Performance evaluation in soccer

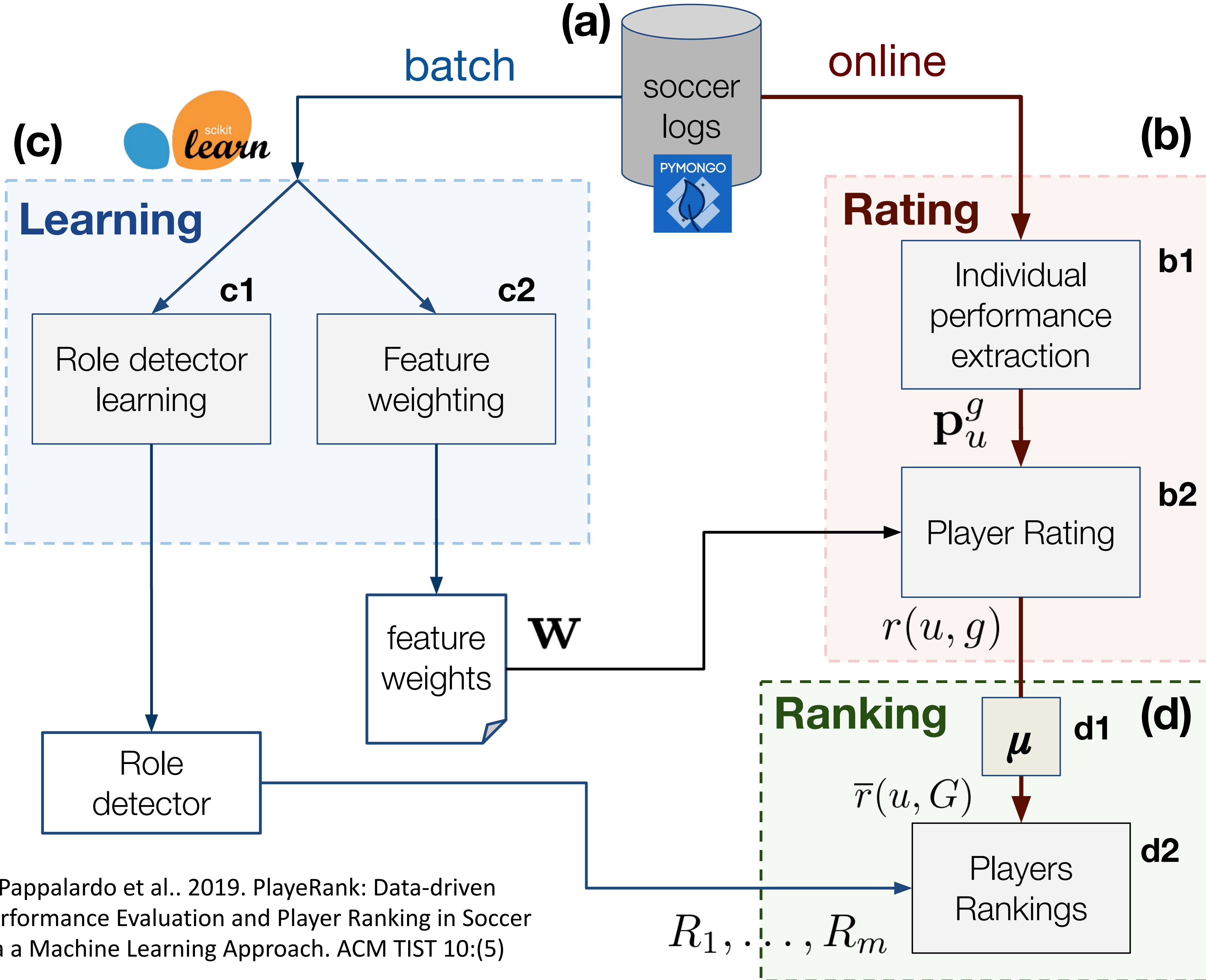
from **human** mechanisms to **data-driven** algorithms

How to *automatically* evaluate performance?

solution:

~~imitate the human~~

make it data-driven



L. Pappalardo et al.. 2019. PlayeRank: Data-driven Performance Evaluation and Player Ranking in Soccer via a Machine Learning Approach. ACM TIST 10:(5)

