Apache Mahout

Bringing Machine Learning to Industrial Strength

Presented by: Isabel Drost
neofonie

Agenda

- What is machine learning all about?
- The mission of Mahout.
- Let's get to work:
  - Grouping data into topics.
  - Assigning data to pre defined categories.
  - Recommend items to users.
- Example applications!

Problem setting

- Huge amounts of data at our fingertips.

Web pages

Mail archives

Pictures tagged with topics

Search engine logs

Wiki collaboration data

Source control logs

Information on proteins

News articles

Corporate job postings

Social network graphs

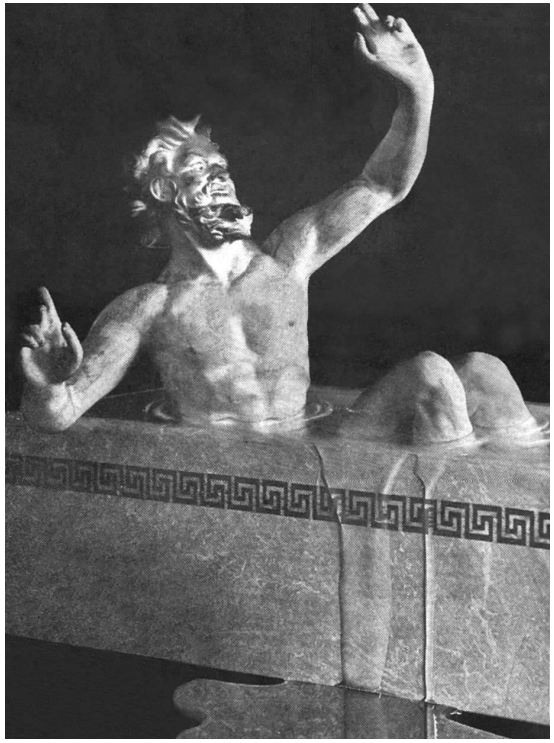
Traffic data

Tagged and rated videos

- Need means to deal with all the data.

Problem setting

- Nature generates data.



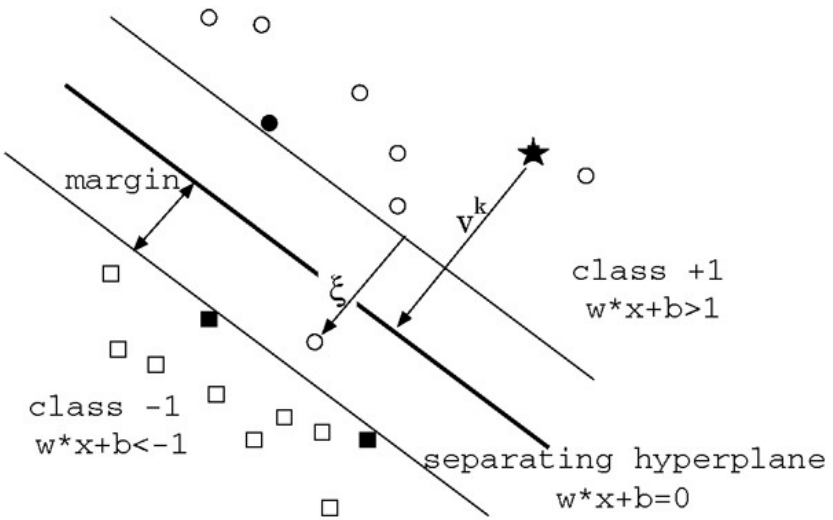
- Archimedes generates model.

$$\frac{Density\ of\ Object}{Density\ of\ Fluid} = .$$

$$\frac{Weight}{Weight - Apparent\ immersed\ weight}$$

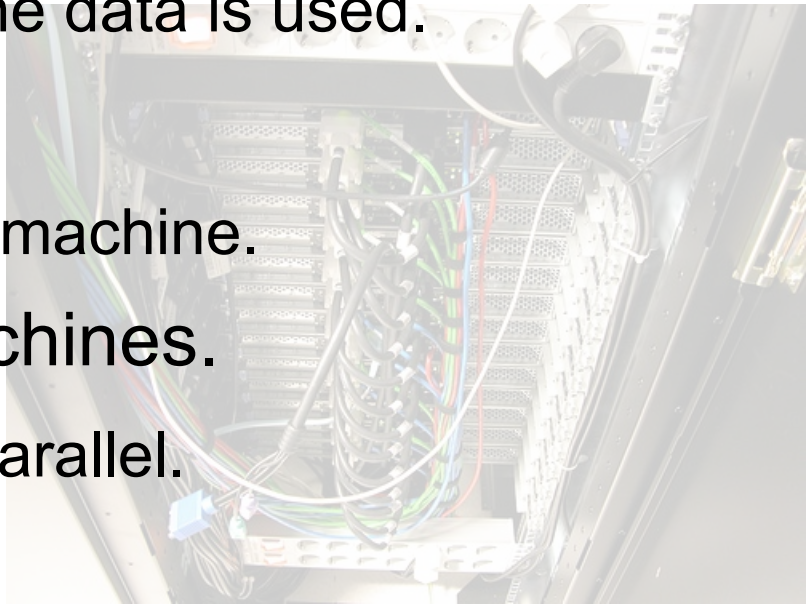
Problem setting

- Nature generates data.
- ML generates models.



Dataset sizes.

- Dataset usually are huge.
- Solution one: Use only a sample of the data.
  - Problem: Not all information in the data is used.
- Solution two: Use all the data.
  - Problem: Takes too long on one machine.
- Our Solution: Use multiple machines.
  - Handle all data, but process in parallel.



Our mission

- Build data (text) mining algorithms that are scalable.
- Context:

Hadoop – one way of parallelizing algorithms

Once upon a time

- How it all began:
  - Summer 2007: Crazy developers needed scalable ML.
  - Mailing list and wiki followed quickly.
- Rather large community even before project start.
- 25.01.2008: Project Mahout launched.
- Today: Mahout @ FrOSCon.

Who we are

Dawid Weiss
Carrot2

Karl Wettin
Lucene

Grant Ingersoll
Lucene PMC

Otis Gospodnetic
Lucene

Jeff Eastman
Welcome!

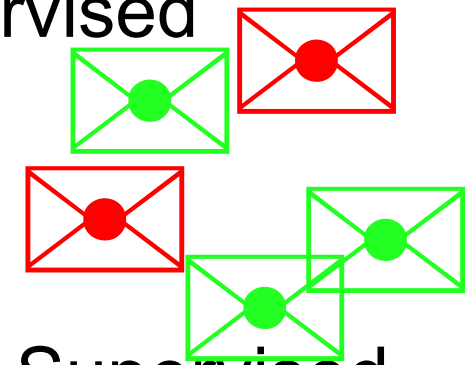
Ted Dunning
The Veoh guy

Erik Hatcher
Lucene

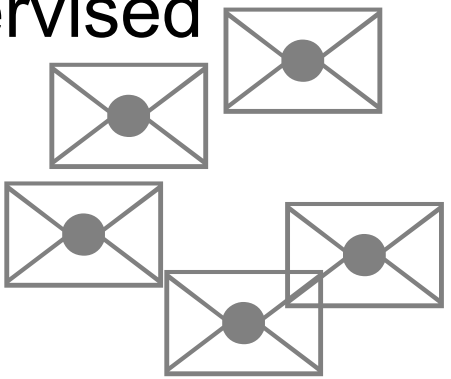
Isabel Drost
(that would be myself)

Types of learning tasks

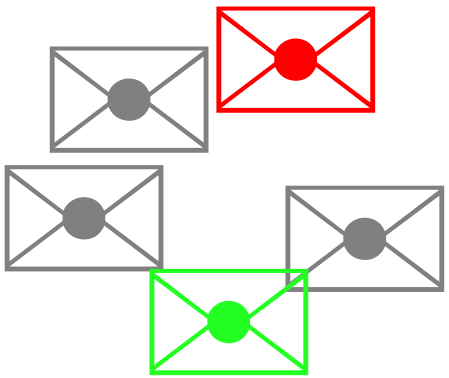
- Supervised



- Unsupervised

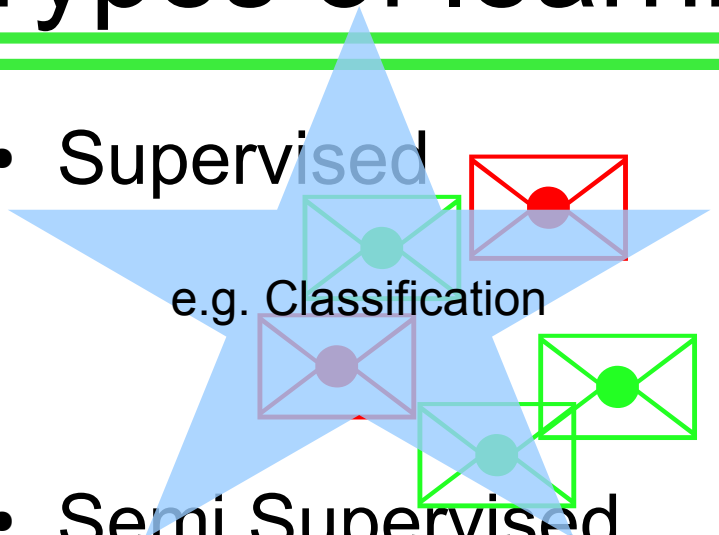


- Semi Supervised

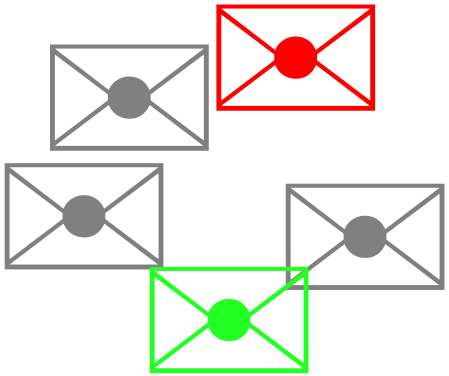


Types of learning tasks

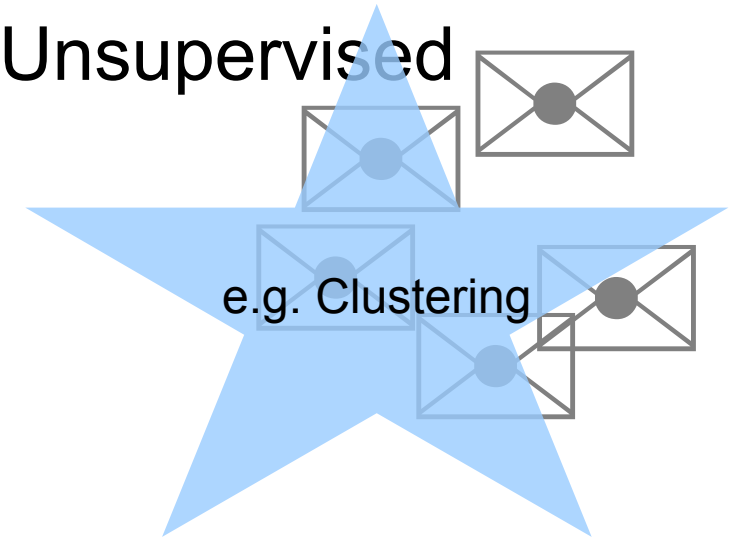
- Supervised



- Semi Supervised



- Unsupervised



Template for learning

- Get the data.
- Transform data to machine understandable form.
- Choose an appropriate algorithm based on problem.
- Search for best parameters.
- Combine features, found parameters, and algorithm.

