Masaryk University

Department of Theoretical Physics and Astrophysics

Master's Thesis

Virtual Observatory and Data Mining

Jaroslav Vážný

Brno 2011

for future generations

What is it that makes us human? It's not something you can program. You can't put it into a chip. It's the strength of the human heart. The difference between us and machines.

Marcus Wright

Acknowledgements

I would like to thank Filip Hroch and Petr Škoda for their remarkable support and patience not only during this project. I greatly appreciated comments and help from following friends: Tereza Jeřábková, Josef Pacula and Petr Šafařík.

Funding for the SDSS and SDSS-II has been provided by the Alfred P. Sloan Foundation, the Participating Institutions, the National Science Foundation, the U.S. Department of Energy, the National Aeronautics and Space Administration, the Japanese Monbukagakusho, the Max Planck Society, and the Higher Education Funding Council for England. The SDSS Web Site is http://www.sdss.org/.

The SDSS is managed by the Astrophysical Research Consortium for the Participating Institutions. The Participating Institutions are the American Museum of Natural History, Astrophysical Institute Potsdam, University of Basel, University of Cambridge, Case Western Reserve University, University of Chicago, Drexel University, Fermilab, the Institute for Advanced Study, the Japan Participation Group, Johns Hopkins University, the Joint Institute for Nuclear Astrophysics, the Kavli Institute for Particle Astrophysics and Cosmology, the Korean Scientist Group, the Chinese Academy of Sciences (LAMOST), Los Alamos National Laboratory, the Max-Planck-Institute for Astronomy (MPIA), the Max-Planck-Institute for Astrophysics (MPA), New Mexico State University, Ohio State University, University of Pittsburgh, University of Portsmouth, Princeton University, the United States Naval Observatory, and the University of Washington.

This research has made use of the SIMBAD database, operated at CDS, Strasbourg, France.

This research has made use of The WEKA Data Mining Software [Hall et al., 2009].

This research has made use of Free software [**?**].

Abstract

Modern astrophysics and natural sciences in general become extensively penetrated by Computer Science. Petabyte scale databases, GRID Computing and Data Mining become routine part of scientific work. Skills related to information technologies are now essential. The new concept of data infrastructure named Virtual Observatory naturally emerged from digitalized surveys. Based on proven standards offers ideal basis for dealing with distributed heterogeneous data. This thesis is a case study of using Virtual Observatory and Data Mining technologies to proceed automatics classification of Be stars. Photometric and spectra classification were done on large scale sample of almost 200 000 spectra from SDSS Segue survey.

Many byproduct originated during the work on the thesis. Wiki pages, Virtual Observatory and Data Mining documentation and about dozen of programs for data manipulation, spectra fitting and result publishing. Everything is given free to the public on the web of the project.

http://physics.muni.cz/~vazny/wiki/index.php/Diploma_work

Contents

Contents v

List of Figures vii

Nomenclature viii

1 Virtual Observatory (VO) 3
- 1.1 Data avalanche: Opportunity or disaster? . . . 3
- 1.2 International Virtual Observatory Alliance (IVOA) . . . 4
- 1.3 Architecture . . . 4
- 1.4 VOResources . . . 6
- 1.5 Data Access Protocols . . . 8
  - 1.5.1 Cone Search Protocol . . . 8
  - 1.5.2 Simple Image Access Protocol . . . 9
  - 1.5.3 Simple Spectra Access Protocol . . . 10
- 1.6 Data Formats . . . 11
  - 1.6.1 VOTable . . . 11
  - 1.6.2 FITS . . . 13
- 1.7 Tools & Libraries . . . 14

2 Data Mining 15
- 2.1 Supervised Methods . . . 15
  - 2.1.1 Decision Tree (DT) . . . 15
    - 2.1.1.1 Cross-validation . . . 16
    - 2.1.1.2 Example: Classifying Galaxies Stars and QSO . . . 17
- 2.2 Existing Projects . . . 19
  - 2.2.1 Weka . . . 19
  - 2.2.2 SVM lib . . . 20
  - 2.2.3 DAME . . . 20

3 Be candidates 21
- 3.1 Be stars . . . 21
- 3.2 Photometric Data Mining . . . 23
  - 3.2.1 Data preprocessing . . . 24
  - 3.2.2 Classification . . . 26
- 3.3 Spectral Data Mining . . . 26

3.3.1 Testing Data . . . 28
3.3.2 Training Data . . . 28
3.3.2.1 Spectra Reduction . . . 29
3.3.3 Spectra Lines Characteristics . . . 31
3.3.3.1 Normalization . . . 31
3.3.3.2 The hight of the Hα line . . . 31
3.3.3.3 The noise level of the spectrum . . . 31
3.3.3.4 The width of the Hα line . . . 31
3.3.4 Data Mining . . . 35
3.3.5 Results . . . 35
3.3.6 Experiment . . . 42
3.3.7 Conclusion . . . 42

4 Conclusion **45**

Appendix1: Spectra of result objects **47**

Appendix2: Spectra of Ondejov Be stars **49**

References **51**

List of Figures

1 Astroinformatics in the context of astronomy [Ball and Schade, 2010] 1

2 Thesis structure 2

1.1 Chapter structure 3

1.2 IVOA members 4

1.3 VO Architecture 5

1.4 UML diagram of VOResource 7

2.1 Chapter structure 15

2.2 Color Diagram of the problem.It shows that individual object classes occupy different regions in the diagram 18

3.1 Chapter structure 21

3.2 Model of a typical Be star. Emission lines coming from an equatorial disk is added to the photo-spheric absorption spectrum. Central B star emits UV (Lyman continuum) and ionizes the disk, which in turn re-emits at high wavelength such as visible domain. [Hirata and Kogure, 1984] 22

3.3 Example of spectra of Be stars based on view angle [Slettebak, 1988] 22

3.4 Schematic diagram of the photometric Data Mining process. The lists of confirmed Be stars consisted of Hipparcos IDs, this was correlated with Hipparcos catalog to obtain right ascension and declination of the objects and subsequently cross-matched with 2MASS catalog to get photometric data. The second set of B stars were acquired in similar manner but using SQL the condition was set to get B type stars different from the list of Be stars. 23

3.5 Color diagram of confirmed Be stars Vs B stars 24

3.6 Color diagram of confirmed Be stars Vs B stars with errors 25

3.7 Schematic diagram of the spectral Data Mining process. Using SSA protocol the spectra from Ondejov server was acquired based on the list from photometric study. SSH Tunneling was necessary since Ondejov spectra are top secret and therefore not available to the public. Convolution had to be performed to ensure compatibility with SDSS. Afterwards desired features were extracted automatically from the spectra after the continuum and Hα line were fitted by appropriate functions. The same was done for spectra from SDSS except the convolution process. 27
3.8 Reduction of Ondejov's spectra of the Be star 4 Hercules. The top figure shows Gaussian function used for convolution with the spectrum, followed by the original spectrum then there is a spectrum after convolution with the Gaussian function. The last is the final spectrum after reduction. 30
3.9 Normalized spectrum of Be star 60 Cyg. The top figure depicts the continuum fit. The bottom figure shows the region (width of the green line) used for extraction. The position of the line correspond to the maximum value in the region of 50$\mathring{A}$. The Gaussian fit is in red. Although the fit is almost perfect, this approach fails to get characteristic "double peak" of the emission line. 33
3.10 Normalized spectrum of Be star HR 8682. The top figure depicts the continuum fit. The bottom figure shows the region (width of the green line) used for extraction. The position of the line correspond to the maximum value in the region of 50$\mathring{A}$. The Gaussian fit is in red. . . . 34
3.11 Spectrum of 37
3.12 Spectrum of 37
3.13 Spectrum of 38
3.14 Spectrum of 38
3.15 Spectrum of 4 Her. Be star. Spectral Type B9pe. 39
3.16 Spectrum of HR 7418 (Albireo B). A fast-rotating Be star, with an equatorial rotational velocity of at least 250 kilometers per second. Its surface temperature has been spectroscopically estimated to be about 13.200 K. Spectral Type B8Ve. 39
3.17 Spectrum of 6 Cepheus. Be star. Spectral Type B3IVe. 40
3.18 Spectrum of Gamma Cassiopeiae. Be Star. Spectral Type B0IVpe C. 40

Introduction

From the dawn of its existence astronomy has always been starving for data but in the last few decades the situation has changed and now we are facing data deluge of biblical proportions. The data are not just increasing in size but also in complexity and dimensionality [Ball and Schade, 2010]. Astroinformatics is the new field of science which has emerged from this technology driven progress. Virtual Observatory, Machine Learning, Data Mining, Grid Computing are just few examples of new tools available to scientists.

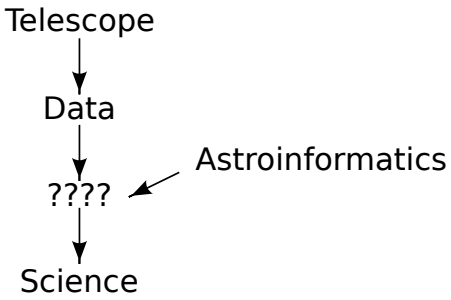


Figure 1: Astroinformatics in the context of astronomy [Ball and Schade, 2010]

The astronomers are not alone and particle physics, biology and other sciences are also in the vanguard of the data intensive science. This is great opportunity for interdisciplinary collaboration.

This work deals with the problem of semi-automatic procedures for finding Be stars [Porter and Rivinius, 2003] candidates in the astronomy surveys. More than straight forward process it's trail and error approach probing new possibilities with rather interesting than useful results.

The aim of this work is to be introductory to the technologies of Virtual Observatory and massive data processing in general.

Chapter one is an introduction to the technologies related to Virtual Observatory. The motivation behind the concept is given without paying too much attention to historical details. Main principles and protocols are discussed and explained. Important aspect are demonstrated on numerous examples. Chapter two is an introduction to Machine Learning and Data Mining in the context of astrophysics. Only methods used in practical part of this work are described in detail: Decision Trees and Support Vector Machines. Examples of several classifications are demonstrated. Third chapter introduces issues of Be stars. Chapter Four is practical application

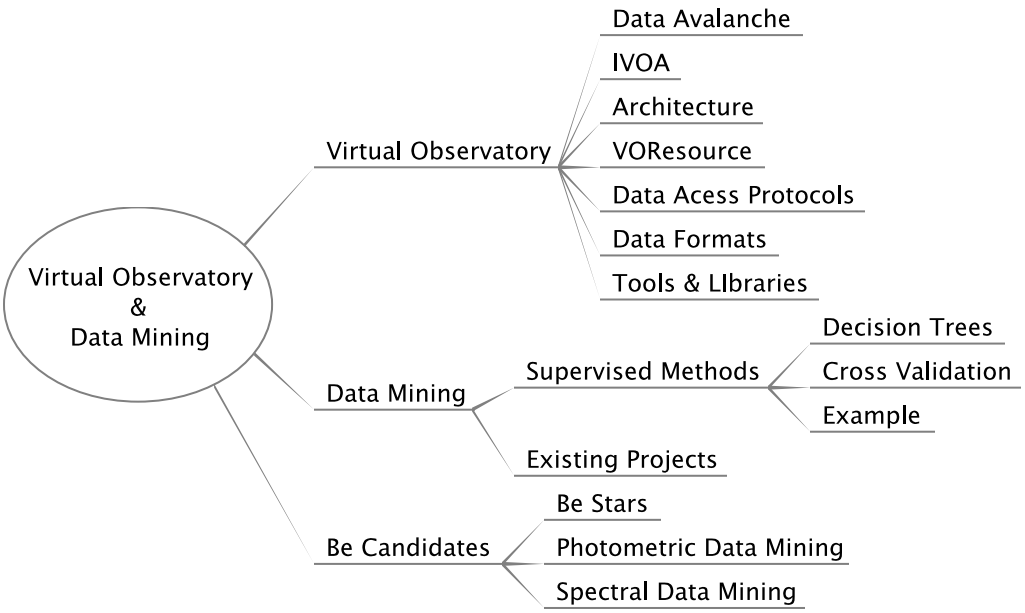


Figure 2: Thesis structure

of previously described technologies and methods. Training data of confirmed Be stars from Ondrejov are correlated with others catalogues to obtain color indexes and spectra. Results are processed by Data Mining algorithms using several libraries and tools. In the last chapter achieved results are critically discussed.

Many scripts were written to achive individual goals. In the text there are numerous commented snippets of codes. Their purpose is to demonstrate the concept and they are thefore short and without auxiliary technicalities such are error handling etc. They are mostly Python and shell scripts. Any interested person can obtain the full source codes (including thesis itself) from GIT repository [1].

Name	Description
analyse	Check the wavelength range, raname according to target name
getSpectraList	Create SSA Compliant list from
getSpectra	Get spectra links from SSA Server
madmax	Extract feautres from spectra
convolve	Reduction of Ondejov's spectra
pf	Print Fits. Shows the spectrum
dm	Perform classification
makeHTML	Creates HTML pages of results

Table 1: Scripts developed within the scope of the thesis.

Activities related to this work went beyond this text. Wiki pages[2] were created to present the results and discuss related topic with supervisor as well as with others scientist around the world. Source codes were maintained by GIT version system allowing easy sharing. All software used and produced are open source.

[1] git://github.com/astar/diplomaWork
[2] http://physics.muni.cz/~vazny/wiki/index.php/Diploma_work

Chapter 1

Virtual Observatory (VO)

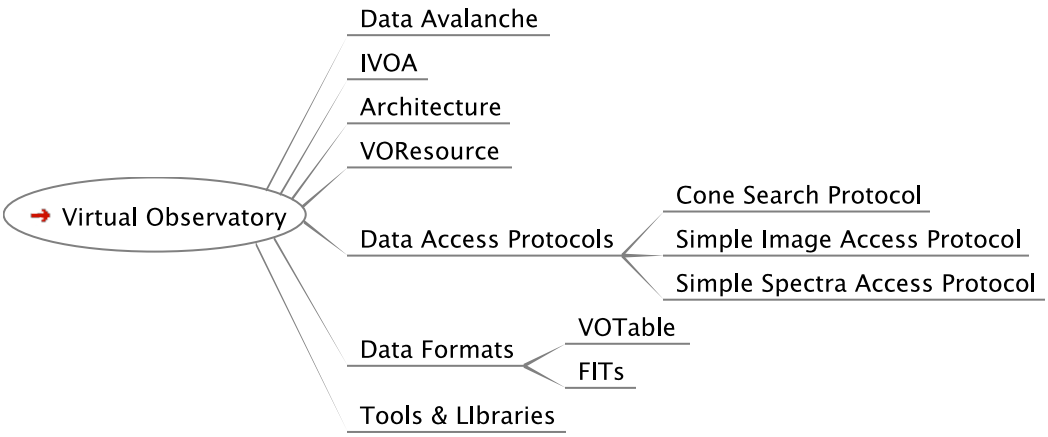


Figure 1.1: Chapter structure

1.1 Data avalanche: Opportunity or disaster?

There are two important trends in current astronomy surveys:

- Size: The cumulative compressed data holdings of the ESO archive will reach 1 PetaByte by 2012 [Hanisch and Quinn, 2010]. Projects like Large Synoptic Survey Telescope (LSST) will produce about 30 TB per night, leading to a total database over the ten years of operations of 60 PB for the raw data [Becla et al., 2006].