Step #1: player performance

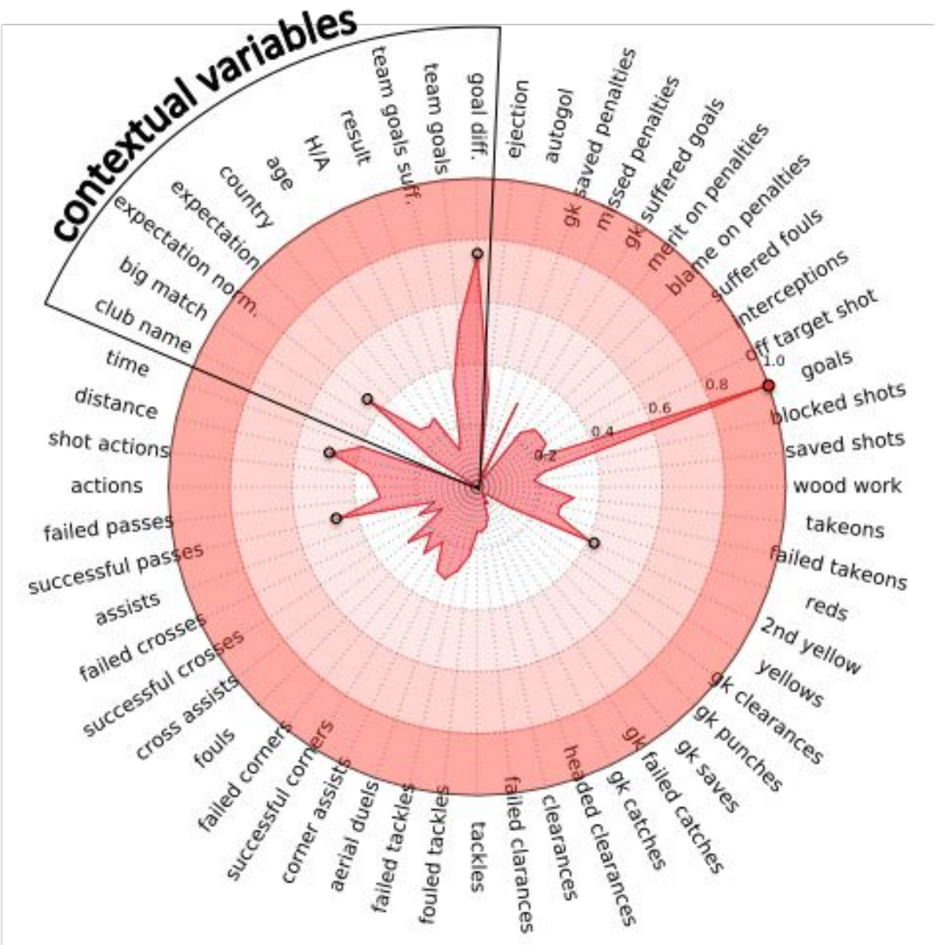
14 million events

150 technical features

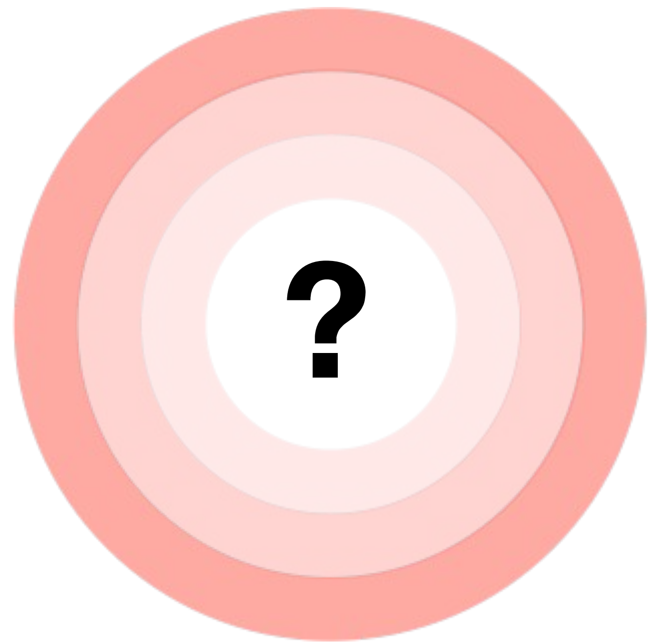
7,304 games

1,192 professional players

Step #2: feature weighting

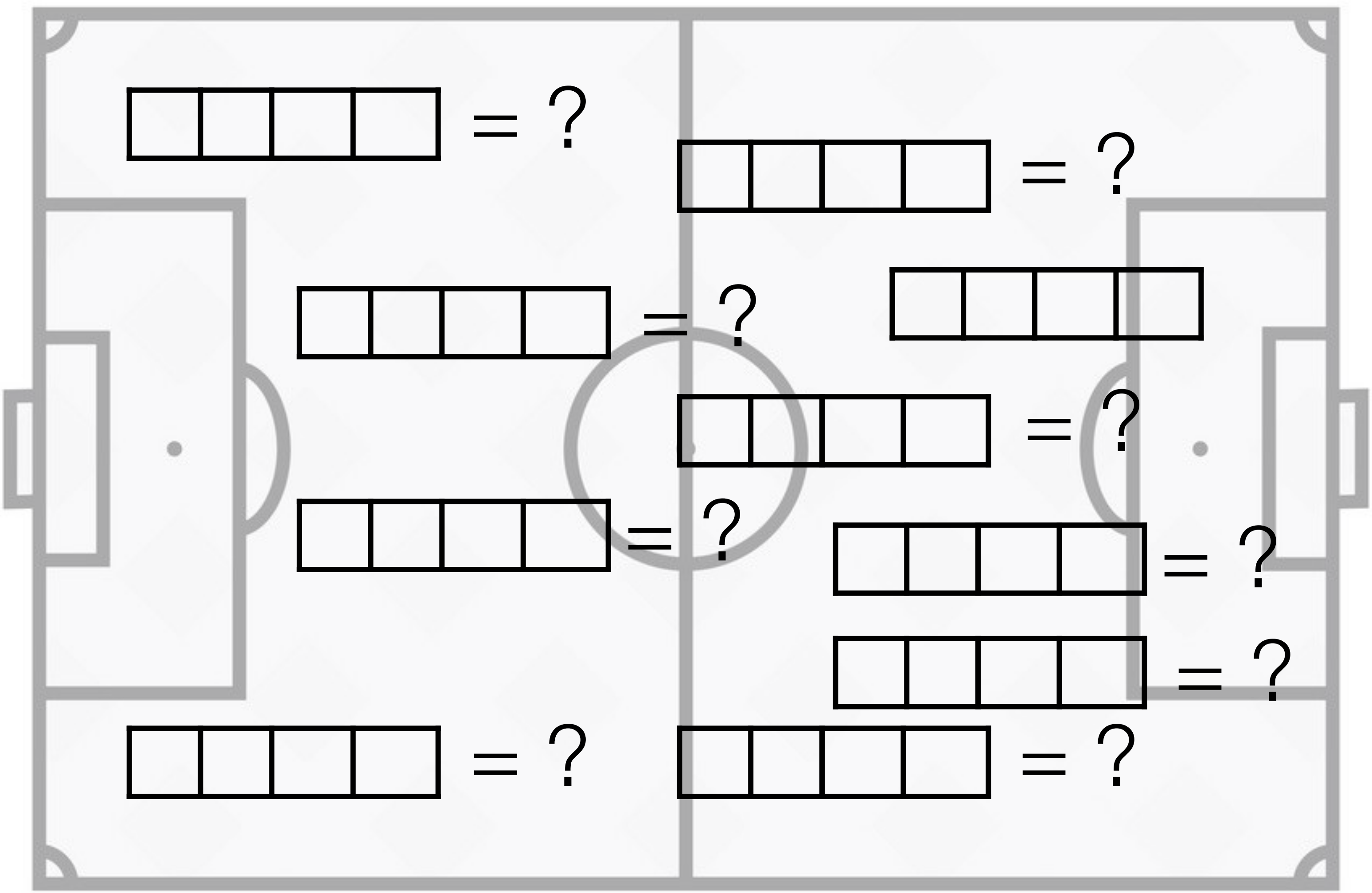


Human rating

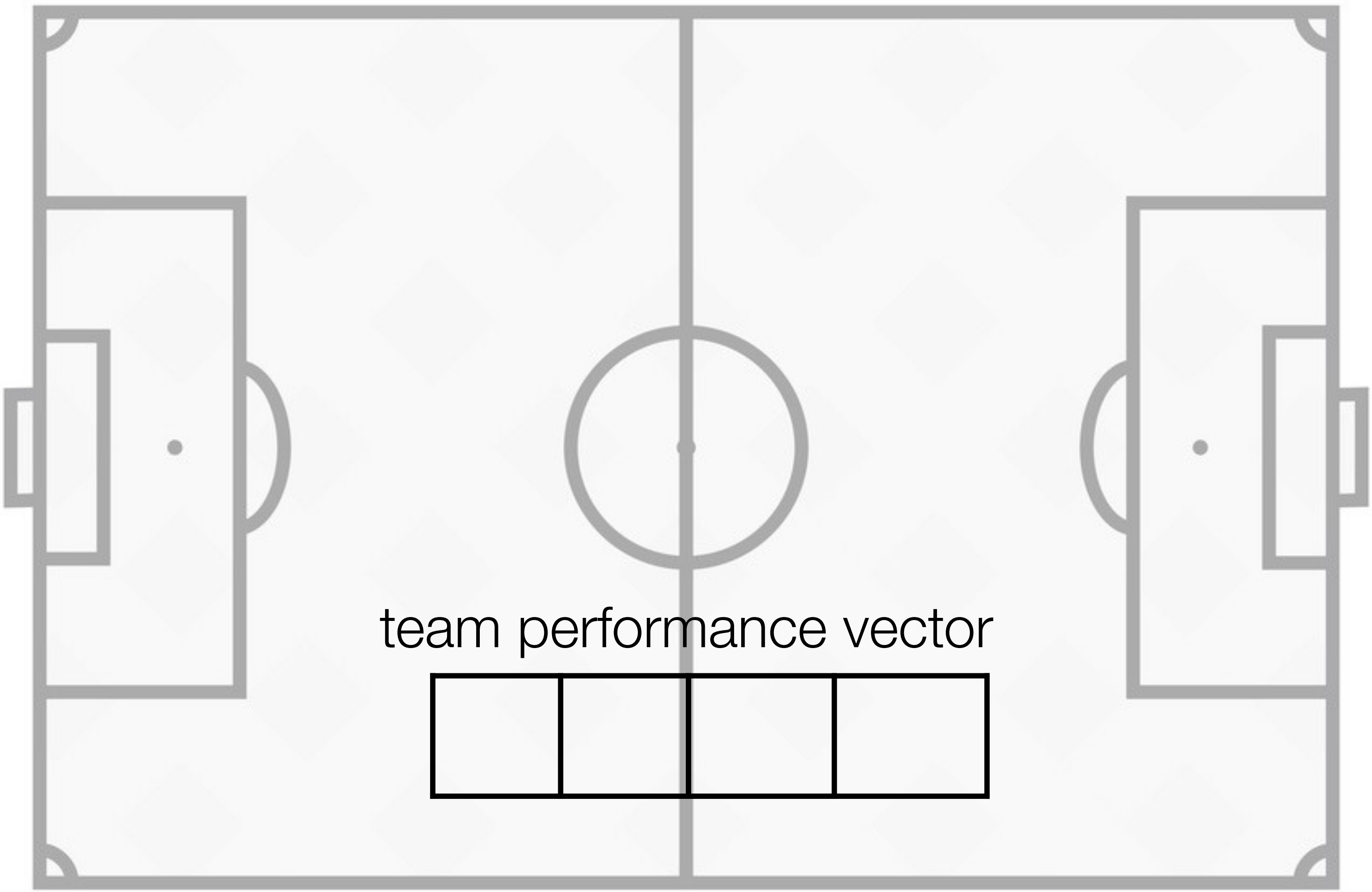


Data-driven rating

Feature Weighting



Feature Weighting



Feature Weighting

team1

passes	xG	pressing	accuracy	...

?

1X2

team2

passes	xG	pressing	accuracy	...

