Lab Efficient Algorithms for Selected Problems: Design, Analysis and Implementation

Magdalena Aretz and Heiko Röglin

Department of Computer Science
University of Bonn, Germany

October 21th, 2013

universitätbonn

Motivation: Why should we look at the k-Means Algorithm aka Lloyd's Method?

Motivation: Why should we look at the `k-Means Algorithm` aka `Lloyd's Method`?

- It was selected as one of the "Top Ten Algorithms in Data Mining" [WKQ+08].

Motivation: Why should we look at the `k-Means Algorithm` aka `Lloyd's Method`?

- It was selected as one of the "Top Ten Algorithms in Data Mining" [WKQ+08].
- It is "by far the most popular clustering algorithm used in scientific and industrial applications" [Ber06] . . .

Motivation: Why should we look at the `k-Means Algorithm` aka `Lloyd's Method`?

- It was selected as one of the "Top Ten Algorithms in Data Mining" [WKQ+08].
- It is "by far the most popular clustering algorithm used in scientific and industrial applications" [Ber06] . . .
- . . . although "thousands of clustering algorithms" [Jai10] have been proposed during the last 50 years.

Main Questions

Quality of the final result

- Worst case?

Main Questions

Quality of the final result

- Worst case? Could be arbitrarily bad

Main Questions

Quality of the final result

- Worst case? Could be arbitrarily bad
- Average / Expected value?
- Real data?

Main Questions

Quality of the final result

- Worst case? Could be arbitrarily bad
- Average / Expected value?
- Real data?

Running time

- Worst case?

Main Questions

Quality of the final result

- Worst case? Could be arbitrarily bad
- Average / Expected value?
- Real data?

Running time

- Worst case? Can be exponential even for $d = 2$ [Vat09]

Main Questions

Quality of the final result

- Worst case? Could be arbitrarily bad
- Average / Expected value?
- Real data?

Running time

- Worst case? Can be exponential even for $d = 2$ [Vat09]
- Average / Expected value?
- Real data?

Goal of this Lab

Investigate one of the central open questions of Machine Learning

Why, when and in what respect does `k-Means` perform good in practice?

Goal of this Lab

Investigate one of the central open questions of Machine Learning

Why, when and in what respect does `k-Means` perform good in practice?

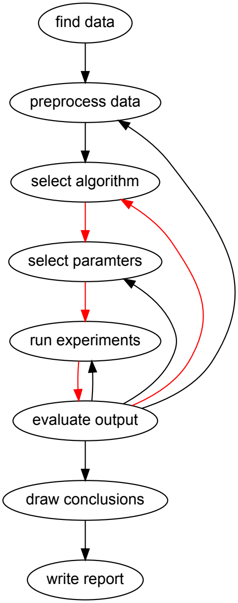
Learn how to work with many useful tools

Java, Eclipse, svn, R, JUnit, gnuplot, graphviz, LaTeX, TikZ, documentation, profiling, how to find and preprocess data, setting up and evaluating experiments, . . .

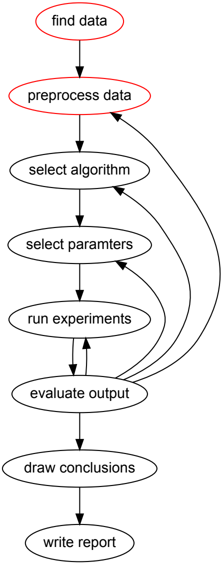
Behaviour of the `k-Means Method` and its variants heavily depends on...

- ... the selection of the data set
- ... general properties of the data
- ... the value of k
- ... the initialization of the centers
- ... the update rule
- ... the pivot rule
- ... the metric used for evaluation.

Workflow

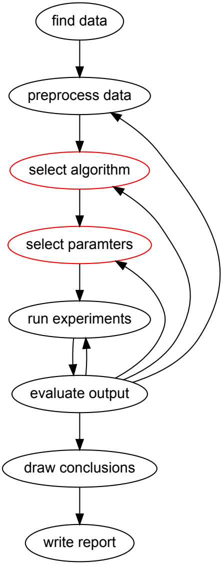

find data
preprocess data
select algorithm
select paramters
run experiments
evaluate output
draw conclusions
write report

Data



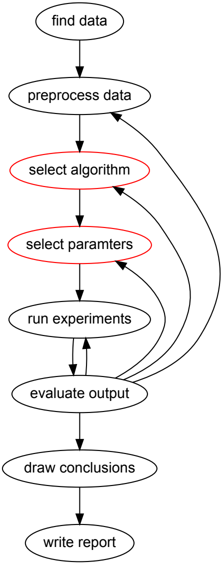
- Sources?
- Suitability?
- Format?
- Missing values?
- Categorical data?
- Dimension reduction?
- Scaling?
- Normalization?
- Sampling?

