SDSS Celestial Objects Classification

Andrea Sessa
Mat. 850082

Abstract

The project aims to study the problem of dimensionality reduction in the field of astronomy. The high number of features involved often makes the job of classifying celestial objects, basing entirely on their spectra, very difficult. The problem of preprocessing high dimensional astronomical data is considered. We refer to the method described in [3] to overcome the problem of missing data and the different redshift factor of different object. The resulting dataset(which is the starting point for the project) contains information of 4000 celestial object(such as galaxies, quasars, stars, etc.) each of the samples consists of 1000 attributes which describe the electromagnetic radiation over different wavelength(3000 to 8000 angstrom). We propose different methodologies to approach the problem of features reduction: PCA, ISOMAP, forward features selection and ANOVA F-Test approach; The goodness of each method is evaluated over Support Vector Machine(soft margin penalty is tuned by 5-Fold cross-validation) considering as metrics accuracy, precision, recall and F1 score. Final results shows, in general, how a very small number of features in the can actually capture an high percentage(90 %) of the variance associated with the data and to results identical or better with respect to the case and high number of features is used. Moreover final results also show how PCA and ISOMAP are quite ineffective on this particular dataset.

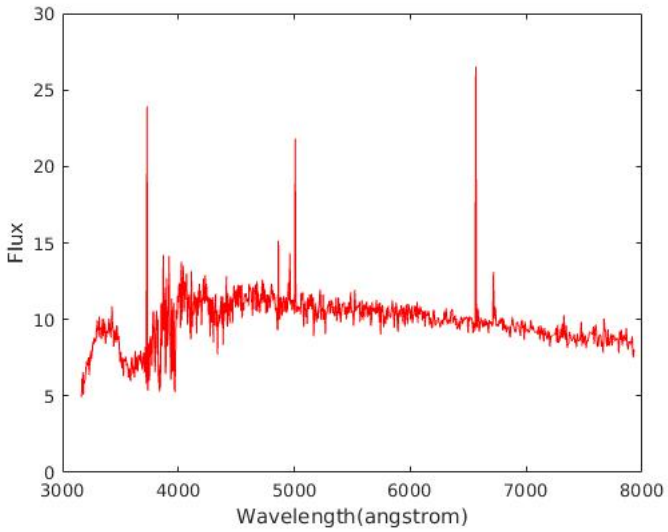
Contents

List of Figures

1 Introduction

Modern astronomy is concerned with the study of very distant celestial objects ie quasars, galaxies, stars, etc.
Often this type of classification is performed by analyzing the spectrum emitted by such objects.
In general the emission spectrum of a chemical element or of a chemical compound is defined as the electromagnetic radiation emitted when an atom or a molecule, of the object that we are observing, perform a transition from an high energy states to a low energy state. During the decadiment the atom or molecule the electromagnetic is irradiated under the form of photon, the associated photon energy (also called flux) is proportional is equal to the energy difference between the two energy states involved in the decadiment.
The important element is that for a given atom there are many possible electron transition, and each of these transition has a specific flux associated.
The Sloan Digital Sky Survey(henceforth referred as SDSS) is a major imaging and spectroscopic survey using a dedicated 2.5-m wide-angle optical telescope at Apache Point Observatory in New Mexico, United States.
The collection of data started in 2000 and continues up to nowadays(latest data release in June 2013). The dataset comprises almost 2 millions of spectra coming from different objects.
Machine learning plays an important role in the task of classifying these objects, given the fact that many times samples include more than 1000 features.
Features in general includes information about the flux associated with a specific wavelength.

Figure 1: Typical emission spectrum for an emission galaxy



If the input feature vectors have very high dimension, the learning problem can be difficult even if the true function only depends on a small number of those features. This is because the many "extra" dimensions can confuse the learning algorithm and cause it to have high variance. Hence the problem of

features selection become very important in this scenario; this project will try to cast some light on the problem by evaluating different approaches over a reduced version of the SDSS dataset.

