

P-Center em Julia

February 11, 2016

1 Trabalho de Implementação

1.1 INF2912 - Otimização Combinatória

1.1.1 Prof. Marcus Vinicius Soledade Poggi de Aragão

1.1.2 2015-2

1.1.3 Ciro Cavani

BigData / Globo.com Algoritmos de clusterização.

1.2 Conteúdo

Esse notebook tem o desenvolvimento e avaliação do algoritmo aproximado do P-Center (algoritmo Farthest-first traversal).

A avaliação do algoritmo é baseada em um mapeamento entre a maioria dos itens que foram atribuídos a um determinado cluster e o correspondente os valores verdadeiros gerados nesse cluster.

O P-Center teve resultados muito bons.

1.3 Dataset

```
In [1]: include("../src/clustering.jl")
import Inf2912Clustering
const Clustering = Inf2912Clustering
dataset = Clustering.load_dataset("small")
Clustering.summary(dataset)
sleep(0.2)
```

```
Number of Groups: 3
Number of Features: 200
Number of Features (group): 40
Probability of Activation: 0.8
Number of Objects (total): 100
Number of Objects per Group (min): 7
Number of Objects per Group (max): 66
Number of Objects in 1: 39
Number of Objects in 2: 17
Number of Objects in 3: 45
```

1.3.1 P-Center - Problema de Localização de Centróides

Consiste em resolver o P-Center determinar os objetos representantes de cada grupo e classificar cada objeto como sendo do grupo com representante mais próximo

https://en.wikipedia.org/wiki/Metric_k-center

https://en.wikipedia.org/wiki/Farthest-first_traversal

```

In [2]: let
    k = 3
    data = map(first, dataset.data)

    centers = Array(Array{Int64,1}, 0)
    i = rand(1:length(data))
    push!(centers, data[i])

    min_dist(v) = minimum(map(c -> norm(c - v), centers))
    max_index() = indmax(map(min_dist, data))

    while length(centers) < k
        i = max_index()
        push!(centers, data[i])
    end

    cluster(v) = indmin(map(c -> norm(c - v), centers))

    assignments = zeros{Int, length(data)}
    for (i, v) in enumerate(data)
        assignments[i] = cluster(v)
    end

    assignments
end

```

```

Out[2]: 101-element Array{Int64,1}:

```

```

1
1
1
3
3
2
2
3
2
1
1
1
3
⋮
2
3
1
1
3
1
1
1
2
1
1
2

```

```

In [3]: function pcenter(dataset, k)

```

```

data = map(first, dataset.data)

centers = Array(Array{Int64,1}, 0)
i = rand(1:length(data))
push!(centers, data[i])

min_dist(v) = minimum(map(c -> norm(c - v), centers))
max_index() = indmax(map(min_dist, data))

while length(centers) < k
    i = max_index()
    push!(centers, data[i])
end

cluster(v) = indmin(map(c -> norm(c - v), centers))

assignments = zeros{Int, length(data)}
for (i, v) in enumerate(data)
    assignments[i] = cluster(v)
end

assignments
end

pcenter(dataset, 3)

```

Out[3]: 101-element Array{Int64,1}:

```

3
3
3
1
1
2
2
1
2
3
3
3
1
⋮
2
1
3
3
1
3
3
3
2
3
3
2

```

In [4]: function pcenter_approx(dataset, k)

```

        assignments = pcenter(dataset, k)
        centermap = Clustering.mapping(dataset, assignments, k)
        map(c -> centermap[c], assignments)
    end

    let
        n = 100
        k = 3
        c = 16
        c_y = 3

        tiny = Clustering.Dataset(size=n, groups=k, features=c, slot=c_y)

        prediction = pcenter_approx(tiny, k)
        Clustering.evaluation_summary(tiny, prediction)
    end

```

Precision: 95.06%
 Recall: 80.21%
 F-score: 0.87

Número de predições: 100
 Acertos: 77 (77.0%)
 Falso negativo: 19 (19.0%)
 Falso positivo: 4 (4.0%)

Cluster 1

Objetos: 14
 Accuracy: 95.0%
 Precision: 90.91%
 Recall: 71.43%
 F-score: 0.8

Acerto positivo: 10 (71.43%)
 Acerto negativo: 85 (98.84%)
 Falso negativo: 4 (21.05%)
 Falso positivo: 1 (25.0%)

Cluster 2

Objetos: 47
 Accuracy: 78.0%
 Precision: 82.05%
 Recall: 68.09%
 F-score: 0.74

Acerto positivo: 32 (68.09%)
 Acerto negativo: 46 (86.79%)
 Falso negativo: 15 (78.95%)
 Falso positivo: 7 (175.0%)

Cluster 3

Objetos: 39
Accuracy: 81.0%
Precision: 70.0%
Recall: 89.74%
F-score: 0.79

Acerto positivo: 35 (89.74%)
Acerto negativo: 46 (75.41%)
Falso negativo: 4 (21.05%)
Falso positivo: 15 (375.0%)