Music Genre Classification

Fan Wenyi
1023041004
Nanjing University of Posts and Telecommunications
Nanjing, China

***Abstract*—Music genre classification is a significant task in the field of audio signal processing and machine learning, aiming to automatically categorize music into predefined genres based on its acoustic features. This paper explores the application of various classification algorithms, including logistic regression and k-nearest neighbors, on a dataset containing diverse musical genres such as classical, jazz, country, pop, rock, and metal. We conduct extensive experiments to evaluate the performance of these algorithms, considering factors like accuracy, precision, and confusion matrices. Additionally, the study investigates the impact of different feature representations, such as Fast Fourier Transform (FFT) coefficients, on the classification accuracy. The results reveal the strengths and limitations of the selected algorithms in the context of music genre classification, providing insights into the optimal approaches for accurate genre prediction. This research contributes to the broader field of audio classification and offers practical implications for automated music organization and recommendation systems.**

***Index Terms*—music, classification, k-nearest, genre, logic regression**

I. Introduction

The dynamic landscape of digital music, characterized by an ever-expanding array of genres, necessitates sophisticated methods for effective music classification. As the volume of musical content continues to grow, the demand for automated systems capable of accurately categorizing songs becomes increasingly critical. Music genre classification, a discipline within the domains of audio signal processing and machine learning, assumes a central role in navigating and organizing this vast musical ecosystem.

Our study embarks on an exploration of the intricate domain of music genre classification, focusing on a diverse dataset that spans genres such as classical, jazz, country, pop, rock, and metal. The overarching goal is to harness the power of machine learning techniques, particularly logistic regression and k-nearest neighbors, to discern the efficacy of these algorithms in accurately categorizing music across diverse genres.

In addition to algorithmic scrutiny, we delve into the realm of feature representations, particularly emphasizing the influence of Fast Fourier Transform (FFT) coefficients. Understanding how different feature representations impact overall classification performance is pivotal for refining algorithms and enhancing the robustness of music classification systems.

The study not only seeks to unravel the technical intricacies of classification algorithms but also aims to shed light on the challenges and opportunities inherent in music genre classification. Through a comprehensive analysis, we aspire to offer valuable insights that contribute to the continual improvement of classification algorithms. Beyond the technical nuances, the research endeavors to enhance automated music recommendation systems, fostering a more personalized and enjoyable music exploration experience for users.

As we navigate through this multifaceted exploration, our research addresses fundamental questions surrounding the adaptability of machine learning algorithms to the intricacies of musical genres. By doing so, we aim to equip researchers, practitioners, and enthusiasts with a deeper understanding of the evolving landscape of music genre classification, fostering advancements that resonate with the diverse tastes of music enthusiasts globally.

II. Related Works

Music genre classification has been a prominent research area within the domains of audio signal processing and machine learning, spurred by the increasing availability of large-scale music dataset. Numerous studies have contributed to advancing the understanding and effectiveness of algorithms in this domain.

Early approaches often relied on handcrafted features derived from signal processing techniques, such as Mel-frequency cepstral coefficients (MFCCs)[3] and chroma features[6]. These features aimed to capture essential characteristics of the audio signal, enabling effective discrimination between genres. Noteworthy contributions include Tzanetakis and Cook's seminal work[7], which explored the application of statistical and spectral features for genre classification.

As the field progressed, machine learning techniques gained prominence, allowing for a more automated and data-driven approach to genre classification. Supervised learning algorithms, including support vector machines (SVMs)[2] and k-nearest neighbors (KNN)[4], have demonstrated success in discerning genre patterns within music data. Additionally, deep learning models, particularly convolutional neural networks (CNNs)[5] and recurrent neural networks (RNNs)[1], have shown promise in learning hierarchical representations from raw audio data, bypassing the need for handcrafted features.

Moreover, research has extended beyond traditional audio features, incorporating additional modalities such as lyrics, album metadata, and user listening patterns to enhance classification accuracy. The integration of multiple modalities has become a burgeoning area, reflecting the multidimensional nature of music classification.

Despite these advancements, challenges persist, including the inherent subjectivity of genre labels, the dynamic nature of musical styles, and the need for robustness to variations in recording quality. This section provides an overview of the diverse approaches and methodologies employed in music genre classification, laying the foundation for our exploration of novel strategies and insights in the subsequent sections of this paper.

III. Problem Statement

Music genre classification is inherently a multifaceted problem, as it involves categorizing songs into predefined genres based on their inherent audio characteristics. Formally, the task can be defined as follows.

A dataset consisting of audio samples from various songs. Each audio sample is associated with a predefined genre label. The task faces amount of challenges. High Dimensionality Challenge: Audio data is often represented in high-dimensional feature spaces, posing challenges in identifying relevant patterns and avoiding the curse of dimensionality.Subjectivity and Ambiguity Challenge: Genre classification is subjective and may vary among listeners. Some songs may exhibit characteristics of multiple genres, introducing ambiguity into the classification task.

Despite all the challenges, we have good hopes for his future.The model should be robust to variations in recording quality, background noise, and diverse musical styles to ensure real-world applicability. Capturing temporal dynamics in music, such as rhythm changes and transitions, is crucial for a holistic understanding of genre characteristics.The primary goal is to design a data mining solution that achieves high accuracy in predicting the genre labels of songs, while also addressing the aforementioned challenges. The solution should leverage advanced techniques in machine learning and signal processing to extract meaningful representations from the audio data.This formalization sets the stage for exploring innovative methodologies and algorithms to tackle the intricacies of music genre classification within the realm of data mining.