2 Problem Formulation

The overall problem is divided into three main sections:

2.1 Features Extraction

The first one is the issue of features selection, what follows is a possible formal formulation of the problem:
The goal is, starting from a number of features $M = 1000$, find $M' \ll M$ such that the selected M' features gives the smallest expected generalization error. In more formal terms:

Given a set of functions $y = f(x, \alpha)$ we want to find a preprocessing over the data, $\boldsymbol{x} \mapsto (\boldsymbol{x} * \sigma)$

$$\tau(\sigma, \alpha) = \int \mathrm{V}(y, f((x * \sigma), \alpha)\, \mathrm{d}p(\mathbf{x}, y) \tag{1}$$

subject to $||\sigma||_0 = M'$, where $p(x, y)$ is unknown, $x * \sigma = (x_1\sigma_1, ..., x_M\sigma_M)$ denotes an element wise product, $V(.,.)$ is a generic loss function
The literature divided the feautures selected methods into two main group

- **Filter methods** rely on general characteristics of the data to evaluate and to select the feature subsets without involving the chosen learning algorithm. Scores are assigned to each features according to some metrics. A widely-used filter method is to apply a classical univariate ANOVA F-Test.
- **Wrapper methods** evaluate subsets of features which allows, unlike filter approaches, to detect the possible interactions between features.
  Sequential feature selection is one of the most widely used techniques. It selects a subset of features by sequentially adding (forward search) or removing (backward search) until certain stopping conditions are satisfied. For this project we will consider a forward search, the algorithm stops when a local minima is obeserved in the metric used to evaluate the goodness of the subset.

2.2 Dimensionality Reduction

The objective of dimensionality reduction differs from the idea behind features selection, while the latter tries to select the best subset, a dimensionality reduction algorithm applies a transformation over the existing features re-projecting the data over a dimensionality reduced dataset.
In more formal terms, we are trying to find a orthogonal transformation matrix W such that:

$$\overline{X} = W^T X \tag{2}$$

$\overline{X}$ is new dimensionality rappresentation of the data.

2.3 Classification - Multi-class SVM

Support Vector Machines (henceforth referred as SVM) are supervised learning models with associated learning algorithms that analyze data used for classification (and regression).
We focus our attention first on the case of binary classification and the we extend it to the general case of multi-class classification.
In a two-class SVM the prediction for a new point is given by:

$$f(x_q) = sign\left(\sum_{m \in M} \alpha_m t_m k(x_q, x_m) + b\right) \tag{3}$$

The objective of a SVM is to maximize the margin that is the distance of the closest point to the separating hyperplane. The problem maximize the margin leads to the following constrained optimization problem:

$$\begin{aligned} \min_x \quad & \frac{1}{2}||w||_2^2 + C\sum_i \xi_i \\ \text{subject to} \quad & t_i(w^T x_i + b) \geq 1 - \xi_i \ \forall i \end{aligned} \tag{4}$$

The previous equation permits to assign to each sample a weight(α) that determine the so-called support vector.
However in our case the problem must be formulated in terms of a generic number of classes($K = 7$ classes).
Different approaches to the problem, for this project we follow a *1 versus 1* approach: for each pairs of classes we solve a binary SVM classification problem as described above. The number of pair to be considered is given by

$$\text{N of comparisons} = \frac{K(K-1)}{2} \tag{5}$$

A comparison with others possible approach [4], eg *1 versus the rest*, shows that *1 versus 1* in general has good performance(very short training time) but in some cases it can lead to situations in which samples are ambiguously classified.

3 Methodology

In this section is included a list(and a brief description) of the approaches that will be considered during the experimentation campaign.
Details and experimentation results are given is section 4.

3.1 Baseline Classifier

The baseline is represented by a standard 7-class SVM classifier, all the dataset is used, no features selection/dimensionality reduction algorithm is applied. The SVM uses a gaussian kernel defined has:

$$k(x, y) = exp\left(\frac{-||x-y||^2}{2\gamma}\right) \tag{6}$$

The hyperparameter γ(the 'spread' of the gaussian curve) and C (the penalty for miss-classified point) are chosen by 5-fold cross validation iterating over a grid of possible value.

