Predicting Year of a song release using timbre features of a song using Million song Dataset(UCI Machine Learning Repository)

A Project report submitted for EE-660 Machine Learning

by

Sahana Venkatesh

Student Id: 6643244225

Department of Electrical Engineering

Viterbi School of Engineering

University of Southern California

December 2016

Contents Pageno

Chapter 1

Predicting Year of a song release using timbre features of a song

1.1 Introduction

Why year prediction

Year prediction has not been studied very much.There is huge untapped potential if we could use year prediction in recommendation system.

How do we label songs?90's song or 80's song.Searching for a song within genre when becomes easier when we have an idea what era a song belongs to.

We could also use it to study evolution of music.So in short,an interesting problem to work on. The dataset I used is the million song dataset of the (UCI Machine Learning Repository).

Problem statement in UCI Machine Learning Repository Prediction of the release year of a song from audio features. Songs are mostly western, commercial tracks ranging from 1922 to 2011, with a peak in the year 2000s.

Attribute Information:Attribute Information: We have 90 attributes, 12 = timbre average, 78 = timbre covariance.Features extracted from the 'timbre' features from The Echo Nest API. We take the average and covariance over all 'segments', each segment being described by a 12-dimensional timbre vector.[3]

what is timbre Timbre is derived from Spectrogram and it is said to be the attribute of a song which enables the listener to judge two non-identical sounds(which have same pitch and loudness) as dissimilar[1].

Songs were divided into segments and you observe each segment through spectrogram.The spectrogram patch is expressed as a combination of the basis[2].[2] describes “the first [dimension of the 12-dimension timbre vector] represents the average loudness of the segment; second emphasises brightness; third is more closely correlated to the flatness of a sound;” etc.
The breakup of the project report:
Chapter 2 introduces the Dataset
Chapter 3 States of the Objectives of the Project.
Chapter 3 Highlights Flow of work
Chapter 4 Discusses the algorithms.
Chapter 5 tests the Algorithms.
Chapter 6 highlights the conclusion of the project.

1.2 Analysis of the Dataset

The literature [2] says the the data set and the features have been preprocessed already.The first 12 features are the mean of 12 the PCA components over all the segments in that song.In total there are 515345 songs.
1)**The first 12 columns** Entries in the column corresponds to the mean timbre value measured for first 12 principal components across all the segments for that particular song
2)**The rest 78 columns** highlight the covariance between the components over the segment In that particular song.
We are trying to predict the year as a function of audio features.If we plot audio features as a function of the years this would give us an hope that target function we are attempting to realise might exist.Audio features are found to be changing with time,so model should utilise and learn these changes as it is trying to predict the year in which song released. 1.1 on the following page.

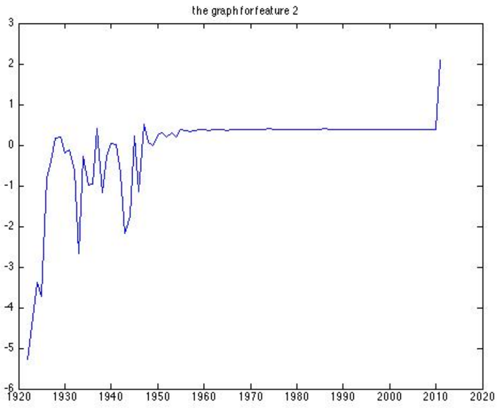


Figure 1.1: Distribution of the feature 2 with respect to the years

1.2.1 Correlation between the features

We can observe the correlation of features using the covariance matrix.The matrix is correlated.The rank is definitely less than 90.The correlation coefficient between columns 15-25 is really high. **Covariance matrix when we consider 90 features**.

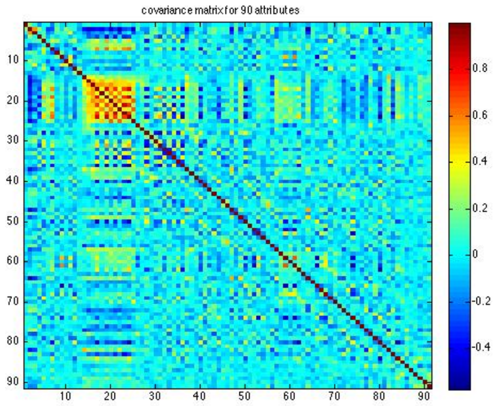


Figure 1.2: Covariance matrix when we consider 90 features

Covariance matrix when we consider first 12 PCA components This matrix is also correlated.The Combination(column 2 and Column 3) is the highest correlation coefficient than any other combination.