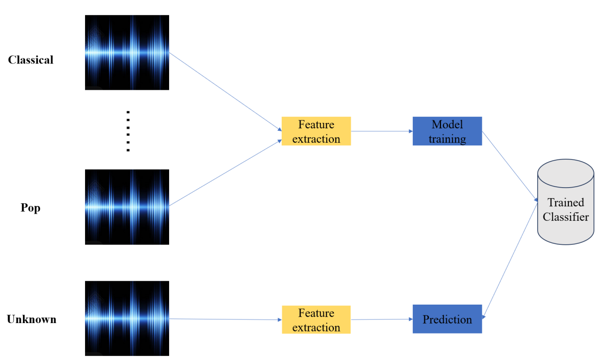


Fig. 1. Music classify pipeline

Algorithms

Addressing the intricacies of music genre classification involves the orchestration of advanced algorithms and innovative solutions. In our pursuit of an accurate and adaptable model, we employ a combination of feature extraction, machine learning, and signal processing techniques.

A. Feature Extraction

To distill meaningful information from the complex audio data, we employ feature extraction methods. These include the extraction of spectral features such as Mel-Frequency Cepstral Coefficients (MFCCs), Fast Fourier Transform (FFT), and Pitch. By capturing key characteristics of the audio signal, these features serve as the foundation for subsequent analysis.

The sound signal is a complex signal that consists of a number of sound waveforms of different frequencies that propagate together by causing perturbations to the medium (changing pressure) to form the sound we hear. When recording sound, we simply capture the synthesized amplitudes of these waveforms . The Fourier transform is a mathematical formula that allows us to break down a signal into individual frequencies and frequency amplitudes. In other words, it converts a signal from the time domain to the frequency domain, and the result is called a spectrum. This is possible because every signal can be broken down into a set of sine and cosine waves that add up to equal the original signal. This is a well-known theorem known as Fourier's theorem. The Fourier Transform is one of the most commonly used acoustic features and is an operation that maps an audio signal from the time domain to the frequency domain. In this paper, we mainly use FFT to extract the audio features (as shown in the visualization of Fig. 2).FFT is equivalent to transforming the signal from the temporal space, which has time as the coordinate, to the frequency space, which has FFT basis as the coordinate.The FFT algorithm is simple and suitable for application scenarios where CPU resources are tight, and the frequency components are better preserved, especially in the high-frequency part.

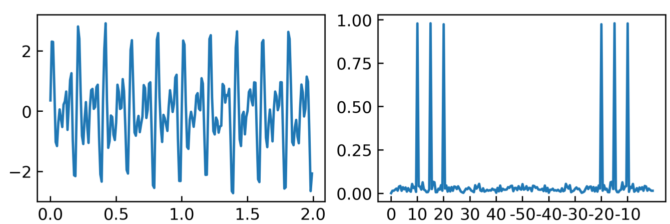


Fig. 2. Fast Fourier Transform feature

B. Machine Learning Models

Our approach embraces the power of machine learning models for classification tasks. Logistic Regression and k-Nearest Neighbors (KNN) models are utilized to discern patterns within the extracted features. Logistic Regression provides a probabilistic framework for multi-class classification, while KNN leverages proximity-based relationships to assign genres based on similarity to training instances.

Logistic regression was originally designed to deal with binary classification problems, but through clever morphing, it can also be applied to multi-category problems. For problems with multiple categories, logistic regression transforms the problem into multiple binary classification problems. Each category is treated as a binary classification task, and the goal of the model is to distinguish the current category from all other categories. For each category, a separate logistic regression model is trained. During the training of each model, samples from the target category are labeled as positive categories, while samples from all other categories are labeled as negative categories. In the prediction phase, input samples are fed into all trained logistic regression models. Each model outputs a probability value indicating the probability that the sample belongs to the current category. Eventually, the category with the highest probability is selected as the predicted category for the sample. The idea of this multi-categorization approach is similar to a hierarchical strategy, where each logistic regression model focuses on distinguishing one specific category, while all other categories are considered as a whole. This approach allows logistic regression to be easily extended to deal with multicategorization problems without the need to redesign the model structure.

The K-nearest neighbor method is an instance-based learning method that can be applied in multiple classification tasks as well. For each sample to be classified, the K nearest neighbors to that sample are found in the feature space. The distance is usually measured using Euclidean distance or other similarity measures. The number of samples belonging to each category in these K neighbors is counted. The majority voting principle is used to determine the category to which the samples to be classified belong, i.e., the category with the most votes is chosen as the prediction. In some variants of KNN, the weights of neighbors can be considered. Closer neighbors may be given higher weights to more strongly influence the voting result.The decision boundaries of KNN are nonlinear and adaptable to complex data distributions.The advantage of KNN is that it is simple and intuitive and does not require an explicit training process.

C. Model Training and Evaluation

In the intricate process of model training and evaluation, our focus extends beyond mere algorithmic implementation. The foundation of our study lies in the meticulous curation of a labeled dataset that faithfully captures the rich diversity of music genres. This dataset becomes the crucible in which our models are forged, offering them the opportunity to glean insights into the nuanced acoustic characteristics that distinguish various musical genres.

As our models traverse the intricacies of training, the learning process is akin to tuning an instrument, with each epoch refining the model's understanding of the intricate interplay of features within the music data. The goal is not just classification but a nuanced comprehension that transcends raw data – an understanding of the rhythmic cadence of jazz, the melodic richness of classical compositions, or the energetic pulse of rock.

To measure the efficacy of our models, a robust evaluation framework is imperative. The partitioning of the dataset into training and test subsets ensures a rigorous examination of the model's generalizability. Through this separation, we gauge the model's ability to extrapolate its learning beyond the confines of the training set, a pivotal characteristic for real-world applicability.

Performance metrics serve as our compass in navigating the evaluation landscape. Accuracy, a fundamental metric, quantifies the overall correctness of our model's predictions. Precision delves into the model's capability to avoid false positives, ensuring that when it asserts a genre classification, it does so with confidence.

As we venture into the realm of evaluation, our study aims not only to quantify performance but also to unravel the subtleties that underpin the success or challenges encountered. This involves a granular examination of the confusion matrix, unraveling instances where our models excel and areas where refinement may be required.

Through this holistic approach to model training and evaluation, our research seeks not just to build accurate classifiers but to contribute to the broader discourse on the application of machine learning in the domain of music genre classification. By delving into the intricacies of algorithmic decision-making, we strive to provide a comprehensive understanding that transcends numerical metrics and resonates with the nuanced artistry embedded in musical genres.

