サッカーのプレーに関するデータを PGXに格納してみた

Miki Kawauchi

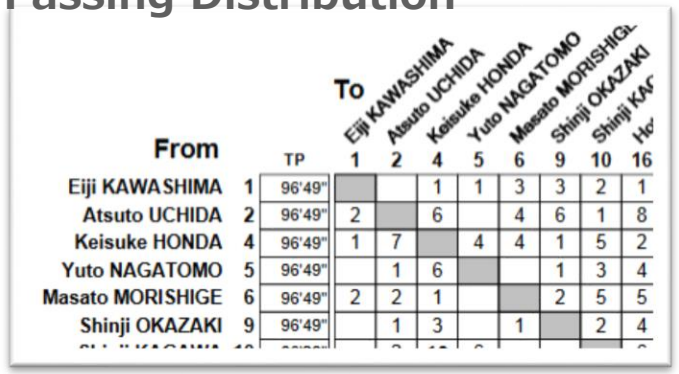
2018/06/20 @PGXユーザ勉強会 #8

ORACLE

準備

- 使用するデータ：
  - 日本代表チーム (**2014** FIFA World Cup Brazil, 3試合分) ※前回の大会です
  - FIFAサイトで各試合のスタッツ → Passing Distribution(PDF資料)から作成
- PGXインストールと使い方はこちらの資料を参考にしました
  - ※PGXはLab版を使用, Oracle Labs PGX (PGX 2.7.0)

Passing Distribution



From		TP	To 1	2	4	5	6	9	10	16
Eiji KAWASHIMA	1	96'49"			1	1	3	3	2	1
Atsuto UCHIDA	2	96'49"	2		6		4	6	1	8
Keisuke HONDA	4	96'49"	1	7		4	4	1	5	2
Yuto NAGATOMO	5	96'49"		1	6			1	3	4
Masato MORISHIGE	6	96'49"	2	2	1			2	5	5
Shinji OKAZAKI	9	96'49"		1	3		1		2	4

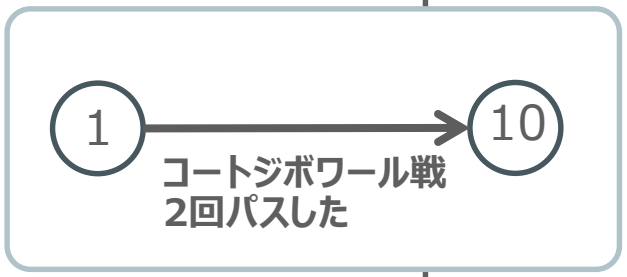
wc2014jpn.csv.json

```
{
        "uri": "wc2014jpn.csv"
,       "format": "edge_list"
,       "node_id_type": "integer"
,       "vertex_props": [
                {"name":"name", "type":"string"}
                ,{"name":"position", "type":"string"}
        ]
,       "edge_props":[
                {"name":"matchid","type":"string"}
                ,{"name":"numofpasses", "type":"integer"}
        ]
,       "separator": ","
}
```

wc2014jpn.csv

```
1,*,KAWASHIMA,GK
2,*,UCHIDA,DF
3,*,GOTOKU,DF
4,*,HONDA,FW
...
1,10,2014-06,2 <-- ※1
1,10,2014-22,1
...
```

※1
<from>, <to>, <MatchID>, <パス回数> を示す
背番号1から背番号10への2014-06(=2014年W杯コートジボワール戦)でのパス回数は2回



やってみる

■グラフをロード

```
pgx> G=session.readGraphWithProperties("wc2014jpn.csv.json")
```

■x->yにパスの回数が多いのはどの組み合わせか？

```
pgx> G.queryPgql(" SELECT x.name, y.name, sum(r.numofpasses) as numofpasses WHERE (x) - [r] -> (y) group by
x.name, y.name order by numofpasses desc").getResults();
==> x.name(STRING)=YOSHIDA y.name(STRING)=KONNO numofpasses(LONG)=37
==> x.name(STRING)=KAWASHIMA y.name(STRING)=YOSHIDA numofpasses(LONG)=30
==> x.name(STRING)=YOSHIDA y.name(STRING)=UCHIDA numofpasses(LONG)=21
…
```

■たくさんパスを出した人

```
pgx> G.queryPgql(" SELECT x.name, sum(r.numofpasses) as numofpasses WHERE (x) - [r] -> () group by x.name
order by numofpasses desc").getResults();
==> x.name(STRING)=YOSHIDA numofpasses(LONG)=139
==> x.name(STRING)=HONDA numofpasses(LONG)=125
==> x.name(STRING)=HASEBE numofpasses(LONG)=118
…
```