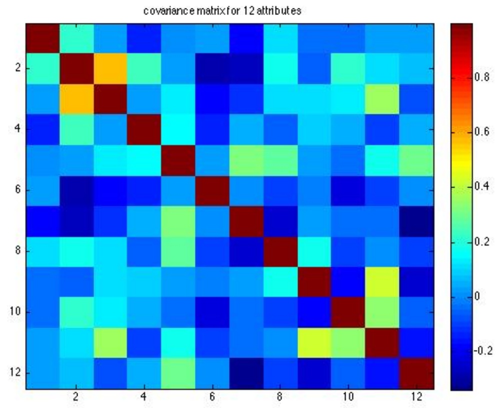


Figure 1.3: Covariance matrix when we consider first 12 PCA components

Variable 2 looks promising predictor.Variable 1 is told be correlated with loudness which means songs (as in shown the sample)are suddenly getting louder.

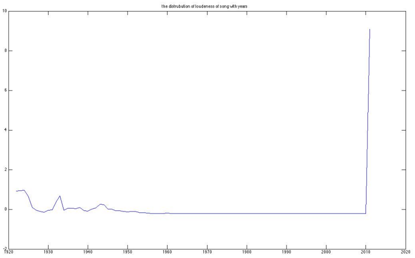


Figure 1.4: Distribution of the loudness with respect to the years

1.2.2 Distribution of the year

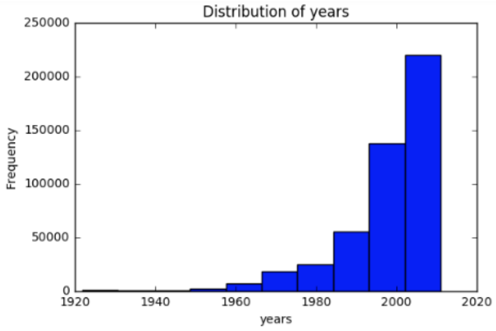


Figure 1.5: Distribution of the samples with respect to years

The statistics of the year distribution is as follows

Mean year is 1998.42004432

The range of the years taken into account is from 1922.0 to 2010.0

The standard deviation of the years is 10.9220052482

The mode is 2007.

If the sample is bias,learning could also be bias

There is a chance that this sample is biased.We don't know whether this abrupt jump is due to the sudden increase in songs or just absence of documentation of songs from early years.There is chance that In sample and Out sample are from totally different distribution.

Checks for sparsity using the function issparse returns that the matrix is not sparse. So,keeping all the analysis in mind we have to formulate the problems we would be addressing.

1.3 Objectives of the project

1)Can we find a model which takes the subset/the entire set of 90 attributes and gives us an estimate of year of song release.

2)Can we predict the year with just knowing the value of the mean of the first 12 principal components.

3)How helpful are the other 78 columns which bring out the covariance between the principal components in each song.

1.4 Flow of work

1.4.1 Requirements

Based on the above analysis.I would choose a model which satisfies or which covers most of the below requirements when compared to other model in my hypothesis set.

1)I need a model to do feature selection on dense matrices (i.e)because not all the features might be equally useful.Reducing the features,reduces the parameters and therefore reduces the complexity of the model.Higher the complexity of the model ,more likely it would over fit.

2)I need a model which would be robust to sample bias.

3)I need a model which performs well on the cross validation set.

4)Model which has low variance.Higher the variance,higher are the chances of the model over fitting.Keeping variance in check also keeps the complexity of the model in check.

1.4.2 Division of training set

Train set : first 363,715 examples.

Test set: last 51,630 examples.

Cross Validation Set:I kept 100000 examples aside to aid me in model selection.