- Complexity: Modern surveys will cover the sky in different wave-bands, from gamma- and X-rays, optical, infrared to radio. The ability to cross correlate these observations together may lead to new understanding of physical phenomenas. [Hanisch and Quinn, 2010]

Such an amount of data is not possible to transfer over the network. Data resources are heterogeneous, distributed and decentralized in nature.

There is an interesting analogy with the problem (and the solution) which scientists discovered during LEP project at CERN. Their problem was too many documents

in different formats. Tim Berners-Lee[1] designed set of protocols (URIs, HTTP and HTML) which allowed link and share documents [Berners-Lee and Cailliau, 1990]. This was recognized as generally useful and World Wide Web was born. An important role plays the World Wide Web Consortium (W3C) in developing Web standards[2].

1.2 International Virtual Observatory Alliance (IVOA)

What is necessary is sets of standards and protocols to deal with heterogeneous distributed data and authority which encourages their implementation. Such an authority is the International Virtual Observatory Alliance (IVOA). It comprises 19 VO programs from Argentina, Armenia, Australia, Brazil, Canada, China, Europe, France, Germany, Hungary, India, Italy, Japan, Russia, Spain, the United Kingdom, and the United States and inter-governmental organizations (ESA and ESO) [Hanisch and Quinn, 2010].

Figure 1.2: IVOA members

Standards and specifications produced by IVOA can be obtained at `http://www.ivoa.net/`.

1.3 Architecture

The Architecture is depicted on the figure 1.3. The level of abstraction goes from top to bottom. Starting with interfaces, used by people or applications to discover resources. Next level is the service layer implemented by standard protocols, followed by the hardware level where actual data are stored. This onion like structure hides the complexity of the lower layer and provide data and meta-data to the higher layer. This concept is similar to TCP/IP [3] protocol.

[1] Sir Timothy John "Tim" Berners-Lee. British engineer and computer scientist and MIT professor credited with inventing the World Wide Web.

[2] Prior to its creation, incompatible versions of HTML were offered by different vendors, increasing the potential for inconsistency between web pages.

[3] TCP/IP (Transmission Control Protocol/Internet Protocol). The basic communication language or protocol of the Internet.

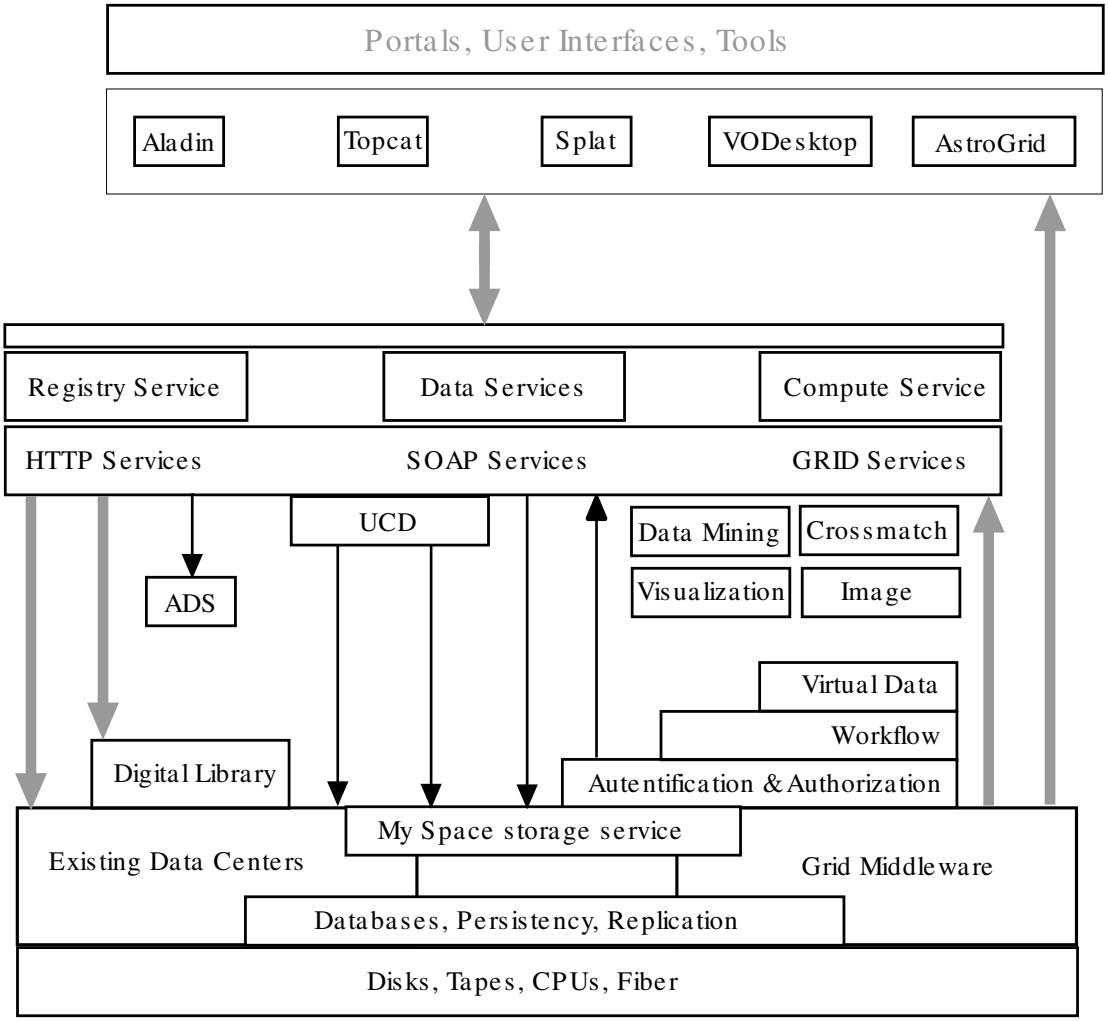


Figure 1.3: VO Architecture

The VO architecture is serviced oriented. Each service is autonomous with well defined defined boundaries. Very important aspect of VO implementation is the adoption of formats and protocols used in astronomy (FITS) and Computer Science (XML [1] , Web service [2] SOAP [3]) for many years. In other words VO does not try to reinvent the wheel but it stands on the shoulders of giants.

1.4 VOResources

A resource is a general term referring to a VO element that can be described in terms of who curates or maintains it and which can be given a name and a unique identifier. Just about anything can be a resource: it can be an abstract idea, such as sky coverage or an instrumental setup, or it can be fairly concrete, like an organization or a data collection. [Benson et al., 2009]

UML[4] diagram of the resource is on the figure 1.4. Next paragraph is an attempt to explain this diagram to non-programmers. Full arrow means generalization, Resource can be a generalization of organization, data collection, application or service. Single arrow means association. Organization can be linked (associated) together with other organization (multiplicity is represented by number 1, 0..). The same is true for data collection. Organization is a generalization of and/or provider which can own zero to N services. Diamond means the aggregation. Publisher can have any resources.

[1] Extensible Markup Language (XML) is a set of rules for encoding documents in machine-readable form.

[2] method of communication between two electronic devices over a network.

[3] Simple Object Access Protocol, is a protocol specification for exchanging structured information in the implementation of Web Services in computer networks.

[4] Unified Modeling Language. Standardized general-purpose modeling language in the field of object-oriented software engineering.

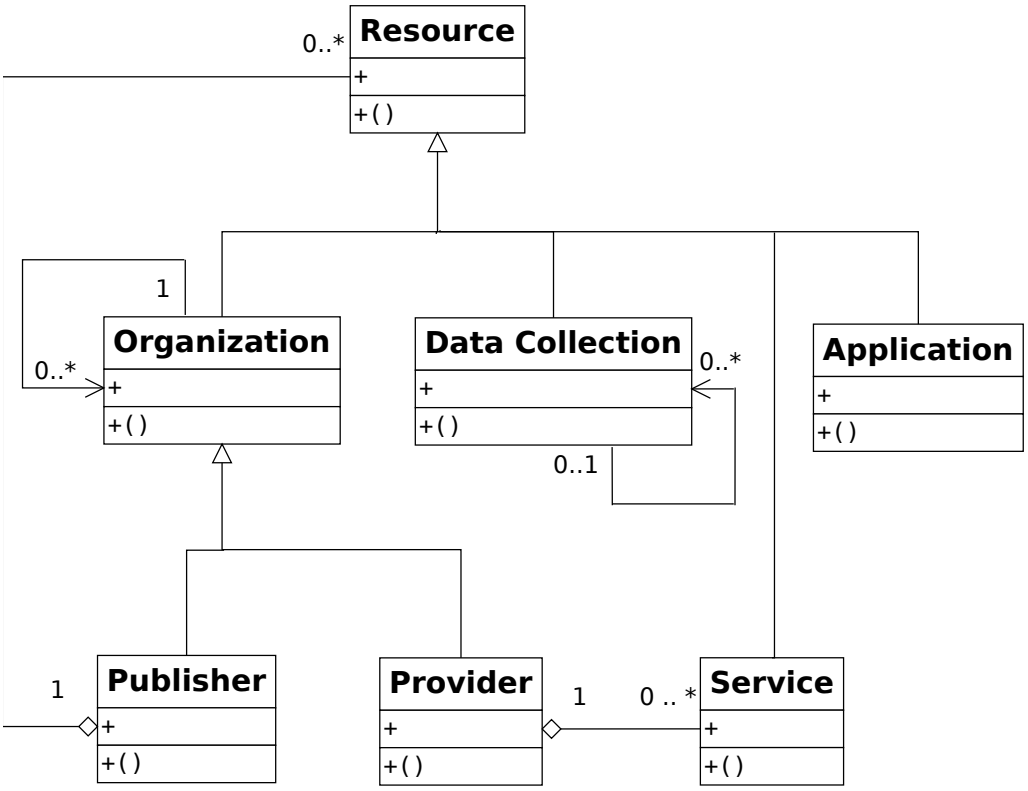


Figure 1.4: UML diagram of VOResource

Following example uses program stilts[1] to query registry with parameter shortName equal to 'AIASCR'[2]. This returns VOTable containing meta-data about the resource.

```
stilts regquery query="shortName like 'AIASCR'"
regurl=http://registry.euro-vo.org/services/RegistrySearch
ofmt=votable-tabledata > resourceExample.vot
```

Rows 1–4 define XML nad VOTable schema with adequate locations (xmlns[3]) followed by informations about the actual resource. The listing is abbreviated.

```
<?xml version='1.0'?>
<VOTABLE version="1.1"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xmlns="http://www.ivoa.net/xml/VOTable/v1.1">
.
<DATA>
<TABLEDATA>
  <TR>
    <TD>ivo://asu.cas.cz</TD>
    <TD>AIASCR</TD>
    <TD>Astronomical Institute of the Academy of Sciences of the Czech
       Republic Naming Authority</TD>
    <TD>http://stelweb.asu.cas.cz/web/index/index-en.php</TD>
    <TD>Petr Skoda &lt;skoda@sunstel.asu.cas.cz&gt;</TD>
```

1.5 Data Access Protocols

Protocols are very important part of Virtual Observatory. Their understanding is key to comprehend the concepts behind VO. They allow to discover resource and obtain desirable data. All of them are based on existing web standards and are designed to be simple and therefor easy to implement on existing astronomical archives. The main idea is simple and universal: HTTP GET request with parameter is sent to the resource and structured document (VOTable) is sent back.

1.5.1 Cone Search Protocol

Cone Search was the first standard protocol of Virtual Observatory. It enables to retrieve records from an astronomical catalog. The input is the query which describes sky position and the radius on the sky. The output is a list of objects whose positions lie in the defined vicinity. The output is formatted as a VOTable. Service compliant with Cone Search Protocol is called Cone Search Service. Only the request and response is specified not the implementation or data storage.

The requirements are:

1. Respond to a HTTP GET request represented by a URL

[1] STIL Tool Set. Set of command-line tools based on STIL, the Starlink Tables Infrastructure Library.

[2] Astronomical Institute of the Academy of Sciences of the Czech Republic

[3] XML namespaces. Provide uniquely named elements and attributes in an XML document.

```
http://<server-address>/<path>?[<extra-GET-arg>&[...]]
```

The constrains are expressed as a list of ampersand-delimited GET arguments. For example:

```
http://simbad.u-strasbg.fr/simbad-conesearch.pl?RA=24.5&DEC=-57.2&
    SR=0.1
```

Where RA is right-ascension, DEC declination and SR the radius of the cone in the ICRS coordinate system in decimal degrees. These parametrers are requied others are optional.

2. Return an XML document in the VOTable format.

   There are several requirements on the contents of the table:

   - UCD fields "ID_MAIN", "POS_EQ_RA_MAIN", "POS_EQ_DEC_MAIN" must be present.
   - Return VOTable with single PARAM element name="Error" in the case of error.

Cone Search is implemented in many software packages. Besides standard VO tools like TOPCAT or STILTS also in MUNIPACK and many others. Following example shows simple query to SIMBAD [Wenger et al., 2000] catalog using method *urlopen* of Python library *urllib2*.

```
import urllib2
response = urllib2.urlopen('http://simbad.u-strasbg.fr/simbad-conesearch.
    pl?RA=24.5&DEC=-57&SR=0.1')
print response.read()
```

The same result can be obtained using program like wget[1] or Web browser.

1.5.2 Simple Image Access Protocol

The key idea behind the SIA Protocol is to allow users and programs to retrieve images created by an image service on-the-fly. From technical point of view it is designed in a similar way as Cone Search Protocol (see 1.5.1), specifically as name-value HTTP GET requests and the VOTable XML format output. The user specifies ideal image coverage (position and the size) he wants to receive and the image service produces a list of images it can return in the VOTable format. The user then could issue getImage request to retrieve desirable images.

There are following requirements for compliance. To be a SIA service To be a SIA service, it must support:

- Image Query web method,
- Image Retrieval (getImage) web method.

[1] program for non-interactive download of files from the Web

Furthermore the image service should be registered to be able to locate optimal service. There are several types of image services:

- Image Cutout Service.

  Provides rectangular regions of large images.

- Image Mosaicing Service.

  Size, scale and projection could be specified.

- Atlas Image Archive

  Pre-computed atlas of images.

- Pointed Image Archive.

  Images are not part of a sky survey but rather focused on specific source

To get a list of images query has to send via HTTP GET method. The first part is base URL. The second part are parameters specifying image properties such as position (POS), size (SIZE), etc.

```
1 http://<server-address>/<path>?[<extra GET arg>&[...]]
```

There are two examples of using SIA protocol to obtain image. First one from SDSS, second from Hubble Space Telecope archive.

```
1 http://skyview.gsfc.nasa.gov/cgi-bin/vo/sia.pl?SURVEY=SDSS&POS
     =18.87667,-0.86083&SIZE=1
2 http://hubblesite.org/cgi-bin/sia/hst_pr_sia.pl?POS=83.6,22.0&SIZE=1.0
```

There is more complex example using Astrogrid framework to show how to discover SIA service and obtain an image. First registry method searchSiap is used to find SIA service for SDSS, this is then used in SiapSearch method to obtain result in VOTable format.

```
1 In [1]: from astrogrid import Registry, ConeSearch
2 In [2]: list = reg.searchSiap('SDSS')
3 In [3]: print [p['id'] for p in list]
4 -------> print([p['id'] for p in list])
5 ['ivo://nasa.heasarc/skyview/sdss']
6
7 In [4]: siap = SiapSearch('ivo://nasa.heasarc/skyview/sdss')
8 In [5]: result = siap.execute(18.8, -0.8, 1.0)
```

1.5.3 Simple Spectra Access Protocol

SSA Protocol allows to discover and obtain 1-D spectra from VO Service. It shares many similarities to the previously discussed SIA Protocol.

defines a uniform interface to remotely discover and access simple 1-D spectra.
similar to that of the older Simple Image Access (SIA)
The process to obtain a spectrum compose of following steps:

- Query the resource registry.