3.2 ANOVA F-Test

The procedure calculates the following F-statistics for each feature:

Y = *generic label(0 to 9)*
N_j = *number of samples with* $Y = j$
$\bar{x}_j$ = *the sample mean for features* X *for target class* j
$s_j^2 = \sum_{i=1}^{N_j}(x_{ij} - \bar{x}_j)/(N_j - 1)$
$\bar{x} = \sum_{j=1}^{J} N_j \bar{x}_j / N$

$$F = \frac{\sum_{j=1}^{J} N_j(\bar{x}_j - \bar{x})^2/(J-1)}{\sum_{j=1}^{J}(N_j - 1)s_j^2/N_j - 1} \sim F(J-1, N-J) \tag{7}$$

From the statistics we can assign a p-value to each features and rank them(ascending order); if the p-value is smaller than the significant level(α) then the feature is considered significant(at level α).
After the procedure has individuated the significant subset of feature we proceed by cross-validating the model(always a SVM with gaussian kernel) to determine the optimal value of C and γ
The final model is then evaluated over the test set.

3.3 Forward Features Selection

The idea is to find the best subset of features starting from an empty set and iteratively adding features until a stopping conditions is met.
The following algorithm shows the procedure that has been implemented in Matlab:

```
Data: The train dataset,D
F = ();
metricOld = 0;
metric = 0;
begin
    metricOld = metric;
    repeat
        x = D.sampleFeature();
        F.append(x);
        meric = 5FoldCrossValidate(D,F);
    until
     abs(metricOld − metric) < 0.001 or 100 features have been selected;
end
```

Algorithm 1: FFS algorithm

metric in this case is evaluated as the number of missclassified points over the number of samples in the training dataset. 5-Fold cross-validation is used to obtain an unbiased estimation of the metric for the model.
Once the procedure terminates we use the selected features to cross-validate a SVM(gaussian kernel). The final model is then evaluated over the test set.

3.4 PCA

The standard PCA algorithm requires that the input data are centered around the axes-origin, so we need to subtract the mean to the data such that they are mean centered.
PCA proceeds by calculating the convariance matrix S defined as:

$$S = \frac{1}{N}(x - \bar{x})(x - \bar{x})^T \tag{8}$$

Now we calculates eigenvectors and eigenvalues of S; each eigenvector represents a 'principal direction', the associated eigenvalues has a value that is proportional to the amount of variance along that particular direction. The idea is to select the first n principal eigenvector such that the 95 % of the variance is retained.

3.5 ISOMAP

ISOMAP is a non-linear dimensionality reduction method. One of the main flaw of PCA is that the detected principal components are linear and cannot capture the variance along non-linear direction.
The algorithm provides a simple method for estimating the intrinsic geometry of a data manifold based on a rough estimate of each data point's neighbors on the manifold. Isomap is highly efficient and generally applicable to a broad range of data sources and dimensionalities.
The algorithm can be seen has an extension of and MDS(Multi-Dimensional Scaling) algorithm.
Follows an high level description of ISOMAP:

Data: The train dataset,D
begin
  forall *p in D* **do**
    p.KNN = CalculateKNN()
  end
  //Build a the KNN graph
  $D_X = D\{d_x(i,j)\}$
  $//d_x(i,j)$ is the euclidean distance between point i and j
  //Compute the shortest path(Dijkstra) between each pair of points
  $D_G = D\{d_g(i,j)\}$
  return MDS(D_G)
end

Algorithm 2: High-level ISOMAP

The idea behind MDS can be synthetixe as follows:
Input: nxn matrix of symilarities(shortest path in ISOMAP) between n objects
Output: A configuration in a low-dimensional Euclidean space R^k whose interpoint distances $d(x_i, x_j)$ closely match the symilarities.

