CP218: Theory and Applications of Bayesian Learning
Project-2 Report

Vilasini Ashokan
SR Number: 06-18-01-10-51-24034

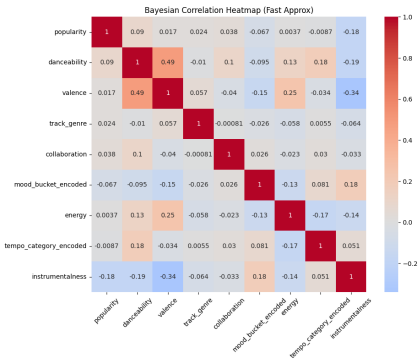
1 Introduction

- **Motivation:** With the current trend of music streaming platforms, understanding the probabilistic relationships between musical features and song popularity is important and helpful. In this project, the underlying factors contributing to song success and artist popularity on Spotify have been analysed using probabilistic methods as they provide better understanding of the data distributions.
- **Problem Statement:** The goal is to find answers for questions such as: What are the latent correlations among musical attributes such as popularity, danceability, valence, energy, and instrumentalness? Which musical genres and mood buckets dominate among popular songs? How can probabilistic modeling explain the distribution of popularity, mood and genre among the songs?
- **Main Modeling Idea:**
  1. **Regression Analysis:** Bayesian Neural Network (BNN) is used to predict song popularity as a function of various other features.
  2. **Clustering:** Gaussian Mixture Model (GMM) is used for clustering data based on mood buckets(created as a part of feature engineering).
  3. **Classification:** Bayesian Logistic regression is used to classify data based on song genres.

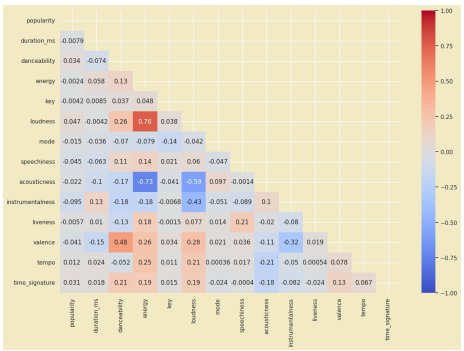
2 Data Description and Analysis:

Spotify dataset has been taken from Kaggle (link). The dataset contains features such as *popularity*, *danceability*, *valence*, *energy*, *tempo*, etc. Preprocessing includes removal of duplicates and missing entries, log transformation and scaling of features based on their distributions. For near-normal features, StandardScaler was used. For light skewed data, Min-Max scaler was used to preserve the shape and for heavily skewed and narrow ranged feature such as *duration_ms*, log transformation followed by Min-Max scaler was used. Categorical features were encoded and new engineered features *tempo_category* and *mood_bucket* were created. The following points are EDA inferences:

1. **Correlation estimation**: To estimate correlation between various features, MCMC sampling has been used to approximate the posterior distribution of correlation coefficient ρ. The mean of posterior has been taken as the estimation of correlation. Many people have done correlation estimation using *Pearson's r* on Kaggle. Below figure shows comparison of the two heatmaps



(a) Heatmap obtained using MCMC sampling



(b) Heatmap obtained using Pearson's r. image source: Kaggle.

Figure 1: Comparison of correlation heatmaps

The Bayesian (MCMC) heatmap shows robust correlations considering uncertainties, whereas Pearson's r presents strong linear relationships without uncertainty. Bayesian approach is better for robust inference, especially with small or noisy datasets. Other inferences from heatmap include: Danceable tracks tend to be happier and more energetic, while instrumental tracks are less popular and less positive. Overall, popularity shows weak correlation with any individual features, indicating that it's influenced by multiple factors.

2. **Which genres do most popular artists work on?** To find the answer, we predict posterior for track_genres among popular songs by taking Dirichlet prior as observed genre counts. It was found that *dance*, *latin* and *hip-hop* are the top genres of popular songs. People on kaggle have estimated top genres with highest popularity score by grouping solely on popularity score. Their conclusion was that pop-film, k-pop and chill are top 3 genres. Bayesian is a better approach as it considers prior knowledge.

