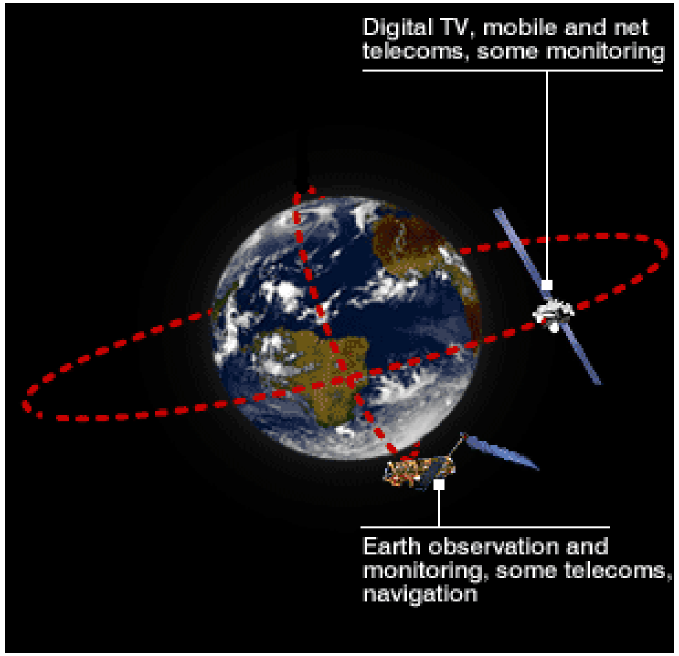
OPTIMIZATIONS USING LINEAR PROGRAMMING

Sai Krishna Kanneti
Yashas JS Shamraju

University at Buffalo The State University of New York



Find an schedule for the satellite that maximizes
the number of photographs taken, subject to the on-board recording capacity

Maximize Returns
Subject to
the investment risk is minimized

Optimization lies at the heart of machine learning:

- **Minimize the loss function subject to overfitting of the model and under fitting of the model.**
- **Find the parameters that minimize loss function**

Similar wide range of optimization problems can be seen in:

- **Supply Chain Industry**
- **Transportation Industry(Uber)**
- **Gambling**

Types of Optimization:

I. Convex Optimization

II. Non Convex Optimization

Examples of convex optimization in ML(Global minimum is guaranteed)

- Linear Regression/ Ridge regression, Lasso, SVM

Examples of non-convex optimization in ML:

- neural networks
- Gaussians mixtures

Support Vector Machines:

- Learning the SVM can be formulated as an optimization:

$$\max_{\mathbf{w}} \frac{2}{||\mathbf{w}||} \text{ subject to } \mathbf{w}^\top \mathbf{x}_i + b \begin{array}{ll} \geq 1 & \text{if } y_i = +1 \\ \leq -1 & \text{if } y_i = -1 \end{array} \text{ for } i = 1 \ldots N$$

- Or equivalently

$$\min_{\mathbf{w}} ||\mathbf{w}||^2 \text{ subject to } y_i \left(\mathbf{w}^\top \mathbf{x}_i + b \right) \geq 1 \text{ for } i = 1 \ldots N$$

Addressing optimization problems:

- linear, quadratic and convex programming methods can be used to solve convex problems.
- linear programming is a technique for the optimization of a linear objective function, subject to linear equality and linear inequality constraints.

$$\begin{array}{ll} \text{maximize} & \mathbf{c}^{\mathrm{T}}\mathbf{x} \\ \text{subject to} & A\mathbf{x} \le \mathbf{b} \\ \text{and} & \mathbf{x} \ge \mathbf{0} \end{array}$$

- Linear Constraints

Fantasy Football using Linear Programming:

- **Build an ideal Fantasy Football team out of 465 players.**

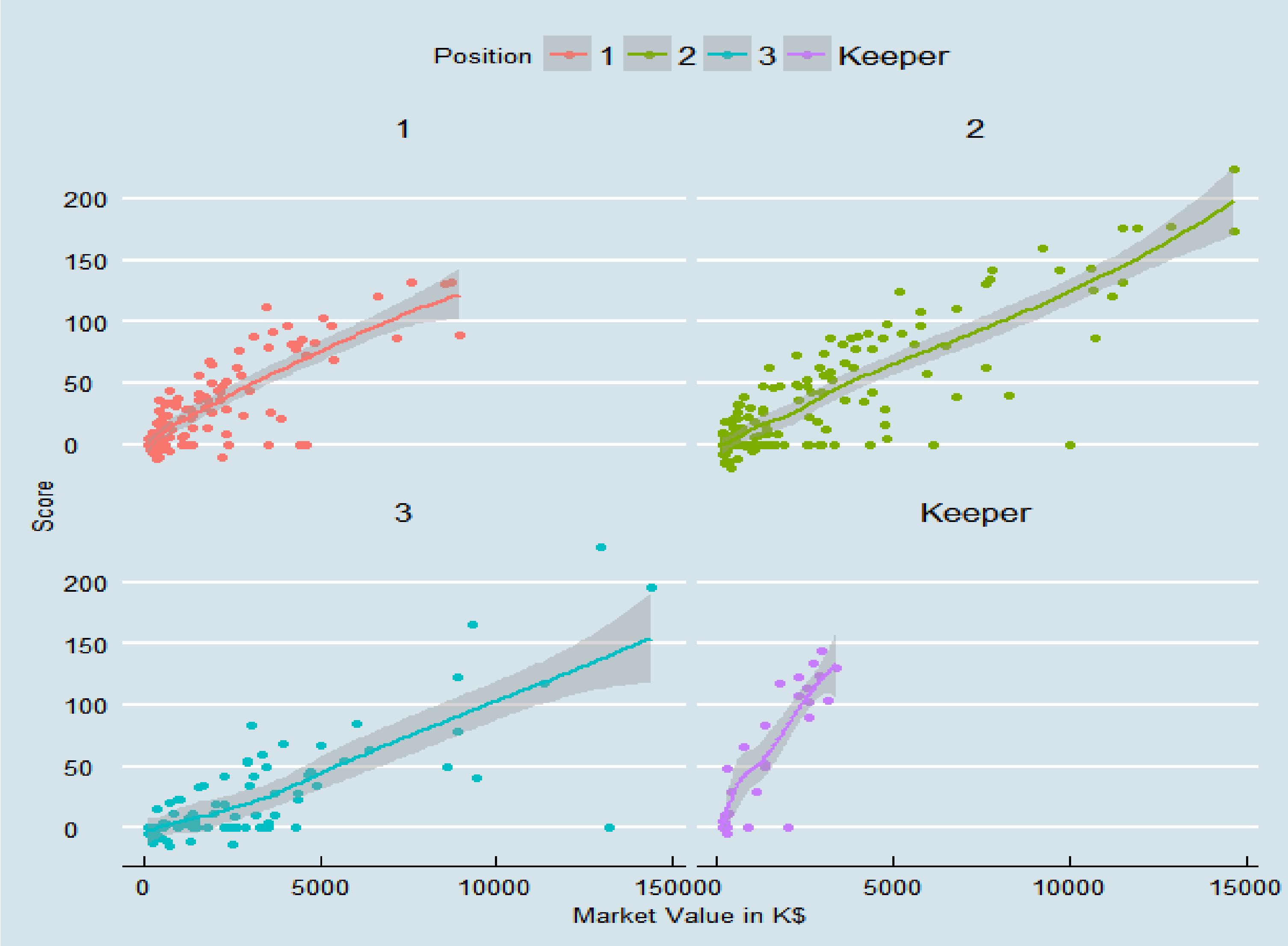
Objective:

- **Maximize the score of the team**
- **Subject to:**
  - **Budget<=20 M$**
  - **1 keeper,**
  - **5 defenders,**
  - **5 midfielders,**
  - **3 scorers.**

Data set: comunio.de, German fantasy football site

	Name	Position	Score	MarketValue	buy
1	Arnautovic	3	68	3980000	0
2	Arnold	2	18	2830000	0
3	Arslan	2	0	380000	0
4	Aubameyang	3	0	13200000	0
5	Audel	2	0	200000	0
6	Avdic	3	0	160000	0
7	Avevor	1	0	170000	0
8	Ayhan	1	0	480000	0
9	B. RÃ¶cker	1	0	220000	0
10	Badelj	2	49	2260000	0
11	Badstuber	1	56	1550000	0
12	Baier	2	72	2260000	1
13	Bajner	3	0	320000	0

ggplot of market value vs score of players:



Solving optimization using lpsolve package in R

```
f.obj <- football_data$Score  ###objective max team score

f.con <- t(football_data$MarketValue)  ### constraints max MV <= Budget
player <- rep(1, nrow(football_data))  ## constraints max number of players!

f.con <- rbind(f.con, player)

## constrain that per postion can only be a certain number of players be set up. (e.g. just one keeper)
## define matrix   - as a one hot (dummy coding what position the player holds)
A <- as.data.frame(model.matrix(MarketValue ~ Position -1, football_data) )

f.con <- rbind(f.con, t(as.matrix(A)))

f.dir <- c("<=", "<=", "=", "<=", "<=" ,"<=")
f.rhs <- c(20000000, 13, 1, 5, 5, 3)  ## right hand side .. not more than Budget, Players, and players per

### solve the problem
solved<- lp("max", f.obj, f.con, f.dir, f.rhs, all.bin=TRUE)  ## just binary variables!
```

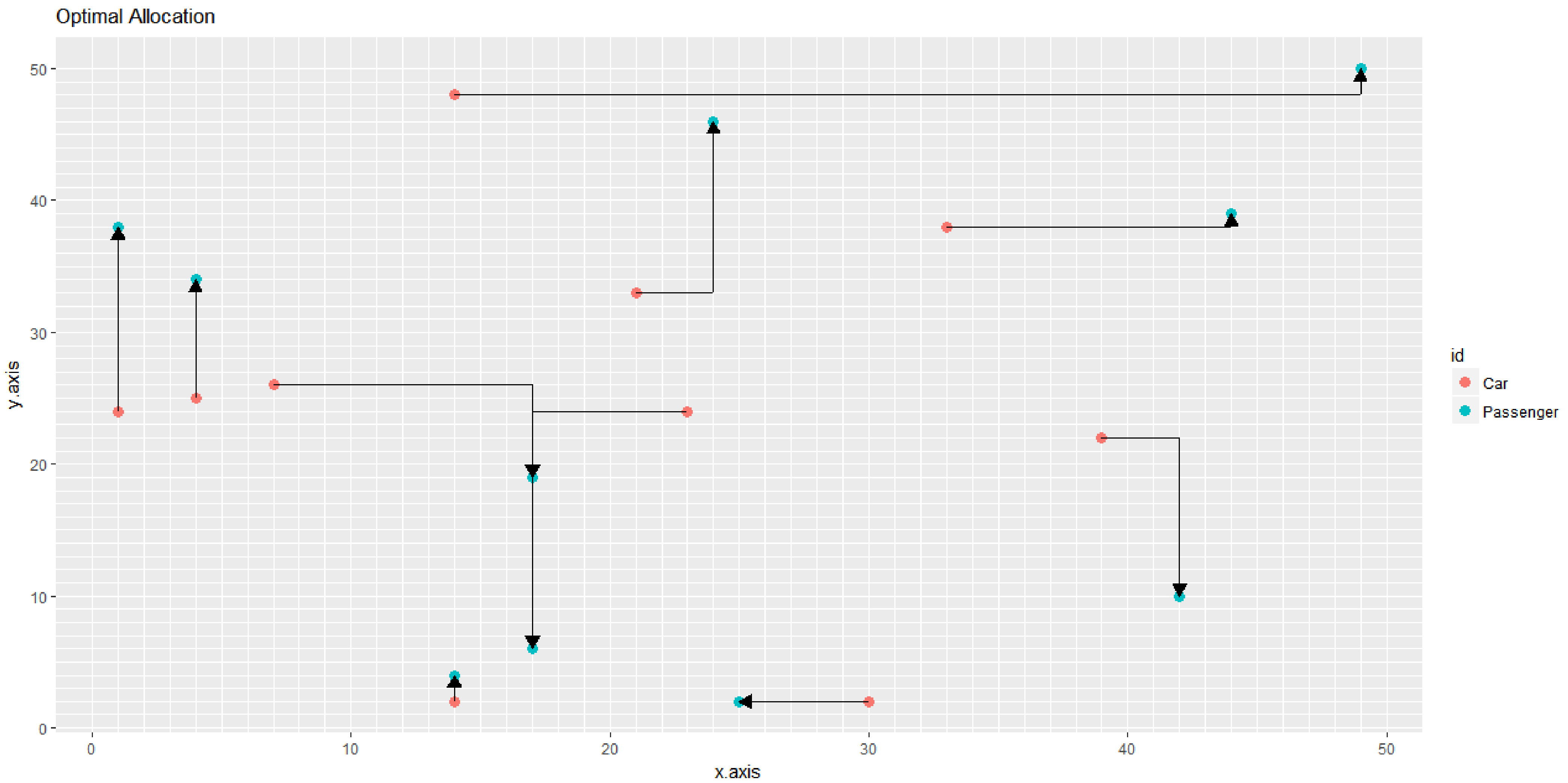
Final team:

- ❖ Team Market value: 19800000
- ❖ Team total score: 784
- ❖ Number of players bought: 13
- ❖ Players:

Badstuber, Baier, Balitsch, Ede,Gomez, Hasebe, klavan, krmas, piszczek,R. SchAfer, Soto, Svensson, Werner

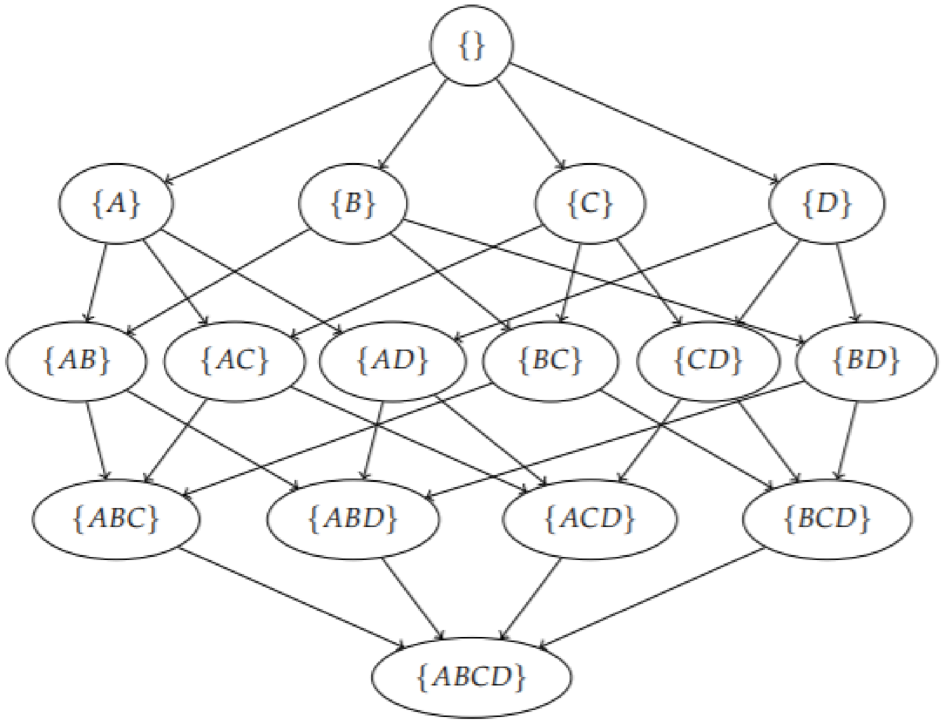
```
> sum(df[df$buy == 1,]$MarketValue)  ## what is the Budget
[1] 19800000
> sum(df[df$buy == 1,]$Score) ## what is the Score
[1] 784
> sum(df[df$buy == 1,]$buy)   ## number of players bought
[1] 13
> paste(df[df$buy == 1,]$Name, collapse=", ")
[1] "Badstuber, Baier, Balitsch, Ede, Gomez, Hasebe, Klavan, Krmas, Piszczek, R. SchÃ¤fer, Soto, Svensson, Werner
"
```

Uber Network Optimization:



Transcript profiling data→ sparse linear genetic network

3. Bayesian Network Structure Learning



linear programming:
estimate the structure of network from: transcript profiling data

maximize Score(G)
s.t. G ∈ DAGS(V)

V:-nodes corresponding to the variables X1, . . . , Xn
DAGS(V) is the space of DAGs over V.

A scoring function is a function whose outcome is a measure related to the probability of the substructure (sv → v) to be in the network, according to the dataset D.

THANK YOU!!!