1.4.3 Model design

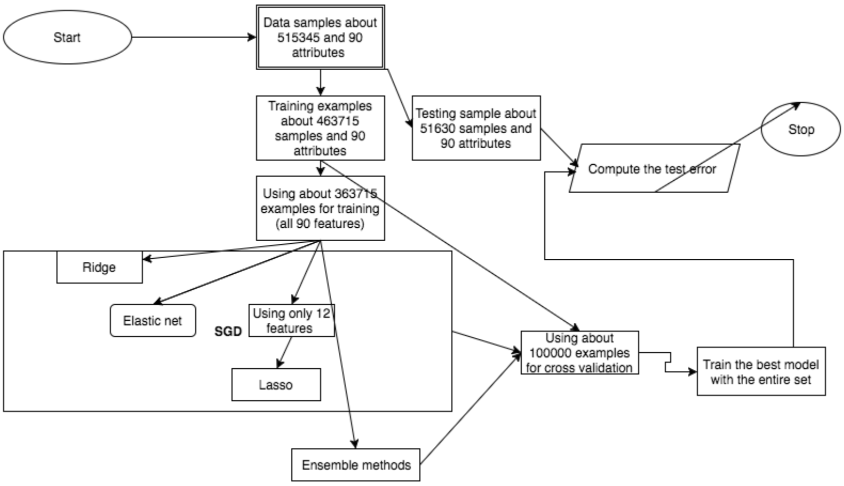

Start
Data samples about 515345 and 90 attributes
Training examples about 463715 samples and 90 attributes
Testing sample about 51630 samples and 90 attributes
Compute the test error
Stop
Using about 363715 examples for training (all 90 features)
Ridge
Elastic net
SGD
Using only 12 features
Lasso
Using about 100000 examples for cross validation
Train the best model with the entire set
Ensemble methods

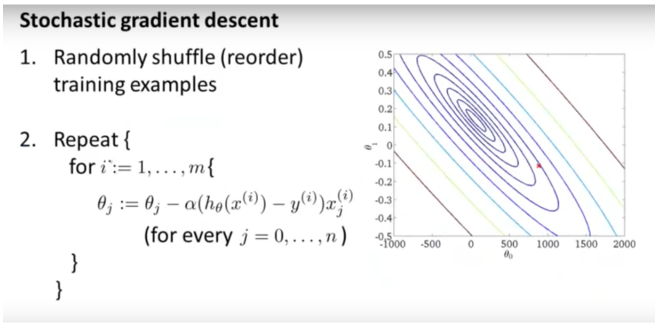
Figure 1.6: Model selection

1.5 Algorithms and their Performance

Stochastic Gradient Descent

Scikit[5] recommends the users to use SGD if the number of samples is quite high(¿10,000). Hence I will implement the models with idea that when it is searching for its optimal parameter it will search with the help of stochastic gradient descent rather than normal gradient descent.The stochastic Gradient descent updates parameter with one example,so that new parameter fits the examples taken one at a time.The batch gradient computes cost across all the training examples.Sums the gradient across all the examples and then carries out the update.The

Stochastic Gradient Descent converges much faster than the gradient Descent.



Stochastic gradient descent

```
1. Randomly shuffle (reorder)
   training examples

2. Repeat {
     for $i := 1, \dots, m${
         $\theta_j := \theta_j - \alpha(h_\theta(x^{(i)}) - y^{(i)})x_j^{(i)}$
             (for every $j = 0, \dots, n$)
     }
   }
```

Figure 1.7: Stochastic Gradient Descent

Courtesy:[https://www.coursera.org/course/ml][6]

1.5.1 Ridge Regression

Ridge regression addresses some of the problems of Ordinary Least Squares by imposing a penalty on the size of coefficients. The ridge coefficients minimize a penalized residual sum of square of the weights.Least square gives us the best way we can represent the target(year) as a linear function of the attributes(features).We impose a penalty on the weights so that introduction of error or small changes in the input does not give rise to huge changes in the predicted target values.It usually does not give you sparser solution but it does give a robust system.

What does the ridge result tell us tell us

1)The true value of the year predicted for a song would be between 8 years of the predicted value with the very high probability given by the Hoeffding equation.

2)The weights assigned to the features are small.

Degrees of freedom of the Ridge regression decreases as lambda increases.It is always lesser than or equal to 90 in this case.

1.5.2 LASSO-Least Absolute Shrinkage and Selection Operation

The Lasso is a linear model that estimates sparse coefficients. It is useful in some contexts due to its tendency to prefer solutions with fewer parameter values, effectively reducing the number of variables upon which the given solution is dependent[5].We impose a l1 penalty on the weights.We try to find the solution at one the corner of the points(i.e one of the co-efficients would be zero). We compute lasso using only first 12(Mean of the PCA components) of the features to judge which features are important.

