Machine Learning: Algorithms and Applications

Quang Nhat Nguyen

Faculty of Computer Science
Free University of Bozen-Bolzano
October, 2007

Content

- **Introduction**
  - **The World Wide Web**
  - **Web mining**
  - **Machine learning**
  - **WEKA tool**
- Web crawling and data pre-processing
- Machine learning techniques and algorithms
- Evaluation

World Wide Web

- The Word Wide Web (a.k.a. the Web)
  - The largest and most widely known repository of hypertext
  - Billions of hypertext documents
  - Authored by millions of diverse people
  - Distributed over millions of computers, connected by variety of media
- Hypertext documents
  - Text
  - Links to other texts
- The term “hypertext” was first introduced by Nelson in 1965

The first hypertext systems

- Memex `[Bush, 1945]`
  - Stands for “memory extension”
  - Created in 1945, by Vannevar Bush
  - A photo-electrical-mechanical storage and computing device
  - To create and help follow hyperlinks across documents
- Xanadu `[Nelson, 1965]`
  - Created in 1965, by Doug Engelbart and Ted Nelson
  - To support robust two-way hyperlinks, version management, controversy management, annotation, and copyright management

History of the World Wide Web

- Initiated in 1980 at CERN (the European Organization for Nuclear Research)
  - By Tim Berners-Lee
  - A program to create and browse along named, typed bidirectional links between documents in a collection
- In 1989, Berners-Lee wrote the "*HyperText and CERN*" proposal
  - → To enable collaboration in the high-energy physics community
  - *HTML*: a markup language for composing and rendering hypertext
  - *HTTP*: a transport protocol for sending HTML documents
  - A platform-independent *web browser*: to receive and interpret data, and display results
- In May 1991, the information-sharing system (CERN HTTPD)
  - A web server of hypertext documents
  - A (client) web browser
  - Using HTML and HTTP

History of Web browsers

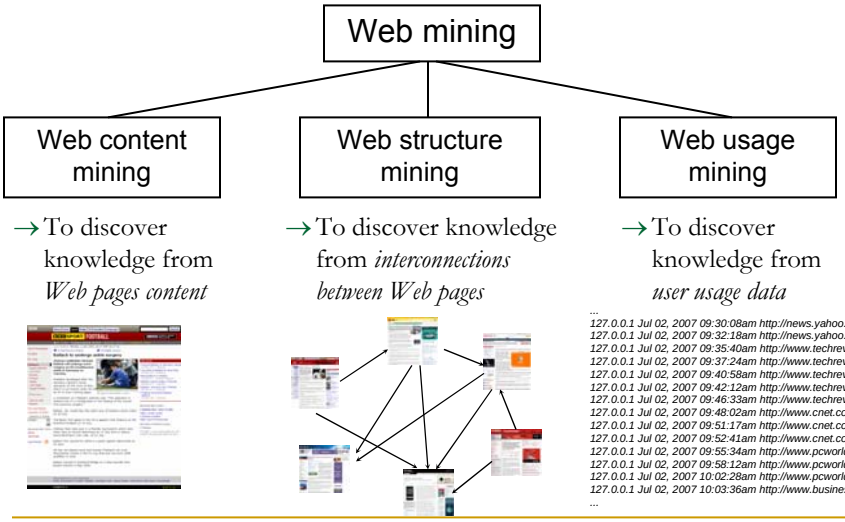
[*http://www.livinginternet.com/w/wi_browse.htm*]

- In 1990: *WorldWideWeb*, by Tim Berners-Lee, the first web browser on a NeXT computer
- In 1991: *Line-mode*, by Nicola Pellow, a web browser aimed to work on any device (from Unix to MS DOS systems)
- In 1992: *ViolaWWW*, by Pei Wei, a browser for Unix systems, capable of displaying graphics and downloading applets
- In 1993: *Mosaic*, by Marc Andreessen and Eric Bina, support for sound, video clips, forms, bookmarks, and history files
- … *(and several others)*
- In 1994: *Mozilla*, by Netscape, the first commercial web browser, in 2004 *Firefox* (open-source web browser) was released
- In 1995: *Internet Explorer*, by Microsoft, passed Netscape in 1999

Introduction of Web mining (1)