Template for learning

- Get the data.
- Transform data to machine understandable form.
- Choose an appropriate algorithm based on problem.
- Search for best parameters.
- Combine features, found parameters, and algorithm.

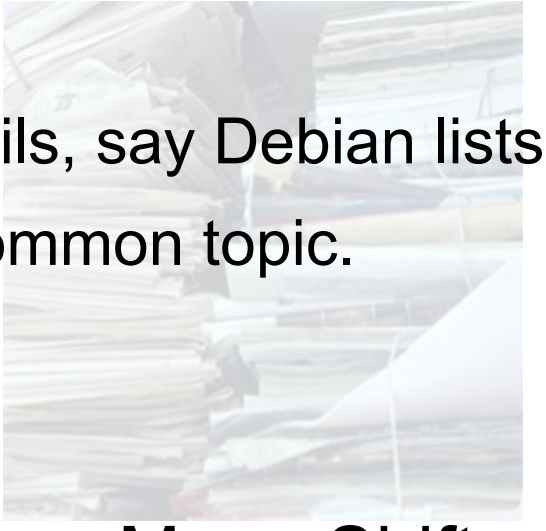
Features

No single best

Evaluate.

Clustering

- Example problem setting:
  - What you have: Huge amount of mails, say Debian lists.
  - What you want: Mails grouped by common topic.



- Algorithms so far: K-Means, Canopy, Mean Shift, Hierarchical.

Clustering

1) Gather the mails (e.g. from the archives).

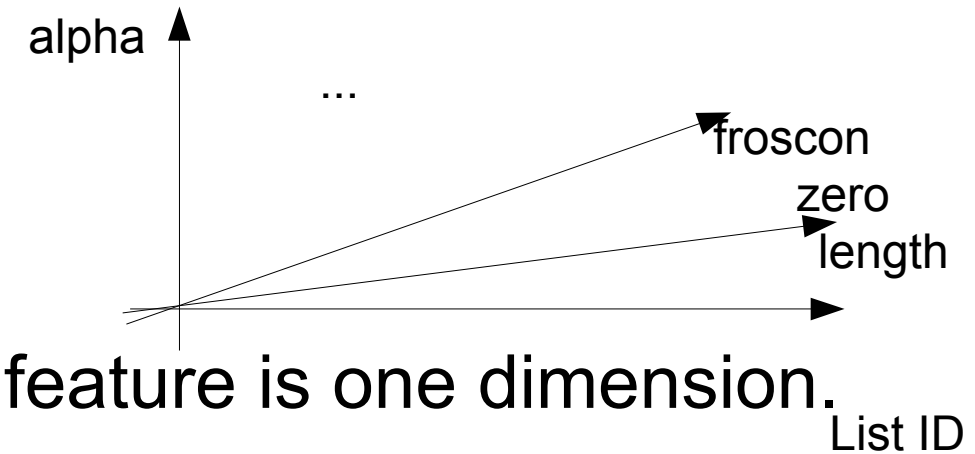
2) Find a way to generate vectors of mail properties.

- Mailinglist ID.
- Length of the mail.
- Parse the text and make features from word occurence.
- Parse the subject line and make features from words.
- ...

3) Apply some clustering algorithm to data vectors.

Clustering – step 2

- Each mail is a point in high dimensional space.



One of your mails:

$$\begin{pmatrix} 0 \\ 1 \\ 10 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{pmatrix}$$

- Each feature is one dimension.
- k-Means: Group points that are close to each other.
- Meaning of “close” depends on use case.

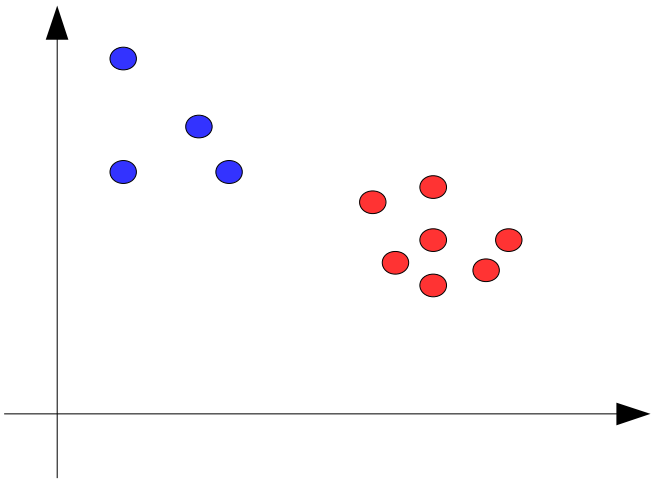
Clustering – step 2

- Common text feature generation schemes:
  - One binary dimension per term.
  - TF – Each dimension counts term occurrences.
  - TFIDF – Weighted number of terms.
- Specific features depend on your application.

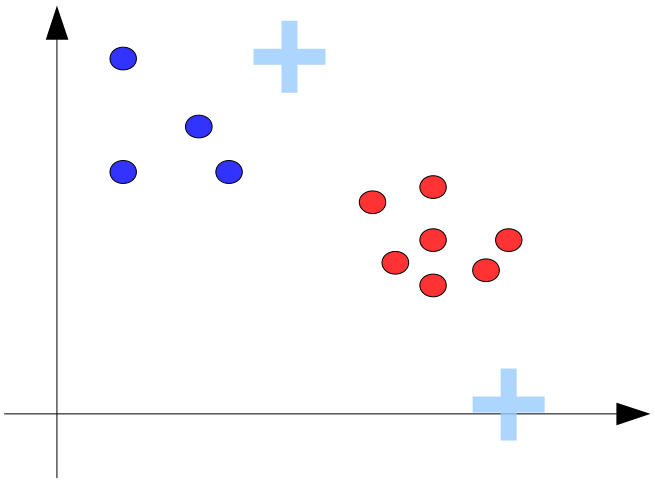
Clustering: k-Means – step 3

- Choose how many clusters/topics you seek (k).
- Choose arbitrary centers.
- Choose your definition of “closeness”.
- Repeat:
  - Assign each point to its closest the center.
  - Use the mean of the point groups as new center.
- Until assignment stable.