- Data discovery to selected service to get aviable resources in VOTable format.
- Download selected spectra using URL.

The spectra could be one of the following types:

- Pre-computed
- Computed on the fly

To be a SSA-compliant, the service must provide:

1. HTTP GET interface, returning the query response encoded as a VOTable document, with at least parameters POS, SIZE, TIME, BAND, and FORMAT.
2. GetData method returning data in at least one of the SSA-compliant data formats (VOTable, FITS)
3. FORMAT=METADATA metadata query feature

Following example show how to discover resources with SSA capability using STILTS program.

```
stilts regquery query="shortName like 'ESO' capability/@standardID =
'ivo://ivoa.net/std/SSA'" ocmd="keepcols 'ShortName accessUrl'"
ofmt=ascii
```

With information of service URL, one can specify a query to obtain a list with aviable spectra in VOTable format. This can be used in Web browser or via programs such *wget* or *curl*.

```
http://archive.eso.org/apps/ssaserver/EsoProxySsap?REQUEST=queryData&POS
    =83.63,22&SIZE=1
```

1.6 Data Formats

Astronomy has always been, by its nature, on vanguard of image producing and processing. This is especially true for the era of digitalization. The situation with data formats in astronomy is unique. There are just few very good standards with variety of implementation in many programming languages. Virtual Observatory takes advantage of this heritage and implement these formats in sensible way into its own standards.

1.6.1 VOTable

Motivation

VOTable is flexible storage and exchange format fundamentally interconnected with Virtual Observatory. It has features for big-data and Grid computing. Data can be stored in different ways in dependence on the charatecter and size. Small tables

can be stored in pure XML [1], while large-scale data can be referenced with the URL [2] syntax protocol://location. It combines web standards (it is based on XML) and astronomy tradition in storing data (it is FITS compatible). Expiration and authentication are also supported.

Structure

Following example of VOTable was created from SDSS FITS file used in this work. First there is an information about XML and VOTable versions and references to corresponding XML Schema [3]. ¡TABLE¿ tag encapsulating tabular data. ¡FIELD¿ tag describe identifinaction (ID), type and precision of columns. ¡DATA¿ tag contains data (here) in TABLEDATA format (other types are FITS and BINARY)

```
<?xml version="1.0" encoding="utf-8"?>
<!-- Produced with vo.table version 0.6
     http://www.stsci.edu/trac/ssb/astrolib
     Author: Michael Droettboom <support@stsci.edu> -->
<VOTABLE version="1.0"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:noNamespaceSchemaLocation="http://www.ivoa.net/xml/VOTable/v1.0"
 xmlns="http://www.ivoa.net/xml/VOTable/v1.0">
 <RESOURCE type="results" >
  <TABLE >
   <FIELD ID="col0" name="wave" datatype="float" unit=""
   precision="F9"/>
  <DATA>
    <TABLEDATA>
     <TR>
      <TD>4012.50757</TD>
     </TR>
 </TABLEDATA>
   </DATA>
  </TABLE>
 </RESOURCE>
</VOTABLE>
```

Examples

All examples were created using ATpy[4]

Following example shows transformation FITS into VOTable.

```
In [1]: import atpy
In [2]: tbl = atpy.Table('spSpec-53401-2052-458.fit',hdu=1)
Auto-detected input type: fits
```

[1] Extensible Markup Language. W3C standard. Set of rules for encoding documents in machine-readable form

[2] Uniform Resource Locator. Uniform Resource Identifier (URI) that specifies where an identified resource is available and the mechanism for retrieving it.

[3] Define the legal building blocks of an XML document. Note: XML schema can be described by XML Schema, Document Type Definition (DTD) or RELAX NG

[4] High-level Python package providing a way to manipulate tables of astronomical data in a uniform way.

```
In [3]: tbl.write('votableExample.xml')
Auto-detected input type: vo
```

1.6.2 FITS

Motivation

"An archival format must be utterly portable and self-describing, on the assumption that, apart from the transcription device, neither the software nor the hardware that wrote the data will be available when the data are read." [Council, 1995]

FITS (Flexible Image Transport System) was originally created for data exchange between WSRT [1] and the VLA [2] [Schlesinger, 1997]. It is now used as a file format to store, transmit, and manipulate scientific data and it is (thanks to it's revolutionary design) de facto standard in astronomy.

Structure

One file can contains several HDUs (Header Data Units).The first part of each HDU is the header, composed of ASCII card images containing keyword=value statements that describe the size, format, and structure of the data that follow.

- Primary header and data unit (HDU).
- Conforming Extensions (optional).
- Other special records (optional, restricted).

Standards and documents related to FITS are maintained by IAUFWG [3] and aviable at http://fits.gsfc.nasa.gov.

Examples

There are many libraries for working with FITS files. The official list is aviable at http://fits.gsfc.nasa.gov/fits_libraries.html. PyFITS, library for Python programming language was used for following examples. PyFITS is a development project of the Science Software Branch at the Space Telescope Science Institute http://www.stsci.edu/resources/software_hardware/pyfits.

Reading FITS headers.

```
In [1]: import pyfits
In [2]: hdulist = pyfits.open('spSpec-53237-1886-248.fit')
In [3]: hdulist.info()
Filename: spSpec-53237-1886-248.fit
No.    Name         Type      Cards   Dimensions   Format
0    PRIMARY    PrimaryHDU    213  (3874, 5)    float32
1                BinTableHDU    54  6R x 23C     [1E, 1E, ...
2                BinTableHDU    54  44R x 23C    [1E, 1E, ...
3                BinTableHDU    18  1R x 5C      [1E, 1E, ...
```

[1] Westerbork Synthesis Radio Telescope
[2] Very Large Array
[3] International Astronomical Union FITS

Printing primary HDU.

```
In [4]: print hdulist[0].header
-------> print(hdulist[0].header)
DATE-OBS= '2004-08-20'      / 1st row - TAI date
TAIHMS = '10:36:18.11'      / 1st row - TAI time (HH:MM:SS.SS) (TAI-UT =
    appr
TAI-BEG =      4599713999.00 / Exposure Start Time
TAI-END =      4599717089.00 / Exposure End Time
MJD    =              53237 / MJD of observation
MJDLIST = '53237 '          /
VERSION = 'v3_140_0'        / version of IOP
TELESCOP= 'SDSS 2.5-M'      / Sloan Digital Sky Survey
```

Updating FITS file.

```
  In [1]: prihdr = hdulist[0].header
  In [2]: prihdr.update('observer', 'Astar')
  In [3]: prihdr.add_history('I updated this file 3/27/11')
```

1.7 Tools & Libraries

There are many programs and libraries allowing user to interact with VO services. Such application is called VO Enabled. Thanks to openness and standardisation anyone can develop his own application or enable existing[1] application to interact with VO Sevices . Libraries are also aviable for many programming languages enabling advanced users to interact with VO from scripts and programs. Such diversity is healthy and propably the only possible way to ensure natural evolution of Virtual Observatory. This chapter describes some of the libraries and applications ued during this work. Low level tools and libraries are stressed as opposite to standard introductory texts of Virtual Observatory where the focus is on "user friendly" GUI applications.

[1] For example Astroweka or Mirage.

Chapter 2

Data Mining

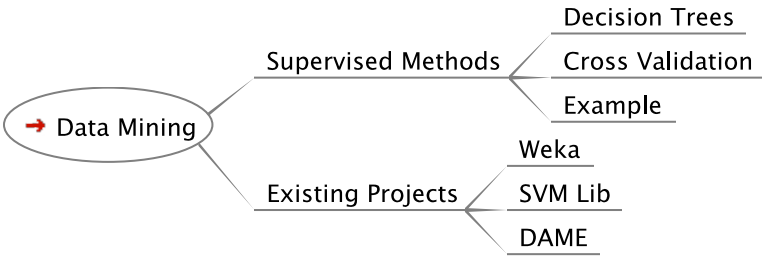


Figure 2.1: Chapter structure

Virtual Observatory may be seen as data infrastructure. It enables astronomers to get data more easily in a uniform way. But there is another and even bigger problem now. How to deal with huge amount of data? Can we change the problem to opportunity? Can we discover new phenomena, new types of objects or exploit natural groups in the data? Data Mining and related techniques are created exactly for such purposes. Used correctly, it can be powerful approach, promising scientific advance. On the other side this field is very complex with dozens of different methods and algorithms. This forms needs and opportunity for interdisciplinary cooperation with Data Mining experts. This can be very beneficially for both fields, providing astronomers with interesting methods for data analysis and computer scientist with large ammout of quality data.

2.1 Supervised Methods

These methods are also known as predictive[Ball et al., 2010]. They rely on training set with known target property. This set must be representative. The selected method is trained on that set and the result is then used on data for which the target property is not known. Among supervised method are classification, regression, anomaly detection and others.

2.1.1 Decision Tree (DT)

Is an example of supervised classification. Based on final number of data $(x^{(1)}, \ldots, x^{(p)})$ with known class $C_1, \ldots, C_m$ classifier is created, i.e. image f classifying any

$x \in \mathcal{X}, f : \mathcal{X} \to \mathcal{Y}$, where $\mathcal{X}$ is a set of possible input vectors and $\mathcal{Y}$ is a set which values represent classes $C_1, \ldots, C_m$ (for example $\mathcal{Y} = 1, \ldots, m$). The model is constructed based on training set as a tree structure, where leaves represent classifications and branches conjunctions of features that lead to those classifications. The main advantages of DT are:

- Simple to understand and interpret.
- Able to handle both numerical and categorical data.
- Uses a white box model.
- Perform well with large data in a short time.

In pseudo-code, the general algorithm for building decision trees is [Kotsiantis et al., 2007]:

1. Check for base cases
2. For each attribute a
   - Find the normalized information gain from splitting on a
3. Let "a best" be the attribute with the highest normalized information gain
4. Create a decision node that splits on "a best"
5. Recur on the sublists obtained by splitting on "a best", and add those nodes as children of node

Furthermore algorithms C4.5 is described for several reasons: Its code is available and free implementations exist (J48 in Weka), is de-facto standard in classification using DT, is used in practical part of this work. The key question of DT algorithm is how to choose attribute for splitting the tree. C4.5 Uses measures based on information entropy:

$$H(X) = -\sum_{i=1}^{n} p(x_i) \log_2 p(x_i), \tag{2.1}$$

where $p(x_i)$ is probability of occurrence of class i and n is the number of classes.

After the tree is created it is optimized by pruning, which prevents over-fitting.

2.1.1.1 Cross-validation

The quality of the training set is crucial to good results. The amount of data for testing is always limited. In general, one cannot be sure whether a sample is representative. If for example certain group is missing, one could not expect a classifier learned from such data to perform well on the examples s of that class. One of the technique used here is cross-validation.

The data is divided into fixed number of partitions and each in turn is used for testing and the reminder is used for training. Finally, the number of partitions error estimates are averaged to yield an overall error. The standard is to used 10-fold cross-validation. This number is a result of tests on numerous data sets [Witten and Frank, 2005]

2.1.1.2 Example: Classifying Galaxies Stars and QSO

There is an example of classifying Galaxies Stars and QSO based on photometric properties using Decision Tree algorithm J48 (C4.5 in Weka). The data come from SDSS (Sloan Digital Sky Survey) DR7. 298 Objects were used (100 Stars, 99 Galaxies, 99 QSO). SDSS Filters u,g,r,i were used as parameters. Data were obtained using SQL query from SDSS CAS.

```
SELECT TOP 100 u-g,g-r,r-i,s.specClass
FROM PhotoPrimary p join SpecPhotoAll s on p.objid=s.objid
WHERE s.specClass in (1)
AND u between 18 and 19
UNION all
SELECT top 100 u-g,g-r,r-i,s.specClass
FROM PhotoPrimary p join SpecPhotoAll s on p.objid=s.objid
WHERE s.specClass in (2)
AND u between 18 and 19
UNION all
SELECT top 100 u-g,g-r,r-i,s.specClass
FROM PhotoPrimary p join SpecPhotoAll s on p.objid=s.objid
WHERE s.specClass in (3)
AND u between 18 and 19
```

The following listing shows the result of classification. The classifier was able to distinguish 95% of the processed objects.

Filter	Wavelength [$\mathring{A}$],
Ultraviolet (u)	3543,
Green (g)	4770,
Red (r)	6231,
Near Infrared (i)	7625,
Infrared (z)	9134,

Table 2.1: SDSS Filters

```
Correctly Classified Instances          96               95.0495 %
Incorrectly Classified Instances         5                4.9505 %
Kappa statistic                          0.9257
Mean absolute error                      0.0669
Root mean squared error                  0.1778
Relative absolute error                 15.0587 %
Root relative squared error             37.6973 %
Total Number of Instances              101
```

The big advantage of Decision Trees over black box algorithms (such as Neural Network) is that one could understand the classification process. The decision tree generated for this example is following:

```
  ug <= 0.663668
|   gr <= -0.191208: 1 (7.0)
|   gr > -0.191208: 3 (104.0/5.0)
ug > 0.663668
```

```
|   ri <= 0.285854: 1 (88.0/5.0)
|   ri > 0.285854
|   |   ri <= 0.314657
|   |   |   gr <= 0.692108: 2 (6.0)
|   |   |   gr > 0.692108: 1 (3.0)
|   |   ri > 0.314657: 2 (90.0/2.0)
```

Useful tool for understanding how classifier was successful on individual classes is the confusion matrix. Columns show how the object was classified and the row what is his actual class. In this example QSO were classified correctly in 100% cases. Distinction between stars and galaxies are a bit worse and the algorithm classify 2 galaxies incorrectly as stars and two stars were confused with galaxies. One stars was incorrectly classified as QSO.

```
 s  g  q   <-- classified as
30  2  1 |  s
 2 33  0 |  g
 0  0 33 |  q
```