■たくさんパスを受けた人

```
pgx> G.queryPgql(" SELECT x.name, sum(r.numofpasses) as numofpasses WHERE (x) <- [r] - () group by x.name
order by numofpasses desc").getResults();
==> x.name(STRING)=HONDA numofpasses(LONG)=126
==> x.name(STRING)=YAMAGUCHI numofpasses(LONG)=117
==> x.name(STRING)=KONNO numofpasses(LONG)=112
…
```

注：LONGといってもデータ型がLONG型であることを示しています(ロングパスかどうかではない)

もうちょっとやってみる

ここまでのページの中で
唯一グラフっぽいクエリ

■たくさんボールにかかわった人（受けた、出した、両方）※無向グラフにしたら両方カウントされる

```
pgx> G.queryPgql(" SELECT x.name, sum(r.numofpasses) as numofpasses WHERE (x) - [r] - () group by x.name
order by numofpasses desc").getResults();
==> x.name(STRING)=HONDA numofpasses(LONG)=251
==> x.name(STRING)=YOSHIDA numofpasses(LONG)=242
==> x.name(STRING)=HASEBE numofpasses(LONG)=216
…
```

■たくさんバックパスした人（GKにパスを出した人）

```
pgx> G.queryPgql(" SELECT x.name, sum(r.numofpasses) as numofpasses WHERE (x) - [r] -> (y),
y.position='GK',r.numofpasses>0  group by x.name order by numofpasses desc").getResults();
==> x.name(STRING)=YOSHIDA numofpasses(LONG)=7
==> x.name(STRING)=UCHIDA numofpasses(LONG)=4
==> x.name(STRING)=KONNO numofpasses(LONG)=2
…
```

■GKからのパスにターゲットになった人

```
pgx> G.queryPgql(" SELECT y.name, sum(r.numofpasses) as numofpasses WHERE (x) - [r] -> (y),
x.position='GK',r.numofpasses>0  group by y.name order by numofpasses desc").getResults();
==> y.name(STRING)=YOSHIDA numofpasses(LONG)=30
==> y.name(STRING)=KONNO numofpasses(LONG)=12
==> y.name(STRING)=HASEBE numofpasses(LONG)=5
==> y.name(STRING)=MORISHIGE numofpasses(LONG)=3
…
```

グラフっぽいことやってみる

※edgeの有無だけで計算しており、パスの回数は考慮されていません

■次数中心性、ページランク、媒介中心性を計算

```
pgx> analyst.inDegreeCentrality(G)
==> VertexProperty[name=in_degree,type=integer,graph=wc2014jp]
pgx> analyst.pagerank(G, 0.0001, 0.85, 100)
==> VertexProperty[name=pagerank,type=double,graph=wc2014jp]
pgx> analyst.vertexBetweennessCentrality(G)
==> VertexProperty[name=betweenness,type=double,graph=wc2014jp]
```

■次数中心性

```
pgx> G.queryPgql("SELECT n.name, n.in_degree where (n) order by n.in_degree desc").getResults();
==> n.name(STRING)=HONDA n.in_degree(INTEGER)=33
==> n.name(STRING)=YAMAGUCHI n.in_degree(INTEGER)=29
==> n.name(STRING)=KAGAWA n.in_degree(INTEGER)=28
…
```

■ページランク

```
pgx> G.queryPgql("SELECT n.name, n.pagerank where (n) order by n.pagerank desc").getResults();
==> n.name(STRING)=HONDA n.pagerank(DOUBLE)=0.07122741492013186
==> n.name(STRING)=UCHIDA n.pagerank(DOUBLE)=0.06271875899374495
==> n.name(STRING)=KAGAWA n.pagerank(DOUBLE)=0.06044043413002474
…
```

■媒介中心性

```
pgx> G.queryPgql("SELECT n.name, n.betweenness where (n) order by n.betweenness desc").getResults();
==> n.name(STRING)=HASEBE n.betweenness(DOUBLE)=15.791310737849
==> n.name(STRING)=YOSHIDA n.betweenness(DOUBLE)=11.953225208
==> n.name(STRING)=UCHIDA n.betweenness(DOUBLE)=11.821359396328
…
```

グラフデータベース的には
「長谷部選手がいないとみんなが困る」という結果

昨日の試合でやってみる (2018/6/19 vsコロンビア)
※先の例を、昨日の1試合分だけのデータを準備・使用して実行

■グラフをロード

```
pgx>  G=session.readGraphWithProperties("wc2018jpn.csv.json")
==> PgxGraph[name=wc2018jpn,N=23,E=108,created=1529472535412]
```

