

Teaser

Teaser

UP TO 1 '064x @ 1.32% SPEEDUP RELATIVE ERROR COMPARED TO K-MEANS++

UP TO 1 '064x @ 1.32% SPEEDUP RELATIVE ERROR COMPARED TO K-MEANS++

+ THEORETICAL GUARANTEES

Fast and Provably Good Seedings for k-Means

Teaser

k-Means clustering

k-Means clustering

Most popular clustering approach (nonconvex)

k-Means clustering

SEEDING

Find initial cluster centers

Most popular clustering approach (nonconvex)

k-Means clustering

SEEDING

Find initial cluster centers

Most popular clustering approach (nonconvex)



FINE-TUNING

Iteratively improve solution

k-Means clustering

SEEDING

Find initial cluster centers

Most popular clustering approach (nonconvex) MANY LOCAL MINIMA MAY EXIST



FINE-TUNING

Iteratively improve solution

k-Means clustering

SEEDING

Find initial cluster centers

Most popular clustering approach (nonconvex) MANY LOCAL MINIMA MAY EXIST



FINE-TUNING

Iteratively improve solution

ENSURES THAT LOCAL MINIMUM IS REACHED

k-Means clustering

SEEDING

Find initial cluster centers

DETERMINES WHICH LOCAL MINIMUM IS REACHED

Most popular clustering approach (nonconvex) MANY LOCAL MINIMA MAY EXIST



FINE-TUNING

Iteratively improve solution

ENSURES THAT LOCAL MINIMUM IS REACHED

k-Means clustering

SEEDING

Find initial cluster centers

DETERMINES WHICH LOCAL MINIMUM IS REACHED

SEEDING IS CRITICAL!

Most popular clustering approach (nonconvex) MANY LOCAL MINIMA MAY EXIST



FINE-TUNING

Iteratively improve solution

ENSURES THAT LOCAL MINIMUM IS REACHED

k-Means algorithms

SEEDING Find initial cluster centers



SEEDING Find initial cluster centers

k-Means++ seeding

k-Means algorithms



SEEDING Find initial cluster centers

k-Means++ seeding

k-Means algorithms



FINE-TUNING

Iteratively improve solution

Lloyd's algorithm

SEEDING Find initial cluster centers

k-Means++ seeding SLOW

k-Means algorithms

FINE-TUNING Iteratively improve solution

Lloyd's algorithm SLOW

SEEDING Find initial cluster centers

k-Means++ seeding **SLOW**

k-Means algorithms



Lloyd's algorithm SLOW Mini-batch k-Means FAST

SEEDING Find initial cluster centers

k-Means++ seeding **SLOW**

??

k-Means algorithms



Lloyd's algorithm SLOW

Mini-batch k-Means FAST

Random seeding



Fast and Provably Good Seedings for k-Means



Random seeding

Sample data points uniformly at random as cluster centers

Fast and Provably Good Seedings for k-Means



Random seeding

Sample data points uniformly at random as cluster centers



Fast and Provably Good Seedings for k-Means







k-Means++ seeding [Arthur et al., 2007]

Sample first center uniformly at random





Sample first center 4 uniformly at random

sample point x with

$$p(x) \propto \mathrm{d}(x, C)^2$$

D²-SAMPLING





Sample first center 4 uniformly at random

sample point x with

$$p(x) \propto \mathrm{d}(x, C)^2$$

D²-SAMPLING





Sample first center 4 uniformly at random

sample point x with

$$p(x) \propto \mathrm{d}(x, C)^2$$

D²-SAMPLING





Sample first center 4 uniformly at random

sample point x with

$$p(x) \propto \mathrm{d}(x, C)^2$$

D²-SAMPLING





Sample first center 4 uniformly at random

sample point x with

$$p(x) \propto \mathrm{d}(x, C)^2$$

D²-SAMPLING



k-Means++ seeding [Arthur et al., 2007]

Sample first center uniformly at random

sample point x with

$$p(x) \propto \mathrm{d}(x, C)^2$$

D²-SAMPLING













Single round of D²-sampling



Single round of D²-sampling

$\begin{array}{l} & \fbox{ Sample each point } \\ & \text{with probability } \\ & p(x) = \frac{\mathrm{d}(x,C)^2}{\sum_{x' \in \mathcal{X}} \mathrm{d}(x',C)^2} \end{array}$



Single round of D²-sampling

$\begin{array}{l} & \fbox{ Sample each point } \\ & \text{with probability } \\ & p(x) = \frac{\mathrm{d}(x,C)^2}{\sum_{x' \in \mathcal{X}} \mathrm{d}(x',C)^2} \\ \\ & \text{REQUIRES LINEAR PASS} \end{array}$



Single round of D²-sampling

Sample each point with probability $p(x) = \frac{d(x,C)^2}{\sum_{x' \in \mathcal{X}} d(x',C)^2}$ REQUIRES LINEAR PASS

? How can we efficiently approximate this step?