Pappalardo and Cintia, (2017) Quantifying the relation between performance and success in soccer, Advances in Complex Systems, doi:10.1142/S021952591750014X

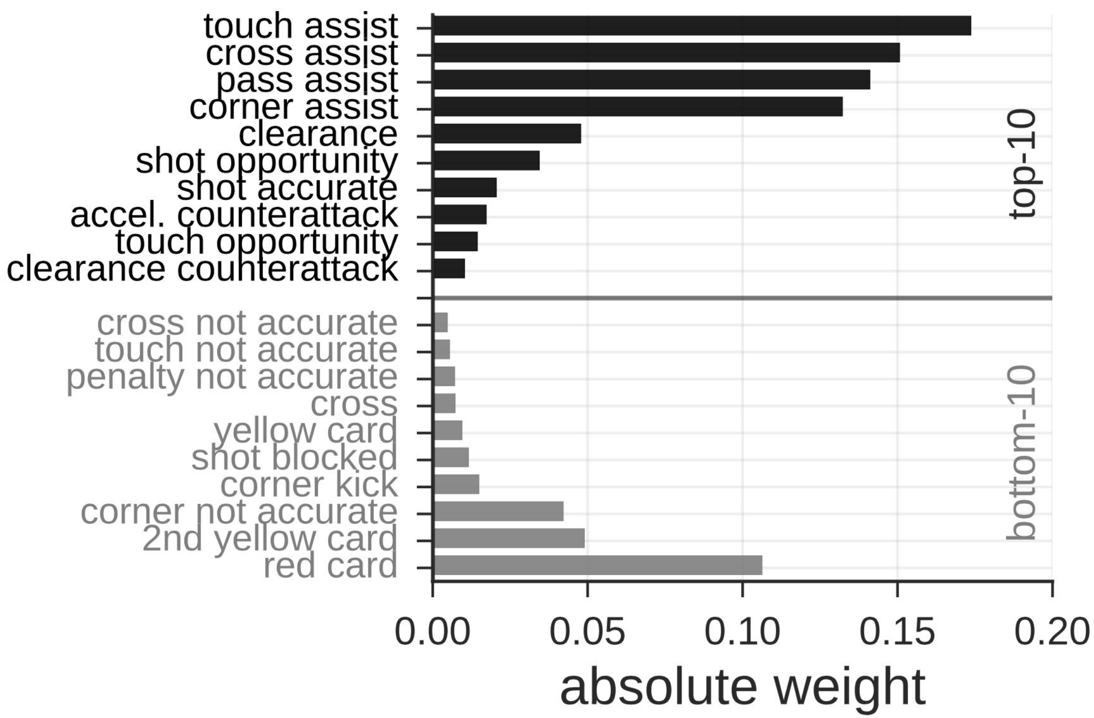
Step #2: feature weighting

```
from pymongo import MongoClient

client = MongoClient('localhost', 27017)
events = client.wyscout.events

res = events.map_reduce(map_agg, reduce_sum)
X, y = extract_data(res)
```

	inaccurate defending duel	intercept	accurate air duel	accelleration	corner assist	missed penalty	foul	corner pass	accurate defending duel	cross key pass	...	outcome
0	-8.0	5.0	2.0	-1.0	0.0	0.0	-3.0	1.0	12.0	2.0	...	w
1	8.0	-5.0	-2.0	1.0	0.0	0.0	3.0	-1.0	-12.0	-2.0	...	n
2	-7.0	-3.0	6.0	0.0	0.0	0.0	5.0	-1.0	-10.0	1.0	...	w
3	7.0	3.0	-6.0	0.0	0.0	0.0	-5.0	1.0	10.0	-1.0	...	w
4	-13.0	-5.0	6.0	1.0	0.0	0.0	-6.0	1.0	-13.0	-2.0	...	w

Step #2: feature weighting

```
from playerank import Weighter
# perform the feature weighting
pw = Weighter()
pw.fit(X, y)
pw.weights_
```

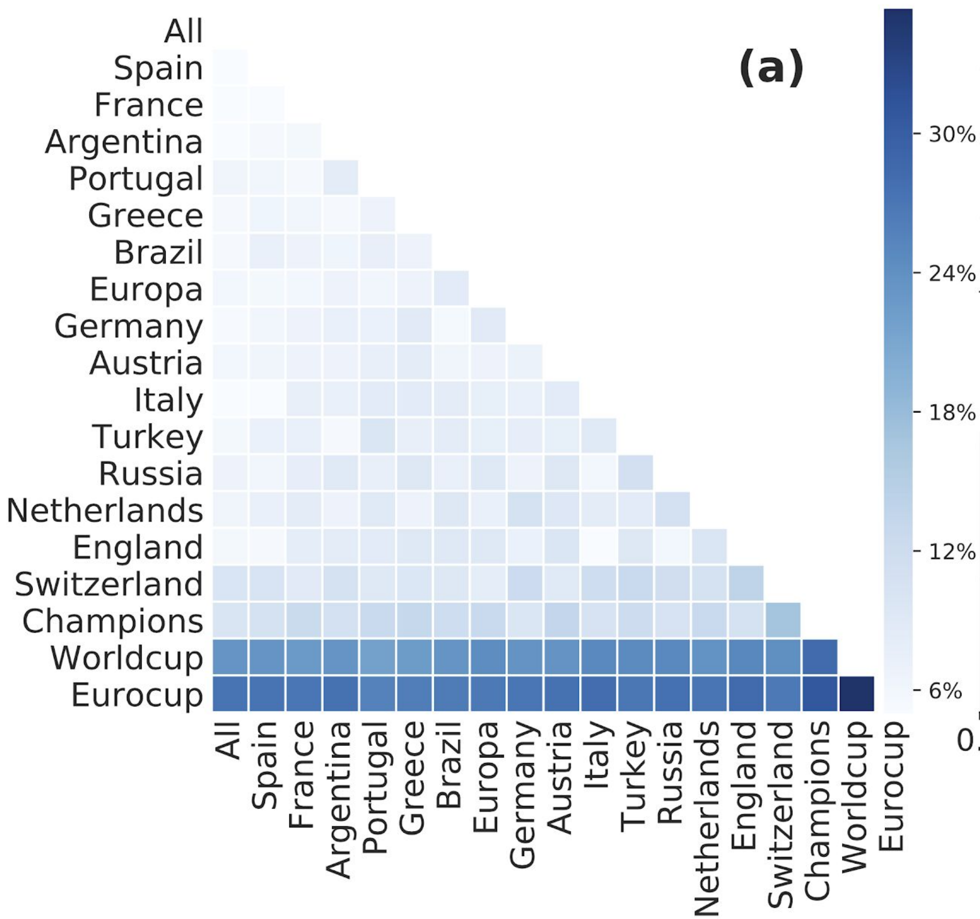


Evaluating the weights

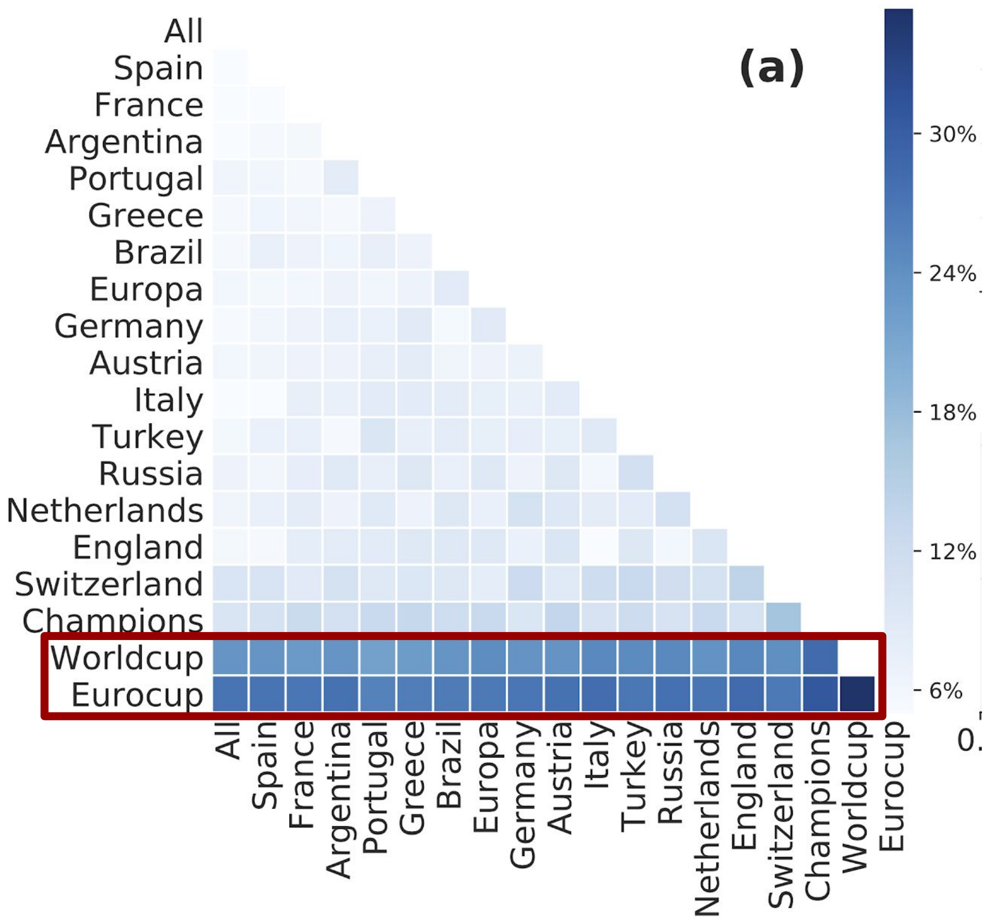
- **stability**
  across competitions and roles