■x->yにパスの回数が多いのはどの組み合わせか？

```
pgx> G.queryPgql(" SELECT x.name, y.name, sum(r.numofpasses) as numofpasses WHERE (x) - [r] -> (y) group by x.name, y.name order by numofpasses desc").getResults();
==> x.name(STRING)=YOSHIDA y.name(STRING)=SHOJI numofpasses(LONG)=24
==> x.name(STRING)=SHOJI y.name(STRING)=YOSHIDA numofpasses(LONG)=21
==> x.name(STRING)=HASEBE y.name(STRING)=YOSHIDA numofpasses(LONG)=21
==> x.name(STRING)=YOSHIDA y.name(STRING)=SHIBASAKI numofpasses(LONG)=18
…
```

昨日の試合でやってみる (2018/6/19 vsコロンビア)
※先の例を、昨日の1試合分だけのデータを準備・使用して実行

■たくさんパスを出した人
```
pgx>  G.queryPgql(" SELECT x.name, sum(r.numofpasses) as numofpasses WHERE (x) - [r] -> () group by x.name
order by numofpasses desc").getResults();
==> x.name(STRING)=YOSHIDA numofpasses(LONG)=85
==> x.name(STRING)=HASEBE numofpasses(LONG)=68
==> x.name(STRING)=SHOJI numofpasses(LONG)=66
==> x.name(STRING)=SHIBASAKI numofpasses(LONG)=60
…
```

■たくさんパスを受けた人
```
pgx> G.queryPgql(" SELECT x.name, sum(r.numofpasses) as numofpasses WHERE (x) <- [r] - () group by x.name
order by numofpasses desc").getResults();
==> x.name(STRING)=YOSHIDA numofpasses(LONG)=80
==> x.name(STRING)=HASEBE numofpasses(LONG)=58
==> x.name(STRING)=SHOJI numofpasses(LONG)=58
==> x.name(STRING)=SHIBASAKI numofpasses(LONG)=55
…
```

■たくさんボールにかかわった人（受けた、出した、両方）
```
pgx>  G.queryPgql(" SELECT x.name, sum(r.numofpasses) as numofpasses WHERE (x) - [r] - () group by x.name
order by numofpasses desc").getResults();
==> x.name(STRING)=YOSHIDA numofpasses(LONG)=165
==> x.name(STRING)=HASEBE numofpasses(LONG)=126
==> x.name(STRING)=SHOJI numofpasses(LONG)=124
==> x.name(STRING)=SHIBASAKI numofpasses(LONG)=115
…
```

昨日の試合でやってみる (2018/6/19 vsコロンビア)

※先の例を、昨日の1試合分だけのデータを準備・使用して実行

■たくさんバックパスした人（GKにパスを出した人）

```
pgx> G.queryPgql(" SELECT x.name, sum(r.numofpasses) as numofpasses WHERE (x) - [r] ->
(y) ,y.position='GK',r.numofpasses>0  group by x.name order by numofpasses desc").getResults();
==> x.name(STRING)=YOSHIDA numofpasses(LONG)=3
==> x.name(STRING)=NAGATOMO numofpasses(LONG)=3
==> x.name(STRING)=SHOJI numofpasses(LONG)=2
==> x.name(STRING)=HASEBE numofpasses(LONG)=1
…
```

■GKからのパスにターゲットになった人

```
pgx> G.queryPgql(" SELECT y.name, sum(r.numofpasses) as numofpasses WHERE (x) - [r] ->
(y) ,x.position='GK',r.numofpasses>0 group by y.name order by numofpasses desc").getResults();
==> y.name(STRING)=YOSHIDA numofpasses(LONG)=7
==> y.name(STRING)=HIROKI numofpasses(LONG)=3
==> y.name(STRING)=SHOJI numofpasses(LONG)=2
==> y.name(STRING)=HASEBE numofpasses(LONG)=1
…
```

昨日の試合でやってみる (2018/6/19 vsコロンビア)

※先の例を、昨日の1試合分だけのデータを準備・使用して実行

■次数中心性、ページランク、媒介中心性を計算　※edgeの有無だけで計算しているので、パスの回数は反映されていない。

```
pgx> analyst.inDegreeCentrality(G)
==> VertexProperty[name=in_degree,type=integer,graph=wc2018jpn]
pgx>   analyst.pagerank(G, 0.0001, 0.85, 100)
==> VertexProperty[name=pagerank,type=double,graph=wc2018jpn]
pgx> analyst.vertexBetweennessCentrality(G)
==> VertexProperty[name=betweenness,type=double,graph=wc2018jpn]
```

