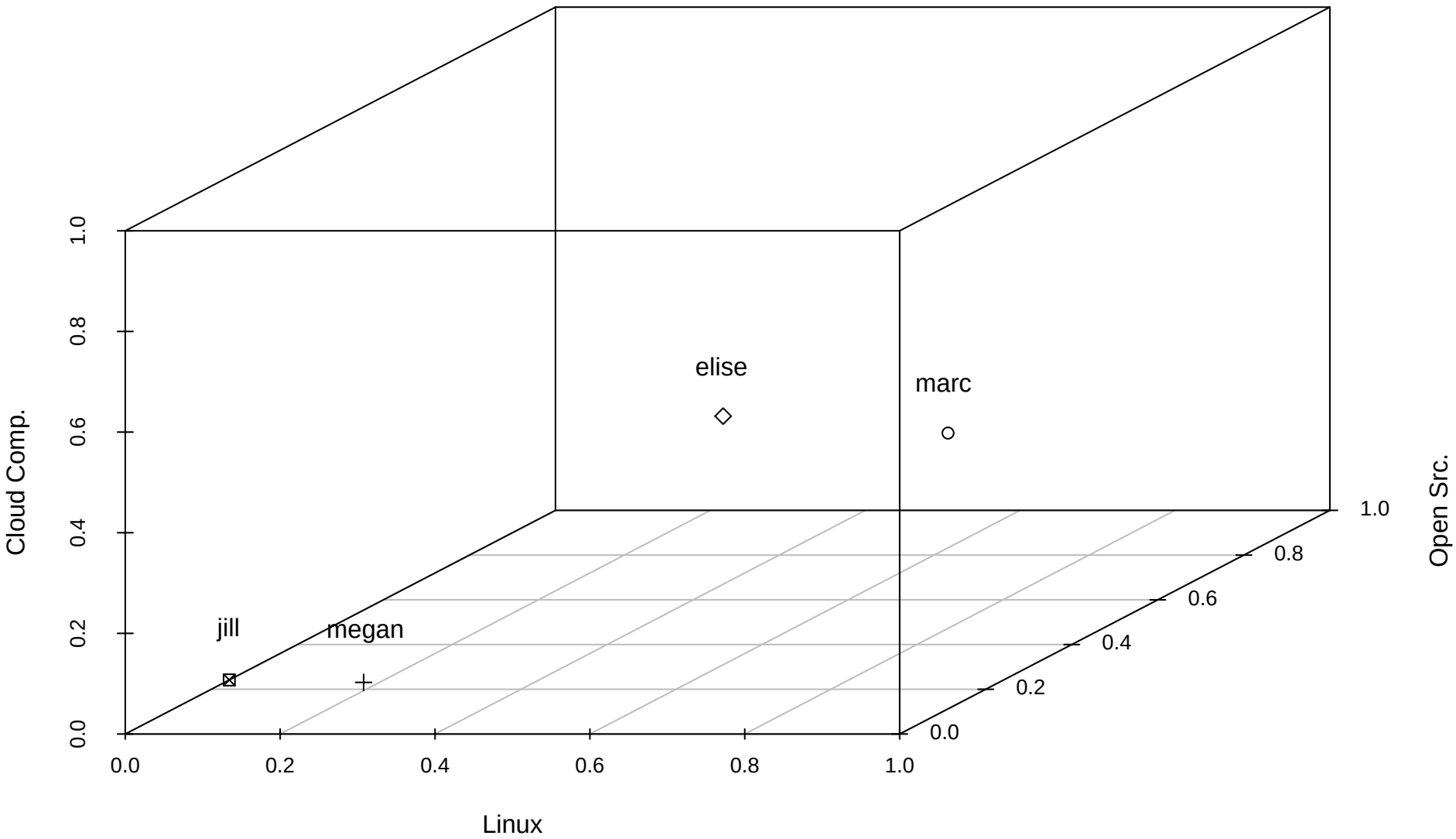
User-Item matrix

Construct a User-Item matrix where each row is a user. Raw counts are not so useful, so normalize them as well.