- evaluation of resulting ranking

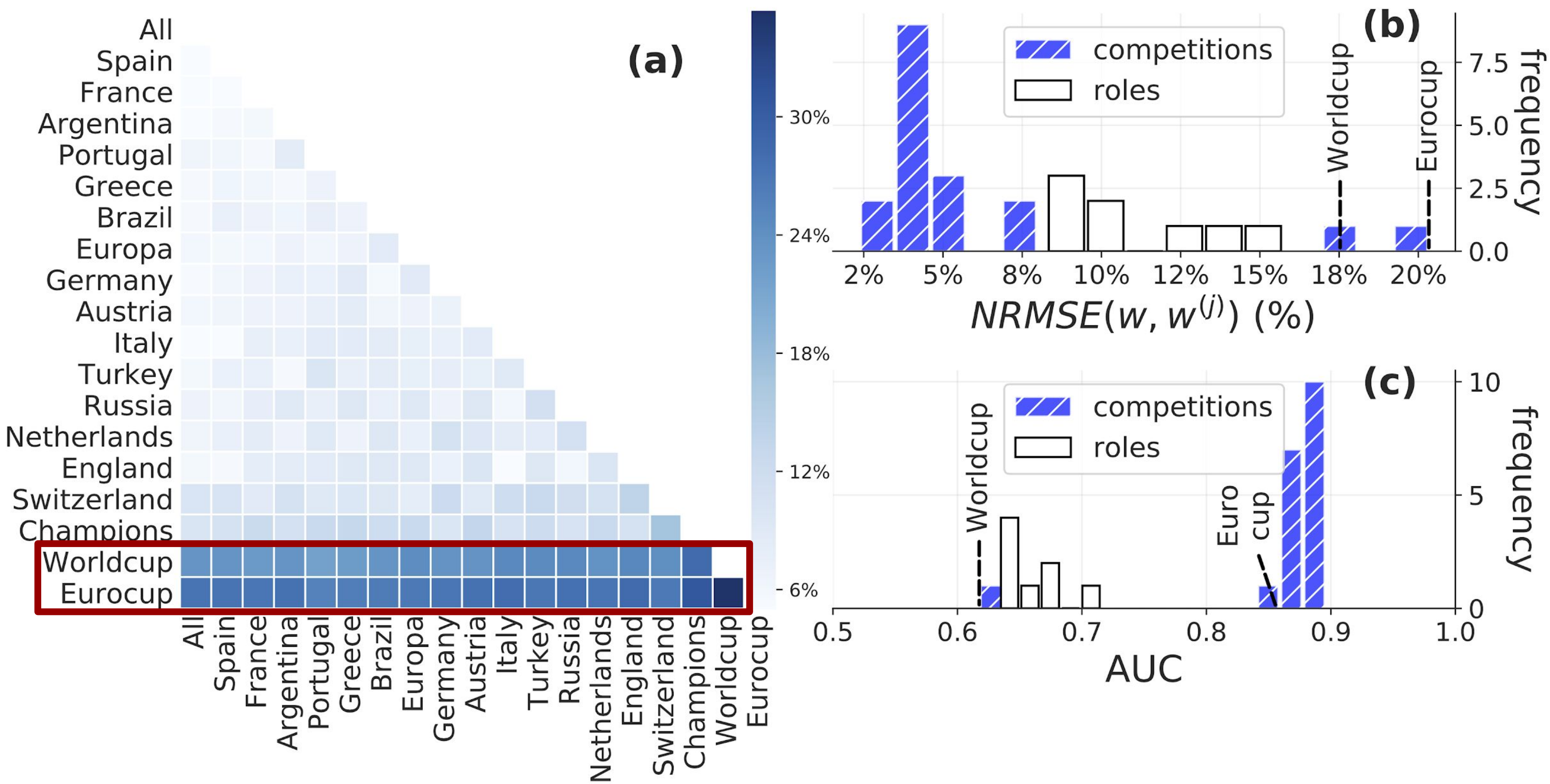
Are these weights “universal”?



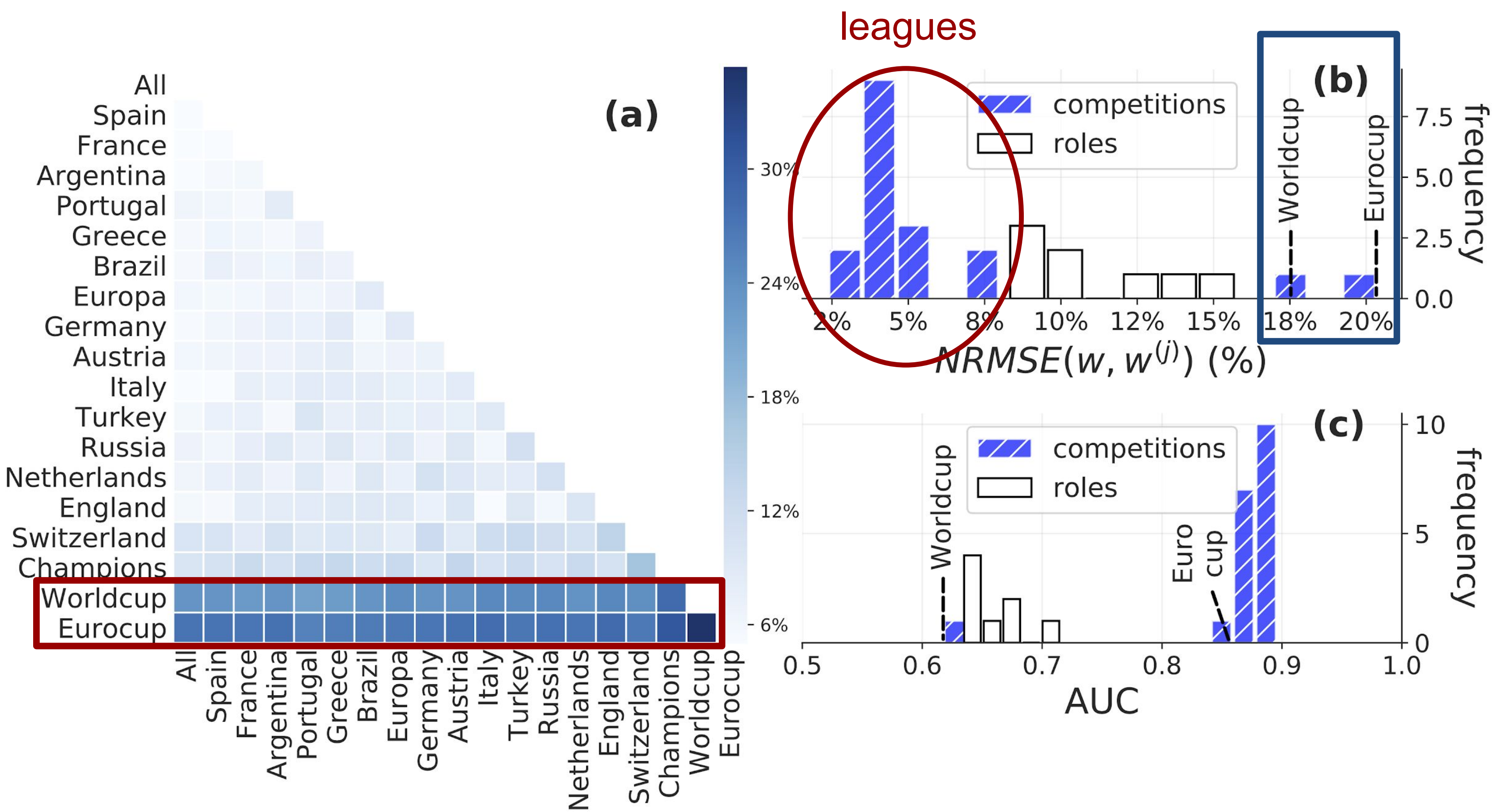
Are these weights “universal”?