Algorithm and Parameters



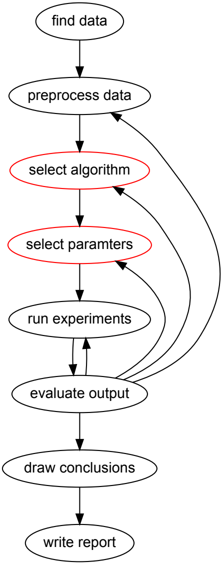
- Choice of k

Algorithm and Parameters



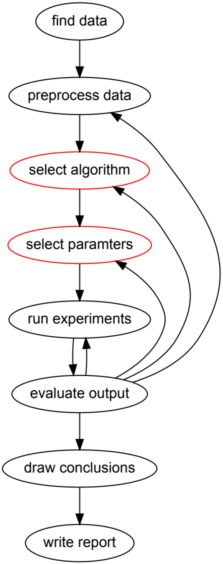
- Choice of k
- Initialization:
  - random subset
  - random assignment
  - random points
  - `k-Means++` [AV07]

Algorithm and Parameters



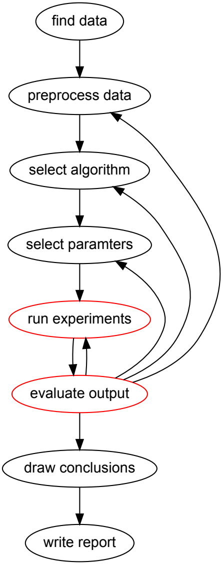
- Choice of k
- Initialization:
  - random subset
  - random assignment
  - random points
  - `k-Means++` [AV07]
- Update rule:
  - `Lloyd's Method`
  - `Swapping` [KMN+04]
  - `SinglePoint`, `Lazy` [HPS05],
  - `Hartigan's Method` [TV10]
  - . . . and many more (see [Jai10])

Algorithm and Parameters



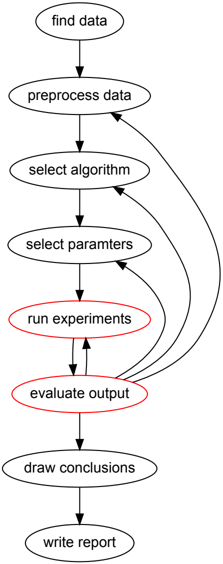
- Choice of *k*
- Initialization:
  - random subset
  - random assignment
  - random points
  - `k-Means++` [AV07]
- Update rule:
  - `Lloyd's Method`
  - `Swapping` [KMN$^+$04]
  - `SinglePoint`, `Lazy` [HPS05],
  - `Hartigan's Method` [TV10]
  - …and many more (see [Jai10])
- Pivot rule:
  - `random`
  - `next`
  - `max`

Metrics



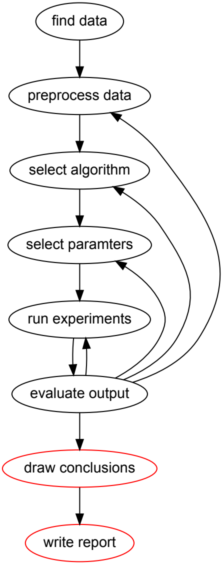
- Final error
  - Minimum?
  - Mean?
  - Number of incorrect assignments?

Metrics



- Final error
  - Minimum?
  - Mean?
  - Number of incorrect assignments?
- Running time
  - Number of steps?
  - Number of reassignments?
  - Running time in ms?

Report



- Find clear central question.
- Document all crucial design choices.
- Select most significant results.
- Find a coherent explanation for these.
- Provide comprehensive visualization.

Idea of this Lab

Investigate one of the central open questions of Machine Learning

Why, when and in what respect does `k-Means` perform good in practice?

Idea of this Lab

Investigate one of the central open questions of Machine Learning

Why, when and in what respect does `k-Means` perform good in practice?

Learn how to work with many useful tools

Java, Eclipse, svn, R, JUnit, gnuplot, graphviz, LaTeX, TikZ, documentation, profiling, how to find and preprocess data, setting up and evaluating experiments, . . .

Organization

- Work together in groups of 2 or 3 students.
- Every group should meet regularly (e.g. 2 or 3 times a week) to compare and combine all results.
- Meet your supervisor every second week to present your results and plan next steps.
- After every meeting you will receive two tasklists:
  - One including assignments to train your **B**asic programming skills and
  - one consisting of concrete features to **I**mplement.
- Each group will give a final presentation of its work at the end of the semester.

Organization

- Each group writes a report about its main results of 8 to 10 pages which has to be handed in 2 Weeks before the date of the presentation.
- Hand in your slides for the presentation 1 Week in advance.
- The presentation should take 30 minutes.
- Your grade will depend on the achievements you made during the semester, your thesis, the slides of your presentation and the quality of the talk per se.
- Please register for this course via BASIS until October 31th. If you do not know how to do this, have a look at `http://wobdoc.iai.uni-bonn.de/pos/msc-computer-science-exam-registration.pdf`.

