Gaussian Mixture Models

2023-06-15

Gaussian Mixture Models allow us to do soft assignments in clusters (i.e., allow for some uncertainty in the clustering results)

Mixture Models

- π_k = mixture proportions (or weights) where $\pi_k > 0$ and $\sum_k \pi_k = 1$

COMPONENT = CLUSTER

To generate a new point:

1. Pick a distribution/component among our K options by introduce a new variable:

z ~ Multinomial($\pi_1, \pi_2, ..., \pi_k$) i.e. a categorical variable saying which group the new point is from

2. Generate an observation with that distribution/component (i.e. x | z ~ f_k)

Assumptions

We assume a **parametric mixture model** with parameters θ_k for the kth component (i.e., a mixture of the K component distributions)

Assume each component is **Gaussian/Normal** meaning that $f_k(x;\theta_k) = N(x;\mu_k,\sigma_k^2)$

We need to estimate each parameter. We do this with the **likelihood function**, i.e., the probability (or density) of observing the data given the parameters (and model).

Expectation-Maximization (EM) Algorithm

Helpful when we have more than one component

We alternative between the following:

- pretending to know the probability each observation belongs to each group, to estimate the parameters of the components
- pretending to know the parameters of the components, to estimate the probability each observation belongs to each group

Similar to K-means algorithm

Expectation step: calculate $\hat{z}_{ik}$ = expected membership of observation i in cluster k

Maximization step: update parameter estimates with **weighted** MLE using $\hat{z}_{ik}$

More Information:

- https://towardsdatascience.com/expectation-maximization-explained-c82f5ed438e5

Relation to Clustering

From the EM algorithm:$\hat{z}_{ik}$ is a **soft membership** of observation i in cluster k

- you can assign observation i to a cluster with the largest $\hat{z}_{ik}$
- measure cluster assignment uncertainty = 1 - $max_k \hat{z}_{ik}$

Multivariate GMMs

Say we have p parameters in our model:

$f_k(x;\theta_k) \sim N(\mu_k, \sum_k)$

- μ_k is a vector of means in p dimensions
- $\sum_k$ is the p by p **covariance** matrix, which describes the joint variability between pairs of variables.

To avoid issues with model fitting and estimation as we increase the number of dimensions

We can use **constraints** on multiple aspects of the k covariance matrices

volume: size of the clusters (i.e., number of observations)

shape: direction of variance (i.e., which variables display more variance)

orientation: aligned with the axes (low covariance) versus tilted (due to relationships between variables)

Bayesian Information Criteria (BIC)

procedure for model selection

BIC is a penalized likelihood measure:

$$BIC = 2 * log(L) - m * log(n)$$

* Log(L) is the log-likelihood of the considered model

- with m parameters and n observations
- penalizes large models with many clusters without constraints
- **we can use BIC to choose the covariance constraints AND number of clusters K**

Mixture Model Example

```
nba_pos_stats <-   read_csv("https://shorturl.at/mFGY2")

# Find rows for players indicating a full season worth of stats
tot_players <- nba_pos_stats %>% filter(tm == "TOT") # Stack this dataset with players that played on j
nba_player_stats <- nba_pos_stats %>%
  filter(!(player %in% tot_players$player)) %>%
  bind_rows(tot_players)

# Filter to only players with at least 125 minutes played

nba_filtered_stats <- nba_player_stats %>%
  filter(mp >= 125)
```

```
head(nba_filtered_stats)
```