- Definitions of Web mining…
  - → The use of data mining techniques to automatically discover Web documents and services, extract information from Web resources, and uncover general patterns on the Web. `[Etzioni, 1996]`
  - → The discovery and analysis of useful information from the World Wide Web. `[Cooley et al., 1997]`
  - → Detecting and exploiting statistical dependencies (i.e., also called patterns) between terms, Web pages, and hyperlinks. `[Chakrabarti, 2003]`
- We adopt the broad definition stated by `[Cooley et al., 1997]`

Introduction of Web mining (2)



Application examples of Web mining

- Web content mining
  - Web pages organization
    →To categorize incoming Web pages in appropriate categories
  - Web pages filtering
    →To filter out those Web pages not interested for a given user
- Web structure mining
  - Ranking of search engine results
- Web usage mining
  - Identification of user browsing patterns

Challenges to Web mining (1)

- The hyperlink structure (i.e., the structure of links between Web pages)
  - A Web page cites another one through a link to that (cited) page
  - A very important property, peculiar to Web mining
  - Very huge and complex
    → Exploiting the link structure is a difficult and challenging task
- Many Web pages are HTML documents
  - Noisier than documents in IR systems (e.g., news articles or research papers)
    e.g., grammar or spelling errors
  - Much less structured than documents in IR systems
    no schema, and very diverse in terms of length and structure
  - Terms (i.e., keywords) vocabulary is typically larger than that in IR systems
  - The topics (i.e., subjects) are more diverse than those in IR systems

Challenges to Web mining (2)

- Parsing markup tags
  - <HTML>, <HEAD>, <TITLE>, <BODY>, <FORM>, <TABLE>, <STYLE>, <SCRIPT…>, <A HREF=…>, <IMG SRC=…>, etc.
  - Some markup tags may, in some cases, introduce importance information

    E.g., a bold (<b>) or italic (<i>) typeface markup may be used to indicate important terms.
  - Simply removing markup tags may cause some irrelevant terms (e.g., "Home", "Site map", "Click here", "Contact us") to be included in the terms vocabulary
- Different content types and formats
  - Content types: text, image, audio, video, etc.
  - Content formats: HTML, XML, PDF, MS Word, MP3, WMA, RA, WMV, AVI, etc.
  - In this course, we focus on the textual content type

Challenges to Web mining (3)

- The huge number of existing Web pages
  - The number of indexable Web pages
    - In 2000: ~2.5 billion, with a rate of growth of 7.3 million pages per day [*http://www2.sims.berkeley.edu/research/projects/how-much-info/internet.html*]
    - End of Jan. 2005: >11.5 billion [Gulli and A. Signorini, 2005]
  - The number of Web sites: >120 million

    (in Jun. 2007, [*http://news.netcraft.com/archives/web_server_survey.html*])
  - The competence in indexing Web pages

    (in Nov. 2004, [*http://blog.searchenginewatch.com/blog/041111-084221*])

Search engine	Reported size (in billion)	Page depth (in Kb)
Google	8.1	101
MSN	5.0	150
Yahoo	4.2 (estimate)	500
Ask Jeeves	2.5	101+

Challenges to Web mining (4)

- The dynamic character of Web pages
  - Documents in IR systems are often static (i.e., unchanged) over time; whereas Web pages are more dynamic (i.e., content may be generated dynamically upon request, or updated over time)
  - A Web page previously available may now be unavailable (i.e., the "dead links" problem)
- Different languages used
  - Different Web pages may use different languages
  - English is still the dominated language used

    (~78% in 2000, [*http://www2.sims.berkeley.edu/research/projects/how-much-info/internet.html*])
- The population of Web users
  - Much larger in the number and more diverse (in goals, interests, experience, etc.) than that of traditional IR systems
  - Collaboration amongst users are promising, but also challenging

Related research areas

- Web retrieval
- Information retrieval
- **Machine learning** ← the focus of this course!
- Statistics
- Text categorization and mining
- Data mining
- Databases

Introduction of Machine learning

- Definitions of Machine learning…
  - → A process by which a system improves its performance `[Simon, 1983]`
  - → Any computer program that improves its performance at some task through experience `[Mitchell, 1997]`
  - → Programming computers to optimize a performance criterion using example data or past experience `[Alpaydin, 2004]`
- Representation of the learning problem `[Mitchell, 1997]`

  Learning = Improving with experience at some task
  - Improve over task **T**
  - With respect to performance measure **P**
  - Based on experience **E**