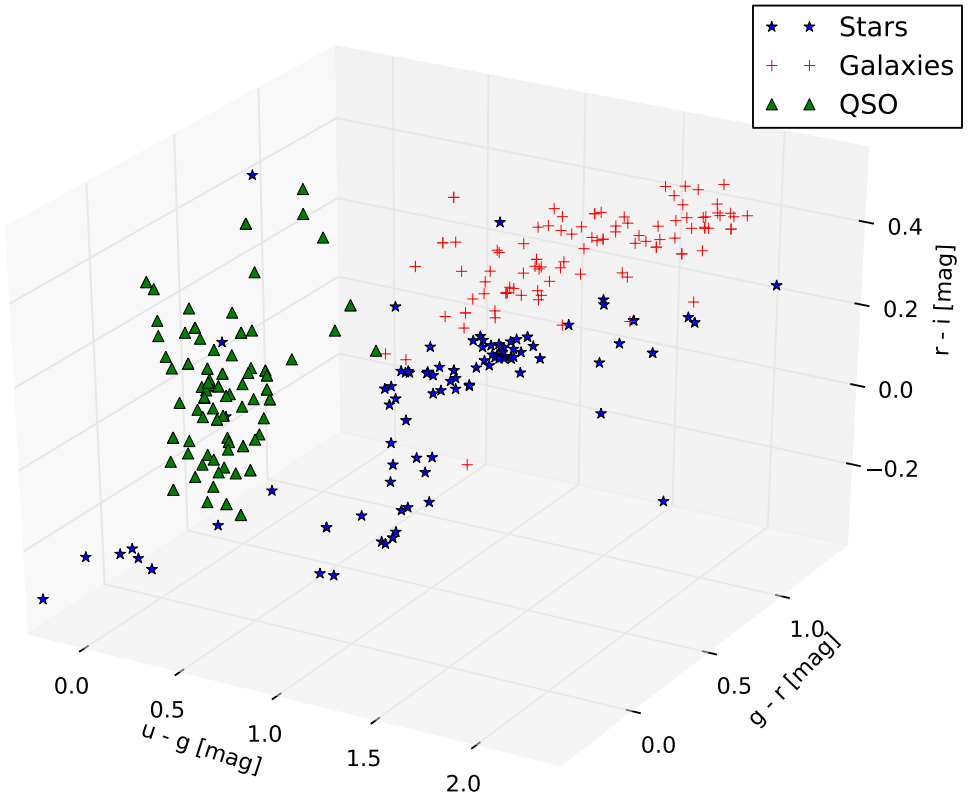


Figure 2.2: Color Diagram of the problem.It shows that individual object classes occupy different regions in the diagram

```
Dataset                 (1) trees.J4 | (2) bayes (3) meta. (4) lazy. (5) lazy. (6) rules
--------------------------------------------------------------------------------
Galaxy-Star-QSO         (100)  93.02 |  87.98 *  94.40    94.19    94.69    33.56 *
STAR-B-BE               (100)  70.78 |  68.11 *  72.02    65.22 *  69.67    52.85 *
STAR-AB                 (100)  69.66 |  64.96 *  69.79    65.97 *  70.45    50.76 *
STAR-BE-O               (100)  99.28 |  72.39 *  93.86    85.28 *  96.71    56.92 *
--------------------------------------------------------------------------------
                              (v/ /*) |  (0/0/4)  (0/4/0)  (0/1/3)  (0/4/0)  (0/0/4)
Key:
(1) trees.J48 '-C 0.25 -M 2' -217733168393644444
(2) bayes.NaiveBayes '' 5995231201785697655
(3) meta.RotationForest '-G 3 -H 3 -P 50 -F \"unsupervised.attribute.PrincipalComponents -R 1.0 -A 5 -M -1\" -S 1 -I 10 -W trees.J48 -- -C 0.25 -M 2' -3255631880798499936
(4) lazy.IBk '-K 1 -W 0 -A \"weka.core.neighboursearch.LinearNNSearch -A \\\"weka.core.EuclideanDistance -R first-last\\\"\"' -3080186098777067172
(5) lazy.KStar '-B 20 -M a' 332458330800479083
(6) rules.ZeroR '' 48055541465867954
```

2.2 Existing Projects

There are many open source projects related to Machine Learning and Data Mining. I would like to mention those which were used during experiments related to this work.

2.2.1 Weka

Weka is a collection of machine learning algorithms developed at University of Waikato, New Zealand. It includes functions for preprocessing, clustering, classification, regression, visualization, and feature selection. Originally designed as a tool for analyzing data from agricultural domains become extremely popular in data mining community because of its quality, openness, perfect documentation, and multi-platform implementation. Weka can be obtained at http://www.cs.waikato.ac.nz/~ml/weka/

2.2.2 SVM lib

Is library implementing Support Vector Machine with following properties

- Different SVM formulations
- Efficient multi-class classification
- Cross validation for model selection
- Probability estimates
- Various kernels (including precomputed kernel matrix)
- Weighted SVM for unbalanced data
- Both C++ and Java sources
- GUI demonstrating SVM classification and regression
- Python, R, MATLAB, Perl, Ruby, Weka, Common LISP, CLISP, Haskell, and LabVIEW, interfaces. C# .NET code and CUDA extension is available. It's also included in some data mining environments: RapidMiner and PCP.
- Automatic model selection which can generate contour of cross valiation accuracy.

The project is hosted on http://www.csie.ntu.edu.tw/~cjlin/libsvm/. The part of this project is useful and well written practical guide to SVM classification.

2.2.3 DAME

Is great example of full understanding of the paradigm shift in astronomy. The project implements Neural Networks and Support Vector Machines algorithms and it is VO-compatible. The documentation includes scientific use cases, masters and PhD thesis and lectures. As it is typical in these projects it exceeds its original domain of astronomical data into general platform. The projects is available at http://dame.dsf.unina.it/

Chapter 3

Be candidates

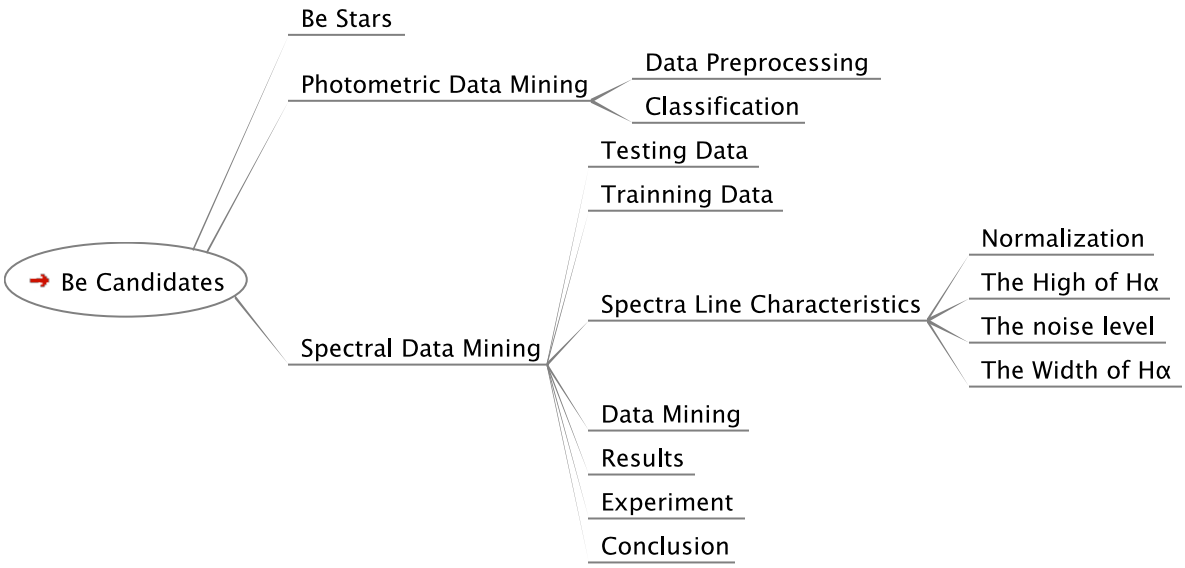


Figure 3.1: Chapter structure

Astronomical objects used in this work to demonstrate some of the discussed technologies and methods were Be stars. The goal was to develop a process of finding new candidates in the available data. Several approaches were considered and two of them are discussed in the rest of this text. First one utilizes photometric properties of Be stars, second uses spectra characteristics.

3.1 Be stars

The first example of Be star was reported by Padre Angelo Secchi in his letter to the Astronomische Nachrichten in 1866.

Classical Be stars are non-supergiant B-type stars whose spectrum has or had at some time, one or more Balmer lines in emission. The current accepted explanation of this phenomena is circumstellar gaseous component in the form of equatorial disk. Rapidly rotating central star is important feature of these objects, which may be important contributor of the circumstellar medium [Porter and Rivinius, 2003].

The important characteristic used later in the work is the Hα emission. The explanation of its origin is on following pictures.

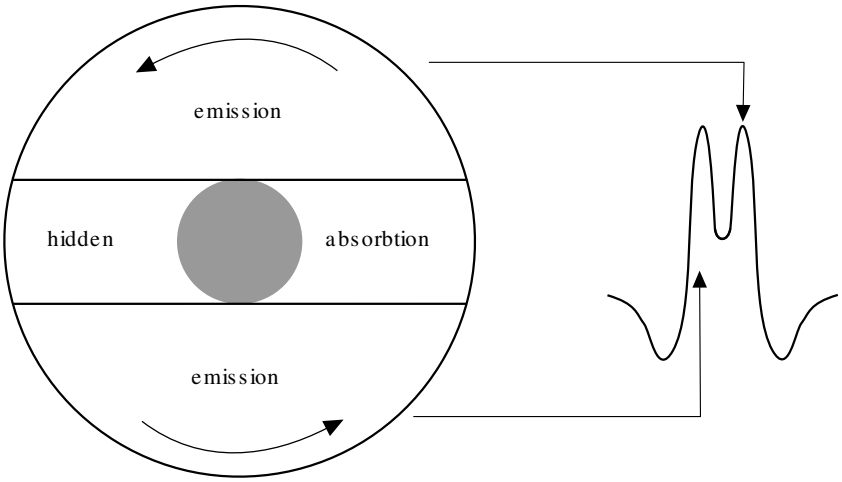


Figure 3.2: Model of a typical Be star. Emission lines coming from an equatorial disk is added to the photo-spheric absorption spectrum. Central B star emits UV (Lyman continuum) and ionizes the disk, which in turn re-emits at high wavelength such as visible domain. [Hirata and Kogure, 1984]

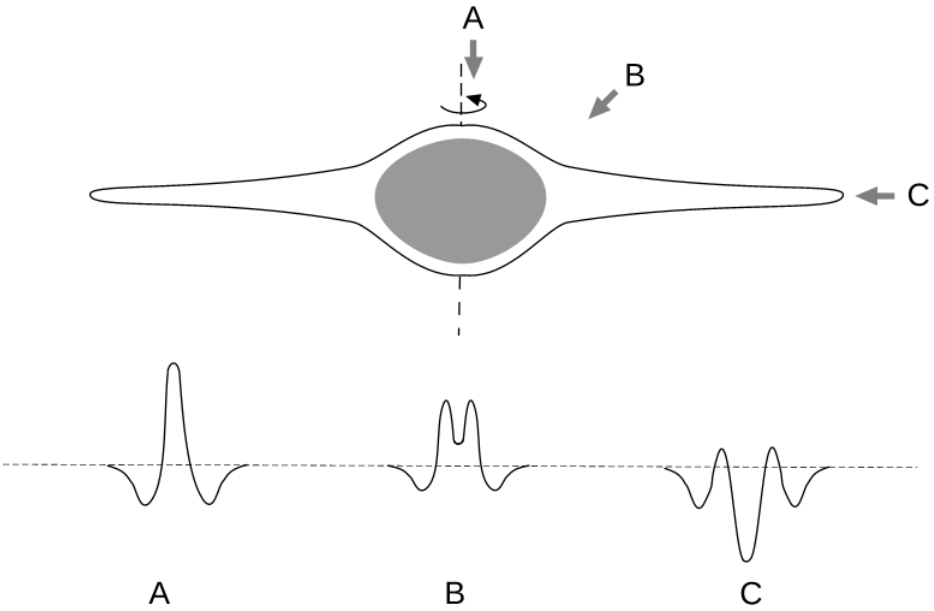


Figure 3.3: Example of spectra of Be stars based on view angle [Slettebak, 1988]

There are still many open questions related to their rotation, evolutionary status, presence and origin of the magnetic fields, mass and angular momentum transfer and others, therefore the process for automatic discoveries of Be phenomena in the digitalized surveys and obtaining new candidates could help answer these questions.

3.2 Photometric Data Mining

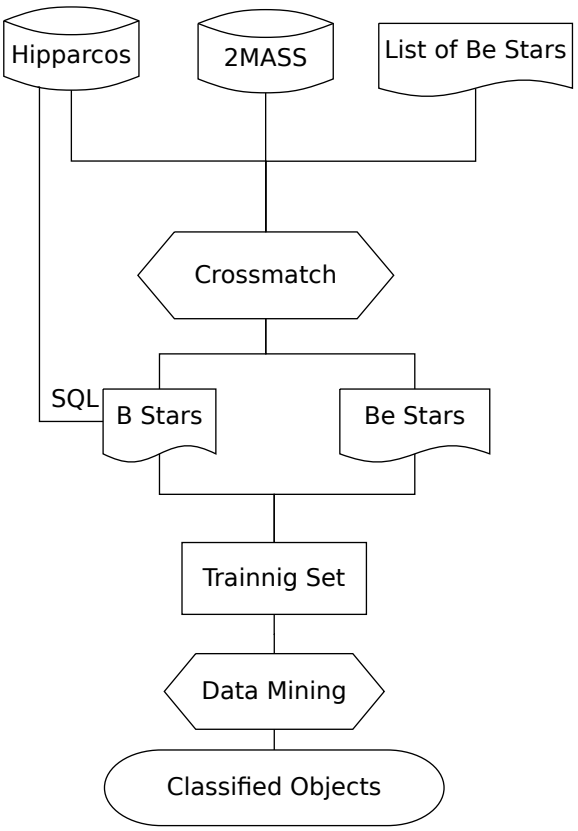


Figure 3.4: Schematic diagram of the photometric Data Mining process. The lists of confirmed Be stars consisted of Hipparcos IDs, this was correlated with Hipparcos catalog to obtain right ascension and declination of the objects and subsequently cross-matched with 2MASS catalog to get photometric data. The second set of B stars were acquired in similar manner but using SQL the condition was set to get B type stars different from the list of Be stars.

Classification based on photometric properties is very attractive from several points of view. There are much more available photometric then spectral data and they are easier accessible. Because they are easier to gain the disproportion between photometric and spectral data will probably increase in the future as well. The distinction between Be and other types of stars also should be theoretically possible since the Be stars exhibits infrared excess correlated to the Hα emission [Van Kerkwijk et al., 1995].