4 Experiments

4.1 Dataset Description

The dataset that has been used for this project is a reduced version of the SDSS spectroscopic dataset: it consisted of 4000 spectroscopic samples, each of this sample formed by 1000 features that describes the spectrum over the different wavelength.
Objects in the dataset belong to 7 different categories:

1. **STAR**: Generic stellar objects
2. **ABSORPTION GALAXY**: Galaxies that show relatively homogeneous spectra, dominated by absorption features from cool giant stars
3. **GALAXY**: Generic galaxy(neither absorption nor emission)
4. **EMISSION GALAXY**: Galaxies whose spectrum is dominated by emission features
5. **NARROW-LINE QSO**: Narrow emission quasars
6. **BROAD-LINE QSO**: Broad emission quasars
7. **LATE-TYPE STAR**: Cooled starts(type K or type M)

4.1.1 Preprocessing

This project do not use the original SDSS dataset(which contained more than 4000 features per sample!).
The samples present in the original dataset had two main problems:

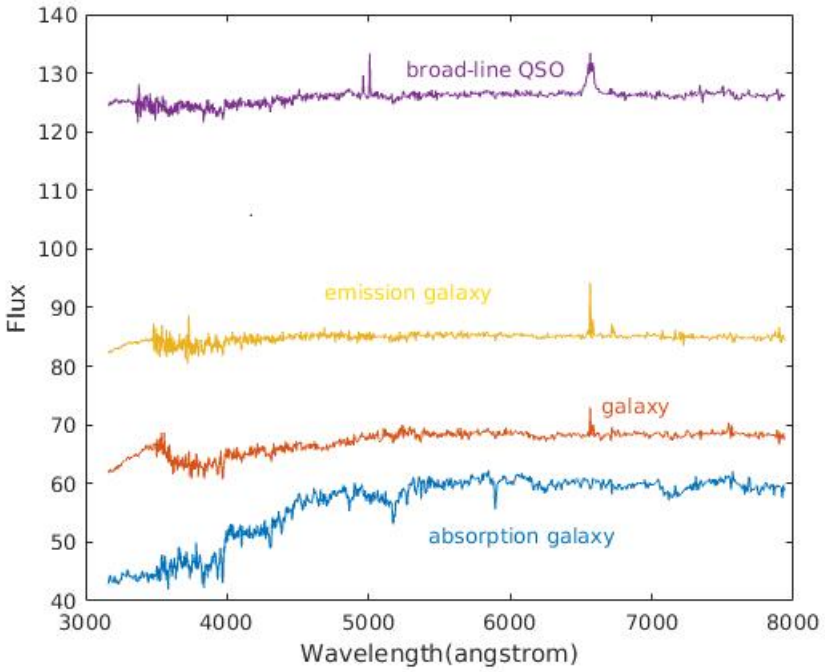
- Each individual spectra present a different redshift factor(z). This phenomena is due to the fact that the object that was being observed, was moving during while emitting light, this cause the frequency of light to 'shift' toward lower energy wavelength ie. toward 'red'.
- Missing data: There are several reasons for gaps to exist: The removal of skylines, bad pixels on the CCD chips all leave gaps at different frame wavelengths for each spectrum, general technical problem, etc. All these factor can contribute to incomplete spectra.
  To overcome this problem a Principal Component Analysis(henceforth referred as PCA) is used to reproject the data on their principal dimension and then used to fill the missing gap. The entire procedure is described in detail in [3]

In addition to the two above stated procedure the dataset has been further downsampled till to reduce the number of significant features to 1000(which is still quite high!).
Finally, given the fact that the spectra belong to very different celestial object at very different distance(light years order) from the observation point, a final step of normalization is applied that is Z-Score and mean centering(making them suitable for a subsequent PCA analysis).