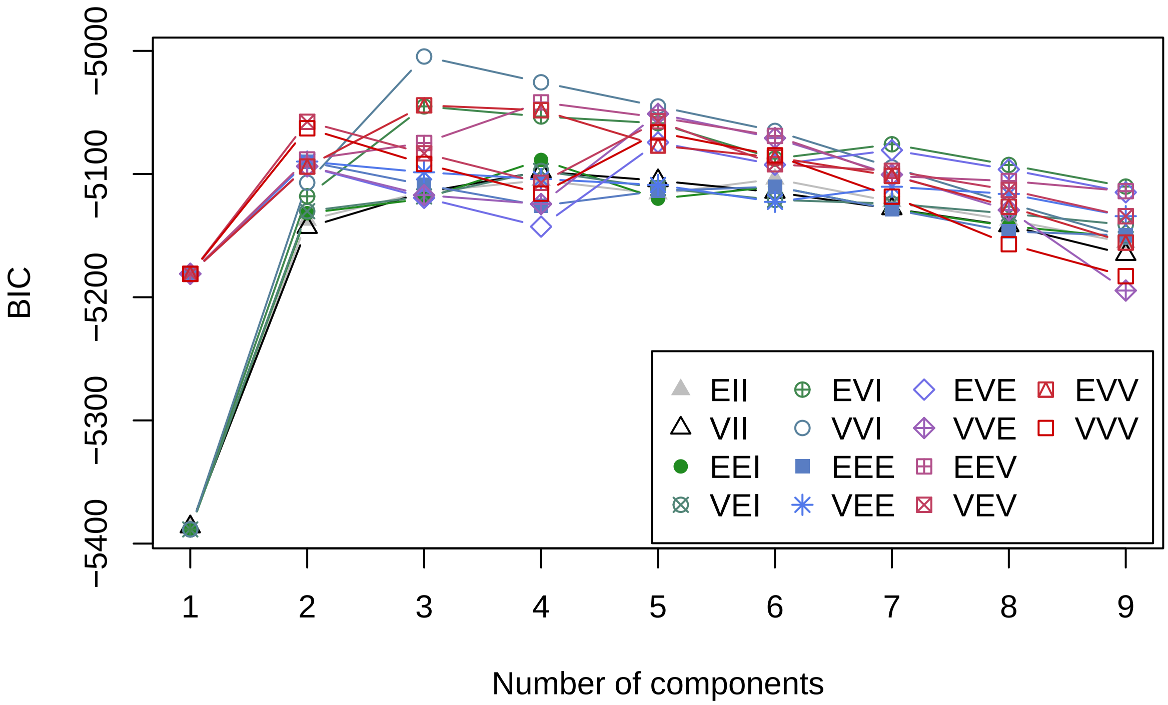
```
## # A tibble: 6 x 31
##   player    pos     age tm        g    gs    mp    fg   fga fgpercent   x3p  x3pa
##   <chr>     <chr> <dbl> <chr> <dbl> <dbl> <dbl> <dbl> <dbl>     <dbl> <dbl> <dbl>
## 1 Preciou~ C        22 TOR      73    28  1725   7.7  17.5     0.439   1.6   4.5
## 2 Steven ~ C        28 MEM      76    75  1999   5     9.2     0.547   0     0
## 3 Bam Ade~ C        24 MIA      56    56  1825  11.1  20       0.557   0     0.2
## 4 Santi A~ PF       21 MEM      32     0   360   7    17.5     0.402   0.8   6.4
## 5 LaMarcu~ C        36 BRK      47    12  1050  11.6  21.1     0.55    0.6   2.1
## 6 Grayson~ SG       26 MIL      66    61  1805   6.8  15.1     0.448   4.2  10.4
## # i 19 more variables: x3ppercent <dbl>, x2p <dbl>, x2pa <dbl>,
## #   x2ppercent <dbl>, ft <dbl>, fta <dbl>, ftpercent <dbl>, orb <dbl>,
## #   drb <dbl>, trb <dbl>, ast <dbl>, stl <dbl>, blk <dbl>, tov <dbl>, pf <dbl>,
## #   pts <dbl>, x <lgl>, ortg <dbl>, drtg <dbl>
```

```
nba_mclust <- Mclust(dplyr::select(nba_filtered_stats, x3pa, trb))
```

```
summary(nba_mclust)
```

```
## ----------------------------------------------------
## Gaussian finite mixture model fitted by EM algorithm
## ----------------------------------------------------
##
## Mclust VVI (diagonal, varying volume and shape) model with 3 components:
##
##  log-likelihood   n df       BIC       ICL
##        -2459.03 483 14 -5004.581 -5141.138
##
## Clustering table:
##   1   2   3
##  52 276 155
```

```
plot(nba_mclust, what = 'BIC',
     legendArgs = list(x = "bottomright",
                       ncol = 4))
```