■次数中心性

```
pgx> G.queryPgql("SELECT n.name, n.in_degree where (n) order by n.in_degree desc").getResults();
==> n.name(STRING)=HASEBE n.in_degree(INTEGER)=12
==> n.name(STRING)=SHIBASAKI n.in_degree(INTEGER)=10
==> n.name(STRING)=OSAKO n.in_degree(INTEGER)=9
==> n.name(STRING)=HIROKI n.in_degree(INTEGER)=9
…
```

■ページランク

```
pgx> G.queryPgql("SELECT n.name, n.pagerank where (n) order by n.pagerank desc").getResults();
==> n.name(STRING)=HASEBE n.pagerank(DOUBLE)=0.0659039014873658
==> n.name(STRING)=SHIBASAKI n.pagerank(DOUBLE)=0.0563965515516788
==> n.name(STRING)=OSAKO n.pagerank(DOUBLE)=0.05054382647635229
==> n.name(STRING)=HIROKI n.pagerank(DOUBLE)=0.049409090782279876
…
```

■媒介中心性

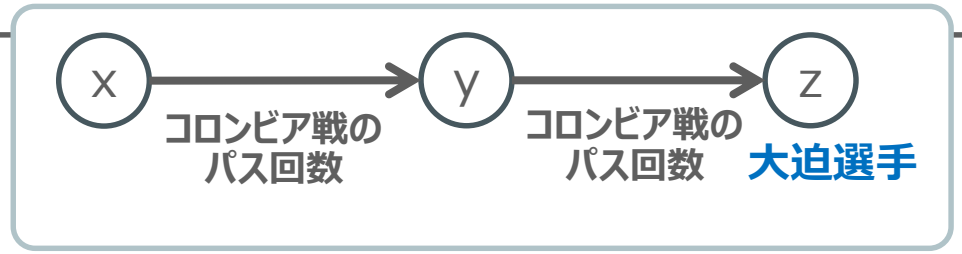
```
pgx>  G.queryPgql("SELECT n.name, n.betweenness where (n) order by n.betweenness desc").getResults();
==> n.name(STRING)=YOSHIDA n.betweenness(DOUBLE)=19.05
==> n.name(STRING)=HASEBE n.betweenness(DOUBLE)=15.461111111111111
==> n.name(STRING)=HONDA n.betweenness(DOUBLE)=13.427777777777777
==> n.name(STRING)=SHIBASAKI n.betweenness(DOUBLE)=9.577777777777
…
```

グラフデータベース的には
「吉田選手がいないとみんなが困る」という結果

昨日の試合でやってみる (2018/6/19 vsコロンビア)

■コロンビア戦で大迫選手へつながる2ホップの組み合わせで、パス回数合計値の多い順に並べてみる

```
PGX> G.queryPgql(" SELECT x.name, y.name, z.name, r1.numofpasses, r2.numofpasses, r1.numofpasses +
r2.numofpasses as totalpasses WHERE (x) - [r1] -> (y) - [r2] -> (z), z.name='OSAKO', r1.matchid='2018-16',
r2.matchid='2018-16' order by totalpasses desc").getResults();
==> x.name(STRING)=YOSHIDA y.name(STRING)=SHOJI z.name(STRING)=OSAKO r1.numofpasses(INTEGER)=24
r2.numofpasses(INTEGER)=2 totalpasses(INTEGER)=26
==> x.name(STRING)=YOSHIDA y.name(STRING)=SHIBASAKI z.name(STRING)=OSAKO r1.numofpasses(INTEGER)=18
r2.numofpasses(INTEGER)=3 totalpasses(INTEGER)=21
==> x.name(STRING)=SHOJI y.name(STRING)=NAGATOMO z.name(STRING)=OSAKO r1.numofpasses(INTEGER)=12
r2.numofpasses(INTEGER)=5 totalpasses(INTEGER)=17
==> x.name(STRING)=SHOJI y.name(STRING)=HASEBE z.name(STRING)=OSAKO r1.numofpasses(INTEGER)=14
r2.numofpasses(INTEGER)=1 totalpasses(INTEGER)=15
==> x.name(STRING)=YOSHIDA y.name(STRING)=HASEBE z.name(STRING)=OSAKO r1.numofpasses(INTEGER)=14
r2.numofpasses(INTEGER)=1 totalpasses(INTEGER)=15
==> x.name(STRING)=YOSHIDA y.name(STRING)=HASEBE z.name(STRING)=OSAKO r1.numofpasses(INTEGER)=14
r2.numofpasses(INTEGER)=1 totalpasses(INTEGER)=15
…
```




ORACLE®