D. Model Adaptability

Given the dynamic nature of music genres and the constant changes in listener preferences, our model can be highly adaptive. In the future, regular updating and retraining of the model with new data can ensure that the model always adapts to emerging music genres and changes in musical styles.

In the symphony of algorithms and solutions, our methodology strives for a harmonious blend of precision, adaptability, and interpretability, paving the way for an enriched music genre classification experience.

IV. Evaluation

A. Dataset

The efficacy of our music genre classification system is contingent on the diversity and representativeness of the dataset employed. We curated a comprehensive dataset spanning genres such as classical, jazz, country, pop, rock, and metal. Each genre comprises 100 audio samples, ensuring a robust and balanced representation across the spectrum of musical styles.

The dataset encompasses variations in tempo, instrumentation, and tonality, mirroring the heterogeneity inherent in real-world musical collections. This diversity is crucial for training a model capable of generalizing well to unseen instances and addressing the challenge of genre ambiguity.

B. Experimental Results

We conducted experiments on music classification models using two algorithms, logistic regression and K-nearest neighbor method, and evaluated them on the same test set. The experimental results show that the model accuracy of the logistic regression algorithm is around 50 percent, while the K-nearest neighbor method achieves a model accuracy of around 67 percent. Clearly, the K-nearest neighbor method outperforms the logistic regression algorithm on this task.

However, despite the better performance of the K-nearest neighbor method, we also found that there is room for improvement in its model. To gain a deeper understanding of the classification results, we further calculated the confusion matrix (shown in Fig. 3). Observing the confusion matrix, we found that the music classification model of the logistic regression algorithm is prone to misclassification errors on the three categories of country, pop, and rock, incorrectly predicting them as other categories. In contrast, the model of the K-nearest neighbor method makes classification errors on the three categories jazz, country, and pop, incorrectly predicting them as other categories.

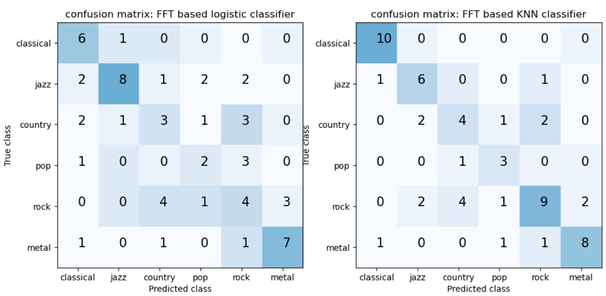


Fig. 3. Confusion Matrix

These observations provide valuable clues for further optimization and improvement of our music classification model, especially for the case of classification errors of both algorithms on specific categories.

V. Conclusion

In conclusion, the exploration of music genre classification within the context of data mining unveils a myriad of challenges and opportunities. As we traverse the landscape of audio data, the intricate nature of musical content, subjectivity in genre perception, and the dynamic interplay of various musical elements necessitate sophisticated data mining approaches.

Throughout this study, we have observed the significance of leveraging advanced techniques in machine learning, signal processing, and feature extraction to unravel patterns within high-dimensional audio data. The quest for a robust and accurate music genre classification model has driven the development of innovative methodologies capable of addressing challenges such as high dimensionality, subjectivity, and temporal dynamics.

However, it is crucial to acknowledge the evolving nature of music and the perpetual emergence of new genres. As the landscape of musical expression evolves, so too must our data mining models adapt to encompass a broader spectrum of genres and account for evolving listener preferences.

In the future, further research and development in music genre classification can benefit from interdisciplinary collaborations, incorporating insights from music theory, psychology, and human-computer interaction. The pursuit of real-world applicability demands continuous refinement of models, ensuring their adaptability to diverse musical styles, recording conditions, and evolving listener perceptions.

As we navigate the harmonious intersection of music and data mining, the journey unfolds not only as a technical pursuit but as a continuous exploration of the multifaceted nature of musical expression, enriching our understanding of both the art and science encapsulated within each note and genre label.

References

[1] Anirudh Ghildiyal, Komal Singh, and Sachin Sharma. “Music Genre Classification using Machine Learning”. In: *2020 4th International Conference on Electronics, Communication and Aerospace Technology (ICECA)*. 2020, pp. 1368–1372. DOI: 10.1109/ICECA49313.2020.9297444.

[2] De Rosal Ignatius Moses Setiadi et al. “Effect of Feature Selection on The Accuracy of Music Genre Classification using SVM Classifier”. In: *2020 International Seminar on Application for Technology of Information and Communication (iSemantic)*. 2020, pp. 7–11. DOI: 10.1109/iSemantic50169.2020.9234222.

[3] D Pradeep Kumar et al. “A comparative study of classifiers for music genre classification based on feature extractors”. In: *2016 IEEE Distributed Computing, VLSI, Electrical Circuits and Robotics (DISCOVER)*. 2016, pp. 190–194. DOI: 10.1109/DISCOVER.2016.7806258.

