P-Center em Julia

February 14, 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
```

```
WARNING: redefining constant srcdir
WARNING: redefining constant default_datasetdir
```

```
Out[1]: Inf2912Clustering
```

```
In [2]: dataset = Clustering.dataset_tiny()
        Clustering.summary(dataset)
        sleep(0.2)
```

```
Clusters: 3
Dimension (features): 16
Features per Cluster: 3
Probability of Activation: 0.8

Size: 100
Min Cluster size: 20
Max Cluster size: 40
Cluster 1 size: 37
Cluster 2 size: 36
Cluster 3 size: 27
```

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 [3]: let
            k = 3
            data = dataset.input.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[3]: 100-element Array{Int64,1}:
         3
         2
         1
         1
         1
         2
         3
         1
         1
         2
         2
         1
         2
         1
         1
         2
         1
         1
         ⋮
         1
         3
```

```
 1
 2
 2
 3
 2
 2
 3
 2
 1
 1
 2
 2
 3
 1
 3
```

In [4]:

```
import Clustering: Input, Dataset

"Algoritmo de clusterização P-Center (algoritmo Farthest-first traversal)."
function pcenter(input::Input, k::Int)
    data = input.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::Dataset, k::Int) = pcenter(dataset.input, k)

pcenter(dataset, 3)
```

Out[4]:

```
100-element Array{Int64,1}:
 1
 2
 2
 3
 2
 2
 3
 2
 1
 1
 1
 2
 1
 1
 2
 1
 2
 1
 ⋮
 1
 2
 2
 2
 2
 1
 2
 2
 1
 2
 1
 1
 1
 1
 1
 1
 2
```

```
In [5]: import Clustering.mapping

        "Algoritmo de clusterização P-Center (algoritmo Farthest-first traversal) \
        aproximado para os grupos pré-definidos do dataset."
        function pcenter_approx(dataset::Dataset, k::Int)
            assignments = pcenter(dataset, k)
            centermap = mapping(dataset, assignments, k)
            map(c -> centermap[c], assignments)
        end

        let
            k = dataset.clusters
            @time prediction = pcenter_approx(dataset, k)
            Clustering.evaluation_summary(dataset, prediction; verbose=true)
            sleep(0.2)
        end
```

```
0.109623 seconds (117.02 k allocations: 5.600 MB)
Confusion Matrix:

[19 11 7
 7 25 4
 6 4 17]
```

```
Size: 100
Correct: 61
Mistakes: 39
Accuracy: 61.0%

Cluster 1

Size: 37
Accuracy: 69.0%
Precision: 59.38%
Recall: 51.35%
F-score: 0.55

True Positive: 19 (51.35%)
True Negative: 50 (79.37%)
False Negative: 18 (46.15%)
False Positive: 13 (33.33%)

Cluster 2

Size: 36
Accuracy: 74.0%
Precision: 62.5%
Recall: 69.44%
F-score: 0.66

True Positive: 25 (69.44%)
True Negative: 49 (76.56%)
False Negative: 11 (28.21%)
False Positive: 15 (38.46%)

Cluster 3

Size: 27
Accuracy: 79.0%
Precision: 60.71%
Recall: 62.96%
F-score: 0.62

True Positive: 17 (62.96%)
True Negative: 62 (84.93%)
False Negative: 10 (25.64%)
False Positive: 11 (28.21%)
```

```
In [6]: Clustering.test_dataset("small", pcenter_approx)
        sleep(0.2)
```

```
0.018788 seconds (33.10 k allocations: 11.160 MB)
Confusion Matrix:

[367 0 0
 0 265 1
 2 1 364]
```

```
Size: 1000
Correct: 996
Mistakes: 4
Accuracy: 99.6%

Cluster 1

Size: 367
Accuracy: 99.8%
Precision: 99.46%
Recall: 100.0%
F-score: 1.0

True Positive: 367 (100.0%)
True Negative: 631 (99.68%)
False Negative: 0 (0.0%)
False Positive: 2 (50.0%)

Cluster 2

Size: 266
Accuracy: 99.8%
Precision: 99.62%
Recall: 99.62%
F-score: 1.0

True Positive: 265 (99.62%)
True Negative: 733 (99.86%)
False Negative: 1 (25.0%)
False Positive: 1 (25.0%)

Cluster 3

Size: 367
Accuracy: 99.6%
Precision: 99.73%
Recall: 99.18%
F-score: 0.99

True Positive: 364 (99.18%)
True Negative: 632 (99.84%)
False Negative: 3 (75.0%)
False Positive: 1 (25.0%)
```

```
In [7]: Clustering.test_dataset("large", pcenter_approx)
        sleep(0.2)
```

```
0.213963 seconds (339.12 k allocations: 111.643 MB, 35.51% gc time)
Confusion Matrix:

[3804 1 9
 5 3968 0
 10 3 2200]

Size: 10000
```

```
Correct: 9972
Mistakes: 28
Accuracy: 99.72%

Cluster 1

Size: 3814
Accuracy: 99.75%
Precision: 99.61%
Recall: 99.74%
F-score: 1.0

True Positive: 3804 (99.74%)
True Negative: 6171 (99.76%)
False Negative: 10 (35.71%)
False Positive: 15 (53.57%)

Cluster 2

Size: 3973
Accuracy: 99.91%
Precision: 99.9%
Recall: 99.87%
F-score: 1.0

True Positive: 3968 (99.87%)
True Negative: 6023 (99.93%)
False Negative: 5 (17.86%)
False Positive: 4 (14.29%)

Cluster 3

Size: 2213
Accuracy: 99.78%
Precision: 99.59%
Recall: 99.41%
F-score: 1.0

True Positive: 2200 (99.41%)
True Negative: 7778 (99.88%)
False Negative: 13 (46.43%)
False Positive: 9 (32.14%)
```