What does the lasso result tell us tell us

1)The true value of the year predicted for a song would be between 8 years of the predicted value with the very high probability given by the Hoeffding equation.

2)The weights assigned to the features are non-zero indicating that all the features are important and I cannot ignore one of them.

1.5.3 Elastic net

ElasticNet is a linear regression model trained with L1 and L2 prior as regularizer. This combination allows for learning a sparse model where few of the weights are non-zero like Lasso, while still maintaining the regularisation properties of Ridge [5].We control the convex combination of L1 and L2 using the l1$_{r}atioparameter[5].Elastic - netisusefulwhentherearemultiplefeatureswh$ $netislikelytopickboth.$

What does the Elastic net result tell us tell us Results:Weights are small and some of the weights are zero. Sparse coefficients which is amazing using which we can reduce the size of the matrix. Comparing the weights of lasso,ridge and Elastic net

1.5.4 Comparing the weights of lasso,elastic net and ridge regression

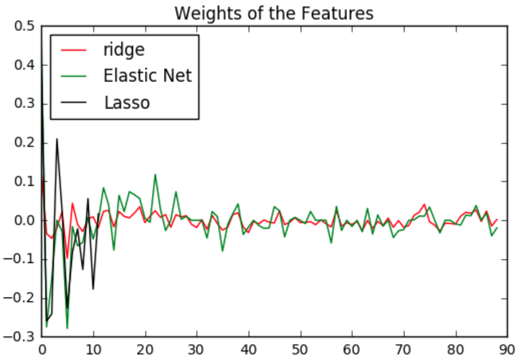


Figure 1.8: Weights of the columns by various algorithms

Key takeaways

1. The weights of the columns as decided by lasso are bigger than ones offered by the ridge or elasticnet.This shows that all of my 12 PCA components are important.

2. The weights of the ridge and elasticnet are quite close to each other.The weights of many columns (given by ridge and elastic net) is close to zero.

1.5.5 Ensemble methods

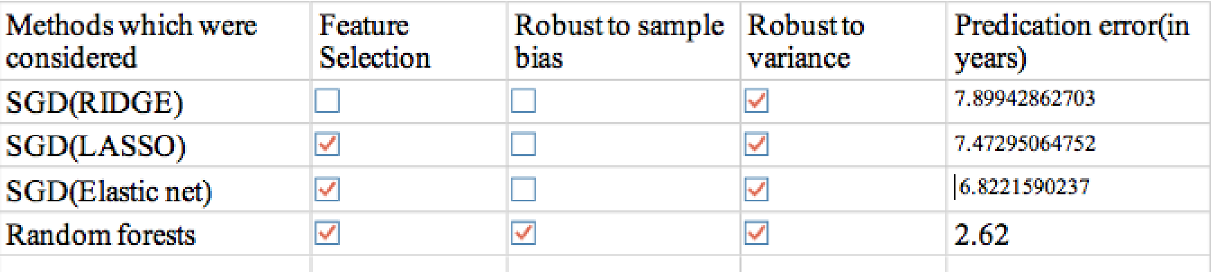
The article [3] is one among the few articles which suggested to use random forests or carts to deal with sampling bias.Since my sample might suffer from sample bias it is important to consider a model which would address the problem. The main idea behind the ensemble methodology is to aggregate multiple weighted models to obtain a combined model that outperforms every single model in it.Increasing the number of trees leads to low variance and Increasing the depth leads to low bias.I delt with 20 trees.The function RandomTreesRegressor can also give you an idea about the important features.

1.6 Testing the algorithm

I separated out around 100,000 data samples for cross validation.According to hoeffding inequality how my algorithms perform in cross validation set is pretty close approximate of how they would perform on real life data.

$$\mathbb{P}\left[|E_{\text{in}}(h) - E_{\text{out}}(h)| > \epsilon\right] \leq 2e^{-2\epsilon^2 N} \qquad \text{for any } \epsilon > 0,$$

The random forests look like a good fit to the problem since it gives me really low accuracy.The problem it takes so much time to train and as the no of samples increases it would be better if we could have model which runs faster.The dataset has the potential of growing and we need a model which we can update without worrying much about the execution.The elastic net seems like the next best thing.The error is around 7 years which is higher when you compare it with random forests. The errors obtained for various algorithms is shown in the table below