3.2.1 Data preprocessing

I was provided with a list of confirmed Be stars from Academy of Science Ondejov. This list consist of 625 manually chosen objects. Data were correlated with Hipparcos [?] catalog to obtain RA, DEC and then with 2MASS[?] catalog to obtain J,H,K Colors using method of multi-cone search in Virtual Observatory. The second set was acquired from Hipparcos catalog using following SQL query:

```
1  Select *
2  From maincat as m, hipva1 as h
3  Where (m.HIP=h.HIP )
4  And h.SpType Like 'B%'
```

The result was cross-correlated with 2MASS catalog to obtain the same colors as for the confirmed Be stars. Color digram of this two sets are on the figure 3.5

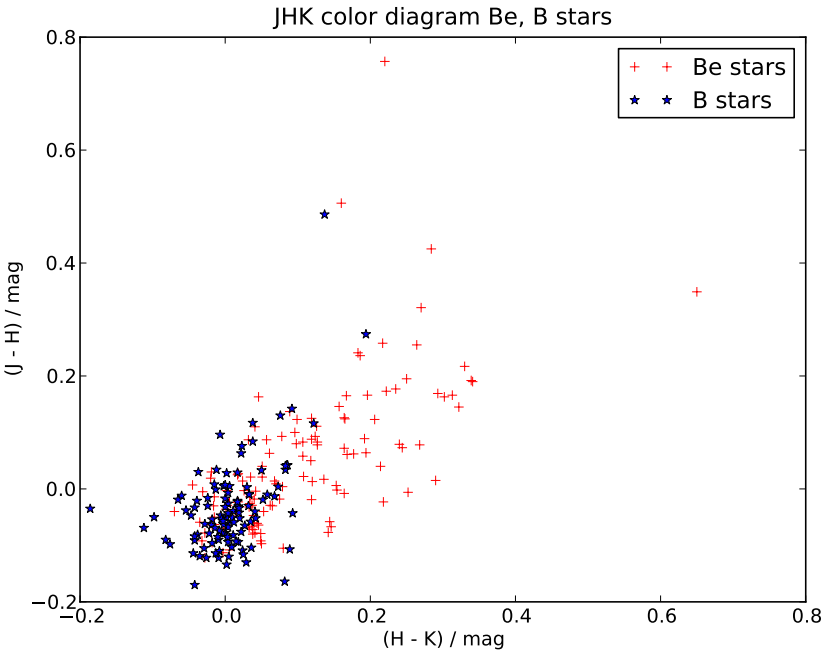


Figure 3.5: Color diagram of confirmed Be stars Vs B stars

The uncertainties were computed for each object using propagation of error. These errors and depicted on the figure 3.6. Although the uncertainties are significant certain trends are presented.

$$\delta_{(j-h)} = \sqrt{\left(\frac{\partial(j-h)}{\partial j}\right)^2 \delta_j^2 + \left(\frac{\partial(j-h)}{\partial h}\right)^2 \delta_h^2}$$

$$\frac{\partial(j-h)}{\partial j} = 1, \frac{\partial(j-h)}{\partial h} = -1$$

$$\delta_{(j-h)} = \sqrt{\delta_j^2 + \delta_h^2}$$

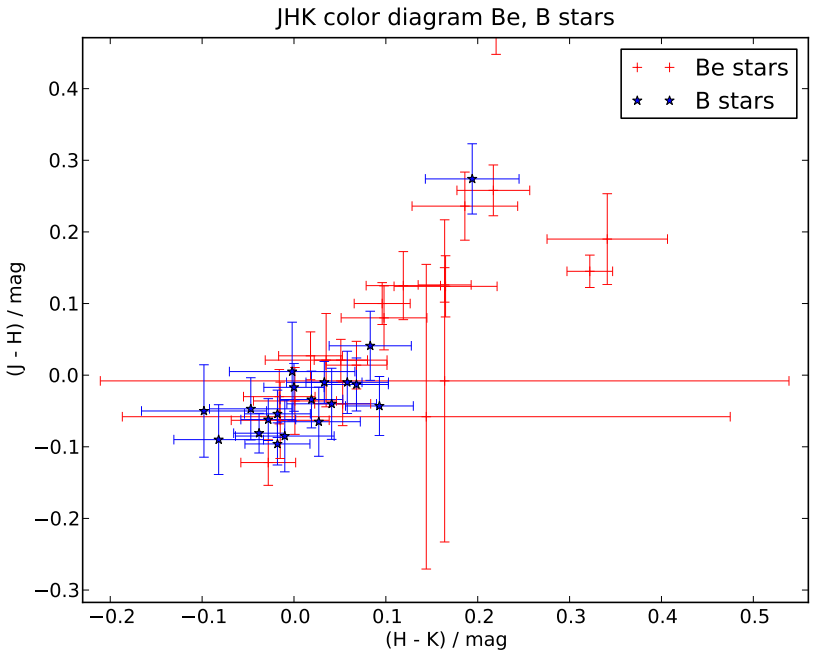


Figure 3.6: Color diagram of confirmed Be stars Vs B stars with errors

3.2.2 Classification

Data were transformed from original VOTable obtained from Virtual Observatory tools to arff[1] format used in Weka Data Mining system. Algorithm C4.5 (J48) was used to perform actual classification with following result:

```
Correctly Classified Instances         769               73.0989 %
Incorrectly Classified Instances       283               26.9011 %
Kappa statistic                          0.4496
Mean absolute error                      0.3843
Root mean squared error                  0.4383
Relative absolute error                 79.4985 %
Root relative squared error             89.1648 %
Total Number of Instances             1052
```

As seen on the first row 73 % from 1052 objects were classified correctly. More details can be obtained from confusion matrix below.

```
  B   Be   <-- classified as
 304 126 |  B
 157 465 |  Be
```

304 of B and 456 of Be stars were classified correctly but 126 of B and 157 of Be stars were classified incorrectly. In virtue of these results one should be sceptical if the distinction based only on photometric properties is significant enough to find relevant new candidates of Be stars. For this reason more sophisticated (and much more complicated) approach using spectra analysis was tested.

3.3 Spectral Data Mining

[1] Attribute-Relation File Format. Developed by the Machine Learning Project at the Department of Computer Science of The University of Waikato.

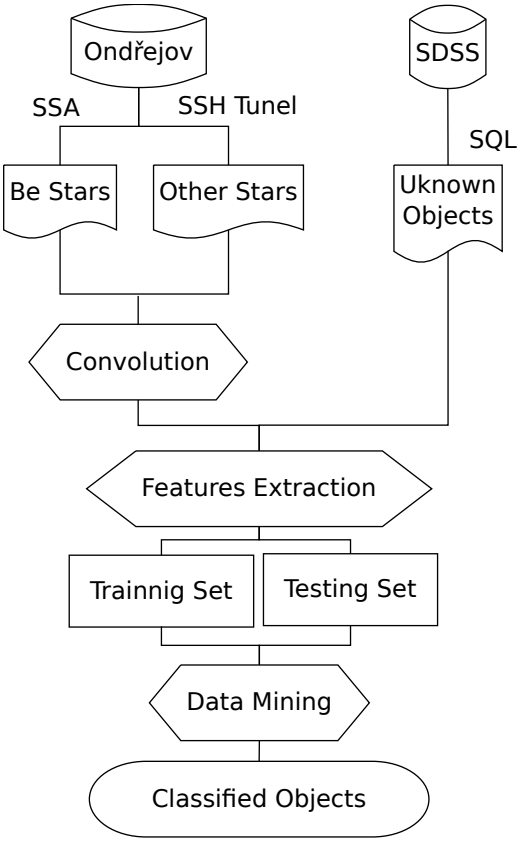


Figure 3.7: Schematic diagram of the spectral Data Mining process. Using SSA protocol the spectra from Ondejov server was acquired based on the list from photometric study. SSH Tunneling was necessary since Ondejov spectra are top secret and therefore not available to the public. Convolution had to be performed to ensure compatibility with SDSS. Afterwards desired features were extracted automatically from the spectra after the continuum and Hα line were fitted by appropriate functions. The same was done for spectra from SDSS except the convolution process.

Spectra provide much wider scientific informations over photometric properties. Spectral lines exhibits many distinguish features and astronomers have long tradition of analysing their properties. On the other side its much complicated to handle them because of different characteristics (resolution, calibration, wavelength range, etc). This is especially true for massive automated processing.

3.3.1 Testing Data

As testing sample the project SEGUE of SDSS were selected. This contains 178315 spectra in DR7. Following SQL query was used to generate the list of URL links for individual FITS files. These files were then downloaded to local sever using wget command.

```
SELECT objid,dbo.fGetUrlFitsSpectrum(s.specObjID)
INTO mydb.segue_1
FROM SpecPhotoAll s, platex p
WHERE s.specObjID is not null
AND s.plateid = p.plateid
AND p.programname LIKE 'segue%'
AND specClass = 1
```

3.3.2 Training Data

The spectra from Ondejov Observatory were used as a training sample. Files were downloaded using SSA protocol. The SSA server is not publicly available, therefore SSH tunneling was used. Two scripts for this process were created. First to construct the list of SSA compliant addresses, the second to analyse acquired response in VOTable format. Then the spectra were downloaded using wget command. The function for constructing the links based on list of the RA, DEC which were obtained from Hipparcos catalog using the specification of IDs from Ondejov's index.

```
def createQuery(data):
    """ From raw data construct ra, dec """
    """ Convert to degrees """
    for line in data:
        ra = ac.AngularCoordinate(line[0:10]).degrees # convert ra to
            degrees
        dec = ac.AngularCoordinate(line[-13:-1]).degrees # convert dec to
            degrees
        ra = line[0]
        dec = line[1]
        ssaTemp = 'http://tvoserver/coude/coude.cgi?c=ssac&n=coude_ssa&
            REQUEST=queryData&POS=<ra>,<dec>&SIZE=1'
        ssaTemp = ssaTemp.replace('<ra>',"%0.3f" % ra)
        ssaTemp = ssaTemp.replace('<dec>',"%0.3f" % dec)
        ssa.append(ssaTemp)
    return ssa
```

The script generate the following output. The same process were used later for obtaining th sample of non Be stars.

```
http://tvoserver/coude/..._ssa&REQUEST=queryData&POS=83.113,-65.582&SIZE
    =60
http://tvoserver/coude/..._ssa&REQUEST=queryData&POS=162.537,148.333&SIZE
    =60
http://tvoserver/coude/..._ssa&REQUEST=queryData&POS=19.907,-73.502&SIZE
    =60
```

3.3.2.1 Spectra Reduction

Because spectra from SDSS and Ondejov Observatory had different resolution, reduction was needed. First the parameter CD1_1 (Coordinate increment per pixel) had to be obtained form FITS file.

```
In [1]: hdu = pf.open('sdss_test.fit')
In [2]: hdu[0].header['CD1_1']
Out[2]: 0.0001 # SDSS spectrum
Out[3]: 0.2567 # Onejov spectrum
```

Spectra in SDSS are stored in logarithmic scale thus the value is computed as $10^{CD1\_1} = 1.00$. The ratio is then $CD1\_1_{SDSS}/CD1\_1_{OND} = 3.87$. Based on this computation 4 pixels of Ondejov's spectra were reduced into one. There is the critical part of the reduction program:

```
def convolution(f, g):
   """ Convolve two functions"""
   fg = np.convolve(g,f,'same')
   return fg
def reduce(x,y,bin):
   """ Reduce bin pixel into 1"""
   size = x.size/bin
   l = 0
   xx = x[:x.size-1:bin]
   yy = list()
   for i in range(0,size):
       s = 0
       for j in range(0,bin):
           s = s + y[l]
           l+=1
       yy.append(s/bin)
   return xx, yy
```

Prior to binning pixels convolution with Gaussian function was performed on the spectra. Convolution is defined:

$$(f * g)(t) \stackrel{\text{def}}{=} \int_{-\infty}^{\infty} f(\tau)\, g(t-\tau)\, d\tau \tag{3.1}$$

Here it was used in it's discrete form

$$(f * g)[n] \stackrel{\text{def}}{=} \sum_{m=-\infty}^{\infty} f[m]\, g[n-m] \tag{3.2}$$

The figure shows the result.

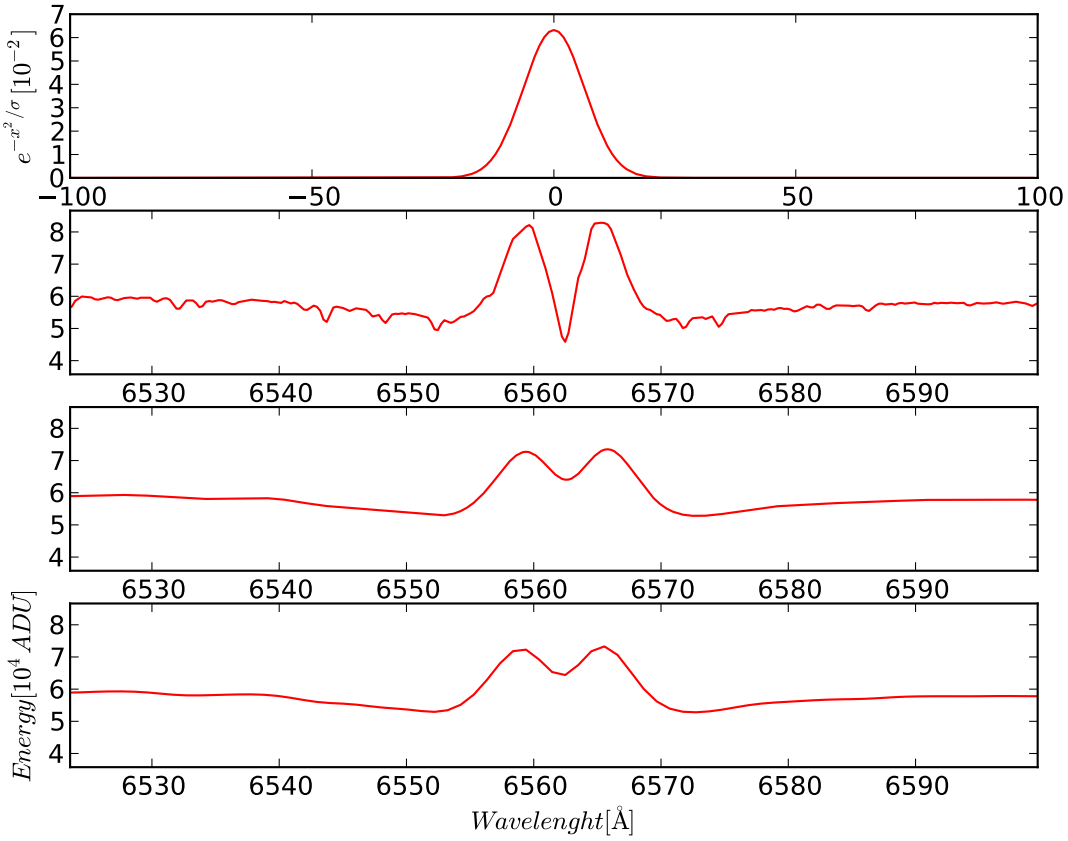


Figure 3.8: Reduction of Ondejov's spectra of the Be star 4 Hercules. The top figure shows Gaussian function used for convolution with the spectrum, followed by the original spectrum then there is a spectrum after convolution with the Gaussian function. The last is the final spectrum after reduction.

3.3.3 Spectra Lines Characteristics

As parameters for Data Mining process characteristic values of Hα line were extracted from the spectra. Many possible characteristics from fitting functions through Wavelets Coefficients and Eigenvalues Values were discussed with experts. Three parameters were finally selected. The hight and the width of the Hα emission line and median absolute deviation as a characterization of the noise level in the spectrum.

3.3.3.1 Normalization

Spectra from SDSS are normalized but the spectra from Ondejov are not. The spectra were divided by it's continuum fit function. This process ensures the compatibility when comparing different spectra. Function polyfit from numpy package was used to perform the fit. The solution minimizes the squared error:

$$\frac{d}{dq}\sum_{i=1}^{n}(y_i - \mathrm{f}(x_i))^2 = 0, \tag{3.3}$$

where f is in our case $\mathrm{f}(x) = q_1 x + q_0$.

3.3.3.2 The hight of the Hα line

The maximum value in the region of 50$\mathring{A}$ were extracted from the spectrum.

```
1  def getMax(x,y,line,range):
2      """ Return maximum value of range in the spectrum"""
3      xrange = x[(x < line + range) & (x > line - range)]
4      yrange = y[(x < line + range) & (x > line - range)] - 1
5      maximum = yrange.max()
6      minimum = yrange.min()
7      if abs(maximum) > abs(minimum):
8          extrem = maximum
9      else:
10         extrem = minimum
11     return xrange, extrem, sgn
```

3.3.3.3 The noise level of the spectrum

The noise in the spectrum contributes to the characteristics of the spectral lines. As an estimator of the noise level the median absolute deviation was used. It is defined as:

$$\mathrm{mad} = \mathrm{median}_i\,(\,|X_i - \mathrm{median}_j(X_j)|\,) \tag{3.4}$$

3.3.3.4 The width of the Hα line

The Gaussian function was fitted to the spectral line. First the robust estimators were computed and used as input parameters for leastsq[1] method from scipy.opt module, which minimize the sum of squares.

