Review of Lone Star Software Symposium: NFJS 2010

Peter Donton

Variety of Sessions and Speakers

- 50 sessions on a wide range of topics presented by knowledgeable professionals in approachable and manageable ways.
  - Java
    - Collections api
    - Unit testing and testing methodologies
    - Refactoring
    - Modular Java: intro to OSGi
    - Upcoming Java additions
  - Web development
    - RESTful applications
    - HTML 5
    - Javascript, jQuery
    - Testing

Variety of Sessions and Speakers

- Scala
- Groovy
- Frameworks: Spring, Grails, J2EE
- Agile Engineering Practices
- Cloud computing
- Maven
- Mobile development on iPhone, Android
- Keynote: Neal Ford – Smithing in the 21st Century

Variety of Sessions and Speakers

- Sessions to highlight
  - Tools and Techniques to build Smart Java Apps
  - Relational Database Alternatives

Smart Java Applications

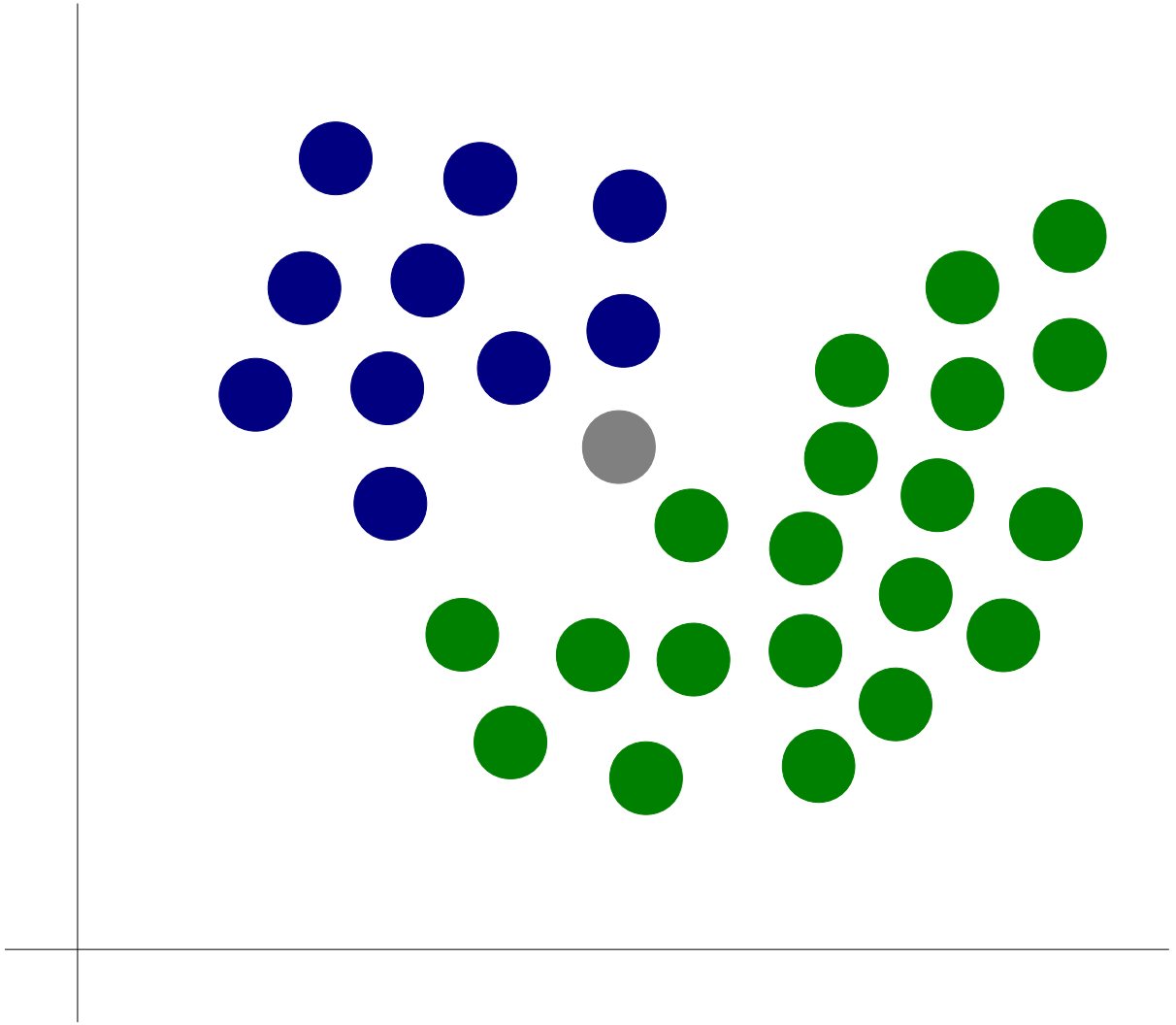
- Session was about the automated processing of information for:
  - Prioritization
  - Grouping
  - Search
  - Routing
- Prime example: “spam or ham” emails