Markov chain Monte Carlo approach

Markov chain Monte Carlo approach

Goal: construct a Markov chain

Markov chain Monte Carlo approach

Goal: construct a Markov chain

Solutions where data points are states

Markov chain Monte Carlo approach

Goal: construct a Markov chain

where data points are states (\checkmark)







Markov chain Monte Carlo approach

Goal: construct a Markov chain

 \oslash where data points are states

 \oslash whose stationary distribution is

$$p(x) = \frac{\mathrm{d}(x, x)}{\sum_{x' \in \mathcal{X}} \mathrm{d}(x)}$$







Markov chain Monte Carlo approach

Goal: construct a Markov chain

 \oslash where data points are states

 \oslash whose stationary distribution is

$$p(x) = \frac{\mathrm{d}(x, x)}{\sum_{x' \in \mathcal{X}} \mathrm{d}(x)}$$



APPROXIMATION



Markov chain Monte Carlo approach

Goal: construct a Markov chain

where data points are states (\checkmark)

whose stationary distribution is (\checkmark)

 \oslash with a fast mixing time

Fast and Provably Good Seedings for k-Means



APPROXIMATION



Markov chain Monte Carlo approach

Goal: construct a Markov chain

where data points are states (\checkmark)

whose stationary distribution is (\checkmark) $p(x) = \frac{\mathrm{d}(x,C)^2}{\sum_{x' \in \mathcal{X}} \mathrm{d}(x',C)^2} \qquad \Longrightarrow \qquad$

 \oslash with a fast mixing time

Fast and Provably Good Seedings for k-Means





GUARANTEES APPROXIMATION

> **GUARANTEES EFFICIENCY**

Markov chain construction



Markov chain construction

Start with an arbitrary initial state X₀



Markov chain construction



Markov chain construction

Propose new candidate y_i according to "some" q(x)



Markov chain construction

Propose new candidate y_i according to "some" q(x)



Markov chain construction

Propose new candidate y_i according to "some" q(x)

Set $\mathbf{x_i} = \mathbf{y_i}$ with probability $\min\left(1, \frac{\mathrm{d}(y_j, C)^2}{\mathrm{d}(x_{j-1}, C)^2} \frac{q(x_{j-1})}{q(y_j)}\right),$ otherwise keep $\mathbf{x_i} = \mathbf{x_{i-1}}$



Markov chain construction

Propose new candidate y_i according to "some" q(x)

Set $\mathbf{x_i} = \mathbf{y_i}$ with probability $\min\left(1, \frac{\mathrm{d}(y_j, C)^2}{\mathrm{d}(x_{j-1}, C)^2} \frac{q(x_{j-1})}{q(y_j)}\right),$ otherwise keep $\mathbf{x_i} = \mathbf{x_{i-1}}$



Markov chain construction



Markov chain construction



Markov chain construction



Markov chain construction



Markov chain construction



Markov chain construction



Markov chain construction



Markov chain construction



Markov chain construction

Repeat m times to create Markov chain of length m

 \oslash Return x_m as cluster center



Markov chain construction

C Repeat **m** times to create Markov chain of length **m**

 \bigotimes Return x_m as cluster center

APPROXIMATE SINGLE STEP OF D² SAMPLING



Algorithm

Fast and Provably Good Seedings for k-Means

Algorithm

Sample first center uniformly

Fast and Provably Good Seedings for k-Means

Sample first center uniformly 4 Compute proposal distribution q(x)

Fast and Provably Good Seedings for k-Means

Algorithm

Sample first center uniformly 4 Compute proposal distribution q(x)Sequentially construct **k-1** independent Markov chains to obtain **k-1** cluster centers

Fast and Provably Good Seedings for k-Means

Algorithm

- Sample first center uniformly 4
- Compute proposal distribution q(x)
- Sequentially construct **k-1** independent Markov chains to obtain **k-1** cluster centers
- Approximation of k-Means++ seeding (\checkmark)