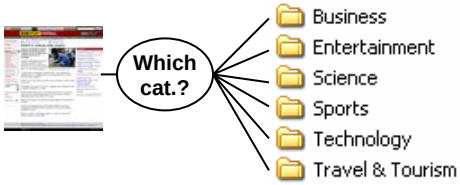
Application examples of ML (1)

Web pages filtering problem

- **T**: to predict which Web pages a given user is interested in
- **P**: % of Web pages correctly predicted
- **E**: a set of Web pages identified as interested/uninterested for the user

Web pages categorization problem

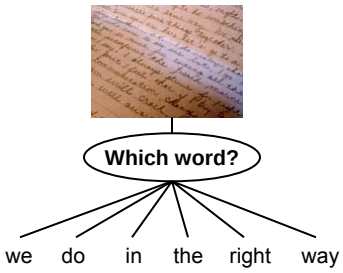
- **T**: to categorize Web pages in predefined categories
- **P**: % of Web pages correctly categorized
- **E**: a set of Web pages with specified categories



Application examples of ML (2)

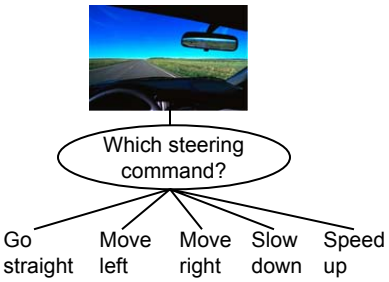
Handwriting recognition problem

- **T**: to recognize and classify handwritten words within images
- **P**: % of words correctly classified
- **E**: a database of handwritten words with given classifications (i.e., labels)



Robot driving problem

- **T**: to drive on public highways using vision sensors
- **P**: average distance traveled before an error (as judged by human overseer)
- **E**: a sequence of images and steering commands recorded while observing a human driver



Key elements of a ML problem (1)

- Selection of the training examples
  - Direct or indirect training feedback
  - With teacher (i.e., with labels) or without
  - The training examples set should be representative of the future test examples
- Choosing the target function (a.k.a. hypothesis, concept, etc.)
  - F: $X \rightarrow \{0,1\}$
  - F: $X \rightarrow$ a set of labels
  - F: $X \rightarrow R^+$ (i.e., the positive real numbers domain)
  - …

Key elements of a ML problem (2)

- Choosing a representation of the target function
  - A polynomial function
  - A set of rules
  - A decision tree
  - A neural network
  - …
- Choosing a learning algorithm that learns (approximately) the target function
  - Regression-based
  - Rule induction
  - ID3 or C4.5
  - Back-propagation
  - …

Issues in Machine Learning (1)

- Learning algorithm
  - What algorithms can approximate the target function?
  - Under which conditions does a selected algorithm converge (approximately) to the target function?
  - For a certain problem domain and given a representation of examples which algorithm performs best?
- Training examples
  - How many training examples are sufficient?
  - How does the size of the training set influence the accuracy of the learned target function?
  - How does noise and/or missing-value data influence the accuracy?

Issues in Machine Learning (2)

- Learning process
  - What is the best strategy for selecting a next training example? How do selection strategies alter the complexity of the learning problem?
  - How can prior knowledge (held by the system) help?
- Learning capability
  - What target function should the system learn?
    Representation of the target function: expressiveness vs. complexity
  - What are the theoretical limits of learnability?
  - How can the system generalize from the training examples?
    To avoid the overfitting problem
  - How can the system automatically alter its representation?
    To improve its ability to represent and learn the target function

WEKA – a ML and DM software toolkit

- WEKA is a ML and DM software tool written in Java
- Main features
  - A set of data pre-processing tools, learning algorithms and evaluation methods
  - Graphical user interfaces (including data visualization)
  - Environment for comparing learning algorithms

WEKA – Main environments

- Simple CLI

  A simple command-line interface

- **Explorer** (we will use this environment!)