Schedule

1. **Week 1 – 4**:
   Basics: Learn how to work with Java, Eclipse, svn, R, the *UCI Machine Learning Repository*.
   Implement: data collection, preprocessing, reading and writing files, `Lloyd's Method` and some variants of it.

Schedule

1. **Week 1 – 4**:
   Basics: Learn how to work with Java, Eclipse, svn, R, the *UCI Machine Learning Repository*.
   Implement: data collection, preprocessing, reading and writing files, `Lloyd's Method` and some variants of it.
2. **Week 5 – 8**:
   Basics: Learn how to implement JUnit tests, write documentation, profile your code, implement visualization.
   Implement: variants of `kMeans`, run and evaluate experiments, visualization.

Schedule

1. **Week 1 – 4**:
   Basics: Learn how to work with Java, Eclipse, svn, R, the *UCI Machine Learning Repository*.
   Implement: data collection, preprocessing, reading and writing files, `Lloyd's Method` and some variants of it.
2. **Week 5 – 8**:
   Basics: Learn how to implement JUnit tests, write documentation, profile your code, implement visualization.
   Implement: variants of `kMeans`, run and evaluate experiments, visualization.
3. **Week 9 – 12**:
   Decide on your own what open questions you want to explore further!

Schedule

1. **Week 1 – 4**:
   Basics: Learn how to work with Java, Eclipse, svn, R, the *UCI Machine Learning Repository*.
   Implement: data collection, preprocessing, reading and writing files, `Lloyd's Method` and some variants of it.
2. **Week 5 – 8**:
   Basics: Learn how to implement JUnit tests, write documentation, profile your code, implement visualization.
   Implement: variants of `kMeans`, run and evaluate experiments, visualization.
3. **Week 9 – 12**:
   Decide on your own what open questions you want to explore further!
4. **Week 13 – 16**:
   Write thesis, prepare presentation, give presentation.

Tasklist B.1

- **Java**:
  Go through Oracle's *Learning the Java Language* tutorial
  `http://docs.oracle.com/javase/tutorial/java/index.html`
- **Eclipse**:
  Lessons 1 – 8 of M. Dexters *Eclipse and Java for Total Beginners* tutorial
  `http://eclipsetutorial.sourceforge.net/totalbeginner.html`
  Lessons 1 – 6 of M. Dexters *Using the Eclipse Workbench* tutorial
  `http://eclipsetutorial.sourceforge.net/workbench.html`
- **svn**:
  Understand the basic concepts of svn, e.g. by reading through this:
  `http://www.eclipse.org/subversive/documentation/index.php`

Tasklist I.1

- Install Java, Eclipse and a plugin for Eclipse that supports working with svn (e.g. subclipse or subversive).
- Create a Java project and check it in to your svn.
- Look for appropriate data on the *UCI Machine Learning Repository*.
  `http://archive.ics.uci.edu/ml/datasets.html`
- Implement and test functions that read / write datasets. Think about dealing with missing or categorical values.
- Implement `Lloyd's Method`. Figure out what values you want to keep and in what format you want to save the output.

Any questions?

Any questions?

⇒ Contact Magdalena Aretz:
aretz@cs.uni-bonn.de

Thank you for your attention!

- David Arthur and Sergei Vassilvitskii, *k-means++: the advantages of careful seeding*, Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms (Nikhil Bansal, Kirk Pruhs, and Clifford Stein, eds.), SIAM, 2007, pp. 1027–1035.
- Pavel Berkhin, *A survey of clustering data mining techniques*, Grouping Multidimensional Data (Jacob Kogan, Charles K. Nicholas, and Marc Teboulle, eds.), Springer, 2006, pp. 25–71.
- Sariel Har-Peled and Bardia Sadri, *How fast is the k-means method?*, Algorithmica **41** (2005), no. 3, 185–202.
- Anil K. Jain, *Data clustering: 50 years beyond k-means*, Pattern Recognition Letters **31** (2010), no. 8, 651–666.
- Tapas Kanungo, David M. Mount, Nathan S. Netanyahu, Christine D. Piatko, Ruth Silverman, and Angela Y. Wu, *A*

local search approximation algorithm for k-means clustering, Comput. Geom. **28** (2004), no. 2-3, 89–112.

Matus Telgarsky and Andrea Vattani, *Hartigan's method: k-means clustering without voronoi*, Journal of Machine Learning Research - Proceedings Track **9** (2010), 820–827.

Andrea Vattani, *k-means requires exponentially many iterations even in the plane*, Symposium on Computational Geometry (John Hershberger and Efi Fogel, eds.), ACM, 2009, pp. 324–332.

Xindong Wu, Vipin Kumar, J. Ross Quinlan, Joydeep Ghosh, Qiang Yang, Hiroshi Motoda, Geoffrey J. McLachlan, Angus F. M. Ng, Bing Liu, Philip S. Yu, Zhi-Hua Zhou, Michael Steinbach, David J. Hand, and Dan Steinberg, *Top 10 algorithms in data mining*, Knowl. Inf. Syst. **14** (2008), no. 1, 1–37.