Algorithm

EFFICIENT IF M IS SMALL ENOUGH

K-MC² [Bachem et al., 2016]



Inform proposal: $q(x) = \frac{1}{\infty}$ \mathcal{N}

Fast and Provably Good Seedings for k-Means

K-MC² [Bachem et al., 2016]



Inform proposal: $q(x) = \frac{1}{m}$

▲ Misses small, far away clusters

Fast and Provably Good Seedings for k-Means

K-MC² [Bachem et al., 2016]



Inform proposal: $q(x) = \frac{1}{\infty}$



Requires assumptions on data or approach fails

Fast and Provably Good Seedings for k-Means

$K-MC^2$ [Bachem et al., 2016]



Assumption Free K-MC² [This paper]



Assumption Free K-MC² [This paper]

Nonuniform proposal: $q(x) = \frac{1}{2n} + \frac{1}{2} \frac{d(x, c_1)^2}{\sum_{x' \in \mathcal{X}} d(x', c_1)^2}$

Fast and Provably Good Seedings for k-Means


Nonuniform proposal: $q(x) = \frac{1}{2n} + \frac{1}{2} \frac{d(x, c_1)^2}{\sum_{x' \in \mathcal{X}} d(x', c_1)^2}$



Nonuniform proposal: $q(x) = \frac{1}{2n} + \frac{1}{2} \frac{d(x, c_1)^2}{\sum_{x' \in \mathcal{X}} d(x', c_1)^2}$

 \bigcirc Provably good w/o assumptions



Nonuniform proposal: $q(x) = \frac{1}{2n} + \frac{1}{2} \frac{d(x, c_1)^2}{\sum_{x' \in \mathcal{X}} d(x', c_1)^2}$

Provably good w/o assumptions





Nonuniform proposal: $q(x) = \frac{1}{2n} + \frac{1}{2} \frac{d(x, c_1)^2}{\sum_{x' \in \mathcal{X}} d(x', c_1)^2}$ COMPUTED ONCE IN SINGLE LINEAR PASS

 \oslash Provably good w/o assumptions





Main theoretical result

Main theoretical result

? Choose an error tolerance $\epsilon > 0$

Main theoretical result

Choose an error tolerance $\epsilon > 0$ (?)**Run algorithm with** $m = 1 + \frac{8}{\epsilon} \log \frac{4k}{\epsilon}$



Main theoretical result

Choose an error tolerance $\epsilon > 0$ (?)



Run algorithm with $m = 1 + \frac{8}{\epsilon} \log \frac{4k}{\epsilon}$ INDEPENDENT OF DATA SET SIZE

Main theoretical result

Choose an error tolerance $\epsilon > 0$ (?)



Expected solution quality:

 $\mathbb{E}[\phi_{\text{AFK-MC}^2}] \le 8(\log_2 k + 2)\phi_{\text{OPT}} + \epsilon \operatorname{Var}(\mathcal{X})$

Run algorithm with $m = 1 + \frac{8}{\epsilon} \log \frac{4k}{\epsilon}$ INDEPENDENT OF DATA SET SIZE

Main theoretical result

Choose an error tolerance $\epsilon > 0$ (?)



Expected solution quality:

$$\mathbb{E}[\phi_{\mathrm{AFK-MC}^2}] \le 8(\log_2 k)$$

SAME AS K-MEANS++

Run algorithm with $m = 1 + \frac{8}{\epsilon} \log \frac{4k}{\epsilon}$ INDEPENDENT OF DATA SET SIZE

 $(+2)\phi_{\mathsf{OPT}} + \epsilon \operatorname{Var}(\mathcal{X})$

Main theoretical result

Choose an error tolerance $\epsilon > 0$ (?)



Expected solution quality: APPROXIMATION $(+2)\phi_{\mathsf{OPT}} + \epsilon \operatorname{Var}(\mathcal{X})$ SAME AS K-MEANS++

$$\mathbb{E}[\phi_{\mathrm{AFK-MC}^2}] \le 8(\log_2 k)$$

Run algorithm with $m = 1 + \frac{8}{\epsilon} \log \frac{4k}{\epsilon}$ INDEPENDENT OF DATA SET SIZE

Main theoretical result

Choose an error tolerance $\epsilon > 0$ (?)