  An environment for exploring data with WEKA

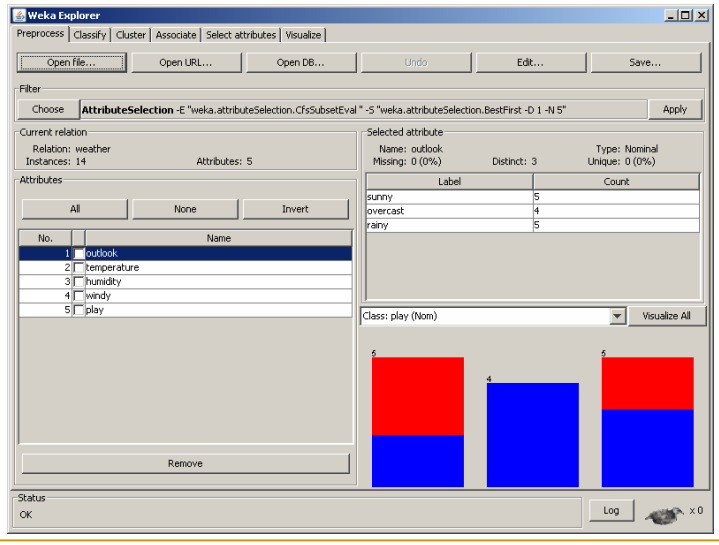
- Experimenter

  An environment for performing experiments and conducting statistical tests between learning schemes

- KnowledgeFlow

  An environment that allow you to graphically (drag-and-drop) design the flows of an experiment

WEKA – The Explorer environment



WEKA – the Explorer environment

- **`Preprocess`**
  To choose and modify the data being acted on
- **`Classify`**
  To train and test learning schemes that classify or perform regression
- **`Cluster`**
  To learn clusters for the data
- `Associate` ← Not discussed in this course!
  To discover association rules from the data
- **`Select attributes`**
  To determine and select the most relevant attributes in the data
- **`Visualize`**
  To view an interactive 2D plot of the data

WEKA – Data set format

- WEKA only deals with flat (text) files in *ARFF* format
- Example of a data set

```
@relation weather

@attribute outlook {sunny, overcast, rainy}
@attribute temperature real
@attribute humidity real
@attribute windy {TRUE, FALSE}
@attribute play {yes, no}

@data
sunny,85,85,FALSE,no
overcast,83,86,FALSE,yes
…
```

Name of the data set

Nominal attribute

Numeric attribute

Classification (i.e., by default, the last defined attribute)

The examples (instances)

WEKA Explorer: Data pre-processing

- Data can be imported from a file in formats: ARFF, CSV, C4.5, binary
- Data can also be read from a URL or from an SQL database using JDBC
- Pre-processing tools in WEKA are called *`filters`*
  - Discretization
  - Normalization
  - Re-sampling
  - Attribute selection
  - Transforming and combining attributes
  - …

→*Now let's look at the WEKA Explorer GUI…*

WEKA Explorer: Classifiers (1)

- Classifiers in WEKA are models for predicting nominal or numeric quantities
- Classification techniques implemented in WEKA
  - Naïve Bayes classifier and Bayesian networks
  - Decision trees
  - Instance-based classifiers
  - Support vector machines
  - Neural networks
  - Linear regression
  - …

WEKA Explorer: Classifiers (2)

- Select a classifier
- Select test options
  - **`Use training set`**. The learned classifier will be evaluated on the training set
  - **`Supplied test set`**. To use a different data set for the evaluation
  - **`Cross-validation`**. The data set is divided in a number of folds, and the learned classifier is evaluated by cross-validation
  - **`Percentage split`**. To indicate the percentage of the data set held out for the evaluation

WEKA Explorer: Classifiers (3)

- **`More options…`**
  - **Output model**. To output (display) the learned classifier
  - **Output per-class stats**. To output the precision/recall and true/false statistics for each class
  - **Output entropy evaluation measures**. To output the entropy evaluation measures
  - **Output confusion matrix**. To output the confusion (classification-error) matrix of the classifier's predictions
  - **Store predictions for visualization**. The classifier's predictions are saved in the memory so that they can be visualized later
  - **Output predictions**. To output the predictions on the test set
  - **Cost-sensitive evaluation**. The errors is evaluated with respect to an indicated cost matrix
  - **Random seed for XVal / % Split**. To specify the random seed used when randomizing the data before it is divided up for evaluation purposes

WEKA Explorer: Classifiers (4)

- **`Classifier output`** shows important information
  - **Run information**. The learning scheme options, name of the dataset, instances, attributes, and test mode
  - **Classifier model (full training set)**. A textual representation of the classifier learned on the full training data
  - **Predictions on test data**. The learned classifier's predictions on the test set
  - **Summary**. The statistics on how accurately the classifier predicts the true class of the instances under the chosen test mode
  - **Detailed Accuracy By Class**. A more detailed per-class break down of the classifier's prediction accuracy
  - **Confusion Matrix**. Elements show the number of test examples whose actual class is the row and whose predicted class is the column