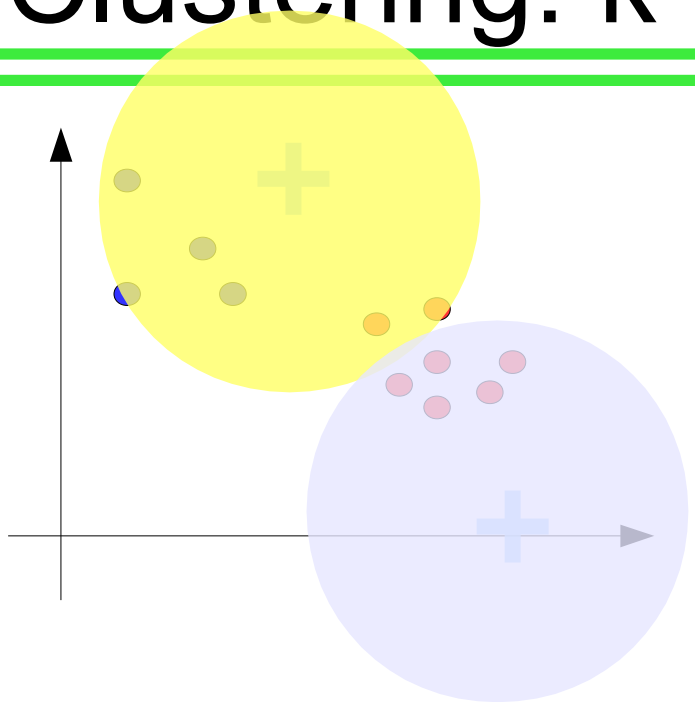
Clustering: k-Means – step 3



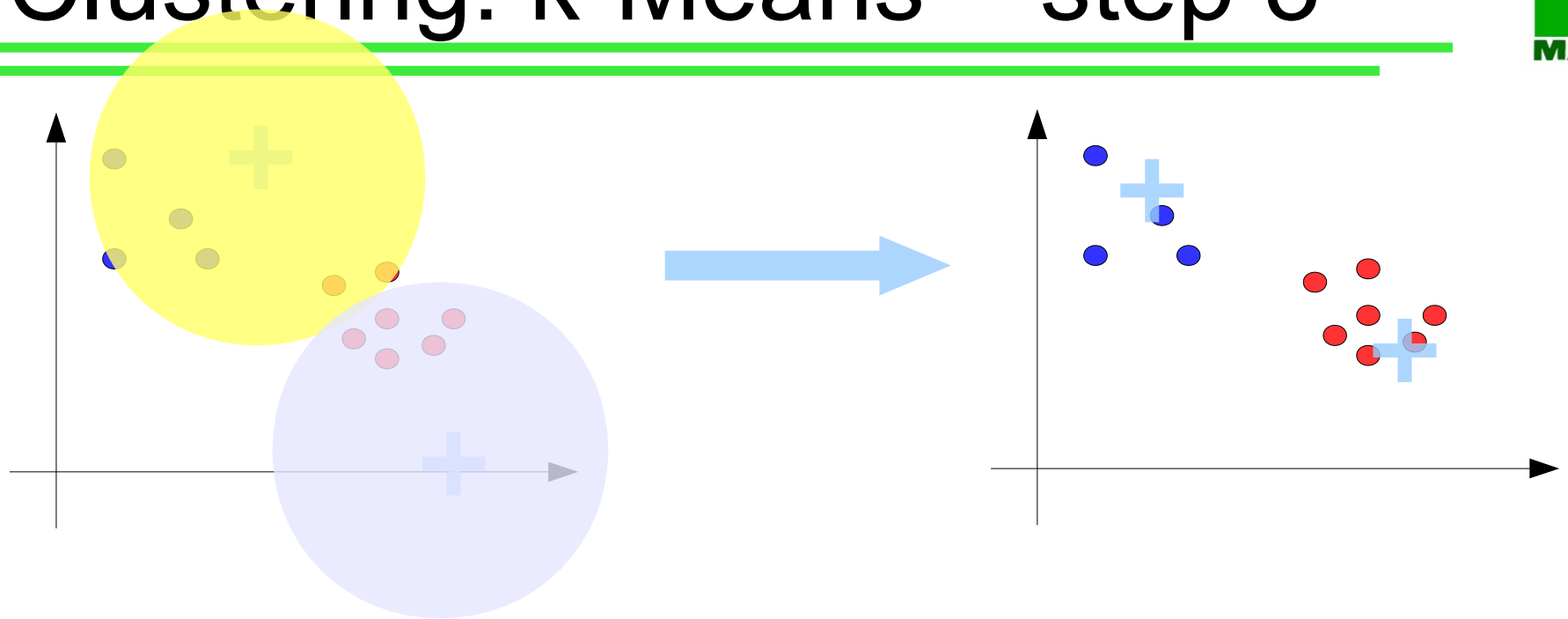
Clustering: k-Means – step 3



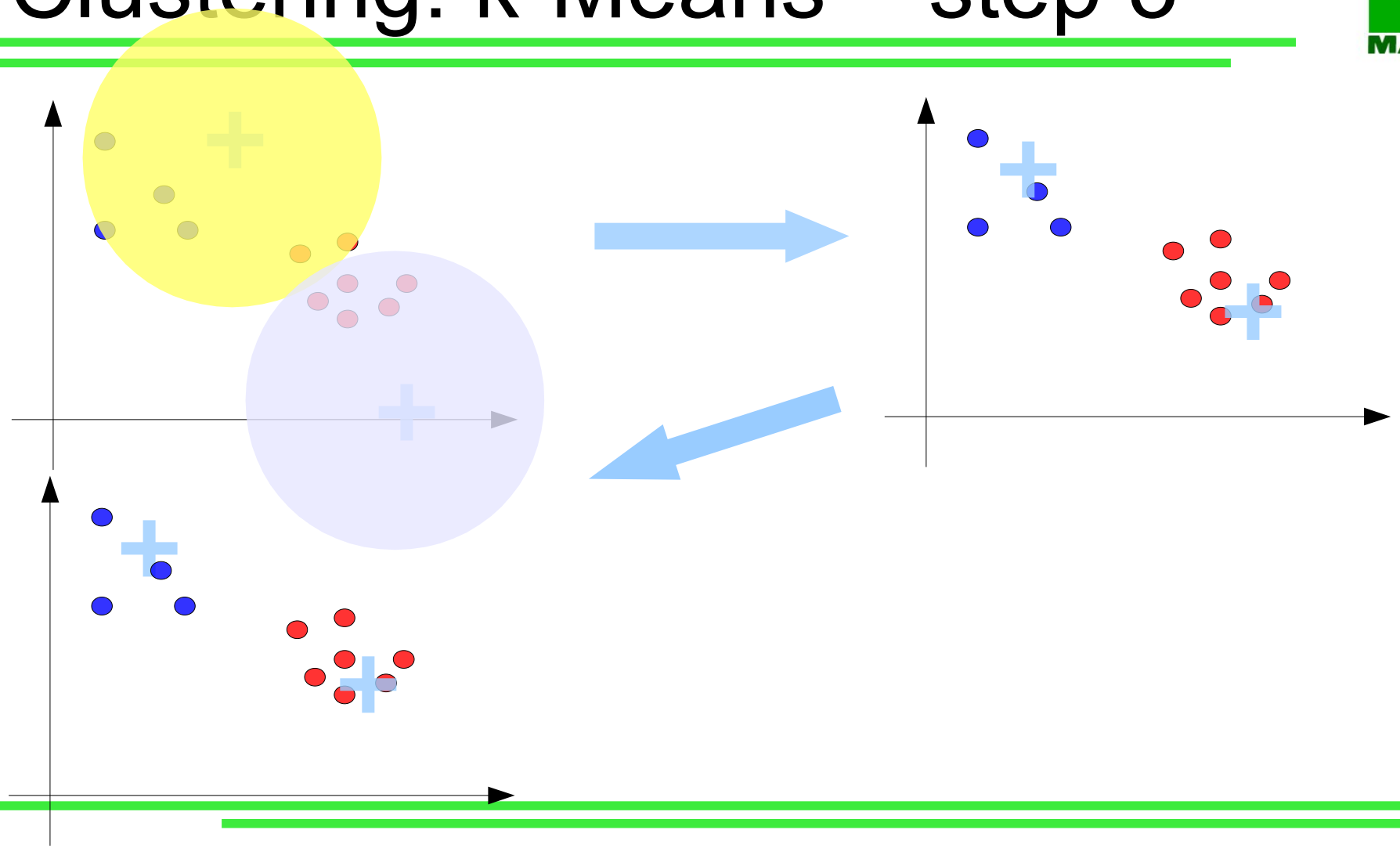
Clustering: k-Means – step 3



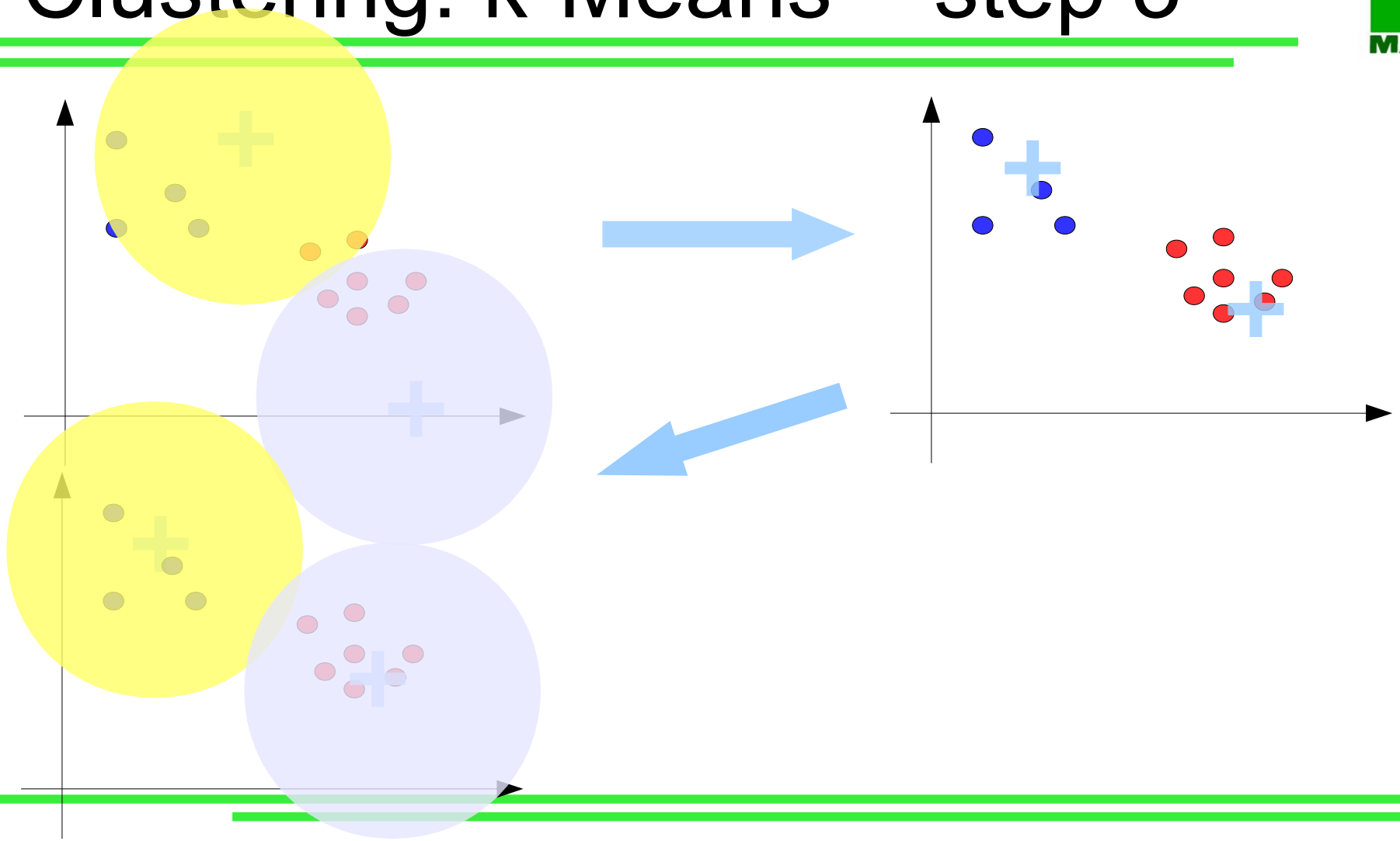
Clustering: k-Means – step 3



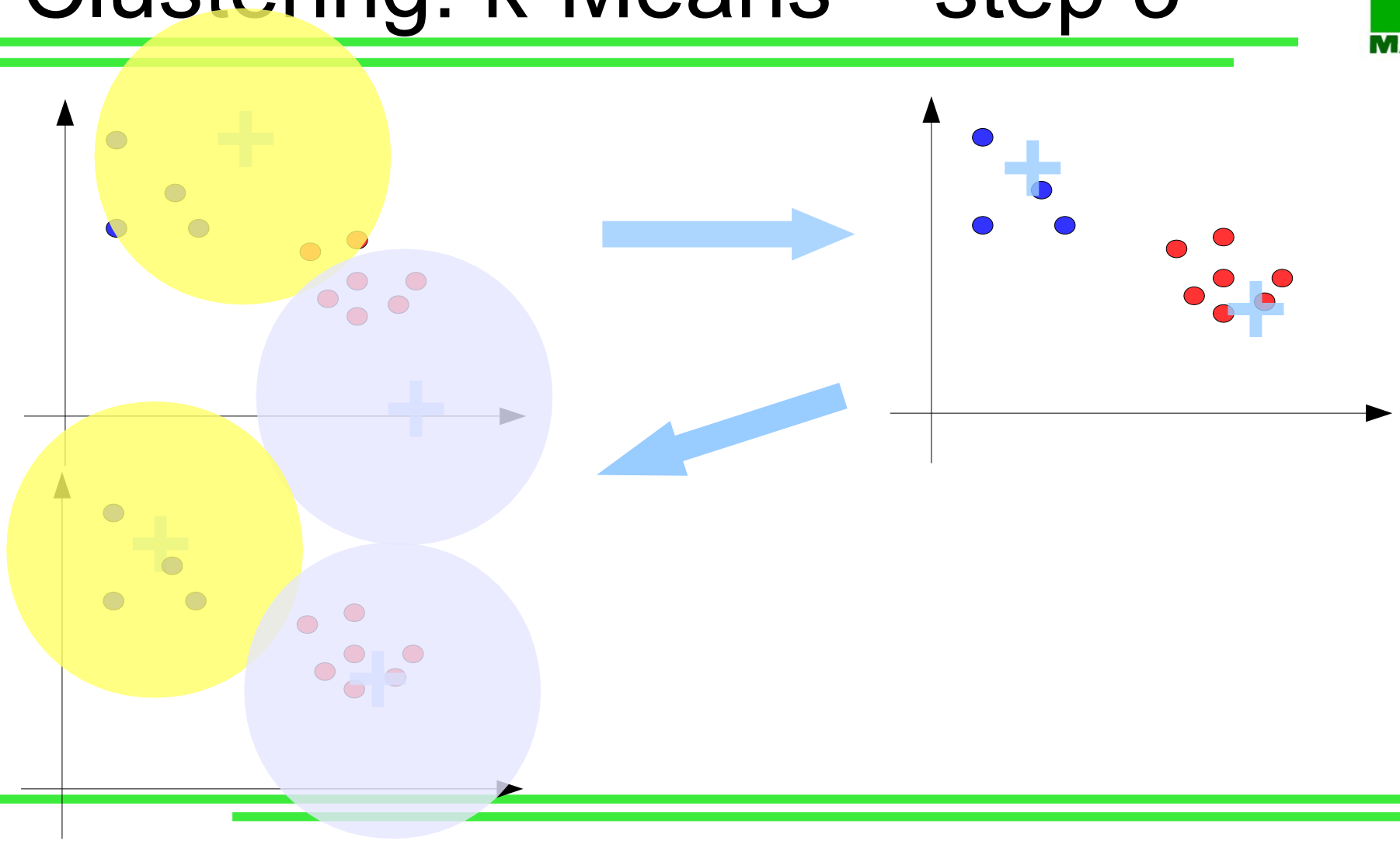
Clustering: k-Means – step 3



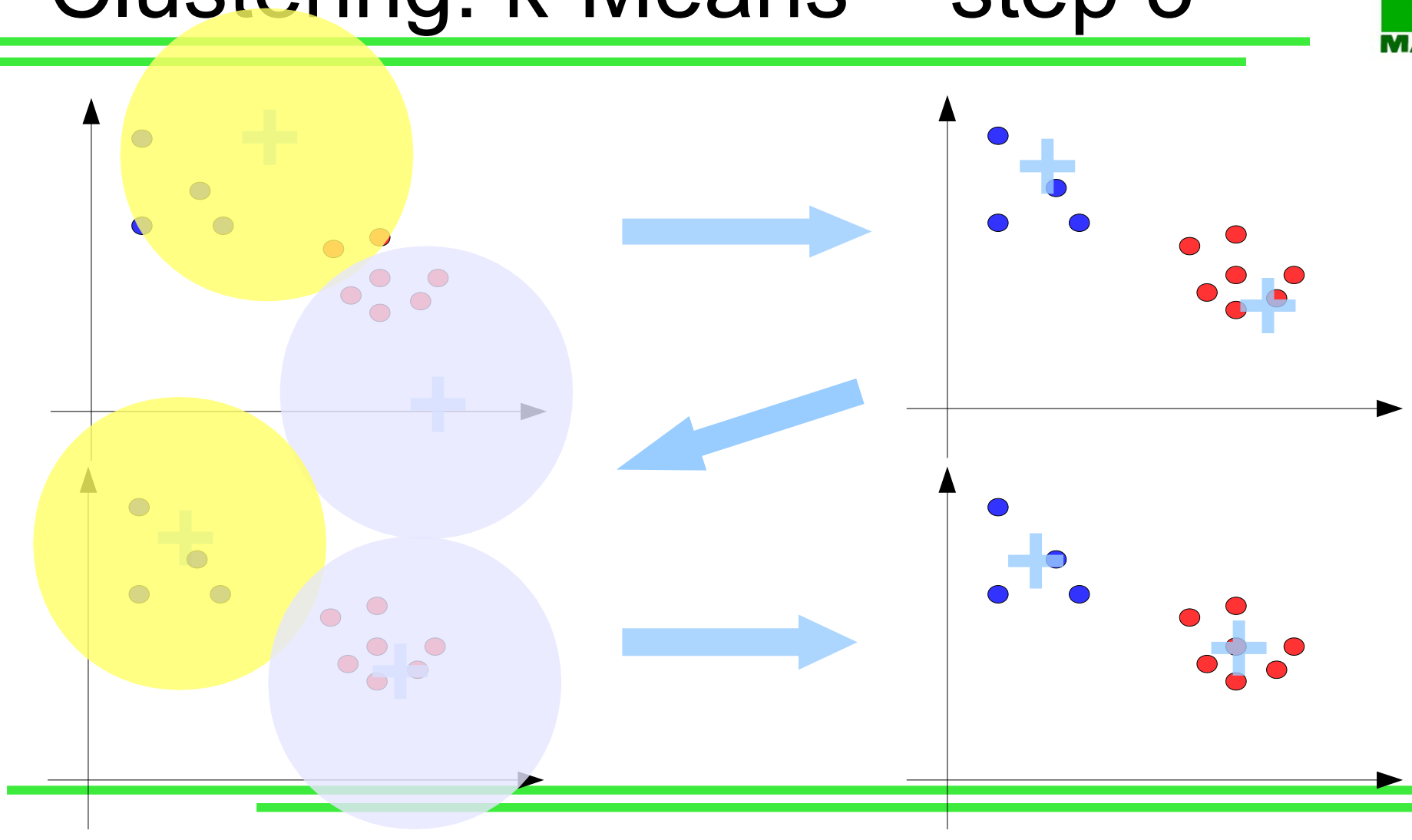
Clustering: k-Means – step 3



Clustering: k-Means – step 3

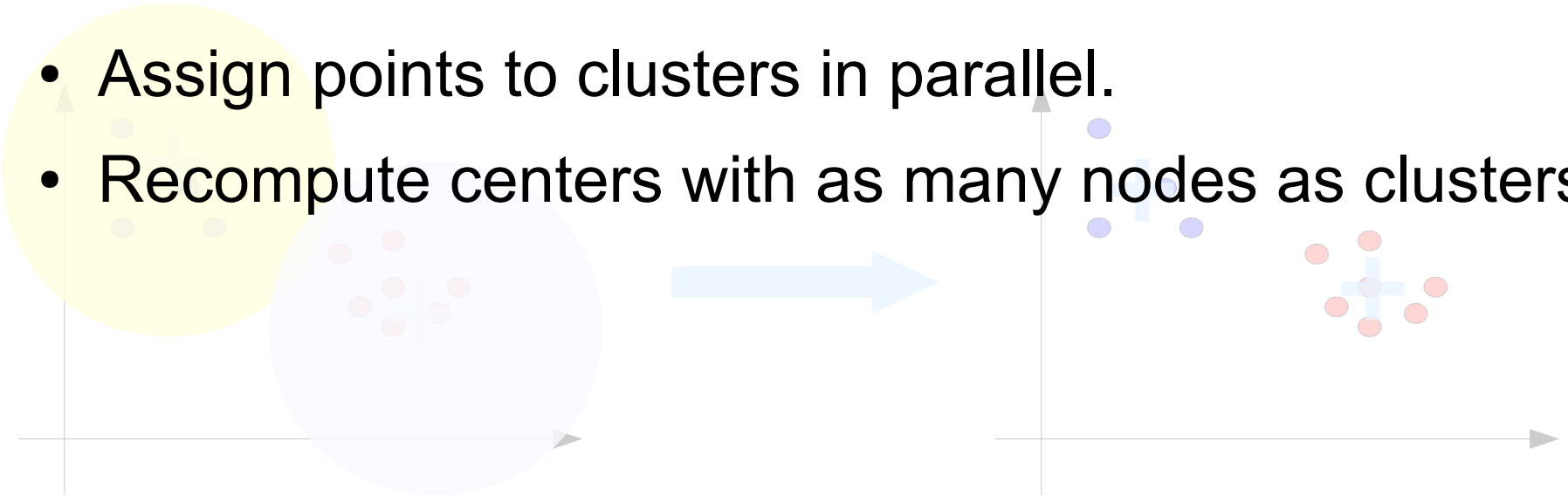


Clustering: k-Means – step 3



Clustering: k-Means – step 3

- Distributed version:
- Assign points to clusters in parallel.
- Recompute centers with as many nodes as clusters.



Points for optimization

- Representation of objects as feature vectors.
- Definition of “distance” of vectors.
- Definition of cluster center.
- Starting points.
- k-Means trivial but not necessarily best algorithm.
- Evaluation: Usually against gold standard.

Points for optimization

- Representation of objects as feature vectors.
- Definition of “distance” of vectors.
- Definition of cluster center.
- Starting points.
- k-Means trivial but not necessarily best algorithm.
- Evaluation: Usually against gold standard.

Features

Parameters.

No single best.

Classification

- Example problem setting:
  - You have: Set up a search engine.
  - You want: To index pages of your favorite topic.

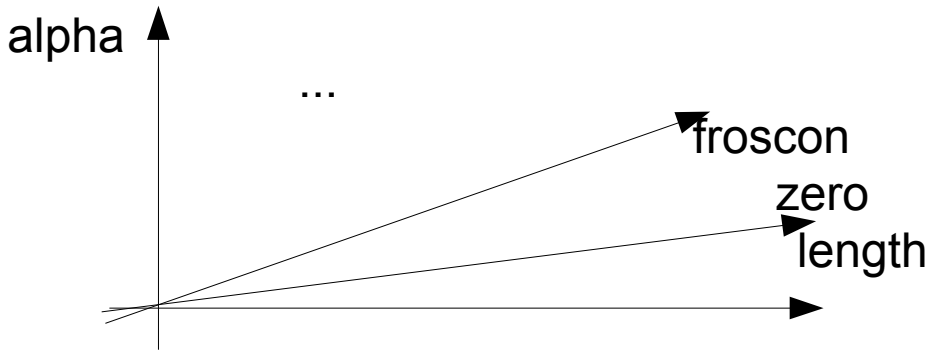
- Algorithms planned: Naïve Bayes, logistic regression, SVM.

Classification

1) Gather web pages.
2) Manually assign labels “off-topic” and “on-topic”.
3) Generate vectors of web page properties.
   - Parse the text and make features from word occurence.
   - Length of web page.
4) Train some classification algorithm to labeled data.
5) Apply the trained algorithm to new incoming data.

Classification

- Each web page is point in high dimensional space:



One of your mails:

$$\begin{pmatrix} . \\ 1 \\ 10 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{pmatrix}$$

- Classifier learns properties relevant for your topic.