Run algorithm with $m = 1 + \frac{8}{\epsilon} \log \frac{4k}{\epsilon}$ INDEPENDENT OF DATA SET SIZE

Expected solution quality: APPROXIMATION $\mathbb{E}[\phi_{\mathrm{AFK-MC}^2}] \le 8(\log_2 k + 2)\phi_{\mathrm{OPT}} + \epsilon \operatorname{Var}(\mathcal{X})$ SAME AS K-MEANS++



 \bigcirc Total runtime: $\mathcal{O}\left(nd + \right)$

$$+\frac{1}{\epsilon}k^2d\log\frac{k}{\epsilon}$$

Main theoretical result

Choose an error tolerance $\epsilon > 0$ (?)

Run algorithm with $m = 1 + \frac{8}{\epsilon} \log \frac{4k}{\epsilon}$ INDEPENDENT OF DATA SET SIZE

Expected solution quality: APPROXIMATION $\mathbb{E}[\phi_{\mathrm{AFK-MC}^2}] \le 8(\log_2 k + 2)\phi_{\mathrm{OPT}} + \epsilon \operatorname{Var}(\mathcal{X})$ SAME AS K-MEANS++



Total runtime: $\mathcal{O}\left(nd + \frac{1}{\epsilon}k^2d\log\frac{k}{\epsilon}\right)$ FASTER $\mathcal{O}(nkd)$

Experimental results



Markov chain length

100



Markov chain length



Markov chain length

Experimental results



Markov chain length

100

Experimental results



Markov chain length

100

Experimental results



Olivier Bachem, Mario Lucic, S. Hamed Hassani, Andreas Krause

Experimental results







Markov chain length

Markov chain length

Markov chain length

Experimental results







Markov chain length

Markov chain length

Experimental results



M=100 IS SUFFICIENT IN PRACTICE

Olivier Bachem, Mario Lucic, S. Hamed Hassani, Andreas Krause

M=100 IS SUFFICIENT IN PRACTICE







CSN

Experimental results

Olivier Bachem, Mario Lucic, S. Hamed Hassani, Andreas Krause

Error vs time tradeoff

Error vs time tradeoff



	8
1	

10M 100M 1M

distance evaluations

Error vs time tradeoff



100M

distance evaluations

Error vs time tradeoff



distance evaluations

100M

Error vs time tradeoff



distance evaluations

100M

Error vs time tradeoff



distance evaluations

100M

Code

C This repository Search	Pull requests Issue	s Gist			♣ +• 11 •			
📮 obachem / kmc2		O	Unwatch - 1	★ Star	1 V Fork 0			
<> Code (!) Issues () [?] Pull reque	ests 0 🔲 Projects 0 💷 Wiki	- Pulse _111	Graphs	🗘 Settings				
Cython implementation of k-MC2 and AFK-MC2 seeding — Edit								
🕝 1 commit	ဖို 1 branch	\bigcirc 0 releases		보 1 cor	ntributor			
Branch: master - New pull request		Create new file	Upload files	Find file	lone or download -			
Tr obachem Initial release Latest commit 6cf35aa 23 days ago								
Jitignore	Initial release				23 days ago			
README.md	Initial release				23 days ago			
E kmc2.c	Initial release				23 days ago			
E kmc2.pyx	Initial release				23 days ago			
Setup.py	Initial release				23 days ago			
E test.py	Initial release				23 days ago			
E README.md								

Fast and Provably Good Seedings for k-Means using k-MC² and AFK-MC²

Introduction

The package provides a Cython implementation of the algorithms k-MC^2 and AFK-MC^2 described in the two papers

Approximate K-Means++ in Sublinear Time. Olivier Bachem, Mario Lucic, S. Hamed Hassani and Andreas Krause. In Proc. Conference on Artificial Intelligence (AAAI), 2016

Fast and Provably Good Seedings for k-Means. Olivier Bachem, Mario Lucic, S. Hamed Hassani and Andreas Krause. To appear in Neural Information Processing Systems (NIPS), 2016.

The implementation is compatible with Python 2.7.

Olivier Bachem, Mario Lucic, S. Hamed Hassani, Andreas Krause

Code

PYTHON IMPLEMENTATION Available at <u>olivierbachem.ch</u> or with

pip install kmc2

FEATURES

 \oslash drop-in replacement for k-means++

- \oslash easy to use (2 lines)
- Compatible with scikit-learn



Poster



Appendix

Comparison to k-Means [[Bachem et al., 2016]