[1] "leastsqi"s a wrapper around MINPACKs lmdif and lmder algorithms.

$$x_0 = \frac{\text{median}(w_j x_j)}{\sum w_i}, \tag{3.5}$$

$$S = \frac{\text{mad}(x_i - x_0)}{\sum w_i}. \tag{3.6}$$

[Launer, 1979]

Part of the script implementing fitting the Gaussian function

```
x0 = np.median(sum(w*x))/sum(w)
S = sum(w*mad((x - x0)))/sum(w)
params = np.array([1, maximum, x0, S], dtype=float)
fit, flag = opt.leastsq(residuals, params, args=(yrange, xrange))
gauss = model(xrange, fit) + 1

def model(t, coeffs):
    return coeffs[0] + coeffs[1] * np.exp( - ((t-coeffs[2])/coeffs[3])**2
        )
def residuals(coeffs, y, t):
    return y - model(t, coeffs)
```

The final result is on the figure **??** and **??**. The script was adjust to work with SDSS and Ondejov's spectra. The whole procedure was performed on all of the cca 200 000 SDSS spectra and few dozens Ondejov's spectra resulting with the ASCII files with the characteristic values used later in Data Mining process.

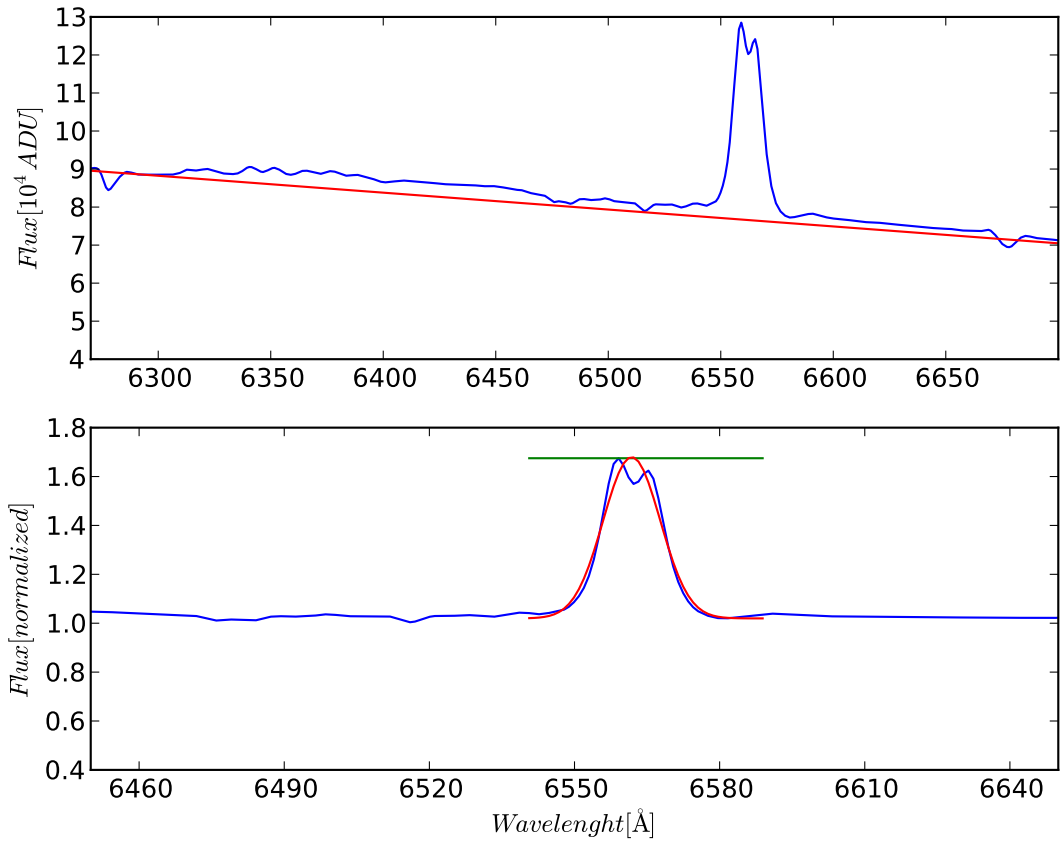


Figure 3.9: Normalized spectrum of Be star 60 Cyg. The top figure depicts the continuum fit. The bottom figure shows the region (width of the green line) used for extraction. The position of the line correspond to the maximum value in the region of 50$\mathring{A}$. The Gaussian fit is in red. Although the fit is almost perfect, this approach fails to get characteristic "double peak" of the emission line.

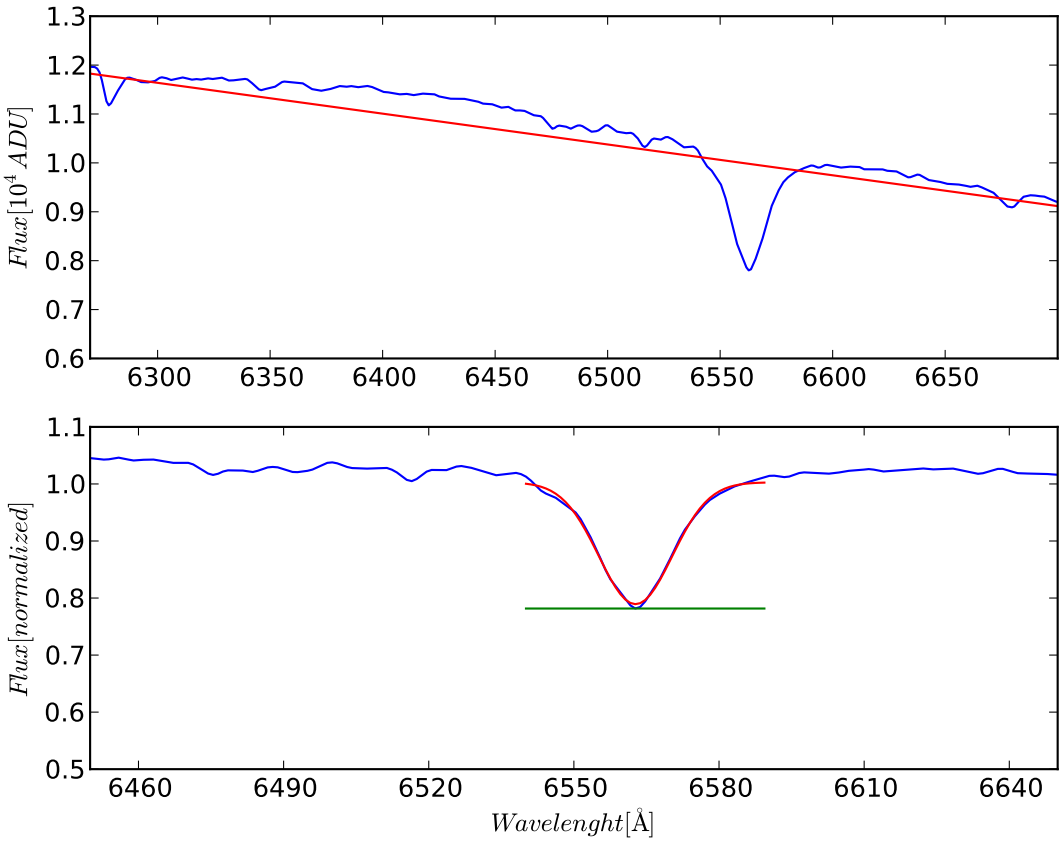


Figure 3.10: Normalized spectrum of Be star HR 8682. The top figure depicts the continuum fit. The bottom figure shows the region (width of the green line) used for extraction. The position of the line correspond to the maximum value in the region of $50\mathring{A}$. The Gaussian fit is in red.

3.3.4 Data Mining

Classification was performed using Weka software with algorithm J48 described in the chapter 2. Training set had 173 and testing set 178314 items. The excerpt from these files follows.

```
@RELATION STAR-B-BE
@ATTRIBUTE name STRING
@ATTRIBUTE alpha NUMERIC
@ATTRIBUTE grp {be,o}
@DATA
10_cas,-0.822196556626,be
11_cyg,1.68689566629,be
```

```
@RELATION STAR-B-BE
@ATTRIBUTE name STRING
@ATTRIBUTE alpha NUMERIC
@ATTRIBUTE grp {be,o}
@DATA
spSpec-53228-1884-001 -0.584628294569 ?
spSpec-53228-1884-002 -0.877184482566 ?
```

The attribute grp is known for the training set but unknown for testing set. The classification process fills this information based on decision tree created during learning phase. To automate the process command line version of Weka software was used.

```
  java -classpath weka.jar
  weka.classifiers.meta.FilteredClassifier -F
  weka.filters.unsupervised.attribute.RemoveType -W
  weka.classifiers.trees.J48 -t $1 -T $2 -p 1
```

3.3.5 Results

The overall fruitfulness of the classification process is almost 84%. 10 folds cross-validation was used to compute the error rate.

```
  === Summary ===
Correctly Classified Instances     145               83.815 %
Incorrectly Classified Instances    28               16.185 %
Kappa statistic                      0.6529
Mean absolute error                  0.1849
Root mean squared error              0.3652
Relative absolute error             39.8819 %
Root relative squared error         75.8919 %
Total Number of Instances          173
```

The classification tree is relatively complicated. But still we can learn few things. It is using all of the parameters putted in so they are choose correctly (if they were irrelevant classifier would not used them). The most important parameter was max which determines the hight of the line above the continuum. This was expected as Hα emission is dominated feature of Be stars. The second important parameter

was the noise of the spectrum expressed in parameter mad. The less important (at least in this example) was the width of the line. It need to be emphasized that the parameter max does not really measure the hight, mad the noise or width the width of the line but there are some simplified (and certainly buggy) versions of real parameters. Though it can give us some physical insight of the studied phenomenas. Decision trees are therefore very powerful compared to black box approaches such are Neural Networks where the classification process is beyond human understanding.

```
  J48 pruned tree
------------------
max <= -0.18843
|   max <= -0.324763: o (46.0/5.0)
|   max > -0.324763
|   |   max <= -0.255475
|   |   |   mad <= 0.004133: o (2.0)
|   |   |   mad > 0.004133: be (13.0/1.0)
|   |   max > -0.255475
|   |   |   mad <= 0.009862: o (10.0)
|   |   |   mad > 0.009862
|   |   |   |   width <= 7.621593: o (3.0/1.0)
|   |   |   |   width > 7.621593: be (2.0)
max > -0.18843
|   mad <= 0.030316
|   |   max <= -0.091726
|   |   |   width <= 5.286489
|   |   |   |   max <= -0.170022: be (2.0)
|   |   |   |   max > -0.170022: o (3.0)
|   |   |   width > 5.286489: be (9.0)
|   |   max > -0.091726: be (76.0)
|   mad > 0.030316
|   |   max <= 6.917615: o (4.0)
|   |   max > 6.917615: be (3.0)
```

```
  === Confusion Matrix ===
 Be Others  <-- classified as
 95 15   | Be
 13 50   | Others
```

The Confusion Matrix evince that the classifier is more successful in assigning Be stars (95/15) than in the case of others types stars where 13/50 were associated with wrong class.

Spectra of some of the objects classified as Be stars are presented here. These samples represent tiny fraction of complete result. The program for generating web pages with thumbnails were created and full result is available on the Wiki pages of this project

http://physics.muni.cz/~vazny/wiki/index.php/Diploma_work.

For comparison there are spectra of know Be stars. It is clear that the profile of the Hα line is complex and just one parameter cannot possibly express it's characteristic. More advanced description such as Wavelets coefficients or theoretical models of the line is needed if we want to create reliable process for identifying Be stars.