3. **Which mood_bucket do most famous songs lie on?** Similar Dirichlet posterior estimation tells us that *Dark Dance*, *Party Vibes*, *Balanced Energy* and *Moody Beats* are popular mood forms.

4. **What tempo are used in most popular songs and by most popular artists?** Similar Dirichlet posterior estimation tells us that slow tempo songs are popular.

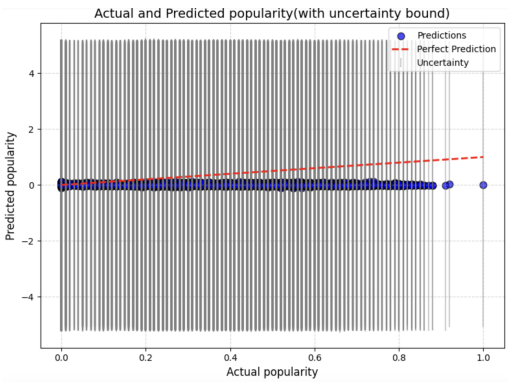
5. **What is the mean energy level(on a scale of 0-1) in popular songs?** MCMC sampling was used to fit Gaussian posterior for energy and it was found that mean energy level is 0.73 for popular songs.

3 Regression Model using Bayesian Neural Network

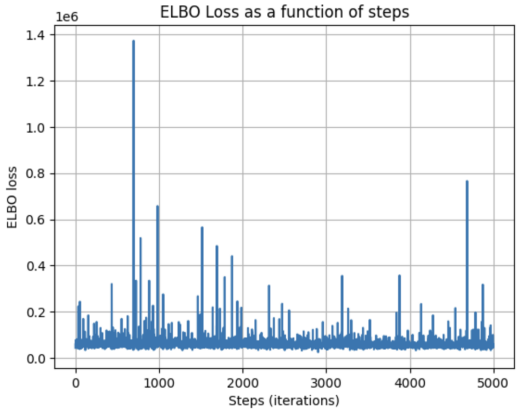
- **Motivation for choosing BNN:** The traditional neural networks make predictions without indicating how certain they are, which can be misleading when dealing with real-world noisy data. BNN learns distribution over the weights. It not only provide predictions but also express the model's confidence in predictions which is helpful for making well-grounded decisions in uncertain situations.

- **Model description:** The model uses 13 features including danceability, energy, tempo, genre encoding, etc to predict popularity. Two hidden layers(with dimensions 64 and 32) are taken with tanh activation function followed by an output layer. For all weights and biases, Gaussian prior $\sim \mathcal{N}(0, 0.5)$ is assumed. The likelihood of observed popularity is modeled as a Normal distribution with the network's output as the mean and noise term $\sigma \sim$ HalfNormal(5) as standard deviation. The posterior over weights is found using Automatic Differentiation Variational Inference which converts the parameters to an unconstrained space and chooses a parameterized family of distributions for posterior. The package pymc was used for implementation.

- **Result:** The model was trained on dataset of size 21000 and was tested on dataset of size 9000(both chosen in a stratified manner). The test MSE was obtained as: 0.1605 and the Average predictive uncertainty (std) obtained was: 2.5724. The following figure shows: (i) actual vs predicted popularity and (ii) ELBO loss a a function of steps.



(a) Actual VS predicted popularity



(b) ELBO loss VS Steps

- **Why model works:** BNN naturally uses prior knowledge and uncertainty, making it suitable for complex, high-dimensional data like music features. By treating weights probabilistically, BNN can generalize better in data-sparse regions, where traditional NNs might overfit or produce over-confident outputs.
- **Advantages:** Provides confidence levels of predictions which is useful in real-world applications like music recommendation or forecasting. The priors over weights act as natural regularizers avoiding overfitting. ADVI are scalable and computationally less expensive than MCMC.
- **Disadvantages:** ADVI just gives a rough approximation of the true posterior, which may miss complex representations like multimodality. BNN is slower than traditional NN due to sampling and posterior approximation. BNN is harder to interpret than simpler models like linear regression or decision trees.