- Which properties are selected depends on labels.

Classification: Naïve Bayes

- Sequential version:
  - For each label count terms per class.
  - Create model that represents label and feature counts.

$$p(C|F_1, \ldots, F_n) = \frac{1}{Z} p(C) \prod_{i=1}^{n} p(F_i|C)$$

- Distributed version:
  - Counting feature occurences for each label.

Points for optimization

- Representation of objects as feature vectors.
- Selection of labeled training data.
- Naïve Bayes easy, but not necessarily best:

SVM

Logistic Regression

Perceptron

Winnow

...

Rocchio

Points for optimization

- Representation of objects as feature vectors.
- Selection of labeled training data.
- Naïve Bayes easy, but not necessarily best:

Features

Parameters.

SVM

Logistic Regression

Perceptron

No single best algorithm.

Winnow

...

Rocchio

Recommendation Mining

- Example problem setting:
  - You have: A video sharing web service + terabytes of log files of user interactions.
  - You want: Recommend users videos they might like.
- Integrated: Taste

Recommendation Mining

1) Gather user interaction logs.
   - For each user store which videos the user watched.
   - For each video store additional information (year, actors...)
2) Create feature vectors from additional information.
3) Recommend videos based on:
   - Similarity to the videos the user watched.
   - Videos other users with same preferences watched.

Points for optimization

- Use video ratings as additional information.
- Find additional video information.
- Use implicit user feedback:
  - How long/often did the user watch the video?
  - Did the user recommend the video to others?
- Definition of user similarity is up to you.
- Definition of video similarity is up to you.

Points for optimization

- Use video ratings as additional information.
- Find additional video information.
- Use implicit user feedback:
  - How long/often did the user watch the video?
  - Did the user recommend the video to others?
- Definition of user similarity is up to you.
- Definition of video similarity is up to you.