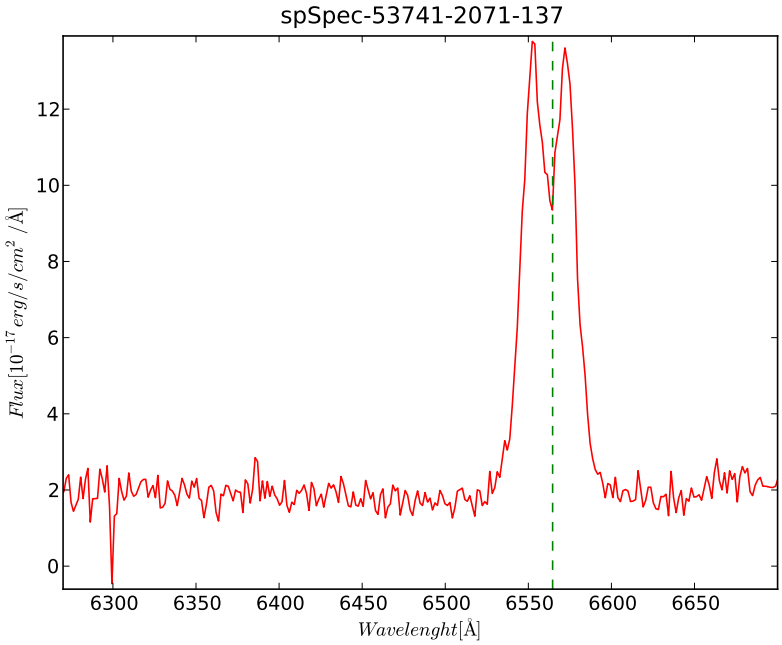


Figure 3.11: Spectrum of

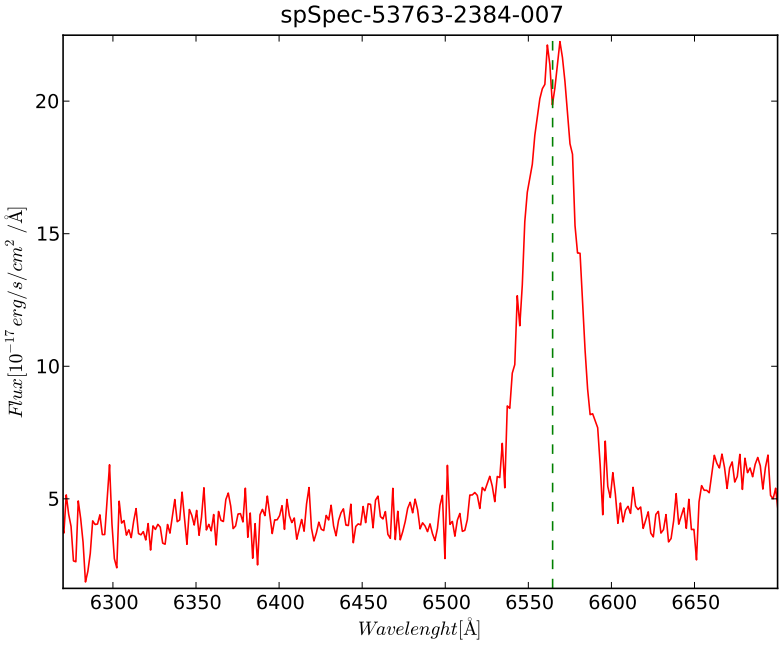


Figure 3.12: Spectrum of

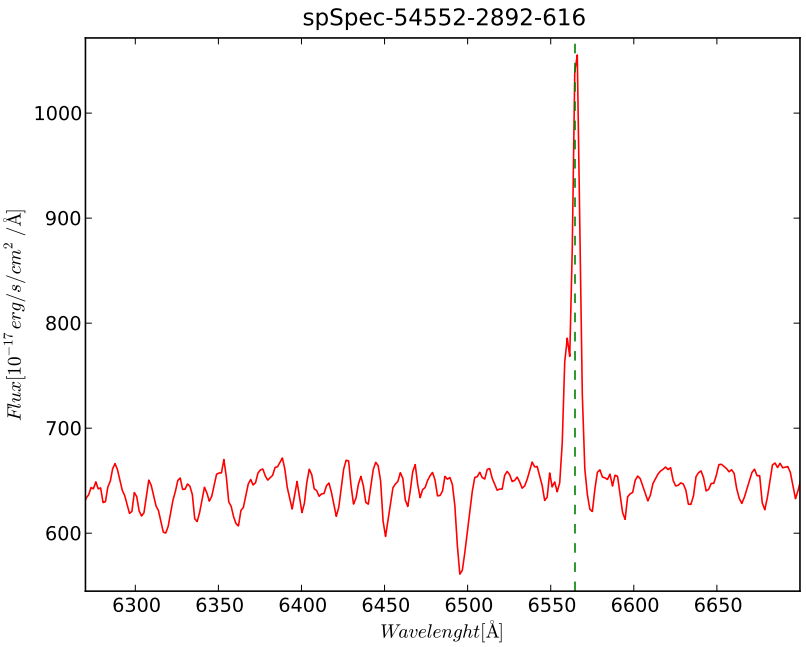


Figure 3.13: Spectrum of

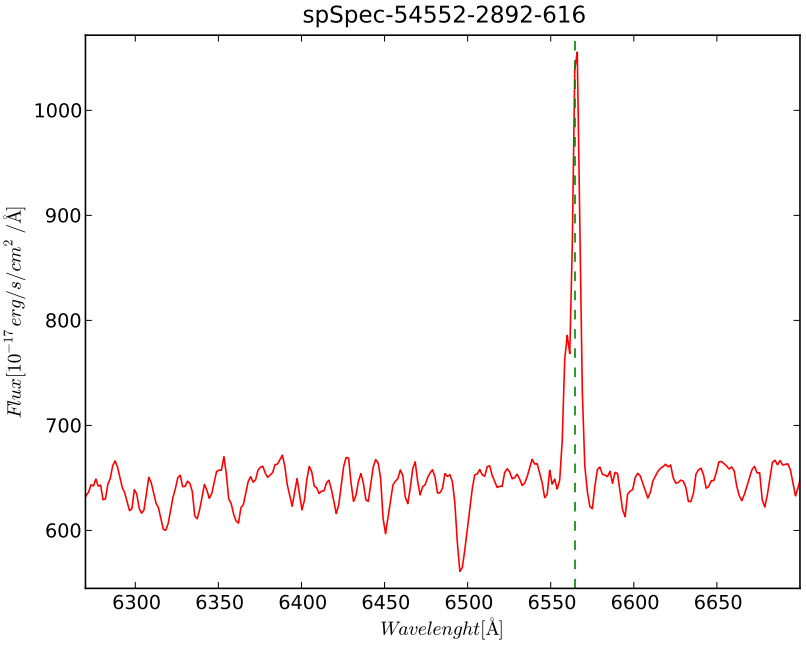


Figure 3.14: Spectrum of

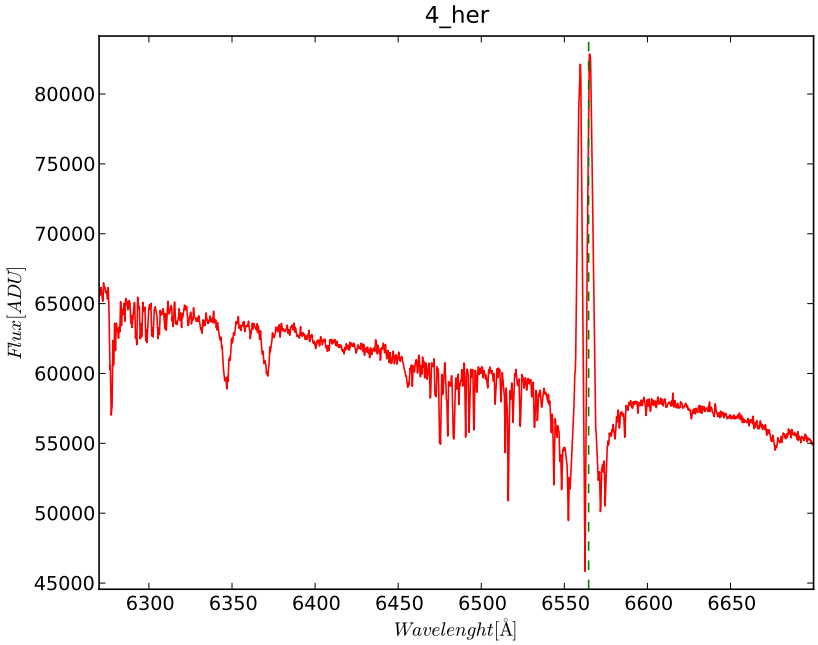


Figure 3.15: Spectrum of 4 Her. Be star. Spectral Type B9pe.

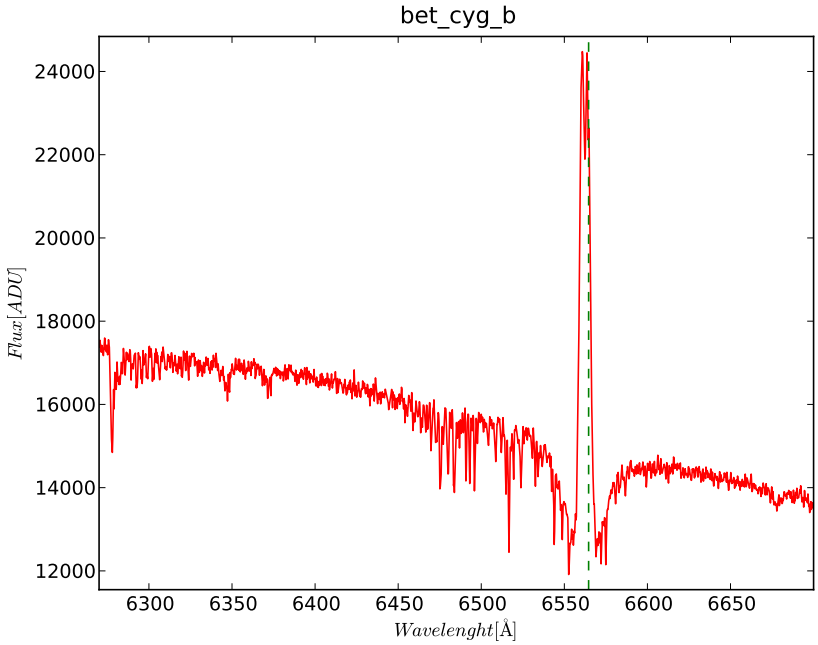


Figure 3.16: Spectrum of HR 7418 (Albireo B). A fast-rotating Be star, with an equatorial rotational velocity of at least 250 kilometers per second. Its surface temperature has been spectroscopically estimated to be about 13.200 K. Spectral Type B8Ve.

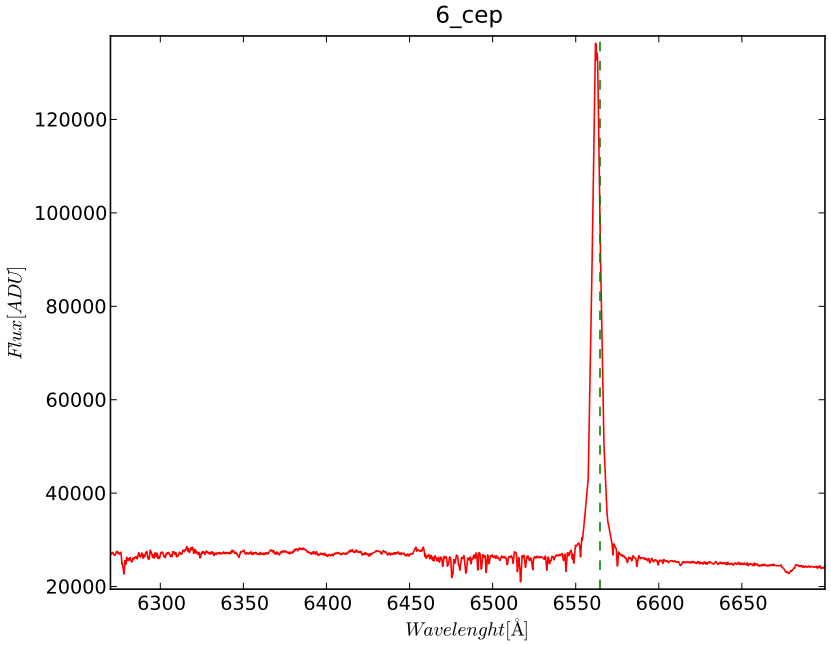


Figure 3.17: Spectrum of 6 Cepheus. Be star. Spectral Type B3IVe.

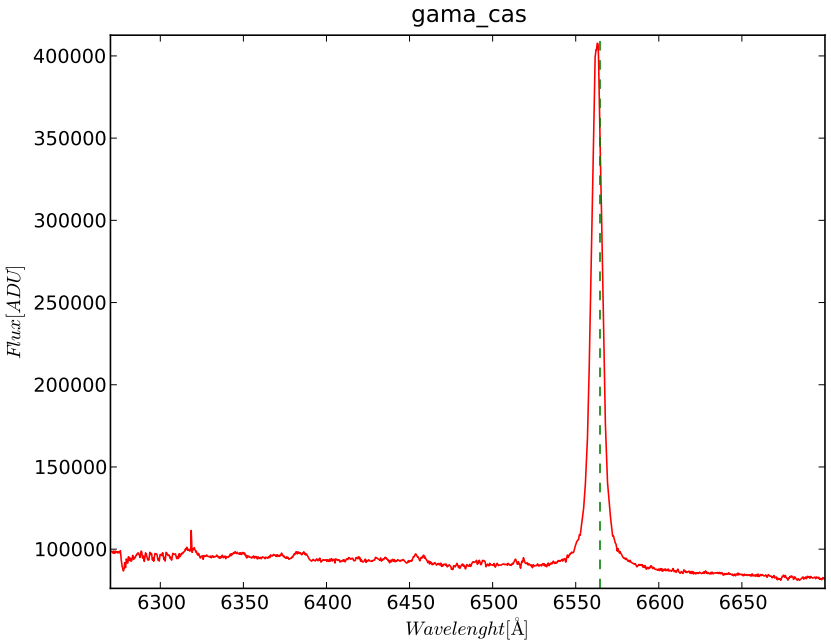


Figure 3.18: Spectrum of Gamma Cassiopeiae. Be Star. Spectral Type B0IVpe C.

#	SDSS name	RA	DEC	u	g	r	i
1	SDSS J035747.16-063850.7	59.44	-6.64	19.83	19.99	19.73	19.86
2	SDSS J094325.89+520128.6	145.86	52.02	16.57	16.42	16.55	16.70
3	SDSS J120729.12+003659.8	181.87	0.62	17.57	15.28	14.30	13.96
4	SDSS J120908.18+194035.8	182.3	19.7	17.87	16.26	15.52	15.19

Table 3.1: Examples of the result

#	link
1	http://cas.sdss.org/dr7/en/tools/explore/obj.asp?sid=583165493179842560
2	http://cas.sdss.org/dr7/en/tools/explore/obj.asp?sid=671267254834298880
3	http://cas.sdss.org/dr7/en/tools/explore/obj.asp?sid=814259934286839808
4	http://cas.sdss.org/dr7/en/tools/explore/obj.asp?sid=814541407275450368

Table 3.2: Links to objects on SDSS Skyserver

3.3.6 Experiment

One could be interested what would happened if we have choose different parameters, used other algorithm, different training set etc. These are perfectly valid questions and it is actually the purpose and essence of Data Mining and computers in general: perform similar task over and over again. With some automation in mind such experiments are easy to do. Here is one of the test I have performed.

Natural idea one could have is using just emission line hight and write a program to check the spectra for condition if emission ¿ threshold. Then we do not need "expensive" Data Mining algorithms to get some interesting results[1]. Lets try to use Classification using just emission line parameter. Here is consequent tree.

```
1  J48 pruned tree
2 ------------------
3 alpha <= -0.464633
4 |   alpha <= -0.676474: be (45.0/18.0)
5 |   alpha > -0.676474: o (46.0/5.0)
6 alpha > -0.464633: be (92.0/16.0)
```

We can see that it is not that simple. In the training set there are some Be stars with extreme hight in the negative direction (absorption line). We have some statistics from classifier. In this example the effectiveness were 77.6%. Also there is Confusion Matrix

```
1  === Confusion Matrix ===
2  Be    O   <-- classified as
3  102   6 |  Be
4   35  40 |  Others
```

which indicates how the classifier will fail to perform in individual cases. Here it shows that it is almost exact when the object is Be star but it confuses others type of stars with Be stars.

This experiment proves that using Data Mining has a sense even in simple cases and can provide non-trivial insight on tested data.

3.3.7 Conclusion

It is evident that dealing with spectroscopic data is much more complicated but also more fruitful. One could extract many characteristic features which fit to the actual problem. The FITs standard is real godsend and makes work with spectra from different sources possible. The results obtained from the Data Miming process are reasonable. During the work it was also "discovered" how humans are good at visual judgment: when thumbnails of result spectra were created it provided much better understanding if something went wrong then statistics and numbers. This is one example how machines and humans could work together when we utilize ours and theirs natural abilities.

There are many aspect which could be done better, some of the considered but not implemented subjects are discussed here.

- Spectral Characteristics

[1] This was actually done in the early stage of this project.

The spectrum was characterized with few, very simple parameters, which can be similar in different types of objects. We have discussed [1] many advanced possibilities such are wavelets eigen values etc. This could be subject of further investigation.

- Continuum fit

  The simple linear function is too rough to capture true continuum features. There is an interesting and effective algorithm discussed in the paper: Advanced fit technique for astrophysical spectra by S. Bukvić et. al. from University of Belgrade [Bukvić et al., 2008] which seems ideal for this purpose.

- More Data Mining algorithms

  Originally more advanced approaches such are Support Vector Machines were considered.

- Larger training sample

  To obtain large enough meaningful training sample of confirmed Be stars was a real problem and many surveys were consider (e.g. IPHAS) but without success.

- Usage of Light curves

  The whole process static but the "Be phenomena" is dynamic in nature. Using light curves could significantly improve the efficiency.

- Unknown errors There are things we do not know we don't

  know. The overall process were very complicated and involved hundreds lines of codes. Any overlooked mistake could effect the results. The absence of evidence is not evidence of absence.

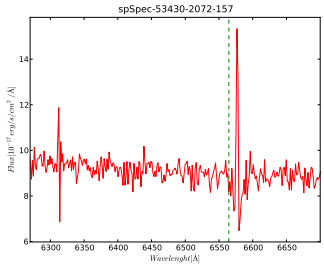
[1] Petr Skoda initiated rich and interesting email conversation with leading experts regarding this topic.