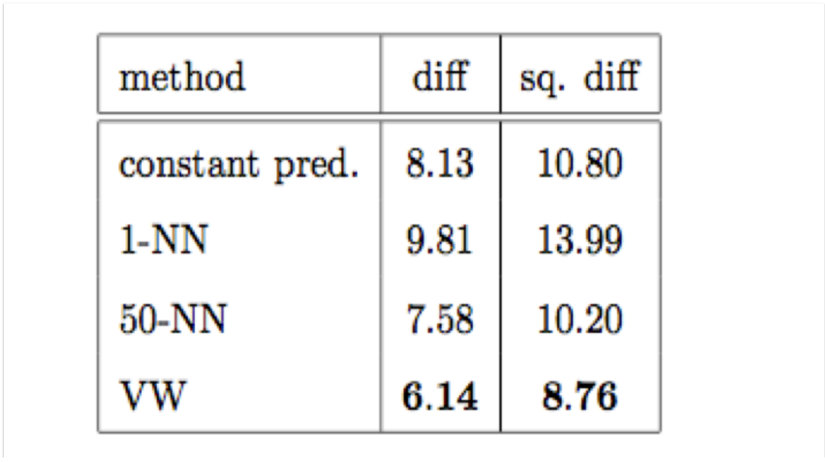
Methods which were considered	Feature Selection	Robust to sample bias	Robust to variance	Predication error(in years)
SGD(RIDGE)	□	□	☑	7.89942862703
SGD(LASSO)	☑	□	☑	7.47295064752
SGD(Elastic net)	☑	□	☑	6.8221590237
Random forests	☑	☑	☑	2.62

1.7 Conclusion

An ideal model should fit for this problem would be one which address the problem Random forests face and Problem Elastic net faces and neatly capture the good part of the above mentioned algorithms.Since the sample set has the potential of expanding further the Elastic net model looks promising. Test data performance= 6.81564491984 years (absolute error) and 90.9447640365(mean squared error)and it is close to the real value with a high accuracy.

1.7.1 Earlier works on same lines

The article [2] carried out work along the same lines . They compared two algorithms: k-NN and Vowpal Wabbit, both chosen because of their applicability to large-scale problems. They also reported the error obtained by the best constant predictor.



method	diff	sq. diff
constant pred.	8.13	10.80
1-NN	9.81	13.99
50-NN	7.58	10.20
VW	**6.14**	**8.76**

Courtesy:Taken from [7]

1.7.2 Answers to my questions

1)Can we find a model which takes the subset/the entire set of 90 attributes and gives us an estimate of year of song release. Yes we can .Elastic net and Random forests perform well and give us an reasonable estimate of the year using the audio features of the song.
2)Can we predict the year with just knowing the value of the mean of the first 12 principal components.
3)How helpful are the other 78 columns which bring out the covariance between the principal components in each song. Both the best performing algorithm needed all the columns.Intuitively we know that knowing one feature is not enough to point out an era,you need combination of features and their covariances.

1.8 Bibliography

1. https://en.wikipedia.org/wiki/Timbre.

2. Thesis titled 'Large-Scale Pattern Discovery in Music' by Thierry Bertin-Mahieux,Columbia University.

3. Lichman:2013 , author = ”M. Lichman”, year = ”2013”, title = ”UCI Machine Learning Repository”, url = ”http://archive.ics.uci.edu/ml”, institution = ”University of California, Irvine, School of Information and Computer Sciences”

4. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4743660/

5. scikit-learn, title=Scikit-learn: Machine Learning in Python, author=Pedregosa, F. and Varoquaux, G. and Gramfort, A. and Michel, V. and Thirion, B. and Grisel, O. and Blondel, M. and Prettenhofer, P. and Weiss, R. and Dubourg, V. and Vanderplas, J. and Passos, A. and Cournapeau, D. and Brucher, M. and Perrot, M. and Duchesnay, E., journal=Journal of Machine Learning Research, volume=12, pages=2825–2830, year=2011

6. https://www.coursera.org/course/ml

7. Bimbot et al., 2011] F. Bimbot, E. Deruty, G. Sargent, and E. Vincent. Methodology and resources for the structural segmentation of music pieces into autonomous and compa-

rable blocks. In BIBLIOGRAPHY 98 Proceedings of the 11th International Society for Music Information Retrieval Conference (ISMIR 2011), 2011