4.1.2 Visualization

In figure 2 are shown some random spectra for different type of objected extracted from the dataset:

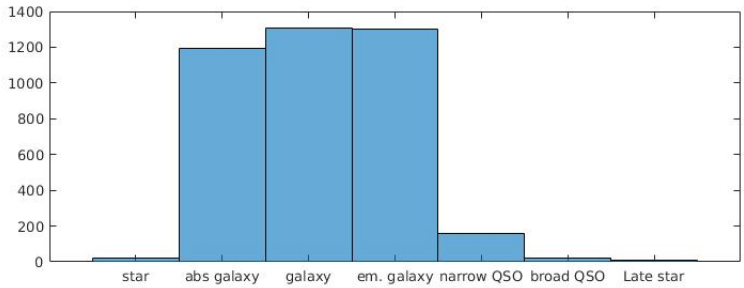
Figure 2: Sample spectrum for different objects



We can observe the different characteristics of the spectra associated with each object:
Broad-line quasars have marked and large peak of emission in the emission, emission galaxies distinguish themself, from other type of galaxies, thanks to the presence of very narrow and high peak of emission(in general higher than narrow-line QSO); absorption galaxies present a series of 'caves' in their spectra where some component of the light is absorbed by particular chemical compounds present in the objects of the galaxy; on the other hand 'standard' galaxies present a spectra that rapresent a trdeoff between the previous cases.

We also include a histogram showing the number of objects per class:

Figure 3: Objects per class



The histogram shows a very irregular distribution of the objects in the class, this is due to two main reasons:

1. Some objects are more difficult to observe with respect to others, quasars are more rare to observe(and to discover) with respect to galaxies

2. The dataset in consideration has a limited amount of samples(4000), this is mainly due to the limited computational power available to the author.

4.2 Metrics

The results of the different alternatives evaluated over the test set are compared according the confusion matrix.
The confusion matrix is KxK matrix (in our case $K = 7$), the generic element e_ij indicates how many samples that comes from class i have been predicted to class j.
The following metric can be evaluated:

$$Accuracy = \frac{\sum_{i=1}^{K} e_{ii}}{N}$$

For each class k we also consider:

$$Precision_k = \frac{tp_k}{fp_k + tp_k}$$

$$Recall_k = \frac{tp_k}{tp_k + fn_k}$$

Where tp_k is the number of true positive for class k, ie the number of samples that belong to class k and has been correctly predicted to class k, and fp_k is the number of false positive, ie the number of samples that does not belong to class k but have been predicted to class k.
The precision express the number of correctly predicted samples for a given class k with respect to the number of true positives plus the number of point that has been uncorrectly predicted belonging to class k.
The recall express the number of correctly predicted samples for a given class k with respect to the number of true positives plus the number of point that has been uncorrectly predicted not belonging to class k.

4.3 Results

4.3.1 Baseline Classifier

The first problem in SVM classification is the choice of the correct kernel:
Three different kernels have been evaluated, as a rough estimation of the performance we consider the classsification accuracy:

- Linear Kernel, Accuracy on the test set: $\sim 67\%$
- Polynomial Kernel(3° polynom), Accuracy on the test set: $\sim 69\%$
- Gaussian Kernel, Accuracy on the test set: $\sim 33.5\%$

The optimal hyper-parameter are chosen by 5-fold cross-validation.
This rough evaluation of these type of kernel shows how a gaussian kernel performs very bad over the test set mainly due to overfitting.
We decide to use a polynomial based kernel, follows a detailed analysis to find the best order for the polynom.
The generic polynomial kernel is in the form:

$$k(x, y) = (\gamma x^T y)^d \tag{9}$$

We perform a 5-fold cross-validation in order to determine the best γ, C and degree(d).
The procedure has individuated the following hyperparameter:

- degree(d) = 3
- $\gamma = 2^{-20} = 0,000000954$
- $C = 2^{15} = 32768$

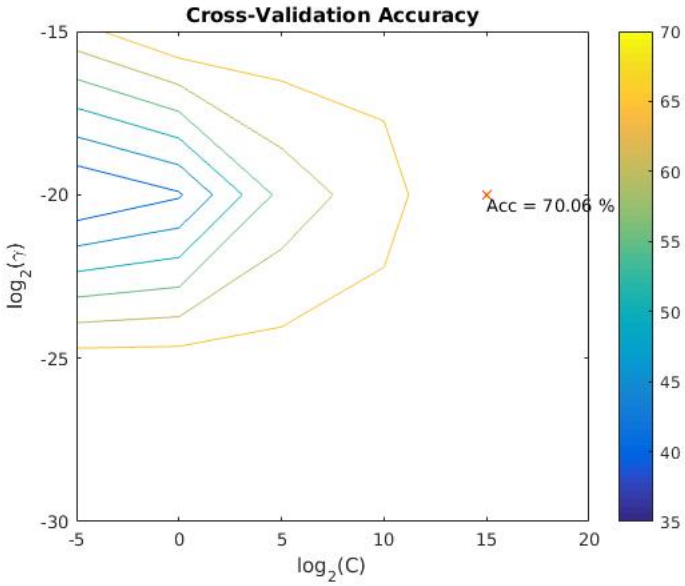


Figure 4: Baseline Contour plot

With the optimal value of C and γ the baseline classifire has been able to achieve on the test set an accuracy of 68.88 %

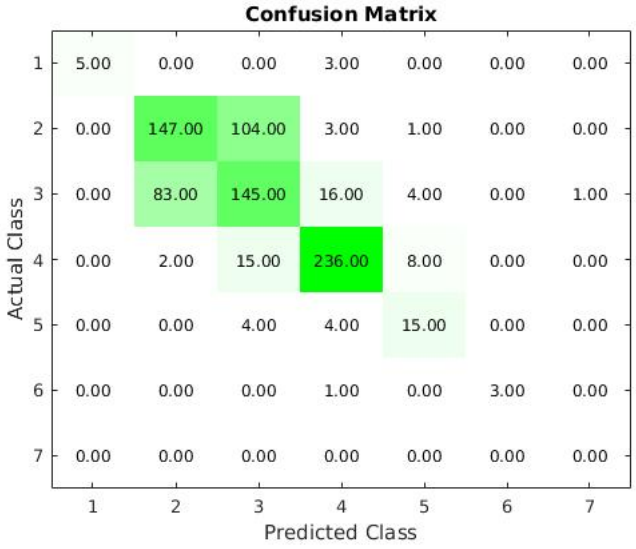


Figure 5: Baseline Heatmap

We also report the recall and precision for each class:

Class	1	2	3	4	5	6	7
Precision	1	0.6336	0.5410	0.8973	0.5357	1	0
Recall	0.6250	0.5765	0.5823	0.9042	0.6522	0.75	*NaN*

4.3.2 ANOVA Classifier

4.4 Notes on LibSVM

Matlab does not include commands that permits to train a multiclass SVM. The external library LibSVM [5] is used in order to train a 7-class SVM. LibSVM sources file has been modfied [6] in order to permit kernel evaluations in training/testing to be parallelized on 6 threads; this small modifications has lead to a significant improvement in the compuatation performance.
TODO: Insert example with numbers!

5 Conclusions

References

[1] Wikipedia, *Emission Spectrum*

[2] Wikipedia, *The Sloan Digital Sky Survey*

[3] C.W. Yip et al, *Spectral Classification of Quasars in the Sloan Digital Sky Survey: Eigenspectra, Redshift, and Luminosity Effects*, Astronomical Journal, 2004

[4] Chih-Wei Hsu, Chih-Jen Lin, *A Comparison of Methods for Multi-class Support Vector Machines*, IEEE Transactions on Neural Networks, 13(2002), 415-425

[5] Chih-Wei Hsu, Chih-Jen Lin, *LibSVM: A Library for Support Vector Machines*, https://www.csie.ntu.edu.tw/ cjlin/libsvm/

[6] Chih-Wei Hsu, Chih-Jen Lin, *Use OpenMP to parallelize LibSVM on a multicore/shared-memory computer*, http://www.csie.ntu.edu.tw/ cjlin/libsvm/faq.html#f432