WEKA Explorer: Classifiers (5)

- **`Result list`** provides some useful functions
  - **Save model**. Saves a model (i.e., a trained classifier) object to a binary file. Objects are saved in Java 'serialized object' form
  - **Load model**. Loads a pre-trained model (i.e., a previously learned classifier) object from a binary file
  - **Re-evaluate model on current test set**. To evaluate a previously learned classifier on the current test set
  - **Visualize classifier errors**. To show a visualization window that plots the results of classification

    Correctly classified instances are represented by crosses, whereas incorrectly classified ones show up as squares
  - …

WEKA Explorer: Cluster builders (1)

- Cluster builders in WEKA are models for finding groups of similar instances in a dataset
- Clustering techniques implemented in WEKA
  - Expectation maximization (EM)
  - k-Means
  - ...
- Clusters can be visualized and compared to "true" clusters (if given)

→*Now let's see a cluster builder in the WEKA Explorer GUI…*

WEKA Explorer: Cluster builders (2)

- Select a cluster builder
- Select cluster mode
  - **Use training set**. The built clusters will be tested on the training set
  - **Supplied test set**. To use a different data set for testing the built clusters
  - **Percentage split**. To indicate the percentage of the data set held out for testing the built clusters
  - **Classes to clusters evaluation**. To compare how well the built clusters match up with the indicated (pre-assigned) classes in the data
- Store clusters for visualization

  →To save the built clusters in the memory for a visualization later
- Ignore attributes

  →To select which attributes are ignored in the clusters building

WEKA Explorer: Attribute selection

- To identify which (subsets of) attributes are the most predictive ones
- In WEKA, a method for attribute selection consists of two parts
  - "*Attribute Evaluator*". An evaluation method for evaluating the appropriateness of attributes
    correlation-based, wrapper, information gain, chi-squared,…
  - "*Search Method*". A search method for determining how (in which order) the attributes are examined
    best-first, random, exhaustive, ranking,…

→*Now let's see an illustration in the WEKA Explorer GUI…*

WEKA Explorer: Data visualization

- Visualization is very useful in practice
  helps to determine difficulty of the learning problem
- WEKA can visualize
  - single attributes (1-D visualization)
  - pairs of attributes (2-D visualization)
- Different class values (labels) are visualized in different colors
- `Jitter` slider supports better visualization when many instances locate (concentrate) around a point in the plot
- Zooming in/out (i.e., by increasing/decreasing `PlotSize` and `PointSize`)

→*Now let's see an illustration in the WEKA Explorer GUI…*

References

- E. Alpaydin. *Introduction to Machine Learning*. The MIT Press, 2004.
- V. Bush. *As We May Think*. The Atlantic Monthly, July 1945.
- S. Chakrabarti. *Mining the Web: Discovering Knowledge from Hypertext Data*. Morgan Kaufmann Publishers, 2003.
- H. Chen and M. Chau. *Web mining: Machine learning for Web applications*. Annual Review of Information Science and Technology, 38: 289-329, 2004.
- R. Cooley, B. Mobasher, and J. Srivastava. *Web mining: information and pattern discovery on the World Wide Web*. In Proc. 9th IEEE Int'l Conf. Tools with Artificial Intelligence, pp. 558-567, 1997.
- O. Etzioni. *The World Wide Web: Quagmire or gold mine*. Communications of the ACM, 39(11): 65-68, 1996.
- A. Gulli and A. Signorini. *The indexable web is more than 11.5 billion pages*. In Proc. 14th Int'l Conf. World Wide Web, pp. 902-903, 2005.
- T. M. Mitchell. *Machine Learning*. McGraw-Hill, 1997.
- T. H. Nelson. *Complex information processing: A File Structure for the Complex, the Changing, and the Indeterminate*. In Proc. 20th Nat'l Conf., pp. 84-100, 1965.
- H. A. Simon. *Why Should Machines Learn?* In R. S. Michalski, J. Carbonell, and T. M. Mitchell (Eds.): Machine learning: An artificial intelligence approach, chapter 2, pp. 25-38. Morgan Kaufmann, 1983.
- *WEKA* tool (http://www.cs.waikato.ac.nz/ml/weka/).