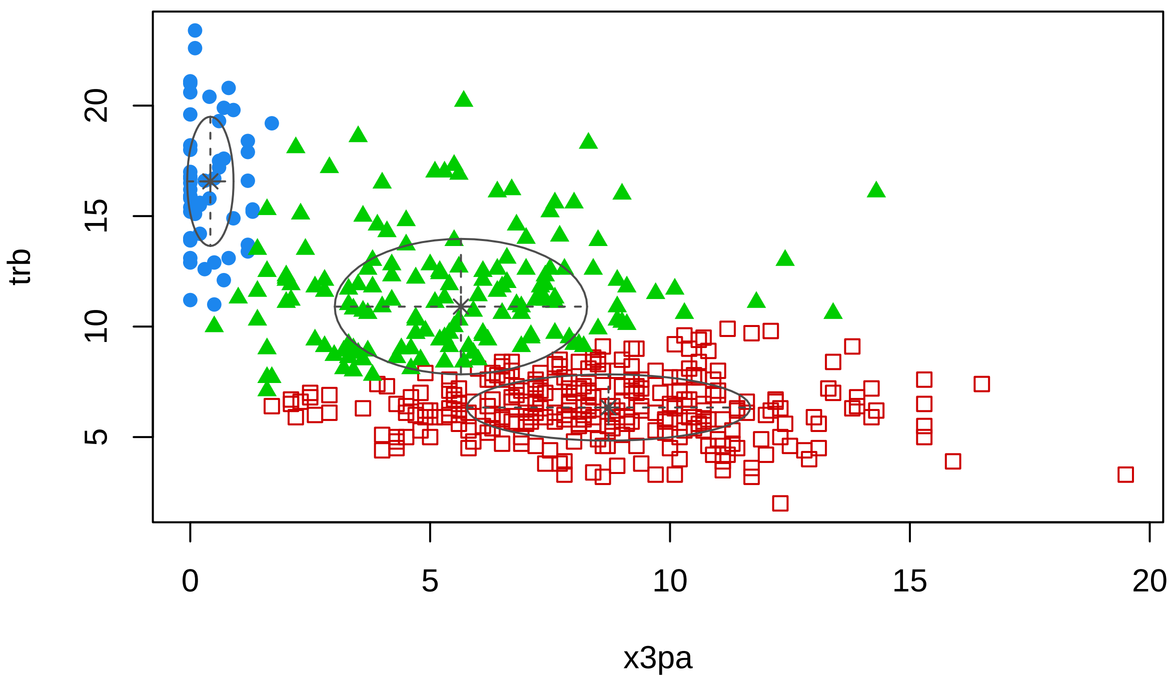


Diagonal versus spherical constraints?

To look at:

- https://alliance.seas.upenn.edu/~cis520/wiki/index.php?n=Lectures.EM

```
plot(nba_mclust, what = 'classification')
```