Features

Parameters

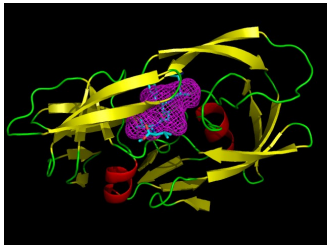
Initial Mahout goals

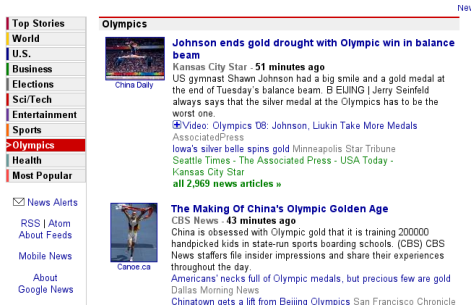
- Clustering (k-Means,Expectation Maximization, Mean Shift, Canopy, Hierarchical Clustering).
- Classification (Naïve Bayes, Logistic Regression, Support Vector Machines).
- Recommendation mining (Taste).
- Regression (Linear Regression).
- Dimensionality reduction (Principal Components Analysis, Independent Components Analysis, Gaussian Discriminative Analysis).

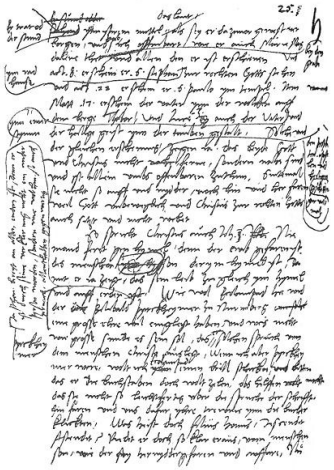
GSoC @ Mahout

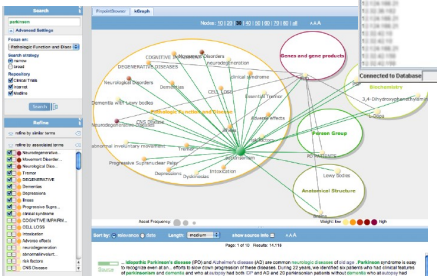
- Genetic Algorithms for Mahout
  - Several ways to parallelize.
  - Good for complex problems.
- Naïve Bayes and Complementary Naïve Bayes.
  - Classification algorithm.
  - Scale well to large amounts of data.
  - Straight forward to parallelize.
- Two advisors: Mahout + University.

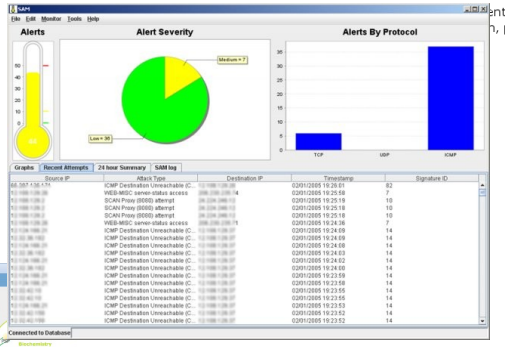
Example Applications











Learn from access patterns

Days of week
Hours
Who:
Countries
- Full list
- Regions
- Cities

Hosts
- Full list
- Last visit
- Unresolved IP Address

Authenticated users
- Full list
- Last visit

Robots/Spiders visitors
- Full list
- Last visit

Navigation:
Visits duration
File type
Viewed
- Full list
- Entry
- Exit

Operating Systems
- Versions
- Unknown

Browsers
- Versions
- Unknown

Screen sizes
Referrers:
Origin
- Referring search engines
- Referring sites

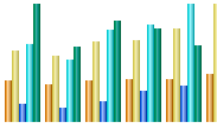
Search

Reported period: Aug 2008 OK

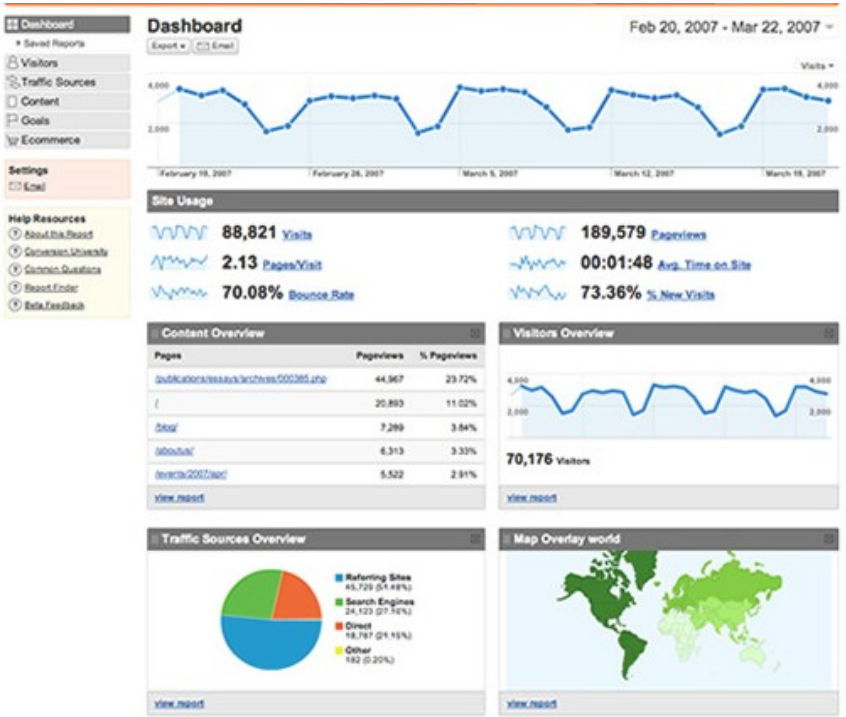
Summary		
Reported period	Month Aug 2008	
First visit	01 Aug 2008 - 00:17	
Last visit	20 Aug 2008 - 05:06	
	Unique visitors	Number of visits
Viewed traffic *	**584**	**913** (1.56 visits/visitor)
Not viewed traffic *		

* Not viewed traffic includes traffic generated by robots, worms, or replies with spe

Monthly history

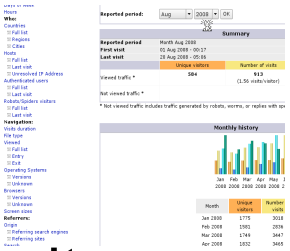


Month	Unique visitors	Number visits
Jan 2008	1775	3018
Feb 2008	1581	2836
Mar 2008	1749	3447
Apr 2008	1832	3465

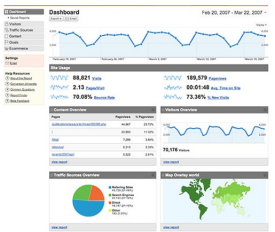


Learn from access patterns

- Input: Files containing server access logs.



- Some interesting tasks:
  - Identify user groups based interest in topics on site.
  - Adjust navigation to rules such as: “People who visit Debian and Mac Book pages also visit the refit pages”.
  - Show different sites to potential developers compared to visitors that are users seeking help.



- So far, a lot is done manually.

Clustering news stories

Ne\

- Top Stories
- World
- U.S.
- Business
- Elections
- Sci/Tech
- Entertainment
- Sports
- >Olympics
- Health
- Most Popular

News Alerts

RSS | Atom
About Feeds

Mobile News

About
Google News

Olympics


China Daily

Johnson ends gold drought with Olympic win in balance beam
Kansas City Star - **51 minutes ago**
US gymnast Shawn Johnson had a big smile and a gold medal at the end of Tuesday's balance beam. B EIJING | Jerry Seinfeld always says that the silver medal at the Olympics has to be the worst one.
Video: Olympics '08: Johnson, Liukin Take More Medals AssociatedPress
Iowa's silver belle spins gold Minneapolis Star Tribune
Seattle Times - The Associated Press - USA Today - Kansas City Star
all 2,969 news articles »

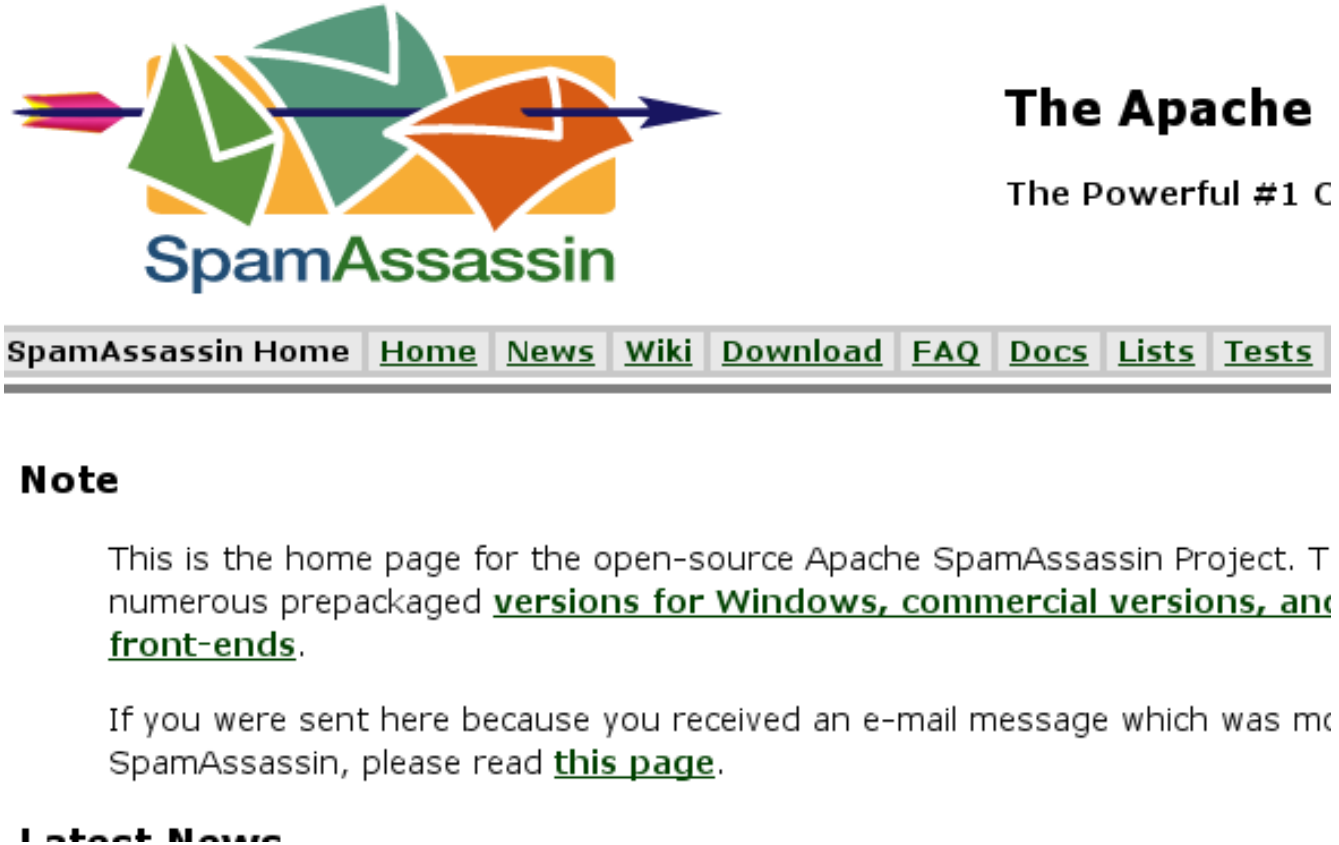

Canoe.ca

The Making Of China's Olympic Golden Age
CBS News - **43 minutes ago**
China is obsessed with Olympic gold that it is training 200000 handpicked kids in state-run sports boarding schools. (CBS) CBS News staffers file insider impressions and share their experiences throughout the day.
Americans' necks full of Olympic medals, but precious few are gold Dallas Morning News
Chinatown gets a lift from Beijing Olympics San Francisco Chronicle