4 Clustering using Gaussian Mixture Model

- **Motivation for choosing GMM:** The mood-related musical features such as danceability and valence exhibits overlapping and continuous distributions. GMM assumes that the data is generated from a mixture of K multivariate Gaussian distributions. Each cluster is defined by a mean vector, covariance matrix, and prior probability of the cluster. Due to this, GMM are capable of capturing soft boundaries between moods unlike k-means clustering. We can model elliptical clusters using full covariance which reflects real-world musical feature distributions well. It provides probabilistic assignments, which is useful for later stages such as playlist personalization or hybrid mood tagging.
- **Model description:** Since mood_bucket was originally created using danceability and valence, we use these features for clustering. 9 clusters were used as the original mood bucket has 9 values. Scikit-learn package was used to implement GMM. GMM uses EM algorithm to find the mean, covariance and mixing coefficient.
  Set-up: The covariance type was chosen to be full as it allows each cluster to have its own general shape, which is important when features are correlated.
  Convergence Diagnostics: GMM was trained with max_iter=100 and n_init=10 which runs with different initializations to avoid local optima. Initially, GMM was not converging fast and was taking up a lot of time. So the training data size and sample size were reduced and max_iter and n_init were introduced for better convergence and clustering.
- **Result:** The clustering is evaluated using Silhouette and ARI score. Both score range in -1 to 1 with higher values representing higher match. For each data point, Silhouette score is calculated as: $s(i) = \frac{b-a}{max(a,b)}$, where a is the average distance between the point and all other points in the same cluster and b is the smallest average distance from the point to all points in any other cluster.
  Silhouette score obtained: 0.334
  ARI score obtained: 0.483
  The following figure shows the actual VS predicted clustering as viewed on dimension pairs: (danceability, energy), (valence, energy), and (danceability, valence).

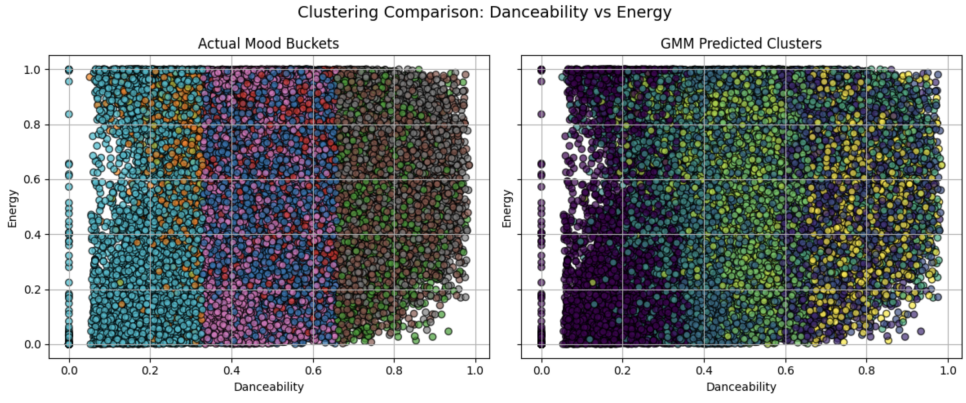


Figure 3: Actual VS Predicted clusters as viewed on danceability-energy space

.