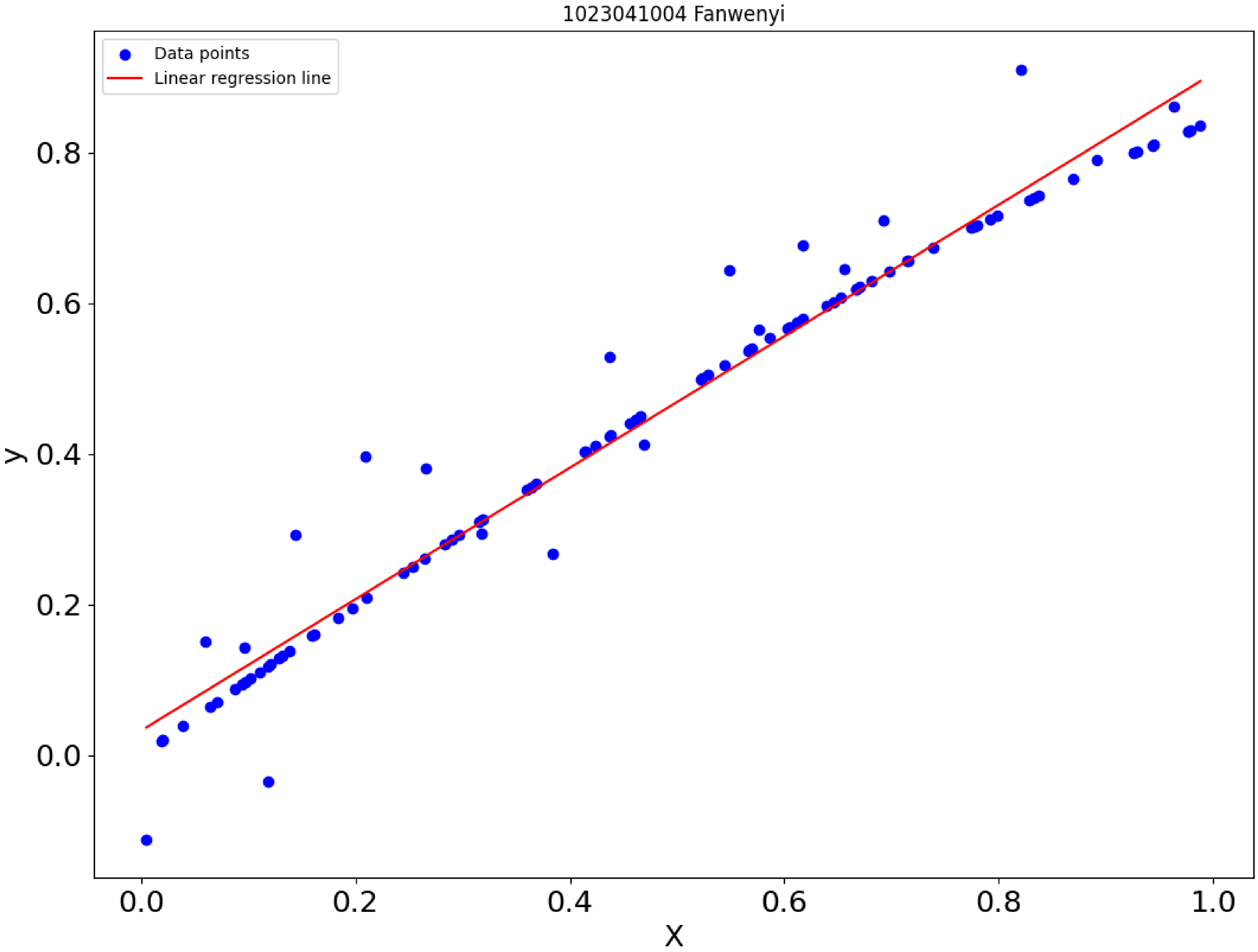
[4] Krittika Leartpantulak and Yuttana Kitjaidure. “Music Genre Classification of audio signals Using Particle Swarm Optimization and Stacking Ensemble”. In: *2019 7th International Electrical Engineering Congress (iEECON)*. 2019, pp. 1–4. DOI: 10.1109/iEECON45304.2019.8938995.

[5] Mitt Shah et al. “Music Genre Classification using Deep Learning”. In: *2022 6th International Conference on Computing Methodologies and Communication (ICCMC)*. 2022, pp. 974–978. DOI: 10.1109/ICCMC53470.2022.9753953.

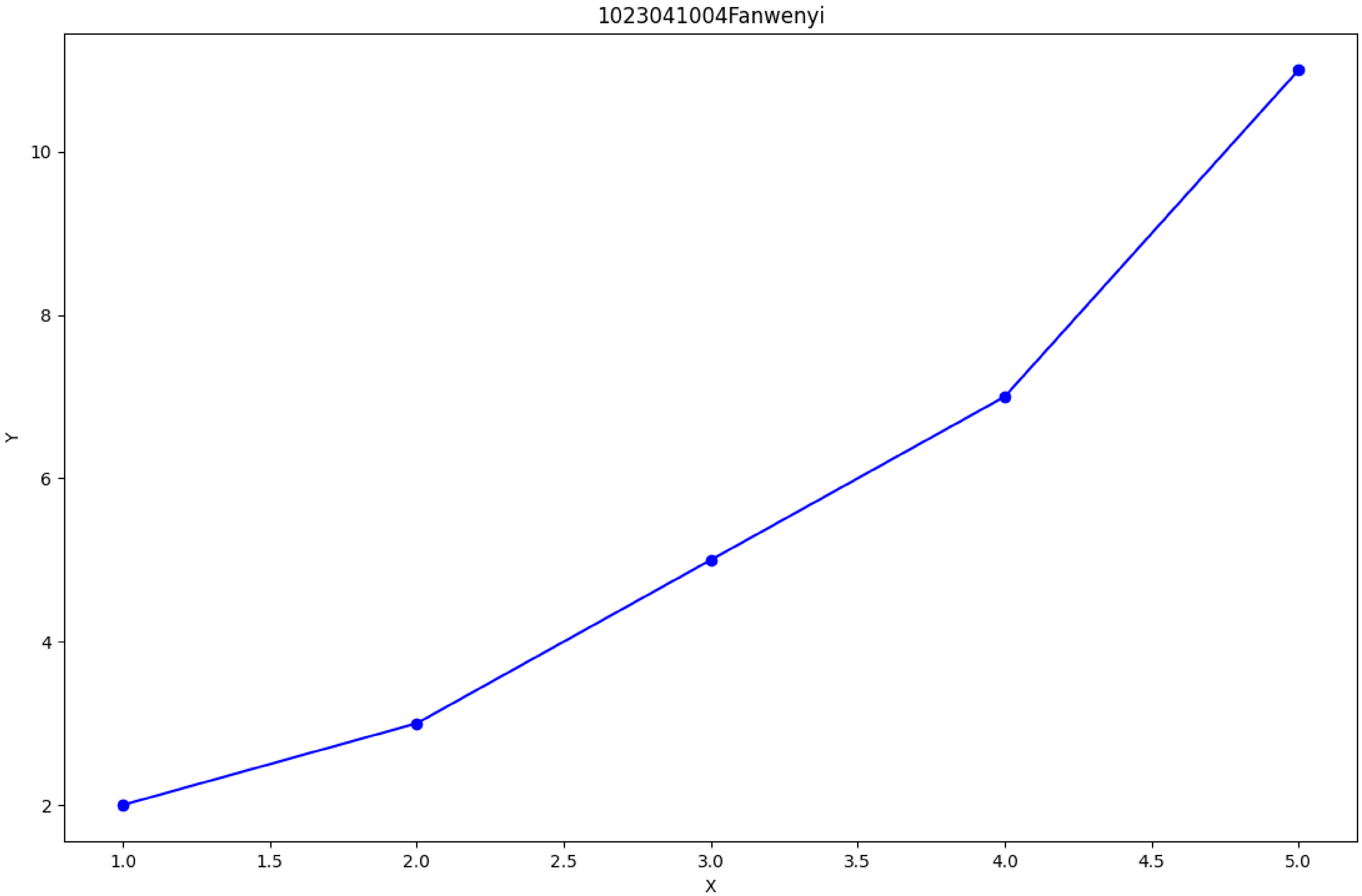
[6] Leisi Shi, Chen Li, and Lihua Tian. “Music Genre Classification Based on Chroma Features and Deep Learning”. In: *2019 Tenth International Conference on Intelligent Control and Information Processing (ICICIP)*. 2019, pp. 81–86. DOI: 10.1109/ICICIP47338.2019.9012215.