Clustering news stories

Identify emerging hot topics from news

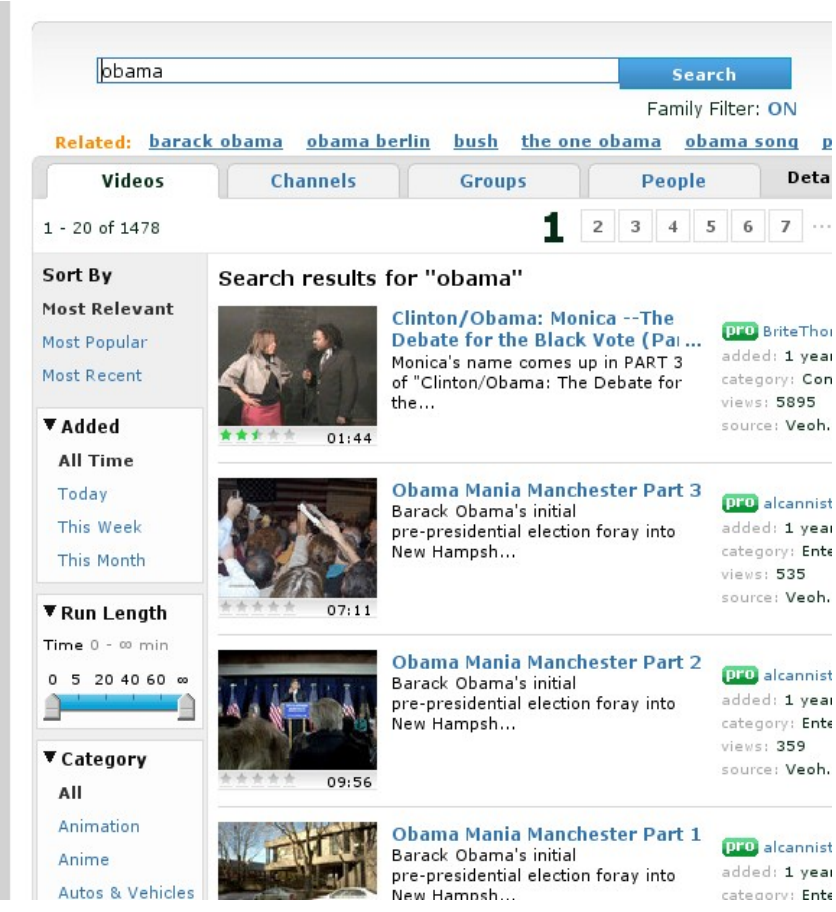
Mail filtering



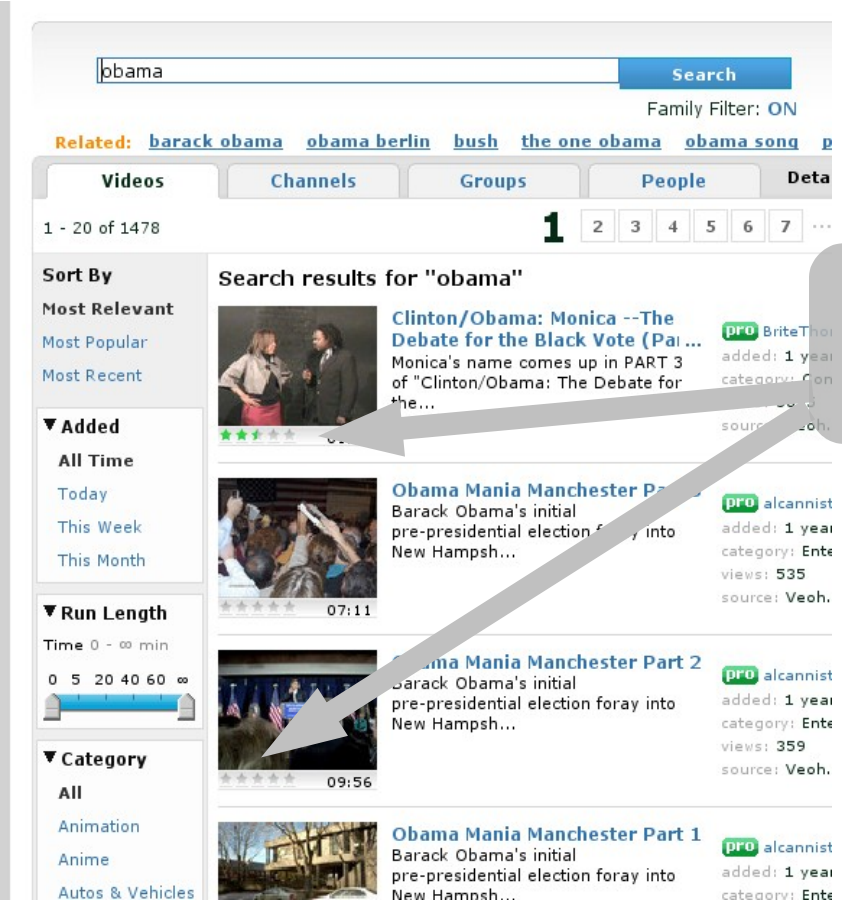
Mail filtering – Special problems

- Spam mails change over time: Adapt to filters.
  - Images that contain the text.
  - Text is modified.
- What is spam for me, might be ham for you.
- Usually users are not willing to provide labels.

Recommend videos

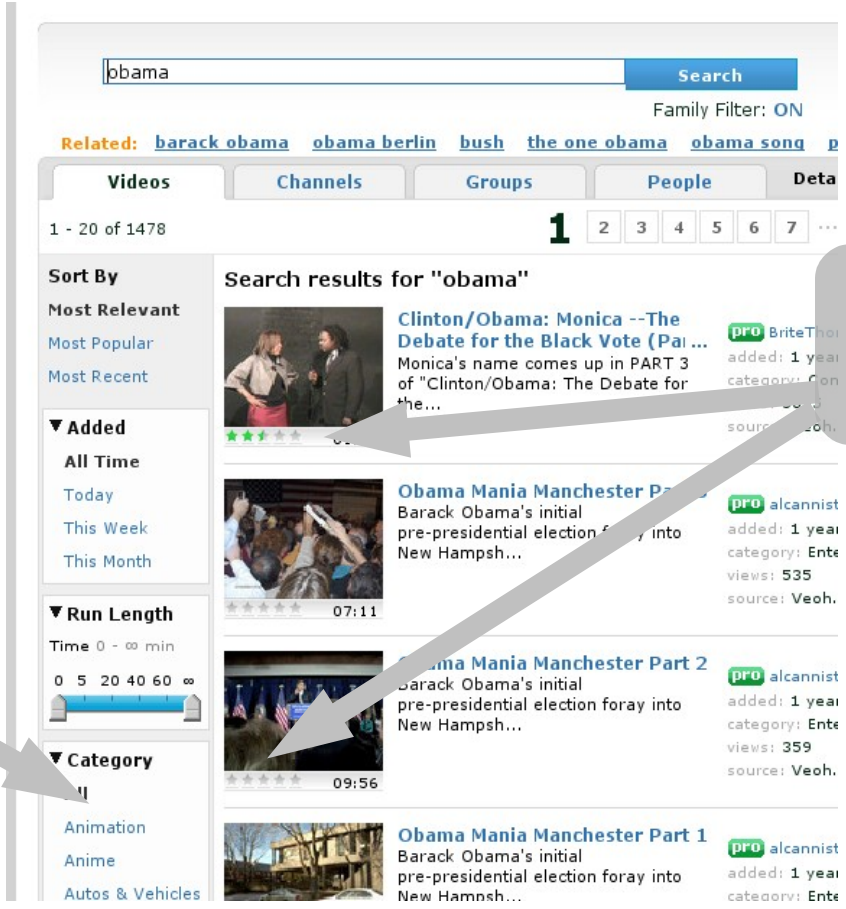


Recommend videos



Provide ratings, where unavailable, infer from usage.

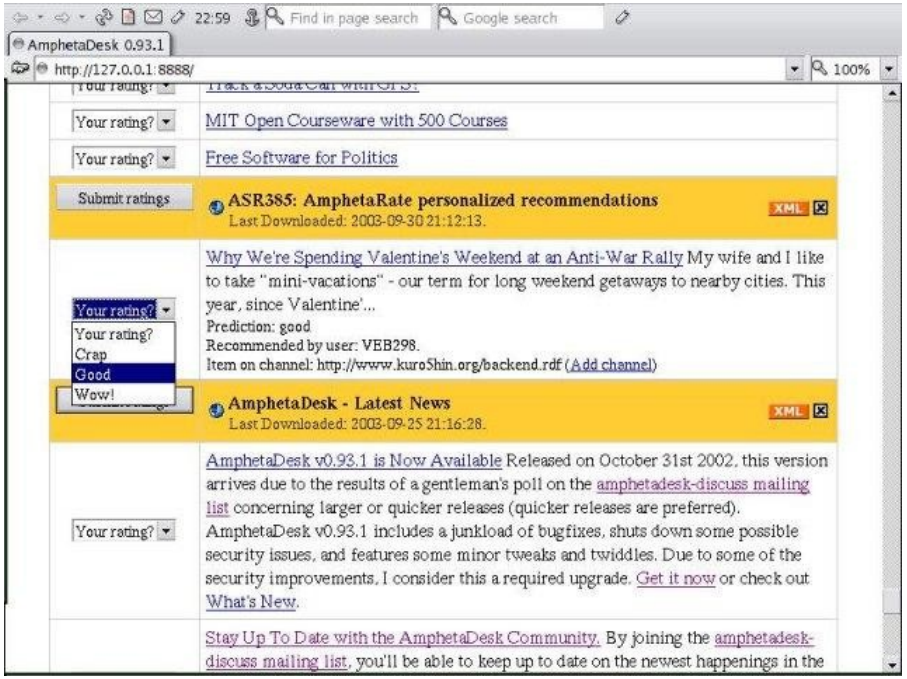
Recommend videos



Provide ratings, where unavailable, infer from usage.