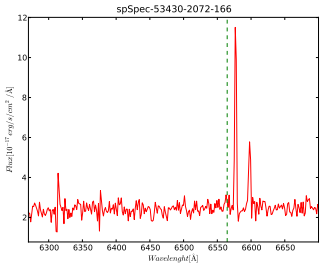
Chapter 4

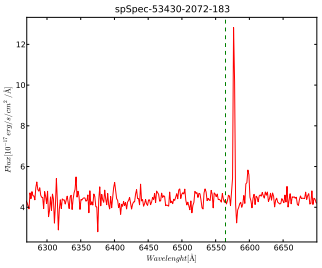
Conclusion

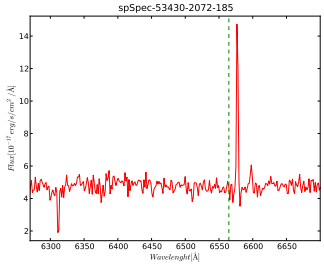
The harvesting of large-scale astronomical data is challenging but solvable problem. The technology of Virtual Observatory offers solid background for data discovery and retrieval. The whole process can be automated using UN*X like approach of small and single purpose scripts or programs. The last stage of choosing the right characteristics and Data Mining method is even more complex task requiring deep understanding of the researched phenomenas and Machine Learning theory and technology. The possible solution can be based on cooperation between experts in scientific and computer science field. Without such collaboration we are missing lots of opportunities.

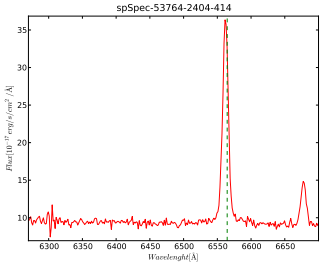
Appendix1: Spectra of result objects

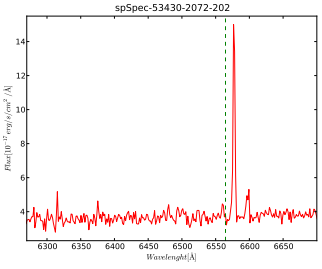


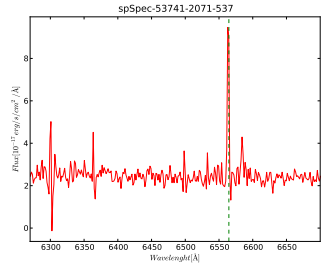


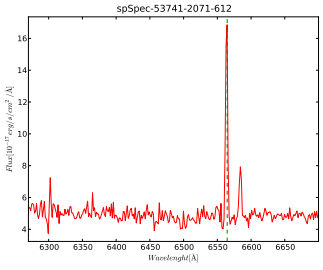


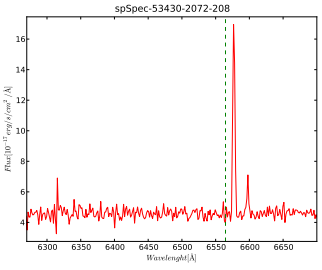




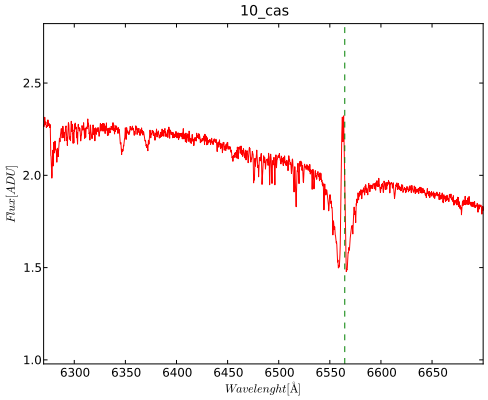


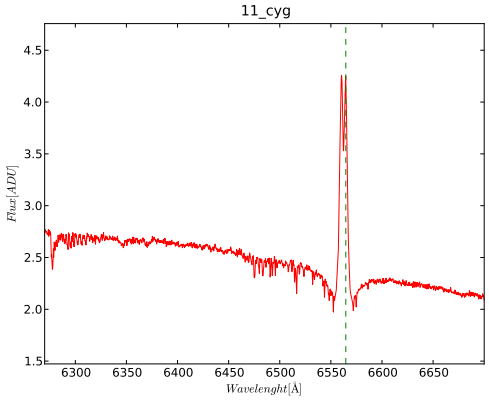


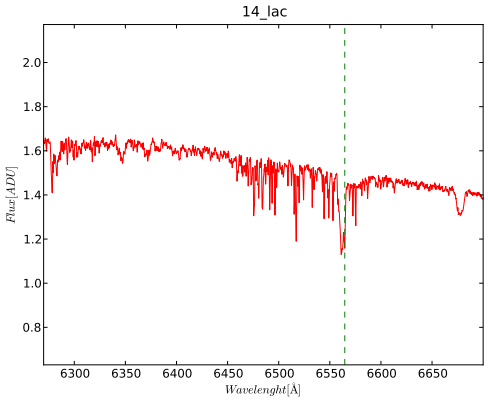


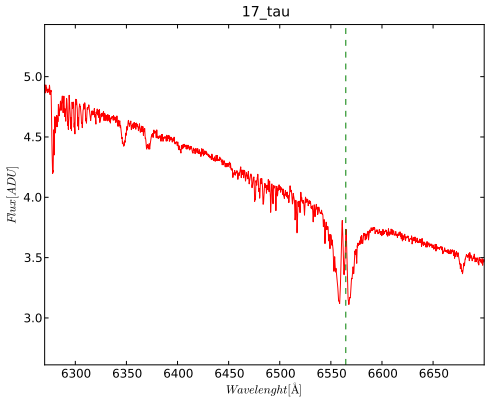


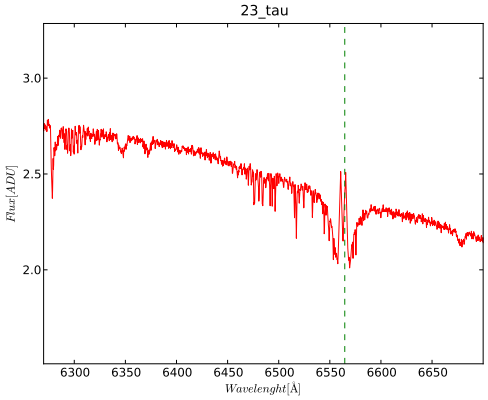
Appendix2: Spectra of Ondejov Be stars

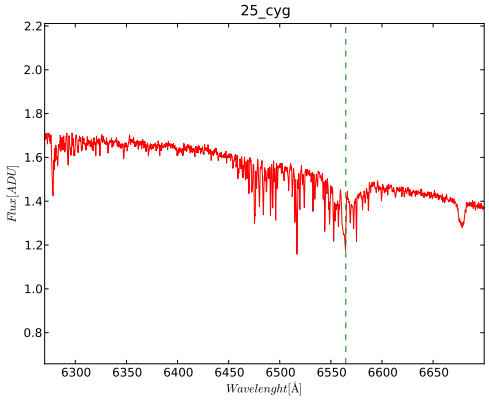


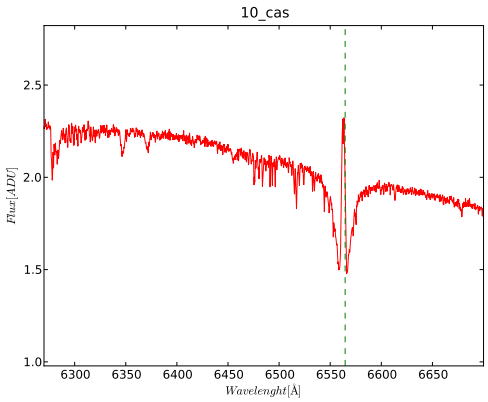


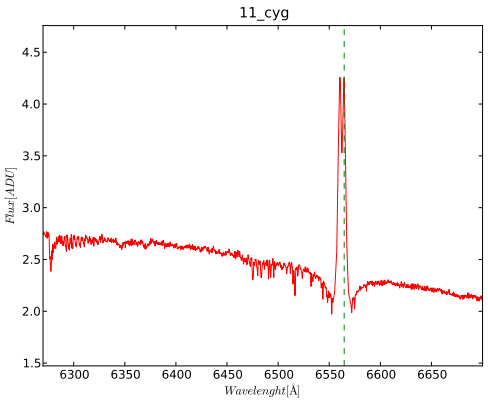


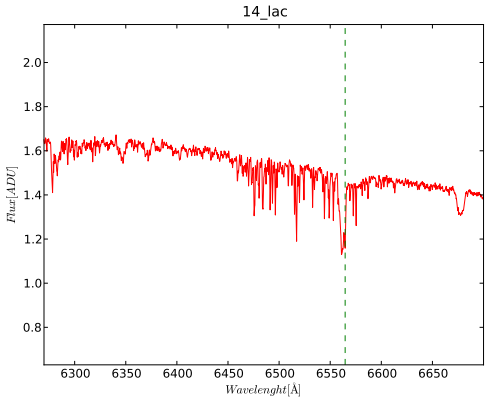


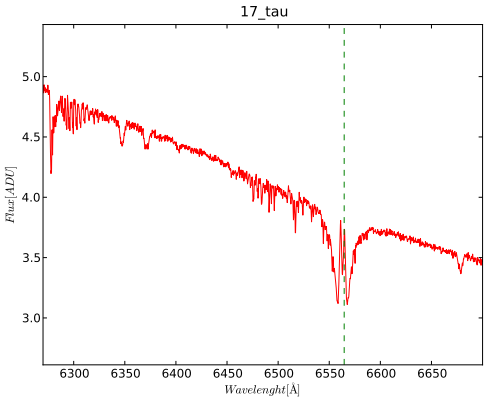


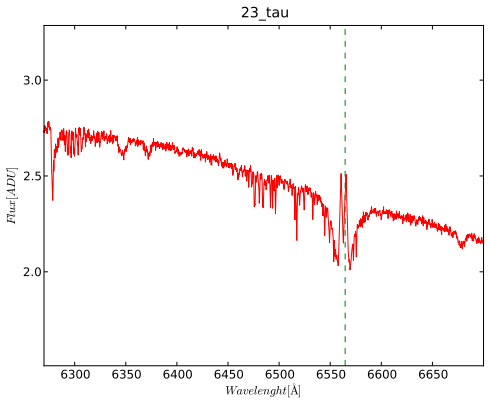


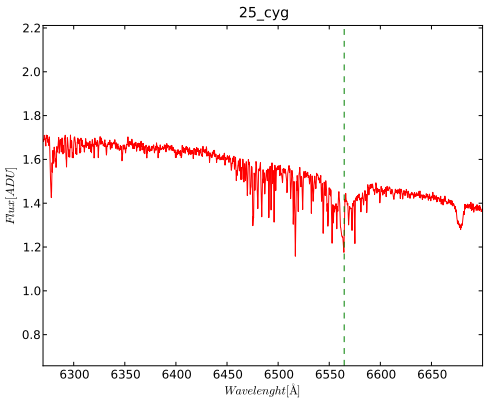

10_cas
Flux[ADU]
Wavelenght[Å]

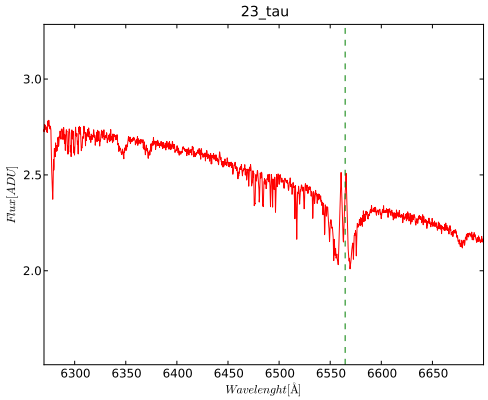

11_cyg
Flux[ADU]
Wavelenght[Å]

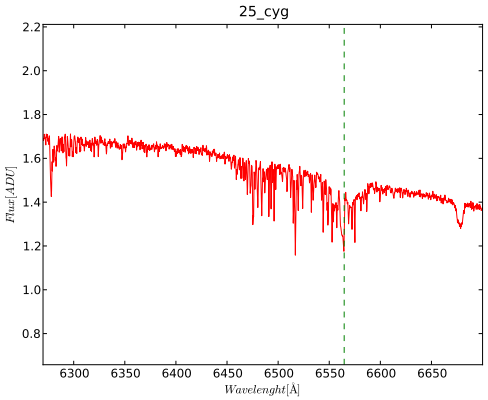

14_lac
Flux[ADU]
Wavelenght[Å]


17_tau
Flux[ADU]
Wavelenght[Å]


23_tau
Flux[ADU]
Wavelenght[Å]


25_cyg
Flux[ADU]
Wavelenght[Å]


23_tau
Flux[ADU]
Wavelenght[Å]


25_cyg
Flux[ADU]
Wavelenght[Å]

References

N.M. Ball and D. Schade. ASTROINFORMATICS IN CANADA. 2010. vii, 1

N.M. Ball, R.J. Brunner, and R. Gregory. Data mining and machine learning in astronomy. *International Journal of Modern Physics D*, 19(7):1049–1106, 2010. ISSN 0218-2718. 15

J. Becla, A. Hanushevsky, S. Nikolaev, G. Abdulla, A. Szalay, M. Nieto-Santisteban, A. Thakar, and J. Gray. Designing a multi-petabyte database for LSST. *Arxiv preprint cs/0604112*, 2006. 3

K. Benson, R. Plante, E. Auden, et al. IVOA Registry Interfaces. *IVOA Working Draft*, 2009. 6

T. Berners-Lee and R. Cailliau. WorldWideWeb: Proposal for a HyperText project. *European Particle Physics Laboratory (CERN)*, 1990. 4

S. Bukvić, D. Spasojević, and V. Žigman. Advanced fit technique for astrophysical spectra. *Astronomy and Astrophysics*, 477(3):967–977, 2008. ISSN 0004-6361. 43

National Research Council. Preserving Scientic Data on our Physical Universe. 1995. 13

M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I.H. Witten. The WEKA data mining software: an update. *ACM SIGKDD Explorations Newsletter*, 11(1):10–18, 2009. ISSN 1931-0145. iii

RJ Hanisch and PJ Quinn. The international virtual observatory. *Retrieved from http://www. ivoa. net/pub/info/TheIVOA. pdf on*, 24, 2010. 3, 4

R. Hirata and T. Kogure. The Be star phenomena. II-Spectral formation and structure of envelopes. *Bulletin of the Astronomical Society of India*, 12:109–151, 1984. ISSN 0304-9523. vii, 22

SB Kotsiantis, ID Zaharakis, and PE Pintelas. Supervised machine learning: A review of classification techniques. *Emerging artificial intelligence applications in computer engineering: real word AI systems with applications in eHealth, HCI, information retrieval and pervasive technologies*, 160:3, 2007. 16

R.L. Launer. *Robustness in statistics: proceedings of a workshop*. Academic Pr, 1979. 32

J.M. Porter and T. Rivinius. Classical Be Stars. *Publications of the Astronomical Society of the Pacific*, 115(812):1153–1170, 2003. ISSN 0004-6280. 1, 21

B. Schlesinger. A Users Guide for the Flexible Image Transport System. 1997. 13

A. Slettebak. The Be stars. *Publications of the Astronomical Society of the Pacific*, 100:770–784, 1988. ISSN 0004-6280. vii, 22

MH Van Kerkwijk, L. Waters, and JM Marlborough. Halpha emission and infrared excess in Be stars: probing the circumstellar disc. *Astronomy and Astrophysics*, 300:259, 1995. ISSN 0004-6361. 24

M. Wenger, F. Ochsenbein, D. Egret, P. Dubois, F. Bonnarel, S. Borde, F. Genova, G. Jasniewicz, S. Lalo
"e, S. Lesteven, et al. The SIMBAD astronomical database. *Arxiv preprint astro-ph/0002110*, 2000. 9

I.H. Witten and E. Frank. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann Pub, 2005. ISBN 0120884070. 16