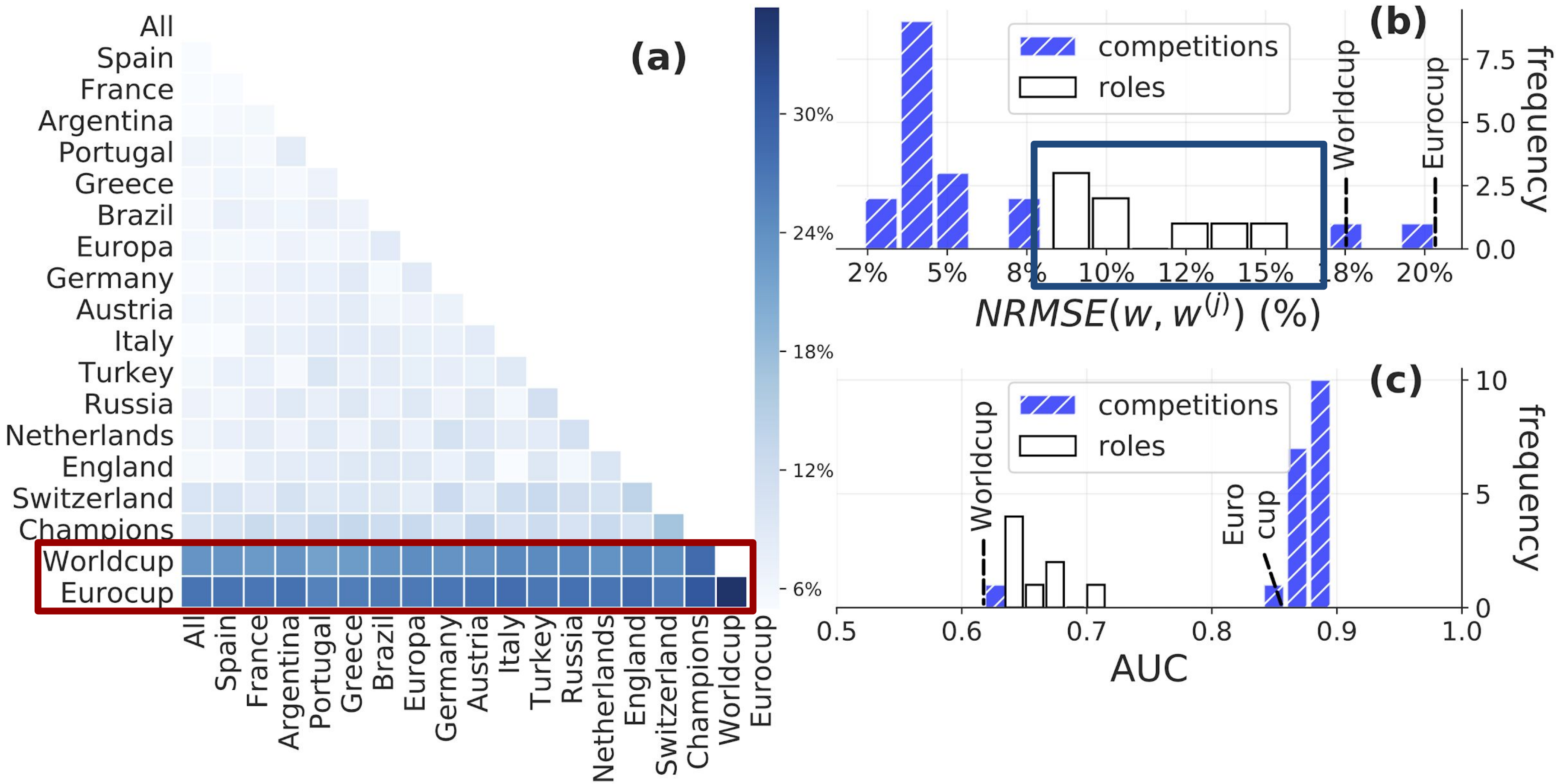
Are these weights “universal”?



Are these weights “universal”?



Are these weights "universal"?



Step #3: role classification

"All ~~animals~~ players are equal, but some ~~animals~~ players are more equal than others."

George Orwell

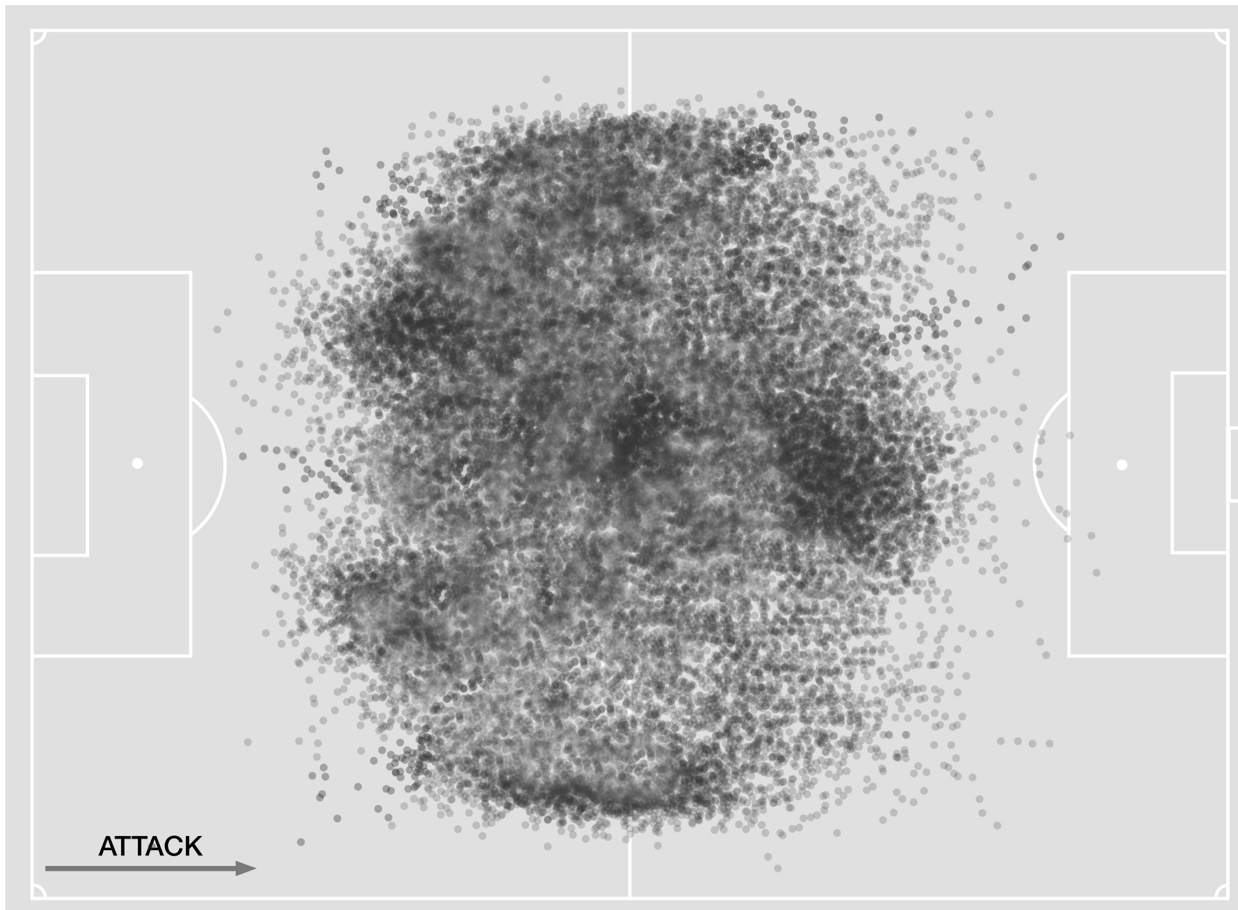
It is meaningless to compare
two players with different roles

Step #3: role classification

```
from pymongo import MongoClient

# load the centers data
res = events.aggregate(pipeline)
X = extract_data(res)
```

```
pipeline = [
    {'$project': {'positions': {'$arrayElemAt': ['$positions', 0]},}},

    {'$group': {
        'x_positions': {'$push': '$positions.x'},
        'y_positions': {'$push': '$positions.y'}
    }},

    {'$project': {
        'avg_x': {'$avg': "$x_positions"},
         'avg_y': {'$avg': "$y_positions"}
     }}]
```