[7] G. Tzanetakis and P. Cook. “Musical genre classification of audio signals”. In: *IEEE Transactions on Speech and Audio Processing* 10.5 (2002), pp. 293–302. DOI: 10.1109/TSA.2002.800560.

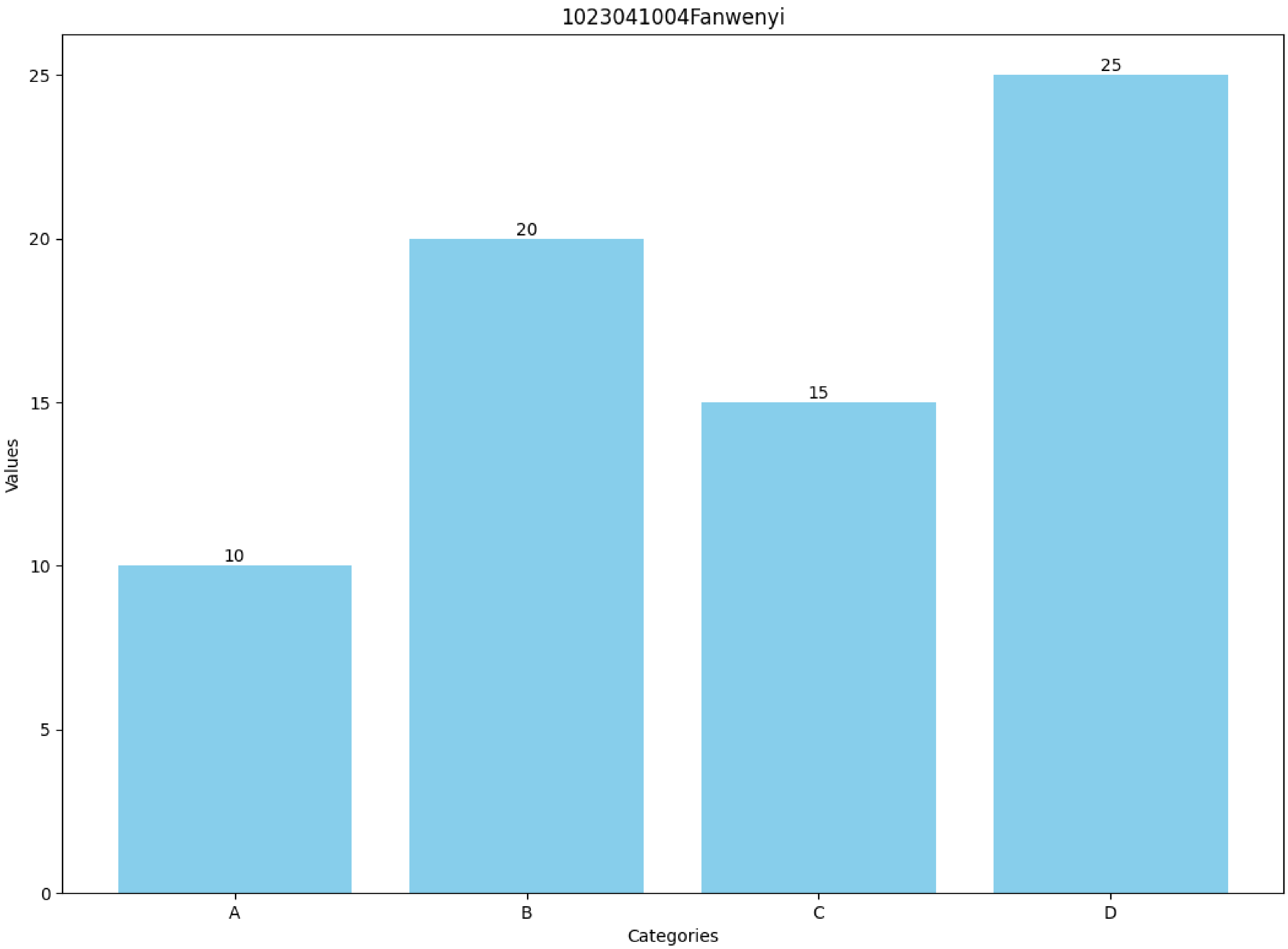
1


1023041004 Fanwenyi
Data points
Linear regression line
y
0.0
0.2
0.4
0.6
0.8
X
0.0
0.2
0.4
0.6
0.8
1.0

2



3


1023041004Fanwenyi
Values
Categories
25
20
15
10
5
0
A
B
C
D
10
20
15
25

多分类问题

科研方法与论文写作2024

范雯怡

南京邮电大学 计算机学院、软件学院、网络空间安全学院

汇报提纲

- 应用场景
- 问题定义
- 模型假设
- Softmax函数优缺
- Softmax与Logistic
- 学习准则
- 优化算法
- 代码实例

应用场景——多分类问题

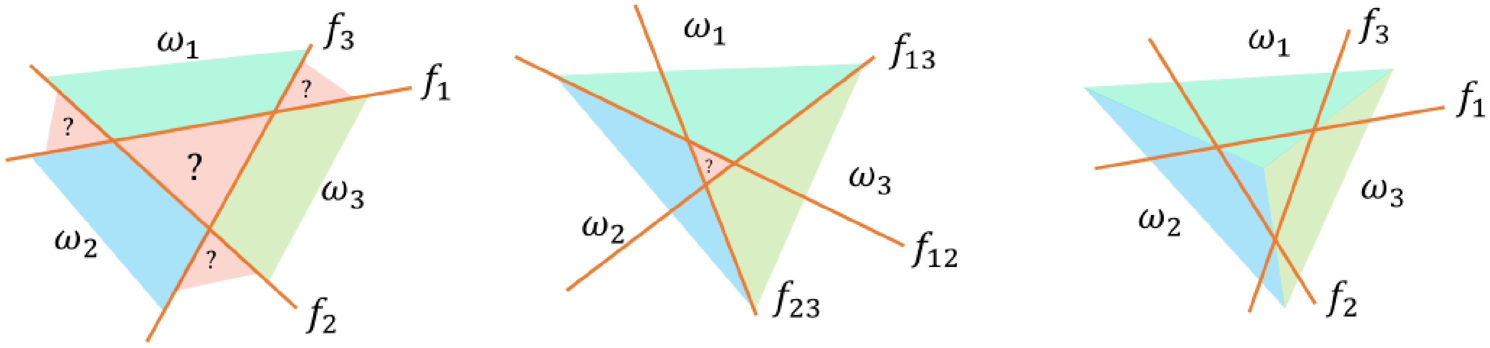
- 某个电子邮件属于垃圾邮件文件夹、工作文件夹还是资讯文件夹?
- 某个声音是儿童、成人还是老人?
- 某个图像描绘的是驴、狗、猫、还是鸡?
- 某个电影属于科幻片还是古装片还是现代片?

问题定义

➢ 给定一个包含N个训练样本的训练集 $\{(x^{(n)},y^{(n)})\}_{n=1}^{N}$ ，其中 $x\epsilon\mathbb{R}^{D}$ ，$y\epsilon\{1,2,.....,C\}$ ，C是一个常数。

也就是需要对这些样本x进行分类，这些样本分别属于不同的C个类别

➢ 多分类问题有3种常见的设计判别函数的方式，而Softmax回归采用第三种方式



(a) “一对其余”方式　(b) “一对一”方式　(c) “argmax”方式

问题定义

➢ Softmax回归基于argmax方式，需要区分C个类别就使用C个判别函数

"*argmax*" 方式：这是一种改进的 "一对其余" 方式，共需要 C 个判别函数

$$f_c(\mathbf{x}; \mathbf{w}_c) = \mathbf{w}_c^{\mathrm{T}} \mathbf{x} + b_c, \qquad c = [1, \cdots, C] \tag{3.10}$$