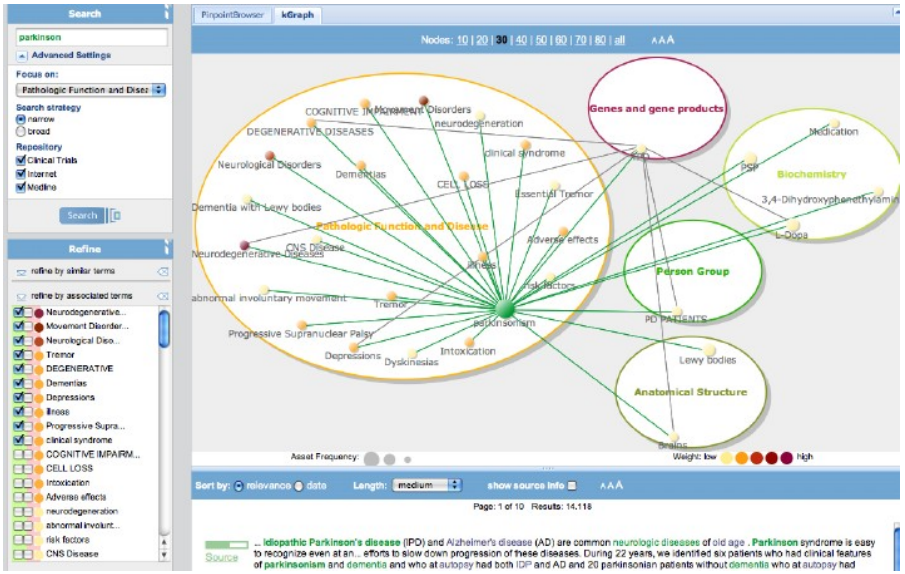
Categorize content.

Recommending RSS feeds



- Input:
  - List of RSS feeds.
  - Maybe ratings for feeds.
- Task:
  - Get me the latest and greatest that I like.

Recommend new papers

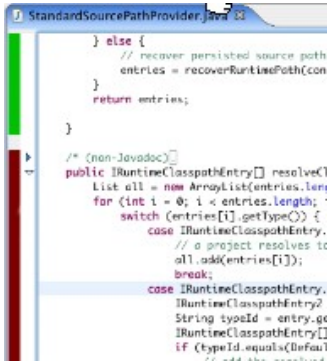


- Input:
  - A list of papers I like.
  - Maybe my own papers.
  - List of new publications.
- Task:
  - Give me all papers relevant for me.

Provide debugging help

Tasks | Error Log | Rose | Progress

Action	Symbol	File	Support	Confidence
ADD_IN	BirdieProcessor.java	\softevo-test\...\birdie\BirdieProcessor.java	4	0.3333
CHG	~usage()	\softevo-test\...\birdie\Birdie.java	4	0.3333
DEL_IN	URLUtils.java	\softevo-test\...\util\URLUtils.java	3	0.25
ADD_IN	URLUtils.java	\softevo-test\...\util\URLUtils.java	3	0.25
CHG	~processFile(String, BirdieProcessor)	\softevo-test\...\birdie\Birdie.java	3	0.25
CHG	~close()	\softevo-test\...\birdie\BirdieProcessor.java	3	0.25
ADD_IN	ArrayUtils.java	\softevo-test\...\util\ArrayUtils.java	2	0.1666



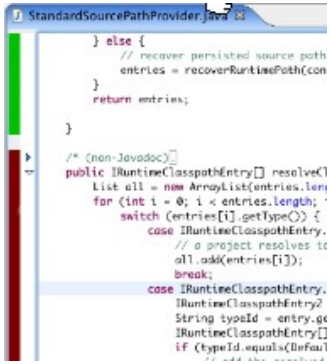
eROSE: Guiding Programmers in Eclipse:
http://www.st.cs.uni-sb.de/softevo/erose/

Provide debugging help

Mark pieces that are edited togehter.

Tasks | Error Log | Rose | Progress

Action	Symbol	File	Support	Confidence
ADD_IN	BirdieProcessor.java	\softevo-test\...\birdie\BirdieProcessor.java	4	0.3333
CHG	~usage()	\softevo-test\...\birdie\Birdie.java	4	0.3333
DEL_IN	URLUtils.java	\softevo-test\...\util\URLUtils.java	3	0.25
ADD_IN	URLUtils.java	\softevo-test\...\util\URLUtils.java	3	0.25
CHG	~processFile(String, BirdieProcessor)	\softevo-test\...\birdie\Birdie.java	3	0.25
CHG	~close()	\softevo-test\...\birdie\BirdieProcessor.java	3	0.25
ADD_IN	ArrayUtils.java	\softevo-test\...\util\ArrayUtils.java	2	0.1666



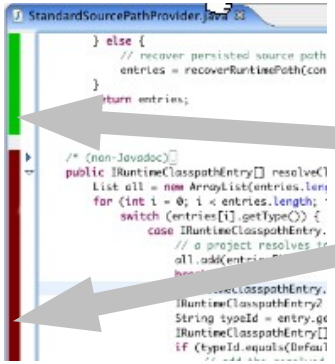
eROSE: Guiding Programmers in Eclipse:
http://www.st.cs.uni-sb.de/softevo/erose/

Provide debugging help

Mark pieces that are edited togehter.

Tasks | Error Log | Rose | Progress

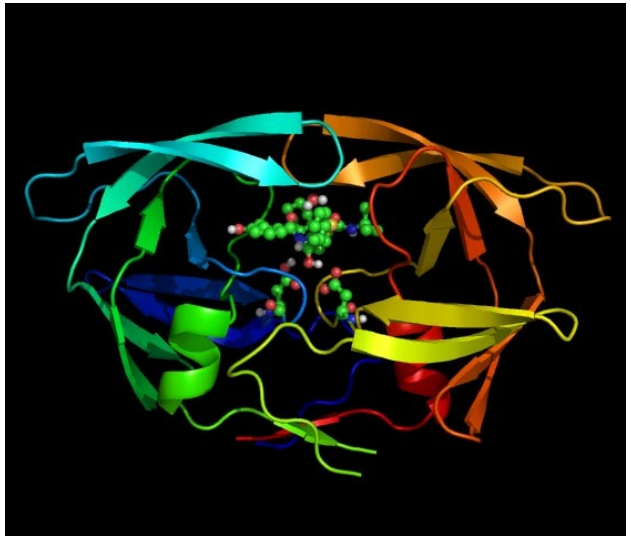
Action	Symbol	File	Support	Confidence
ADD_IN	BirdieProcessor.java	\softevo-test\...\birdie\BirdieProcessor.java	4	0.3333
CHG	~usage()	\softevo-test\...\birdie\Birdie.java	4	0.3333
DEL_IN	URLUtils.java	\softevo-test\...\util\URLUtils.java	3	0.25
ADD_IN	URLUtils.java	\softevo-test\...\util\URLUtils.java	3	0.25
CHG	~processFile(String, BirdieProcessor)	\softevo-test\...\birdie\Birdie.java	3	0.25
CHG	~close()	\softevo-test\...\birdie\BirdieProcessor.java	3	0.25
ADD_IN	ArrayUtils.java	\softevo-test\...\util\ArrayUtils.java	2	0.1666



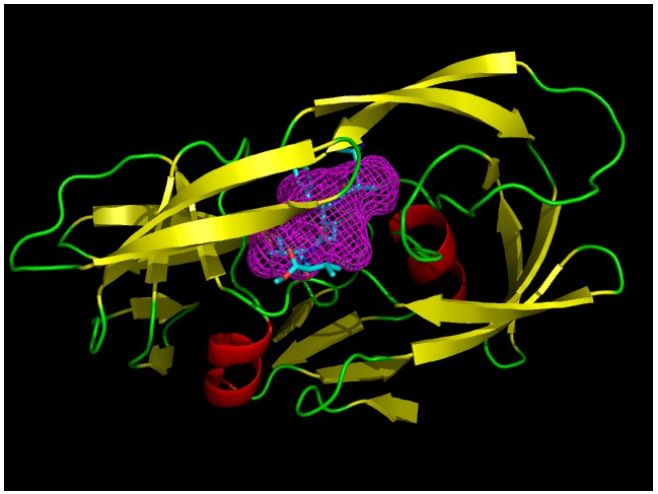
Mark pieces that are related to bugs.

eROSE: Guiding Programmers in Eclipse:
http://www.st.cs.uni-sb.de/softevo/erose/

Find new medications



Analyse the HIV virus to predict drug resistancies.

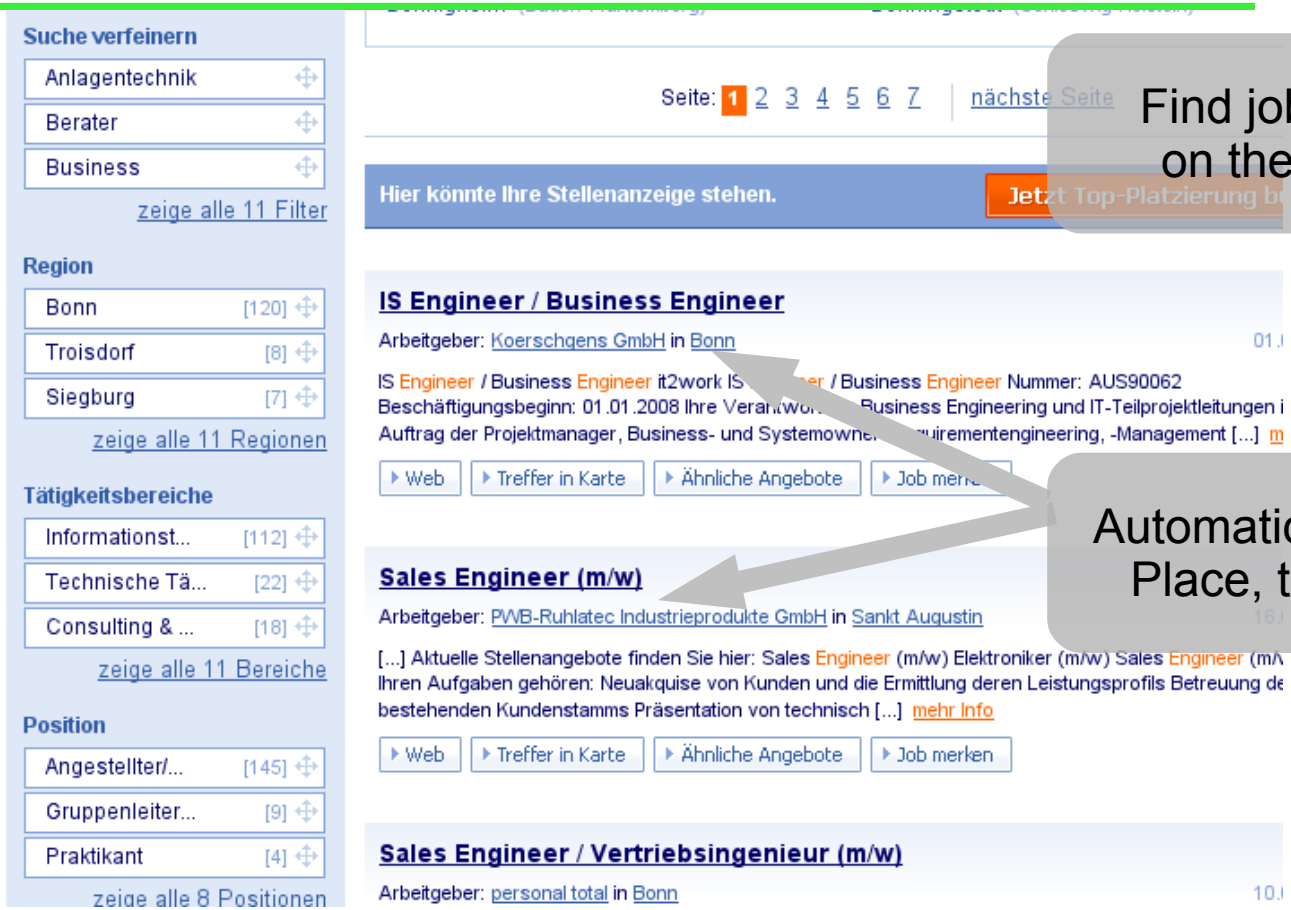


Make job searches easier

Make job searches easier

Find job postings on the internet.