Step #3: role classification

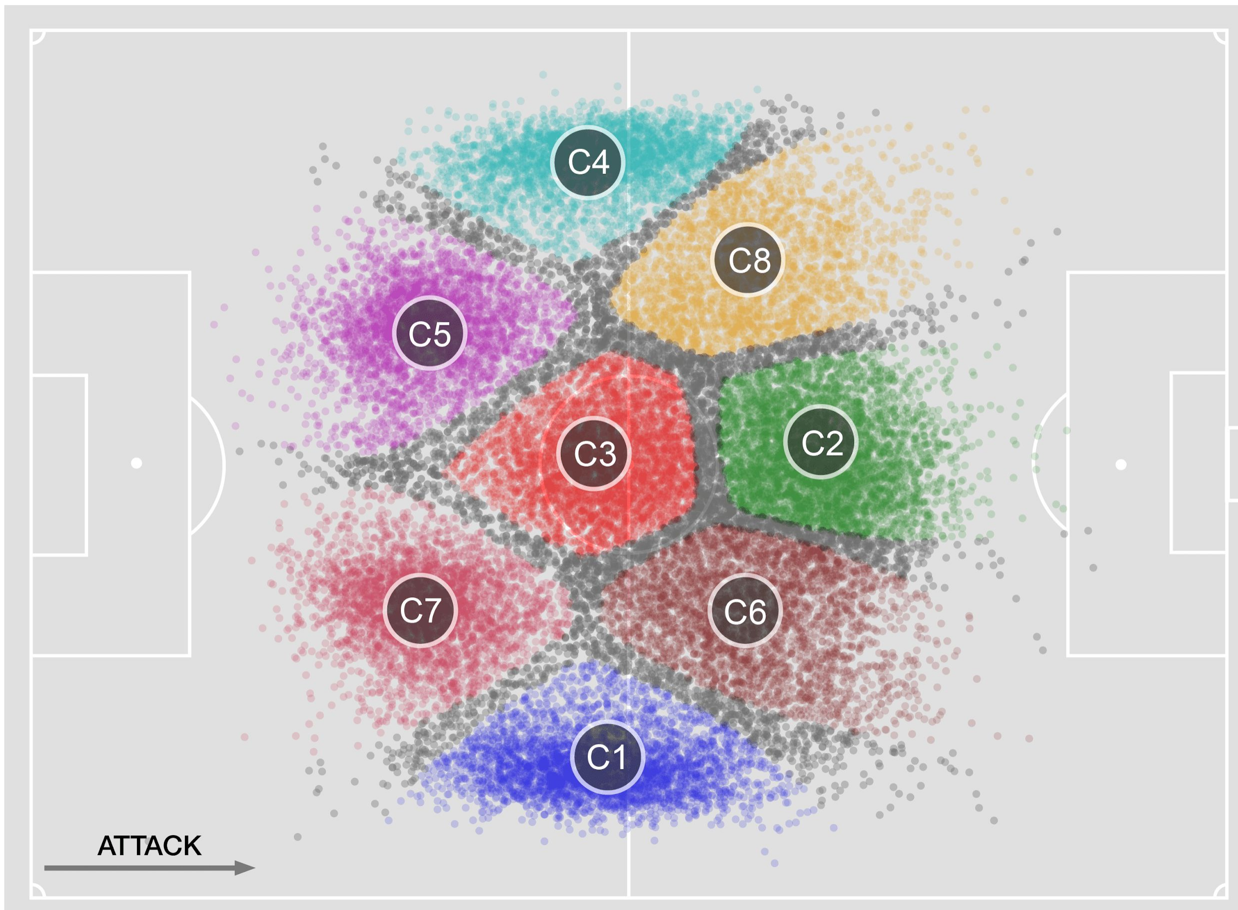


Step #3: role classification

```
from playerank import RoleClusterer

# perform multi-clustering
rc = RoleClusterer(k_range=(2, 20),
   border_threshold=0.2, random_state=42)
rc.fit(X)

rc.labels_
```

```
[[6], [2, 4, 6], [3], [6], [1], [5], [3], [4], [1],
[6], [1], [0], [2, 5], [2], [7], [4], [5], [5], [0],
[4], [5], [4], [6], [3], [5], [1], [6], [4], [0], [7],
[1], [7], [2], [5], [7], [0, 5], ....]
```

Step #3: role classification



Step #4: rating computation

performance rating
of ***u*** in game ***g***

$$r(u,g) = \frac{1}{R}\sum_{i=1}^{n} w_i \times x_i$$

taking into account
the number of goals

$$\alpha \times norm\_goals + (1-\alpha) \times r(u,g)$$

Step #4: rating computation

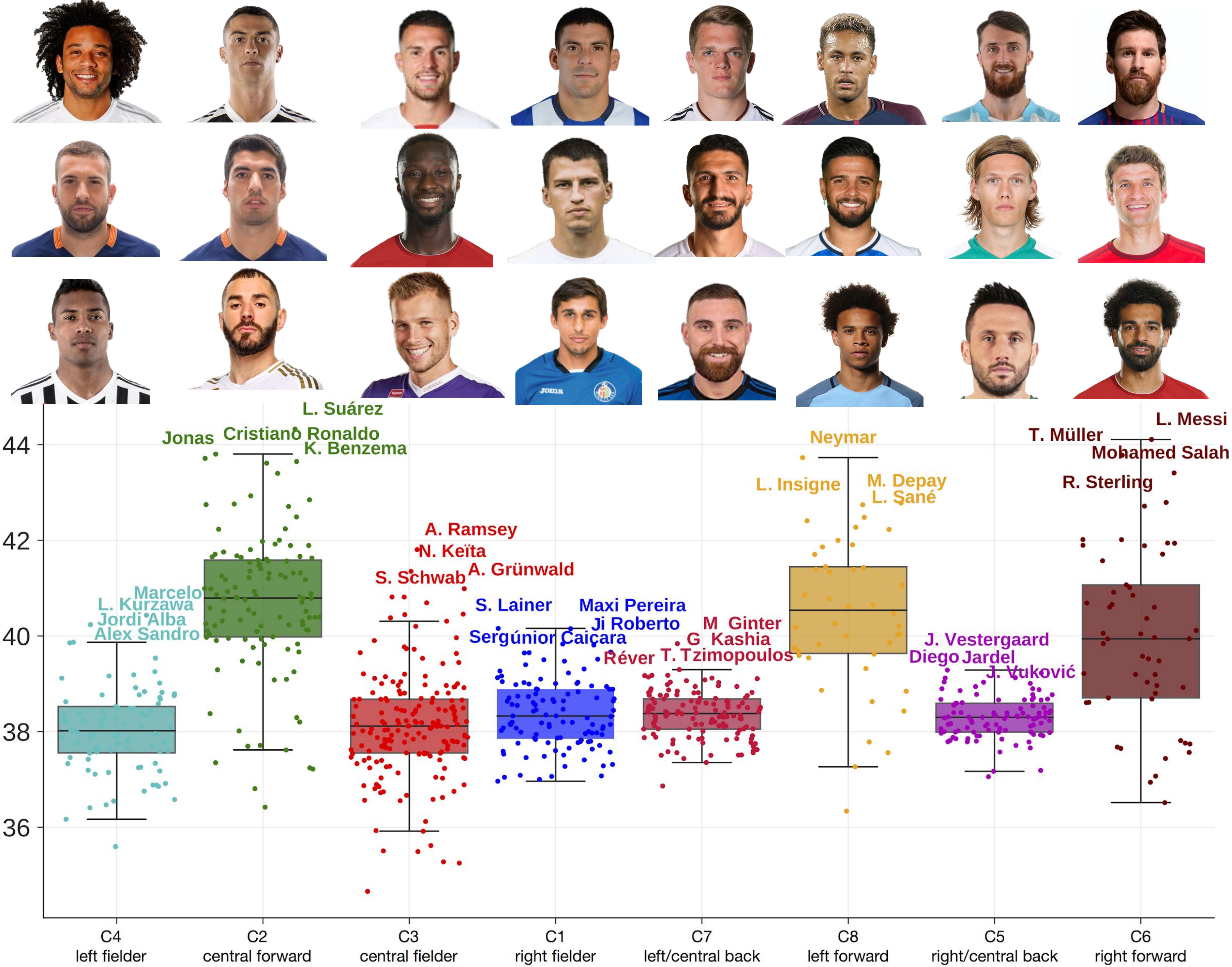
```
from playerank import Rater

res = events.map_reduce(map_aggregate,reduce_sum)
X = extract_data(res)

# rate the performances
rater = Rater(alpha=0.0)
rater.predict(X)
```

goals are not considered

Step #5: player ranking

The ranking of players (by role) can be computed by aggregating over all ratings of the players

```
import pandas as pd

df = pd.read_csv('evaluations.csv')
df.groupby('player_id').mean().sort_values(
              by='rating', ascending=False)
```