如果存在类别 c，对于所有的其他类别 $\tilde{c}(\tilde{c} \neq c)$ 都满足 $f_c(\mathbf{x}; \mathbf{w}_c) > f_{\tilde{c}}(\mathbf{x}, \mathbf{w}_{\tilde{c}})$，那么 $\mathbf{x}$ 属于类别 c。即

$$y = \arg\max_{c=1}^{C} f_c(\mathbf{x}; \mathbf{w}_c). \tag{3.11}$$

模型假设

- 首先假设有C个线性模型 $f_c(x;w_c) = w_c^T x \qquad c\epsilon\{1,...,C\}$
  即C个判别函数
  这样一个样本x作为输入，可以得到C个结果。

  其中w_c是第c类的权重向量

- 将分类问题看作条件概率估计问题

  但是判别函数值域为R，需要建模成条件概率的形式 $p_\vartheta(y=c|x)$
  即给定x后y=c的条件概率，满足值域在[0，1]之间，并且c的所有取值对应的条件概率和为1.
  所以现在要把一个实数区间的值转换成一个概率分布就依靠Softmax函数

- Softmax函数

  对于K个标量$x_1,\ldots,x_K$，转换成值域[0,1],和为1的值

$$Soft\max(x_k) = \frac{\exp(x_k)}{\sum_{i=1}^{K}\exp(x_i)}$$

  使用Softmax函数就可以将K个标量转换成具有K个取值的分布

模型假设

- 综上，给定一个样本$\boldsymbol{x}$， Softmax回归预测的属于类别c的条件概率为：

$$p(y=c|\boldsymbol{x}) = \mathrm{softmax}(\boldsymbol{w}_c^{\top}\boldsymbol{x}) = \frac{\exp(\boldsymbol{w}_c^{\top}\boldsymbol{x})}{\sum_{c'=1}^{C}\exp(\boldsymbol{w}_{c'}^{\top}\boldsymbol{x})}$$

因为用Softmax函数构建了类别的条件概率，所以这个模型称为Softmax回归

- 还可以写成向量形式

$$\hat{\boldsymbol{y}} = \mathrm{softmax}(\boldsymbol{W}^{\top}\boldsymbol{x}) = \frac{\exp(\boldsymbol{W}^{\top}\boldsymbol{x})}{\mathbf{1}_C^{\top}\exp(\boldsymbol{W}^{\top}\boldsymbol{x})}$$