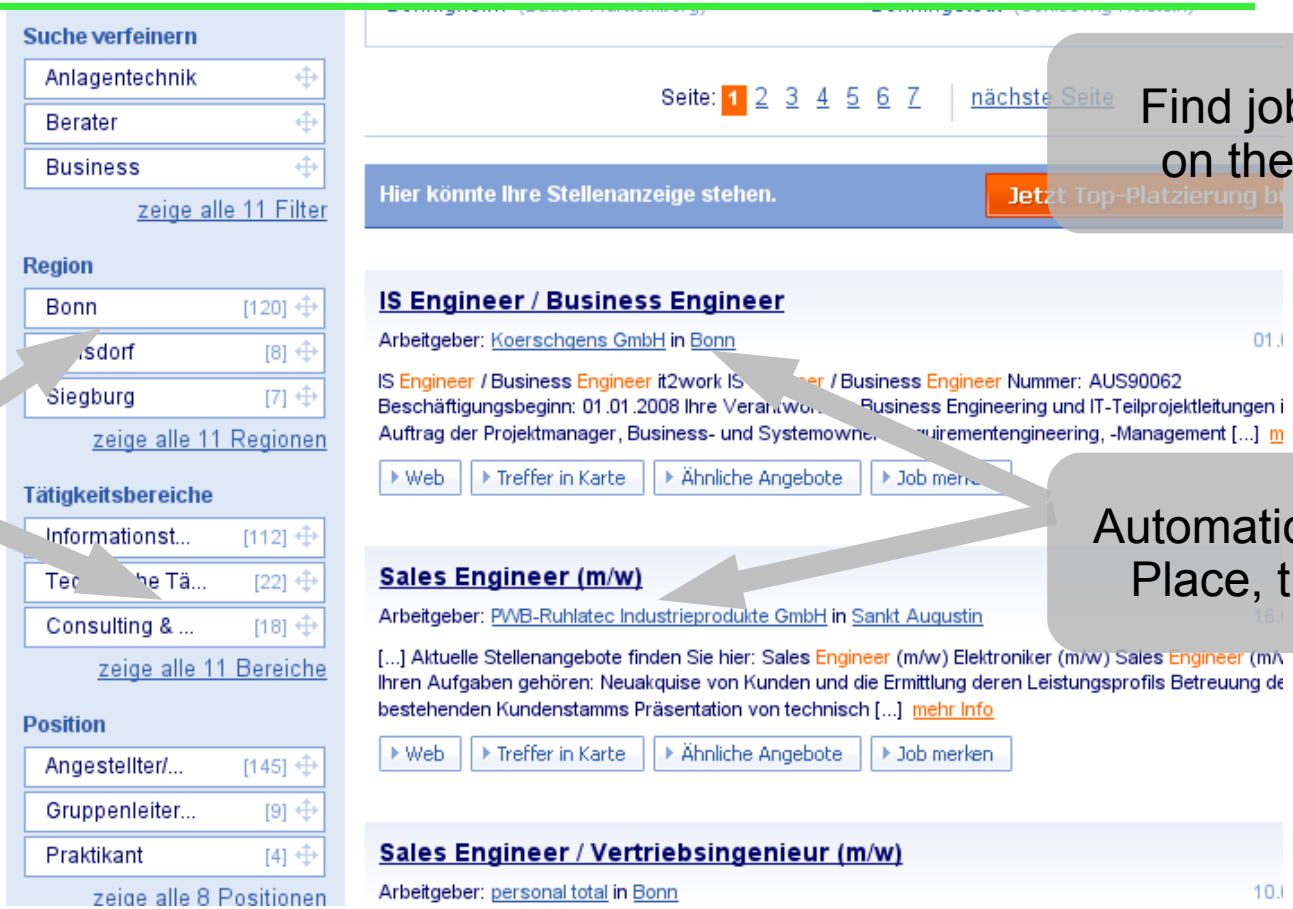
Make job searches easier



Find job postings on the internet.

Automatically extract: Place, title, date...

Make job searches easier

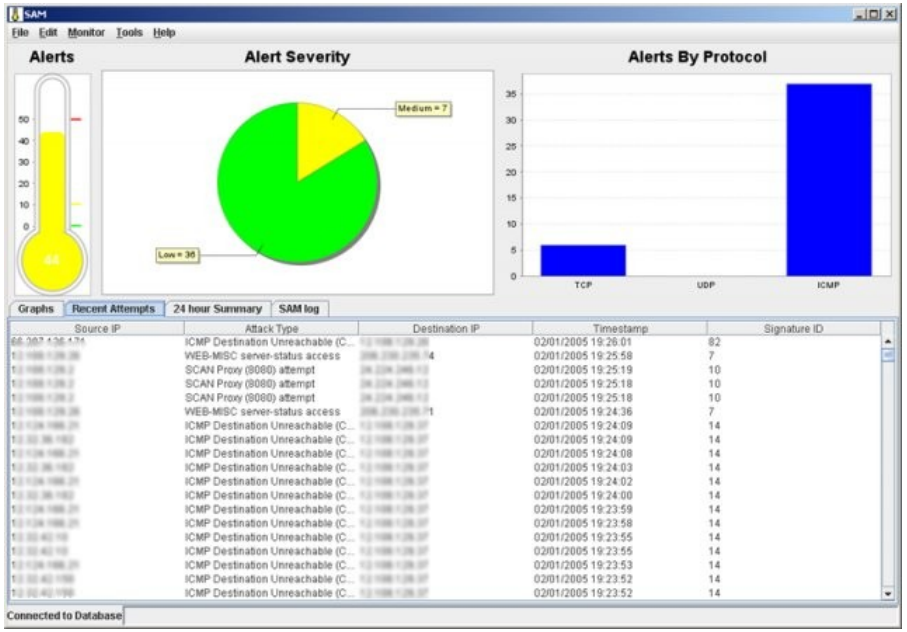


Find job postings on the internet.

Group by categories.

Automatically extract: Place, title, date...

Identify intrusion patterns



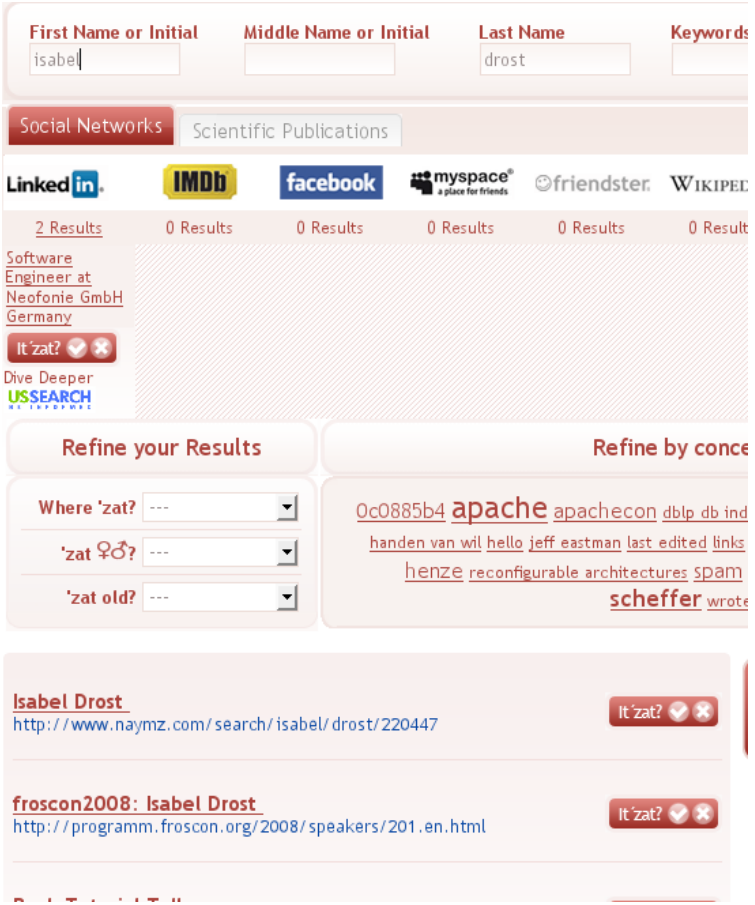
- Input logs for:
  - Normal users.
  - Detected attacks.
- Task:
  - Given live logs, alert me, if someone is attacking.

Learn search rankings

- Input:
  - User votings.
  - Clicks on search results.
  - Query refinement logs.
- Task:
  - Create a perfect ranking.

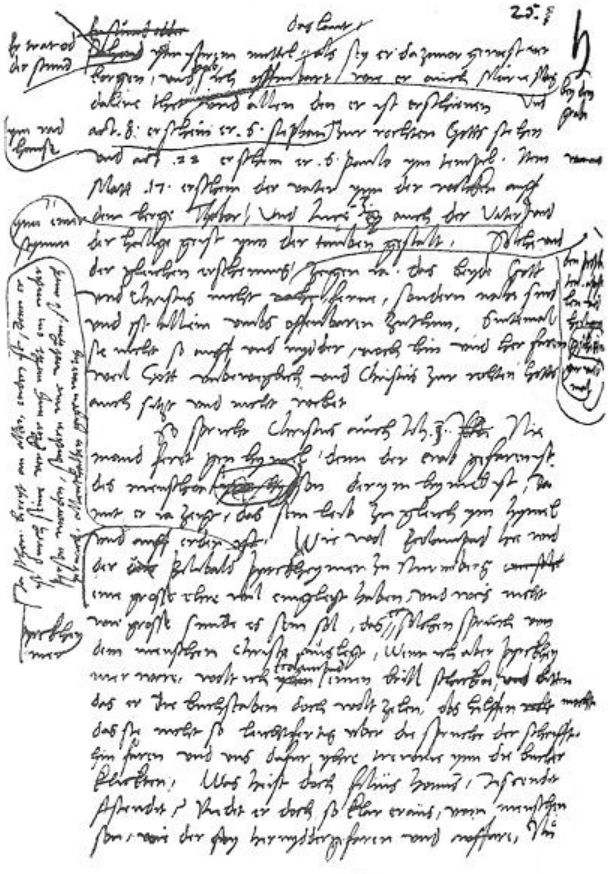
Aggregate information

- Input: e.g. data found in
  - Social networking sites.
  - Regular search engines.
- Task: find information
  - About some person.
  - About a company.



Convert handwritten text

- Input:
  - A handwritten text.
- Task:
  - Convert to (machine-) readable form.



Conclusions

- We are at the beginning.
- High demand for scalable machine learning.
- We need You –
  - Your enthusiasm.
  - Your mathematical knowledge.
  - Your proficiency in or will to learn Hadoop.
  - Your interest in understanding Your data.
- mahout-dev@apache.org mahout-user@apache.org

Some advertising

Berlin - 8th of September at 5p.m.

newthinking store Berlin

Tucholskystr. 48

Hadoop User/Developer Meeting Germany

MAHOUT