Smart Java Applications

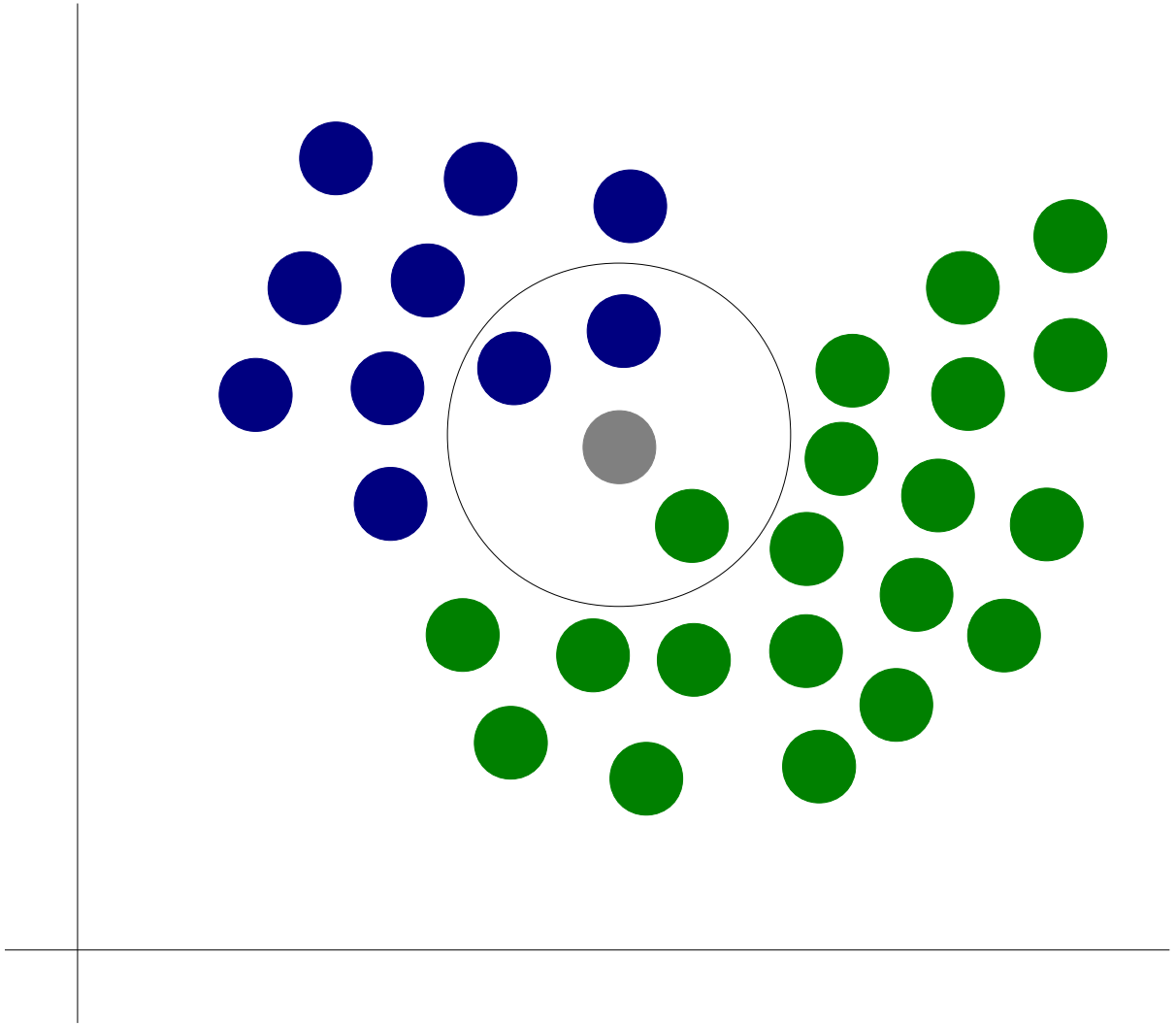
- Elements of machine learning
  - Feature: attribute or variable with heuristic properties
    - Key words
  - Class: belongs to a category, group, label or tag
    - “spam or ham”
  - Feature Selection: using features for classification to reduce the scope of problem space
  - Classification: techniques that use a set of features to assign data to one or more class
    - Bayesian Classification