$W = [w_1, ..., w_C]$

1_C为C维的全一向量

这里左乘全一向量相当于在做求和操作

这个$\hat{y} \in \mathbb{R}^C$,是所有类别的预测条件概率组成的向量，第c维的值是第c类的预测条件概率

模型假设

在上面基础上决策函数可以表示为：

$$\hat{y} = \arg\max_{c=1}^{C} p(y = c|\boldsymbol{x})$$
$$= \arg\max_{c=1}^{C} \boldsymbol{w}_c^{\mathrm{T}}\boldsymbol{x}.$$

选择拥有最大条件概率的类别，作为样本x的最终预测类别。

结果是一个标量，表示样本x对应的类别索引。

Softmax函数优缺

使用Softmax函数完成对判别函数的值域约束，使得结果能以概率的形式出现具有一定的好处：

1. 函数中引入了指数函数，因为指数函数曲线呈现递增趋势，最重要的是斜率逐渐增大，所以在x轴上一个很小的变化，可以导致y轴上很大的变化。这种函数曲线能够将输出的数值拉开距离。
2. 而且在深度学习中通常使用反向传播求解梯度进而使用梯度下降进行参数更新的过程，而指数函数在求导的时候比较方便。

同时也带来了缺点：

因为使用了指数函数，所以若值非常大的话，计算得到的数值也会变的非常大，数值可能会溢出。

有两种解决方法：

1. 所有输入送入Softmax函数之前，先减去所有输入中最大的元素
2. 在经验风险函数中添加正则化项，并不会影响最终结果，而且可以避免数值溢出和过拟合

Softmax与logistic

如果用Softmax回归解决二分类问题，与Logistic回归有什么区别?

1. 判别函数个数。Logistic回归用1个，Softmax回归用2个
2. 概率化时选择的函数不同，一个用Logistic函数，一个用Softmax函数
3. 决策函数有不同。

Softmax的决策函数:

$$\hat{y} = \arg\max_{y\in\{0,1\}} \boldsymbol{w}_y^{\mathrm{T}}\boldsymbol{x}$$

$$= I\left(\boldsymbol{w}_1^{\mathrm{T}}\boldsymbol{x} - \boldsymbol{w}_0^{\mathrm{T}}\boldsymbol{x} > 0\right)$$

$$= I\left((\boldsymbol{w}_1 - \boldsymbol{w}_0)^{\mathrm{T}}\boldsymbol{x} > 0\right)$$

Logistic的决策函数:

$$\hat{y} = I(w^T x > 0)$$

Softmax与logistic

解决多分类问题用多个Logistic回归好，还是用Softmax回归？

- 需要判断类别之间是否有互斥

  例如，如果有四个类别的音乐，分别为：古典音乐、乡村音乐、摇滚乐和爵士乐，那么可以假设每个训练样本只会被打上一个标签（即：一首歌只能属于这四种音乐类型的其中一种），此时使用Softmax 回归比较好。

  如果四个类别如下：人声音乐、舞曲、影视原声、流行歌曲，那么这些类别之间并不是互斥的。例如：一首歌曲可以来源于影视原声，同时也包含人声 。这种情况下，使用 4 个二分类的 Logistic 回归分类器更为合适。这样，对于每个新的音乐作品 ，我们的算法可以分别判断它属于哪些类别。

学习准则

因为要学习，所以先要构建损失函数。这里建立参数化的条件概率和真实条件概率之间的交叉熵，公式如下：

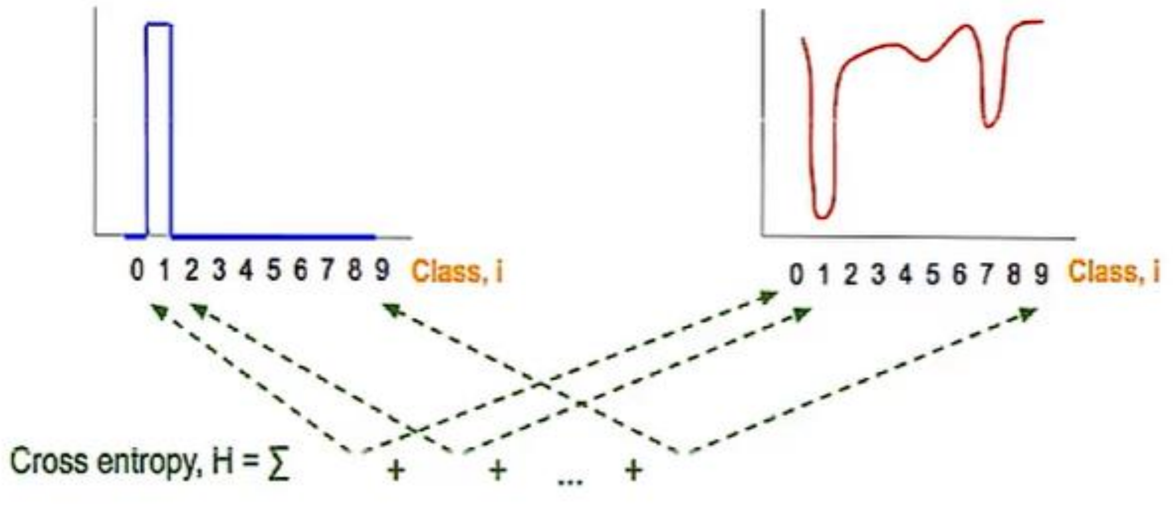
针对某个样本x的交叉熵损失函数：

$$-\sum_{c=1}^{C} y_c^{(n)} \log \hat{y}_c^{(n)}$$

其中$y_c^{(n)}$表示样本x属于类别c的真实概率

其中$\hat{y}_c^{(n)}$表示样本x预测到属于类别c的概率

即存在一个真实概率分布和一个预测概率的负对数分布，将他们对应相乘后再相加而得



学习准则

以某一个样本的标签和预测结果为例，演示交叉熵损失函数的具体计算公式和结果:

- 对于一个三类分类问题，类别为[0,0,1]，预测类别概率为[0.3,0.3,0.4]，则

$$
\begin{aligned}
\mathcal{L}(\theta) &= -\big(0 \times \log(0.3) + 0 \times \log(0.3) + 1 \times \log(0.4)\big) \\
&= -\log(0.4).
\end{aligned}
$$