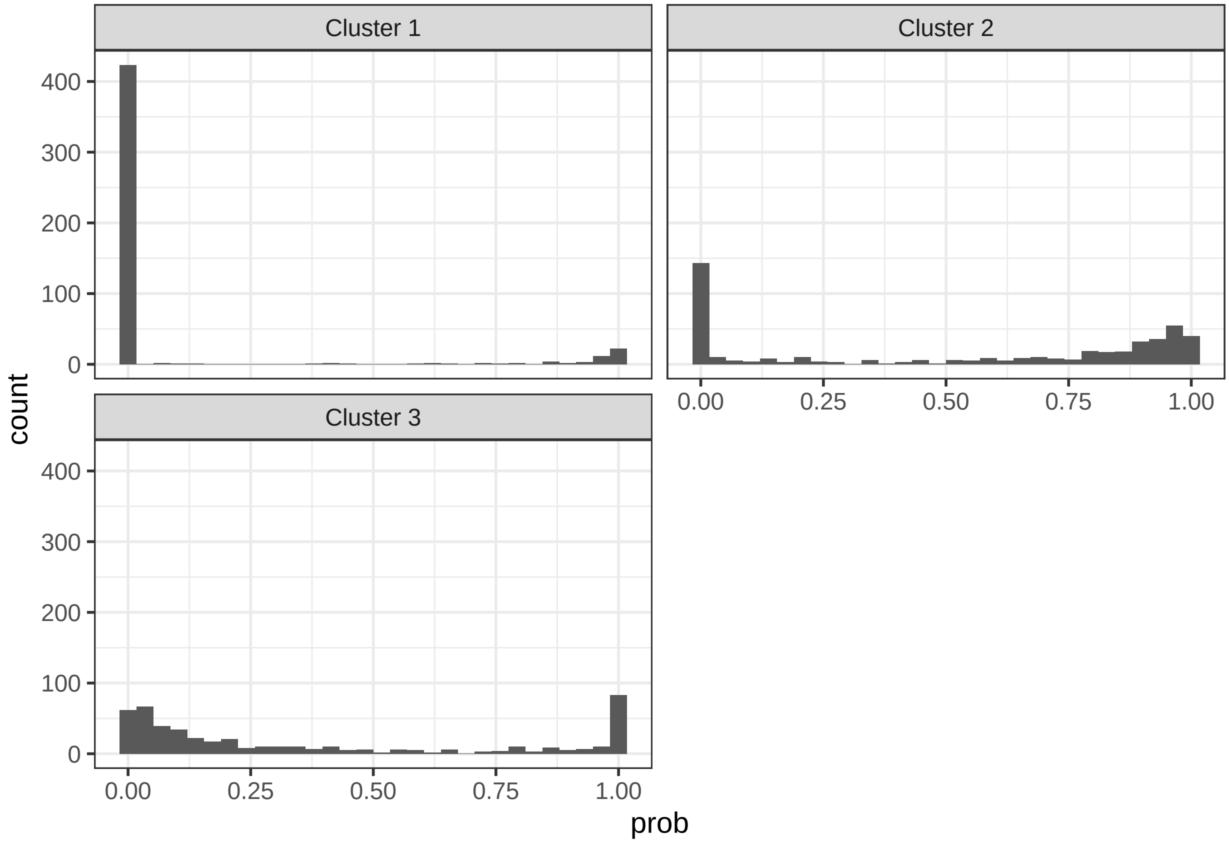

```
table("Clusters" = nba_mclust$classification, "Positions" = nba_filtered_stats$pos)
```

```
##          Positions
## Clusters  C C-PF PF PF-SF PG PG-SG SF SF-SG SG SG-PG SG-SF
```

```
##         1 43     0  9      0  0      0  0      0  0      0      0
##         2  3     0 28      0 84      0 54      5 96      3      3
##         3 39     2 56      1  8      1 38      0  9      0      1
```