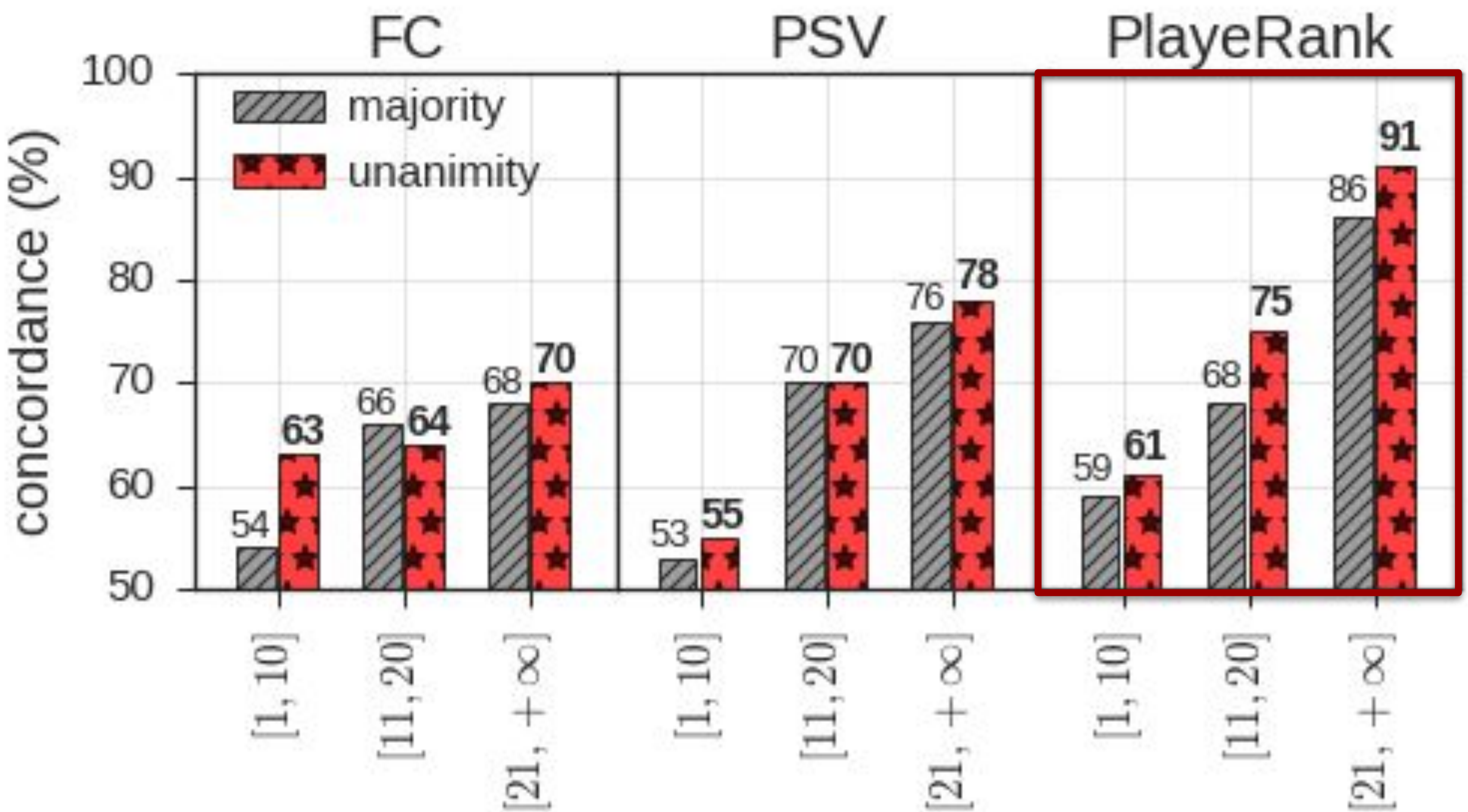
rating
0.44
0.42
0.40
0.38
0.36
L. Suárez
Jonas
Cristiano Ronaldo
K. Benzema
Neymar
L. Messi
T. Müller
Mohamed Salah
L. Insigne
M. Depay
L. Sané
R. Sterling
A. Ramsey
N. Keïta
A. Grünwald
S. Schwab
Marcelo
L. Kurzawa
Jordi Alba
Alex Sandro
S. Lainer
Maxi Pereira
Ji Roberto
Sergúnior Caiçara
M Ginter
G Kashia
Réver
T. Tzimopoulos
J. Vestergaard
Diego Jardel
J. Vuković
C4
left fielder
C2
central forward
C3
central fielder
C1
right fielder
C7
left/central back
C8
left forward
C5
right/central back
C6
right forward

How to evaluate the evaluation?

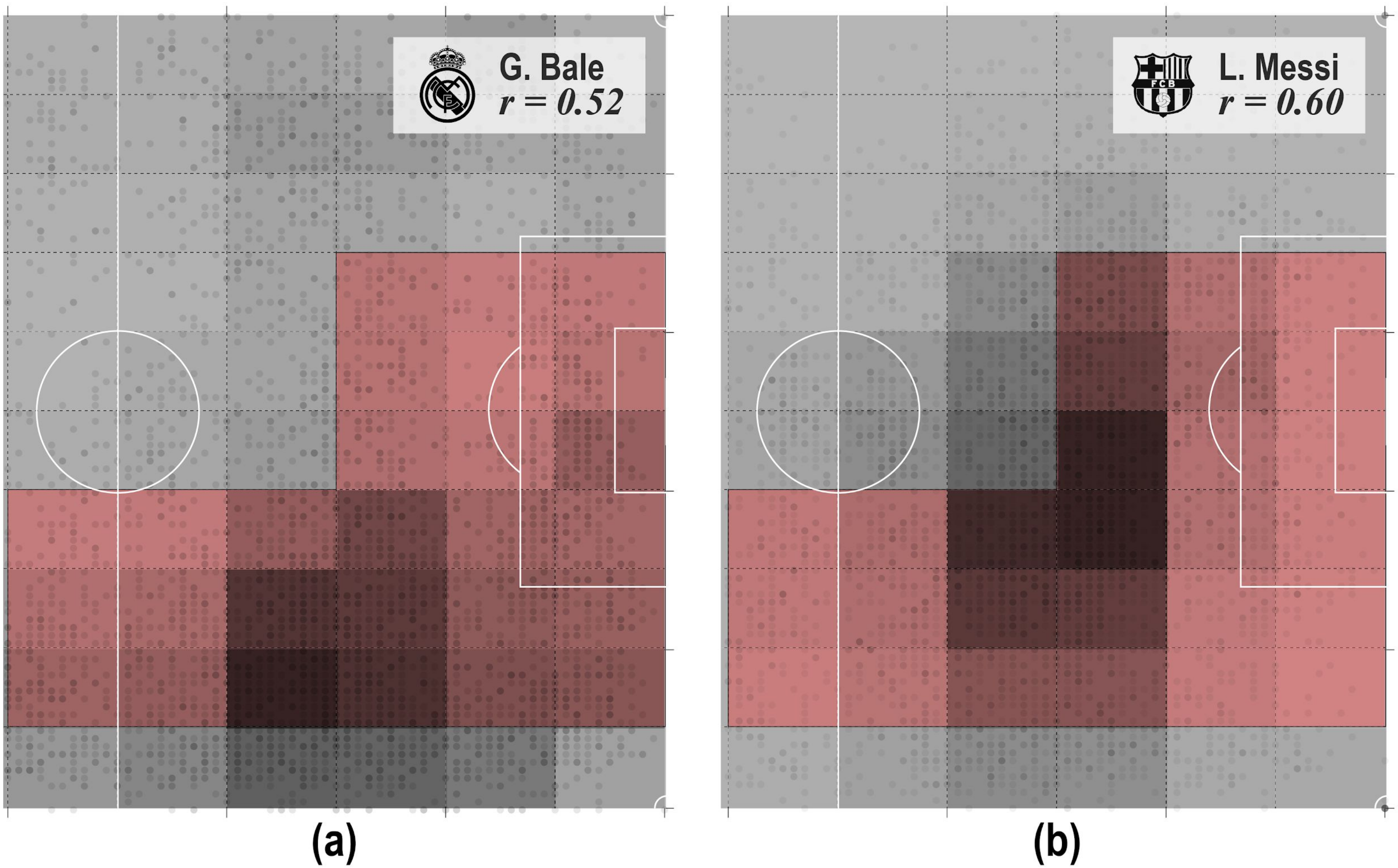
algorithm | expert 1 | expert 2 | expert 3

- majority agreement
- unanimity agreement

Evaluation of 211 pairs



Step #6: the search engine



Step #6: the search engine

	player	$\hat{r}$	r	$\overline{r}$	club
1	L. Messi	**0.28**	0.60	**0.46**	Barcelona
2	A. Robben	0.26	**0.61**	0.43	Bayern M.
3	L. Suárez	0.24	0.54	0.45	Barcelona
4	T. Müller	0.24	0.56	0.43	Bayern M.
5	Mohamed Salah	0.24	0.56	0.43	Liverpool
6	R. Lukaku	0.24	0.56	0.42	Man. Utd
7	A. Petagna	0.23	0.55	0.42	Atalanta
8	D. Berardi	0.22	0.54	0.41	Sassuolo
9	Aduriz	0.22	0.55	0.40	A. Bilbao
10	G. Bale	0.22	0.52	0.43	R. Madrid

