

Aprendizagem não-supervisionada

K-means (2013): Usado para agrupar dados.

São dadas um conjunto QX de instâncias para classificar em diferentes classes. Normalmente, fornece-se também o número de classes k .

Exemplo: Peso da pessoa em kg e classificação em 3 grupos.

Amostra de treinamento (QX):

qx (features, data)
4
15
65
5
18
70

Classifique em $k=3$ classes.

classe 1:

4, 5

classe 2:

15, 18

classe 3:

65, 70

2) Os algoritmos de aprendizagem não-supervisionadas

2a) k-means.

2b) Median cut.

2c) Hierarchical clustering.

2d) Kohonen

```

//kmeans.cpp grad2017
#include <cekeikon.h>

int main() {
    int nl=8;
    int nc=2;

    Mat_<FLT> data = (Mat_<FLT>(nl,nc) <<
        10, 9,
        7, 96,
        10, 8,
        8, 7,
        10, 91,
        8, 97,
        12, 92,
        7, 11);

    Mat_<int> bestLabels(nl,1);
    Mat_<FLT> centers(2,nc);
    TermCriteria criteria(TermCriteria::COUNT | TermCriteria::EPS, 20, 0.1);
    double compactness=kmeans(data,2,bestLabels,criteria,1,KMEANS_PP_CENTERS,centers);
    // double kmeans(InputArray data, int K, InputOutputArray bestLabels,
    // TermCriteria criteria, int attempts, int flags,
    // OutputArray centers=noArray() )

    cout << "Compactness=" << sqrt(compactness/nl) << endl;
    cout << "BestLabels=" << bestLabels << endl;
    cout << "Centers=" << centers << endl;
}

```

Saída:

```

>kmeans
Compactness=2.65165
BestLabels=[1;
0;
1;
1;
0;
0;
0;
1]
Centers=[9.25, 94;
8.75, 8.75]

```

Segmentar feijão pela cor usando k-means (aprendizagem não-supervisionada):

Classificando em duas classes:

```
//kmeanbean1.cpp - 2015
#include <cekeikon.h>

int main()
{ Mat_<COR> qx; le(qx,"f1.jpg");
  Mat_<GRY> qp(qx.rows,qx.cols);

  Mat_<FLT> data(qx.total(),3);
  for (unsigned i=0; i<qx.total(); i++) {
    data(i,0)=qx(i)[0];
    data(i,1)=qx(i)[1];
    data(i,2)=qx(i)[2];
  }

  int K=2;
  Mat_<int> bestLabels(qx.total(),1);
  Mat_<FLT> centers(K,3);
  TermCriteria criteria(TermCriteria::COUNT | TermCriteria::EPS, 80, 2.0);
  kmeans(data,K,bestLabels,criteria,50,KMEANS_RANDOM_CENTERS,centers);

  for (unsigned i=0; i<qp.total(); i++) {
    if (bestLabels(i)==0) qp(i)=0;
    else qp(i)=255;
  }
  imp(qp,"kmeanbean1.pgm");
}
```

Classificando em 7 classes:

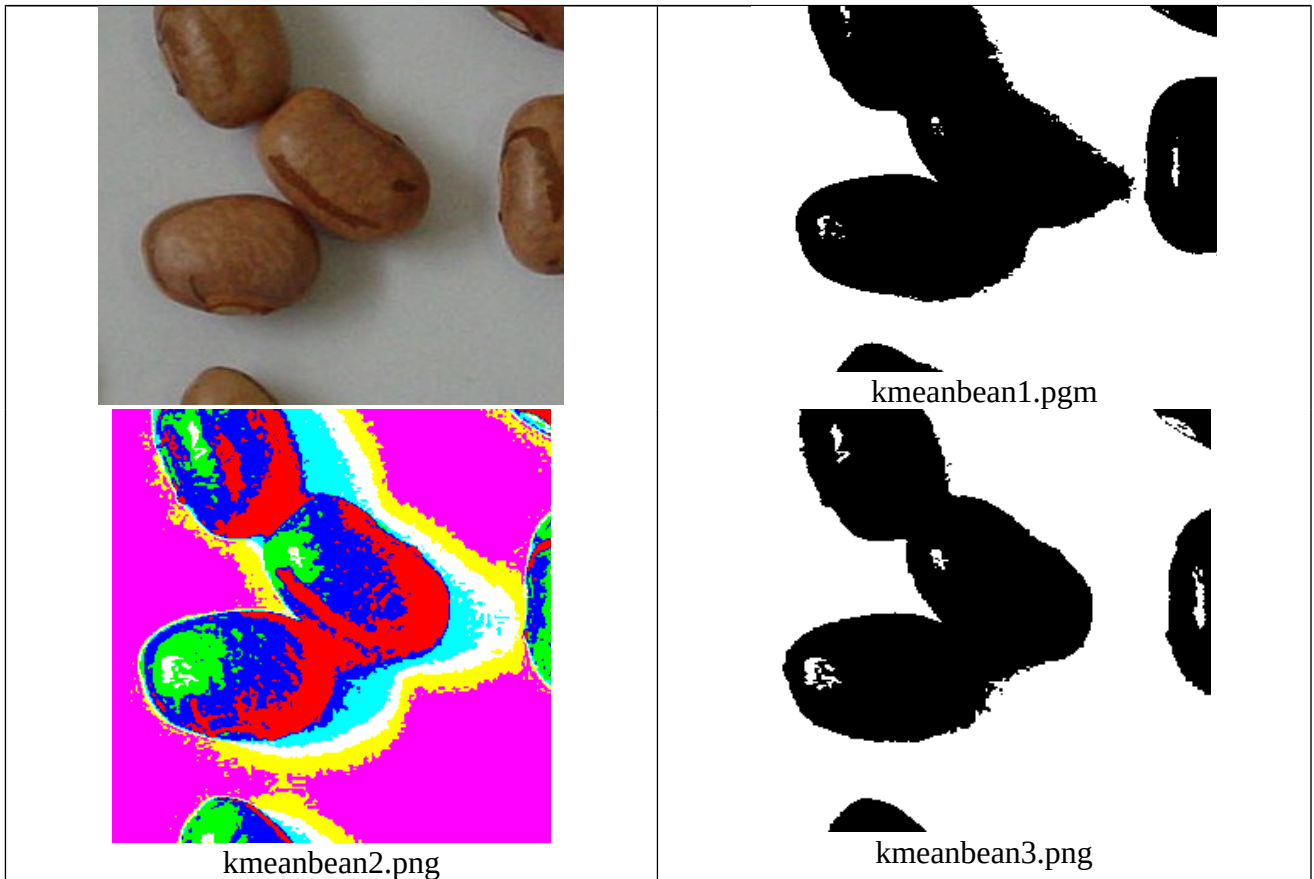
```
//kmeanbean2.cpp - 2015
#include <cekeikon.h>

int main()
{ Mat_<COR> qx; le(qx,"f1.jpg");
  Mat_<COR> qp(qx.rows,qx.cols);

  Mat_<FLT> data(qx.total(),3);
  for (unsigned i=0; i<qx.total(); i++) {
    data(i,0)=qx(i)[0];
    data(i,1)=qx(i)[1];
    data(i,2)=qx(i)[2];
  }

  int K=7;
  Mat_<int> bestLabels(qx.total(),1);
  Mat_<FLT> centers(K,3);
  TermCriteria criteria(TermCriteria::COUNT | TermCriteria::EPS, 20, 5.0);
  kmeans(data,K,bestLabels,criteria,20,KMEANS_RANDOM_CENTERS,centers);

  for (unsigned i=0; i<qp.total(); i++) {
    if (bestLabels(i)==0) qp(i)=COR(0,0,255);
    else if (bestLabels(i)==1) qp(i)=COR(0,255,0);
    else if (bestLabels(i)==2) qp(i)=COR(255,0,0);
    else if (bestLabels(i)==3) qp(i)=COR(0,255,255);
    else if (bestLabels(i)==4) qp(i)=COR(255,0,255);
    else if (bestLabels(i)==5) qp(i)=COR(255,255,0);
    else qp(i)=COR(255,255,255);
  }
  imp(qp,"kmeanbean2.png");
}
```



Kmeanbean1 com 2 classes não consegue segmentar feijões corretamente (sombra é classificada como feijão).

Kmeanbean2 com 7 classes consegue segmentar feijões melhor. No kmeanbean2.png, poderia mapear vermelho, verde e azul em feijão e as outras cores no fundo.

Fazendo esse mapeamento, obtém kmeanbean3.png. Reflexos não são resolvidos. Porém, as sombras são classificadas corretamente como fundo.

```

//kmeanbean3.cpp - pos2016
#include <cekeikon.h>

int main()
{ Mat_<COR> qx; le(qx, "f1.jpg");
  Mat_<COR> qp(qx.rows, qx.cols);

  Mat_<FLT> data(qx.total(), 3);
  for (unsigned i=0; i<qx.total(); i++) {
    data(i, 0)=qx(i)[0];
    data(i, 1)=qx(i)[1];
    data(i, 2)=qx(i)[2];
  }

  int K=7;
  Mat_<int> bestLabels(qx.total(), 1);
  Mat_<FLT> centers(K, 3);
  TermCriteria criteria(TermCriteria::COUNT | TermCriteria::EPS, 20, 5.0);
  kmeans(data, K, bestLabels, criteria, 20, KMEANS_RANDOM_CENTERS, centers);

  for (unsigned i=0; i<qp.total(); i++) {
    if (bestLabels(i)==0) qp(i)=COR(0,0,0); // red
    else if (bestLabels(i)==1) qp(i)=COR(0,0,0); // green
    else if (bestLabels(i)==2) qp(i)=COR(0,0,0); // blue
    else if (bestLabels(i)==3) qp(i)=COR(255,255,255);
    else if (bestLabels(i)==4) qp(i)=COR(255,255,255);
    else if (bestLabels(i)==5) qp(i)=COR(255,255,255);
    else qp(i)=COR(255,255,255);
  }
  imp(qp, "kmeanbean3.png");
}

```

[Escrever sobre PixelMap que faz segmentação usando aprendizagem supervisionada e não-supervisionada.]