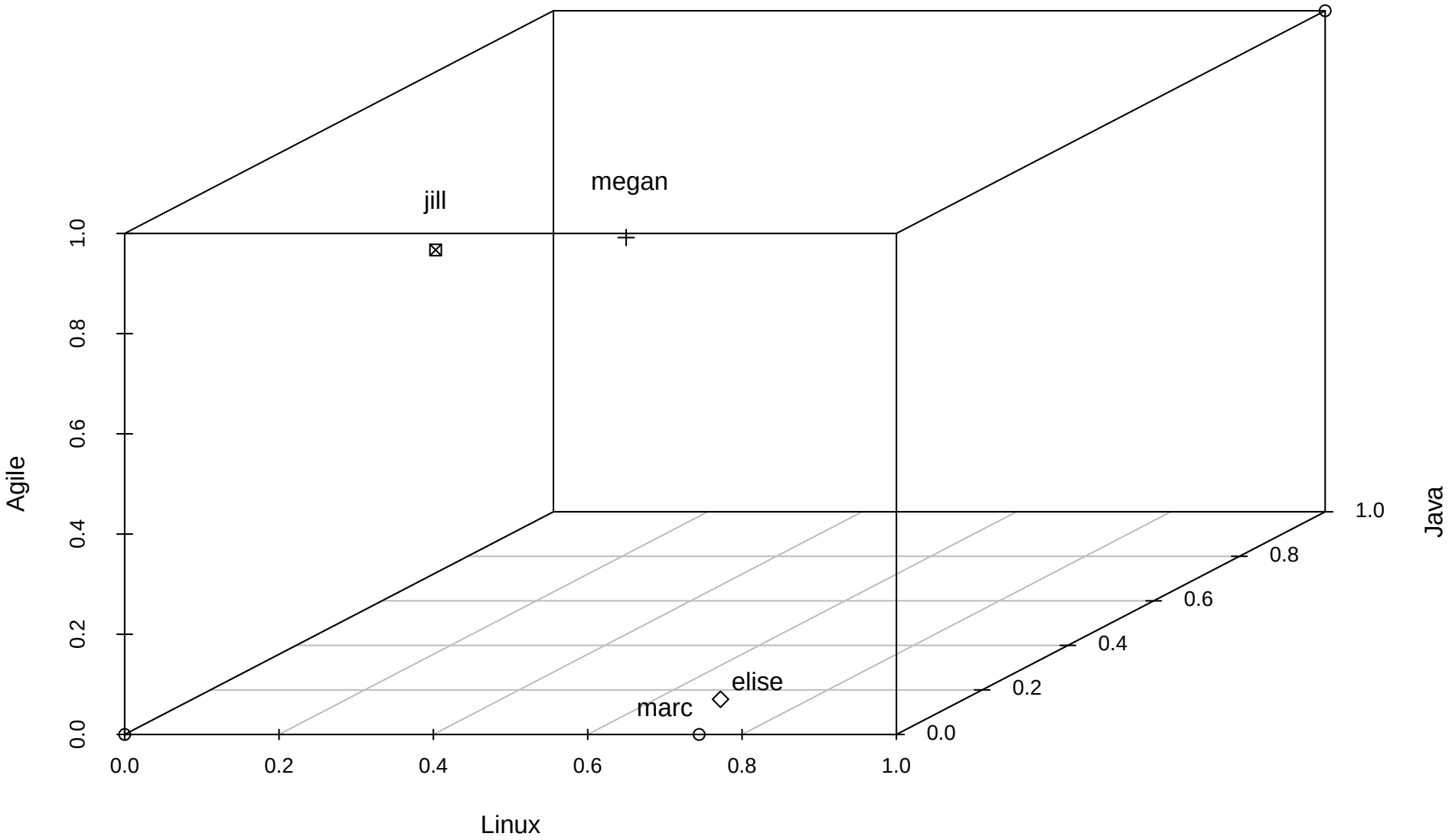
```
unorm <- u
for (i in 1:length(u)) { unorm[,i] <- u[,i] / sqrt(u[,i] %*% u
t(unorm)
```

```
##             Linux Open source Cloud computing      Java
## Marc   0.7443778   0.5725983       0.3435590 0.0000000 0.000
## Megan  0.3077935   0.0000000       0.1025978 0.6155870 0.718
## Elise  0.7720486   0.0000000       0.6316762 0.0000000 0.070
## Jill   0.0000000   0.2417469       0.0000000 0.7252407 0.644
```

Measuring Proximity between Users



Measuring Proximity between Users



Measuring Proximity between Users

- Cosine of the angle between these vectors can be used. Also known as cosine distance.
- The cosine distance between users can be elegantly expressed as $unorm' \times unorm$

```
as.matrix(t(unorm)) %*% as.matrix(unorm)
```

```
##                Marc     Megan      Elise       Jill
## Marc   1.0000000 0.2643631 0.79171393 0.13842387
## Megan  0.2643631 1.0000000 0.35284686 0.90943261
## Elise  0.7917139 0.3528469 1.00000000 0.04524615
## Jill   0.1384239 0.9094326 0.04524615 1.00000000
```

Discussion

- Comparing cosine distances, we see that it agrees with the scatterplot of the reading habits