Smart Java Applications

- Bayesian Classification
  - Bayes' Theorem: published in 1763
  - Gives probability of an event given the measured test probabilities
  - Application in spam filtering attributed to Paul Graham's “A Plan for Spam” in 2002 essay
    - http://www.paulgraham.com/spam.html



10 blue with prior probability of 10/30
20 green with prior probability of 20/30



Using a centroid, we can find the likelihood:

1/20 belonging to green

2/10 belonging to blue.

Smart Java Applications

- Use Bayes' Theorem:
  - Posterior probability = prior probalility x likelihood
  - (20/30) x (1/20) = 1/30 probability gray is green
  - (10/30) x (2/10) = 2/30 probability gray is blue

Smart Java Applications

- Classifier4J
  - Text classifiers using Bayesian or Word Vector techniques
  - Trained with String sentences, text files or JDBC
  - Open source: download at sourceforge

Smart Java Applications

```
 1 String edibleCookies = "A small cake made from stiff, sweet dough rolled"
 2         + "and sliced or dropped by spoonfuls on a large, flat pan and baked.";
 3 String httpCookies = "Consists of one or more name-value pairs containing"
 4         + "bits of information, which may be encrypted for privacy"
 5         + "and data security purposes.";
 6
 7 BayesianClassifier edibleCookieClassifier = new BayesianClassifier();
 8 edibleCookieClassifier.teachMatch(edibleCookies);
 9 edibleCookieClassifier.teachNonMatch(httpCookies);
10
11 String inputA = "I like warm and sweet snacks right from the pan.";
12 String inputB = "Personal information in a web session should be encrypted.";
13
14
15 System.out.println("probability that inputA is about edible cookies: "
16         + edibleCookieClassifier.classify(inputA));
17 System.out.println("probability that inputB is about edible cookies: "
18         + edibleCookieClassifier.classify(inputB));
```

Output:

```
probability that inputA is about edible cookies: .99
probability that inputB is about edible cookies: .01
```

Smart Java Applications

- Classifier4J and Bayes' Theorem
  - Needs training to establish prior probabilities
  - Only as good as the quality and quantity of training data but is surprisingly correct most of the time
  - Can be trained online or in batches

NoSQL for Java

- The session was a survey of non-relational data stores, its purpose, its various types and implementation examples

NoSQL for Java

- NoSQL described:
  - Non-relational, so normalization is not critical; loose or no schema
  - Loose transaction semantics and weaker consistency guarantees
  - Favor simplified reads over complex writes
  - Designed upfront with distributed/clustering/sharding environments in mind
  - Simple interfaces, RESTful

NoSQL for Java

- Brewers CAP Theorem
  - Can have two of consistency, availability, and partition tolerance
- NoSQL stores are BASE (as opposed to ACID)
  - Basically Available, Soft State, Eventually consistent

NoSQL for Java

- Types of NoSQL data stores
  - Key-value
  - Wide column
  - Document oriented
  - Graph
  - Object oriented

NoSQL for Java

- Key-value
  - Associative arrays or hash tables
  - No schema, no tables
  - No SQL for CRUD, just simple api
  - Easily scales horizontally
  - Decentralization, fault tolerant
  - Difficult reporting and anlaytics
  - Integrity is up to the application
- Voldemort created by LinkedIn

NoSQL for Java

- Wide column
  - Indexed columns that hold multiple values
  - Favors high availability over consistency
- Cassandra
  - Data is sorted on insert based on key
  - Fast writes, it does not involve reads or seeks
  - Read can happen on any node

NoSQL for Java

- Document oriented
  - Document is the atomic unit of information
  - No schema, typically with xml or json
  - Can have attributes
  - Self contained, but duplication is common
  - Efficient access to individual documents
- Terrastore
  - Json based; RESTful operations with range and predicate queries

NoSQL for Java

- Graph
- Object oriented

NoSQL for Java

- notes:
  - Long live RDBMS but it may not be suited for all web data store applications
  - Not Only SQL
  - Horizontal scalability is the big motivator
  - “Data is molded for its usage rather than being a slave to a normalized generalized relational data model”