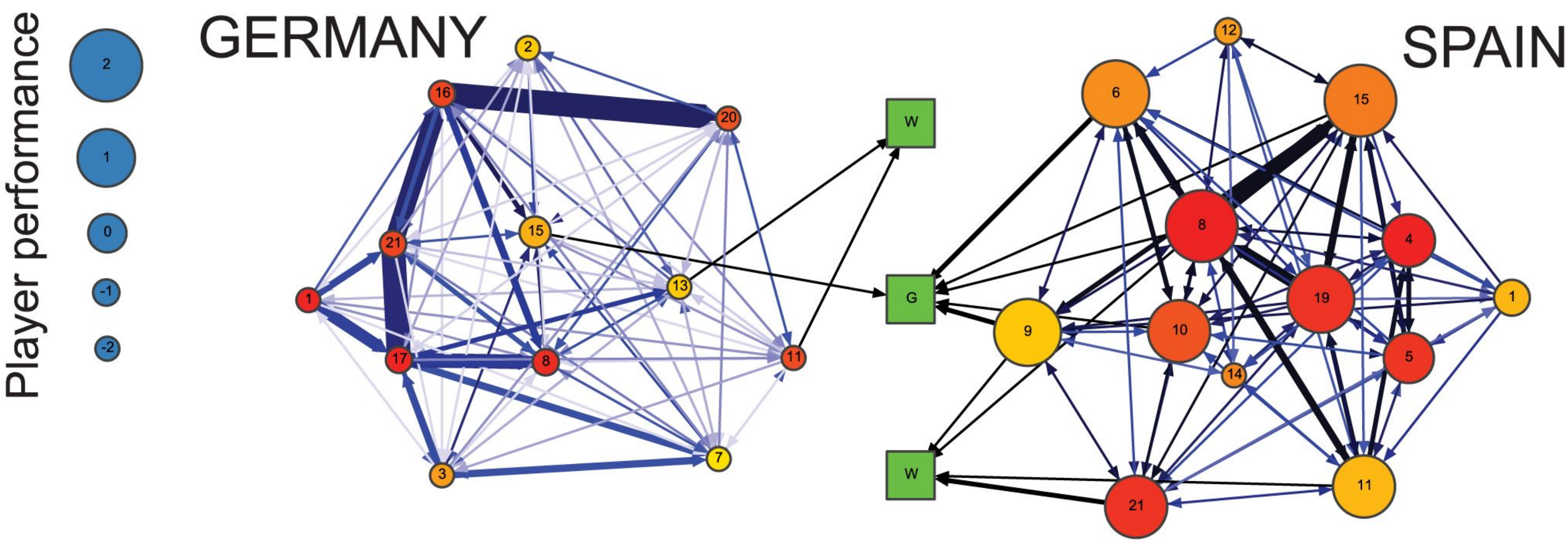
Coming soon:

Soccer Data Challenge @InternetFestival, Pisa, 12-13 October 2018

http://www.internetfestival.it/

Flow Centrality (FC)

Duch et al. (2010) Quantifying the Performance of Individual Players in a Team Activity. PLoS ONE 5(6): e10937.

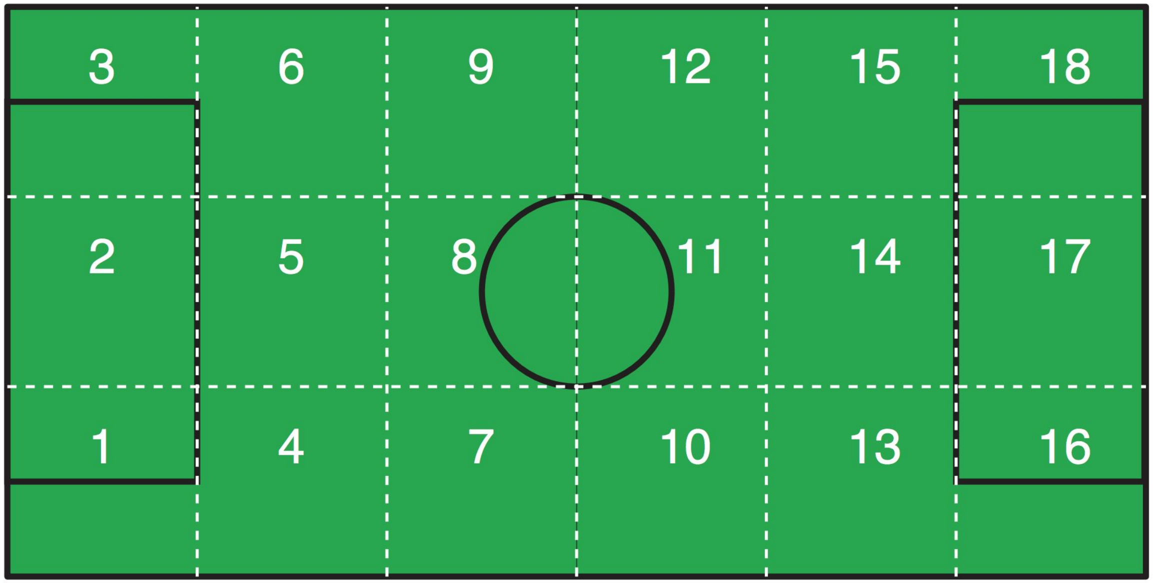


fraction of a player’s accurate shots

Validation: 8 of the 20 players in the list of the competition’s best players

Pass Shot Value (PSV)

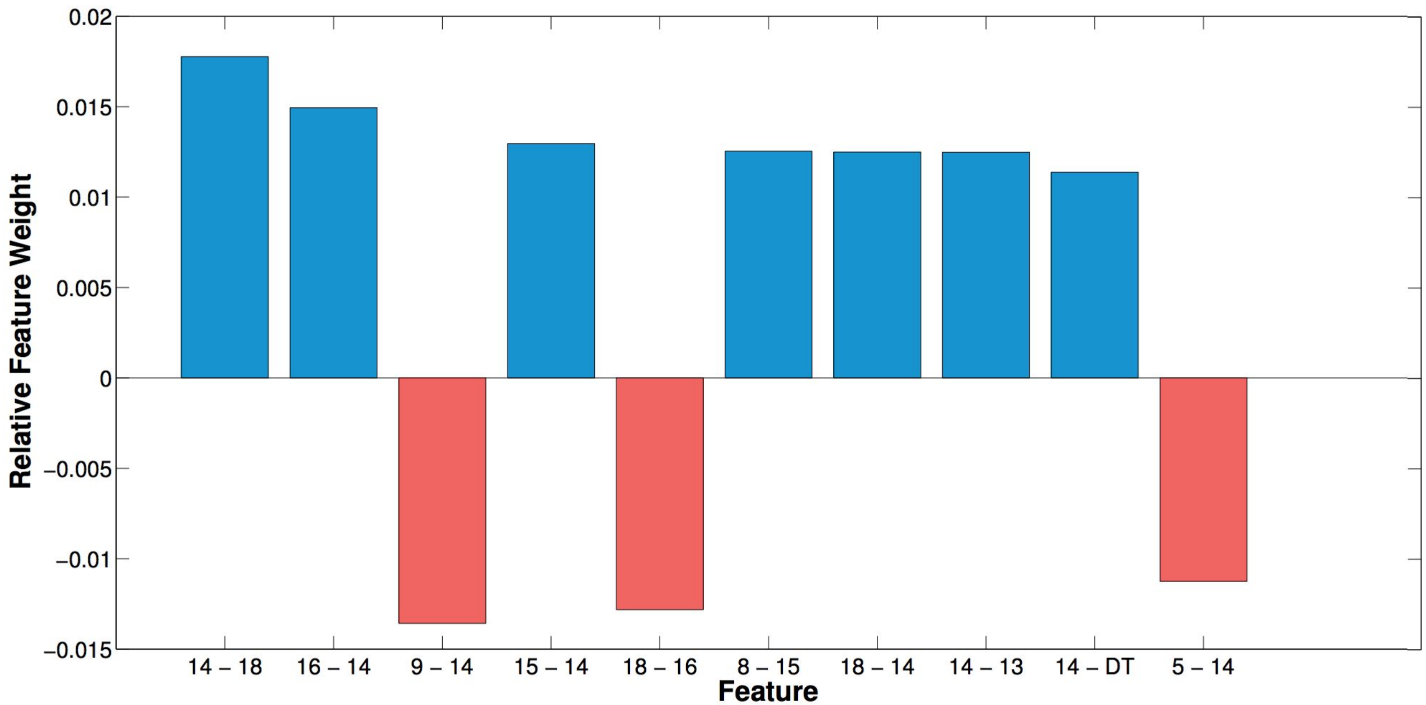
Brooks et al. (2016) Developing a Data-Driven Player Ranking in Soccer using Predictive Model Weights, SIGKDD



each pass is represented as a vector size=360

Pass Shot Value (PSV)

Brooks et al. (2016) Developing a Data-Driven Player Ranking in Soccer using Predictive Model Weights, SIGKDD



predicting if a possession ends in a shot

Validation: correlation with assists and goals