Cluster Probabilites

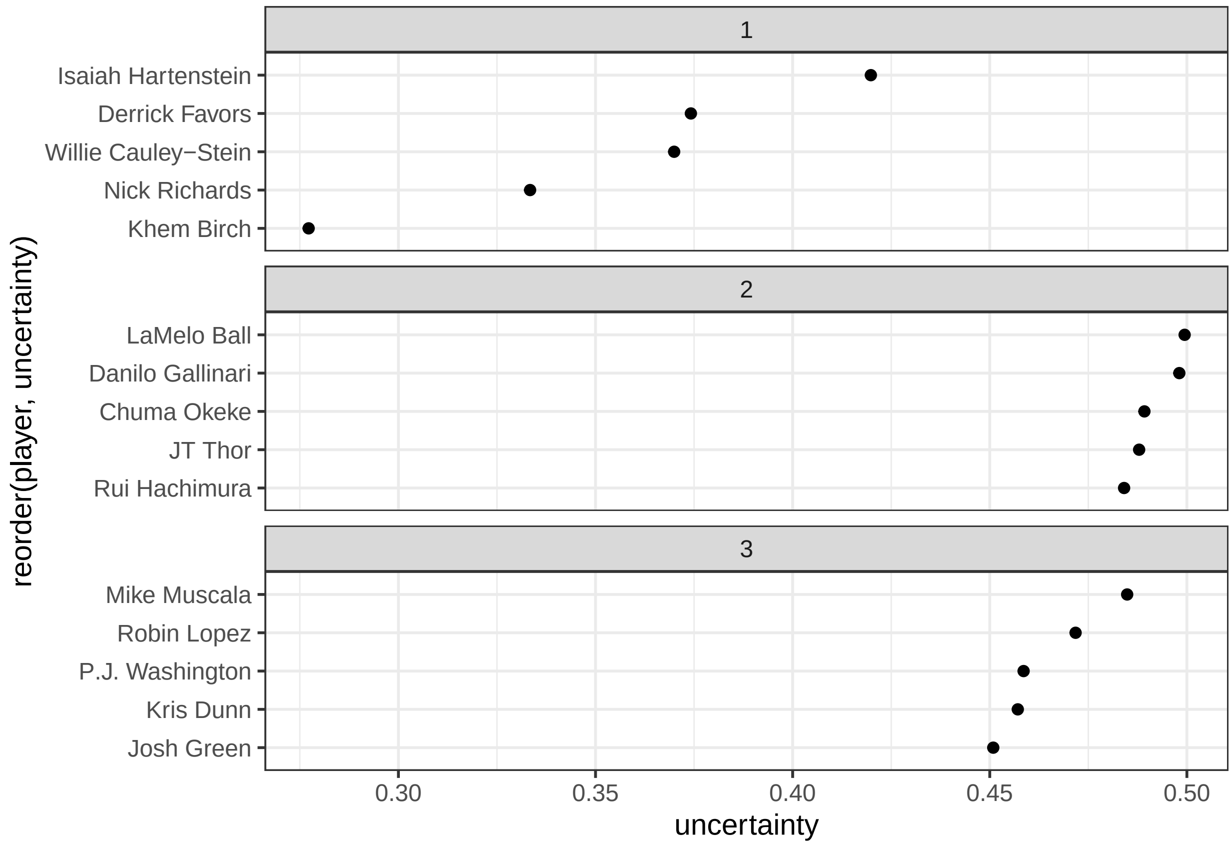
```
nba_player_probs <- nba_mclust$z

colnames(nba_player_probs) <-   paste0('Cluster ', 1:3)

nba_player_probs <- nba_player_probs %>%
  as_tibble() %>%
  mutate(player = nba_filtered_stats$player) %>%
  pivot_longer(contains("Cluster"), names_to = "cluster",              values_to = "prob")

nba_player_probs %>%  ggplot(aes(prob)) +  geom_histogram() +  theme_bw() +  facet_wrap(~ cluster, nrow
```



Player Probabilities

```
nba_filtered_stats %>%  mutate(cluster = nba_mclust$classification,         uncertainty = nba_mclust$unc
  group_by(cluster) %>%
  arrange(desc(uncertainty)) %>%
  slice(1:5) %>%  ggplot(aes(y = uncertainty, x = reorder(player, uncertainty))) +
  geom_point() +
  coord_flip() +
  theme_bw() +
```

```
facet_wrap(~ cluster, scales = 'free_y', nrow = 3)
```



Uncertainty = probability that the players assigned in some cluster i (between 1 and k), would be assigned to any of the other k-1 clusters