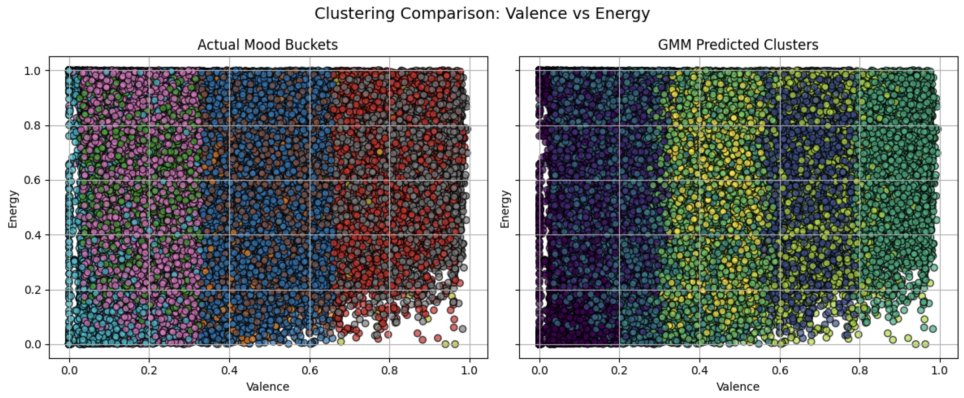


Figure 4: Actual VS Predicted clusters as viewed on valence-energy space

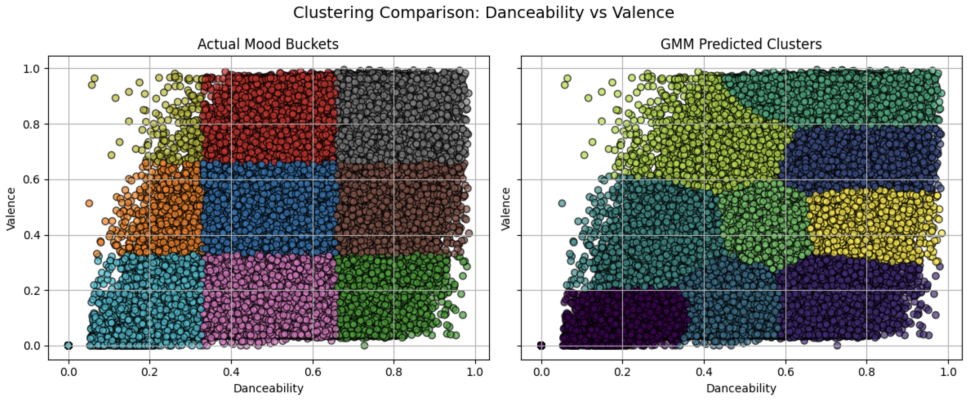


Figure 5: Actual VS Predicted clusters as viewed on danceability-valence space

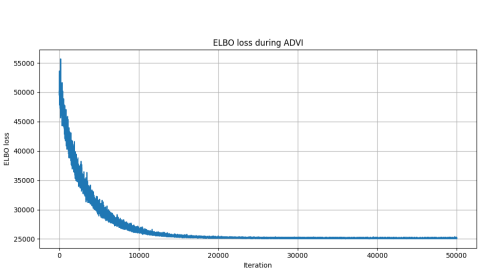
- **Why model doesn't work well:** The real-world mood features like danceability and valence may not follow Gaussian distribution or may not form well-separated elliptical clusters. Using only two features and fixed clusters limits the model's ability. But even using more features and clusters, the model under-performs as prediction deviates from original nature of mood clusters.
- **Advantages:** Songs can belong to multiple mood clusters with varying probabilities. GMM models teh elliptical, complex distributions, unlike spherical K-Means. Each cluster is defined by a mean and covariance, allowing insight into typical mood characteristics. GMM is scalable and efficient with moderate data sizes and feature dimensions.
- **Disadvantages:** GMM will not work well if the data is not Gaussian. Model is sensitive to initialization and EM can converge to local optima and hence multiple runs (n_init) are needed. Also, GMM is slightly computationally expensive than K-Means, especially with full covariance matrices.

5 Classification using Bayesian Logistic Regression

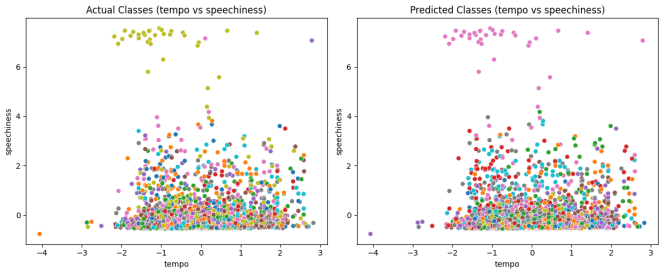
- **Motivation for choosing Bayesian LR:** Unlike classical logistic regression, Bayesian LR provides a full posterior distribution over the parameters which allows uncertainty quantification and interpretation of how each feature contributes to the prediction. Also, placing priors on the parameters acts as regularizer.
- **Model description:** Totally 11 features are used to model track_genre's classification. For weights and biases of each class pair, prior is assumed as $\mathcal{N}(0, 1)$. Later, softmax was used to obtain class probabilities. Categorical likelihood is assumed for the observed class labels. For approximating posterior distributions of weights and biases, ADVI is used. After ADVI, posterior samples are drawn and used to compute the posterior mean estimates of the weights and intercept. PyMC package is used to implement Bayesian Logistic Regression model.
  Set-up: Stratified subset of size 10,000 is chosen out of which 70% is train data and 30% is test data. ADVI is run for 50,000 iterations to estimate the posterior.

Convergence Diagnostics: The convergence of ADVI is monitored through the ELBO loss. During training, the ELBO is plotted over iterations and the stabilizing ELBO curve indicates that the variational approximation has converged. Less data size(10000) has been chosen for faster convergence. Initially MCMC sampling was used to approximate posterior distribution, but was taking up a lot of time and not giving remarkably lesser loss. Thus ADVI was taken as a final choice.

- **Result:** The accuracy scored obtained for test data is 21%. The following figure shows: (i) ELBO loss as function of iterations and (ii) Actual VS predicted classes as viewed on tempo-speechiness space.



(a) ELBO loss VS iterations



(b) Actual VS Predicted classes

- **Why model doesn't work well:** Logistic regression assumes linearity in the log-odds, but the actual relationships between features and the target may be are non-linear. ADVI provides a variational approximation to the true posterior and it might give overly confident predictions. I tried various models for classification such as Bayesian LR with ADVI and MCMC, BNN, and Gaussian process. But every method gave nearly 20-22% accuracy. This might be due to lack of richness in data features, or noise. Music genres can be inherently overlapping or subjective, lacking clear, distinguishable characteristics in the feature space.
- **Advantages:** Provides probability distributions over parameters and predictions leading to well-grounded decisions. The use of informative priors can prevent overfitting in large feature setup.
- **Disadvantages:** ADVI or even MCMC are computationally expensive, especially for large datasets or high-dimensional parameter spaces. Variational inference method (here ADVI) can lead to biased estimates by underestimating the uncertainty in the posterior. The model's performance can be highly sensitive to the choice of priors. Poorly chosen priors can adversely affect the convergence and quality of the posterior estimates. Bayesian methods in general have poor scaling with increase in size of dataset. Due to linearity assumption, Bayesian LR may be ineffective in capturing complex, non-linear relations between features, unless we extend it with kernel methods or additional layers as in BNN.

6 Possible improvements

Additional audio and metadata features can be added to capture the factors affecting popularity and mood in a better way. Non-parametric models like Dirichlet Process GMM or t-SNE along with DBSCAN can be explored for mood clustering as these models have good capability to adapt to non-Gaussian distributions and unknown cluster counts. Also, ADVI can be replaced with Hamiltonian Monte Carlo for more accurate posterior estimates in BNN.

7 Conclusion

The project shows how probabilistic models can offer robust inferences into music data by capturing uncertainty and model variability in the Spotify dataset. The learning process aided in understanding the model diagnostics, hyperparameter tuning, and combining multiple modeling approaches to obtain better interpretability and results.

Self-Reflection: Through this project, I gained significant knowledge in data preprocessing, feature engineering, probabilistic modeling (including MCMC and variational inference), and model evaluation. I understood the importance of uncertainty quantification in real-world data analysis and usefulness of Bayesian approaches.