学习准则

- 为了方便，下面会借助C维的one-hot向量$\boldsymbol{y} \in \{0,1\}^C$来表示类别标签．如果标签属于类别$c$，标签y向量表示为：

$$\boldsymbol{y} = [I(1 = c), I(2 = c), \cdots, I(C = c)]^{\mathsf{T}}$$

- 采用交叉熵损失函数，Softmax回归模型的经验风险函数为：

$$\begin{aligned}\mathcal{R}(\boldsymbol{W}) &= -\frac{1}{N}\sum_{n=1}^{N}\sum_{c=1}^{C} \boldsymbol{y}_c^{(n)} \log \hat{\boldsymbol{y}}_c^{(n)} \\ &= -\frac{1}{N}\sum_{n=1}^{N} (\boldsymbol{y}^{(n)})^{\mathsf{T}} \log \hat{\boldsymbol{y}}^{(n)},\end{aligned}$$

其中$\hat{\boldsymbol{y}}^{(n)} = \mathrm{softmax}(\boldsymbol{W}^{\mathsf{T}}\boldsymbol{x}^{(n)})$为样本$\boldsymbol{x}^{(n)}$在每个类别的后验概率

$y^{(n)}$是样本$x^{(n)}$对应的$onehot$类别标签

优化算法

风险函数有了之后就可以使用优化算法进行参数学习，这里主要采用梯度下降

首先计算得到风险函数$\mathcal{R}(\boldsymbol{W})$关于$\boldsymbol{W}$的梯度为：

$$\frac{\partial \mathcal{R}(\boldsymbol{W})}{\partial \boldsymbol{W}} = -\frac{1}{N}\sum_{n=1}^{N} \boldsymbol{x}^{(n)}\left(\boldsymbol{y}^{(n)} - \hat{\boldsymbol{y}}^{(n)}\right)^{\mathrm{T}}$$

$y^{(n)}$是样本$x^{(n)}$对应的$onehot$类别标签

$\hat{y}^{(n)}$是样本$x^{(n)}$在每个类别预测的条件概率

梯度公式的推导过程见下页

优化算法

先计算每个样本对应的损失函数关于参数W的梯度，稍作更改后就可以得到上述风险函数关于W的梯度值

前提假设：① $y=\mathrm{softmax}(z)$ 那么 $\frac{\partial y}{\partial z}=\mathrm{diag}(y)-yy^T$

② $z=W^Tx=[w_1^Tx, w_2^Tx, \dots, w_C^Tx]^T$ 那么 $\frac{\partial z}{\partial w_c}=[\frac{\partial w_1^Tx}{\partial w_c}, \frac{\partial w_2^Tx}{\partial w_c}, \dots, \frac{\partial w_C^Tx}{\partial w_c}]$

$=[0,0,\dots,x,\dots,0]$

$\triangleq \mathbb{M}_c(x)$ 是个第c列为x，其余为0的矩阵

$$\frac{\partial z^{(n)}}{\partial w_c}=\mathbb{M}_c(x^{(n)}) \qquad \frac{\partial \hat{y}^{(n)}}{\partial z^{(n)}}=\mathrm{diag}(\hat{y}^{(n)})-\hat{y}^{(n)}\hat{y}^{(n)T}$$

$$\therefore \frac{\partial \mathcal{L}^{(n)}(W)}{\partial w_c}=-\frac{\partial((y^{(n)})^T\log\hat{y}^{(n)})}{\partial w_c}$$

$$=-\frac{\partial z^{(n)}}{\partial w_c}\frac{\partial \hat{y}^{(n)}}{\partial z^{(n)}}\frac{\partial \log\hat{y}^{(n)}}{\partial \hat{y}^{(n)}}y^{(n)}$$

上下同乘 $\partial z^{(n)}$、$\partial\hat{y}^{(n)}$ ∵ $y^{(n)}$ 是标签与 w_c 无关 ∴可以直接从∂中提出来

$$=-\mathbb{M}_c(x^{(n)})\left(\mathrm{diag}(\hat{y}^{(n)})-\hat{y}^{(n)}(\hat{y}^{(n)})^T\right)\left(\mathrm{diag}(\hat{y}^{(n)})\right)^{-1}y^{(n)}$$

$$=-\mathbb{M}_c(x^{(n)})\left(I-\hat{y}^{(n)}(\hat{y}^{(n)})^T(\mathrm{diag}(\hat{y}^{(n)}))^{-1}\right)y^{(n)}$$

∵ $y^T\mathrm{diag}(y)^{-1}=1_C^T$ 全1行向量

$$=-\mathbb{M}_c(x^{(n)})\left(I-\hat{y}^{(n)}1_C^T\right)y^{(n)}$$

$$=-\mathbb{M}_c(x^{(n)})\left(y^{(n)}-\hat{y}^{(n)}1_C^Ty^{(n)}\right)$$

∵ $y^{(n)}$ 是one-hot列向量 ∴ $1_C^Ty^{(n)}=1$

$$=-\mathbb{M}_c(x^{(n)})\left(y^{(n)}-\hat{y}^{(n)}\right)$$

$$=-x^{(n)}[y^{(n)}-\hat{y}^{(n)}]_c$$

因此 $\frac{\partial \mathcal{L}^{(n)}(W)}{\partial W}=-x^{(n)}(y^{(n)}-\hat{y}^{(n)})^T$

优化算法

采用梯度下降算法时，Softmax回归的训练过程为，先初始化W_0 ,然后用下面的式子进行迭代更新：

$$\boldsymbol{W}_{t+1} \leftarrow \boldsymbol{W}_t + \alpha \left(\frac{1}{N} \sum_{n=1}^{N} \boldsymbol{x}^{(n)} \left(\boldsymbol{y}^{(n)} - \hat{\boldsymbol{y}}_{\boldsymbol{W}_t}^{(n)} \right)^{\top} \right)$$

其中α为学习率，$\hat{y}_{W_t}^{(n)}$是参数为W_t时，$Soft\max$回归模型的输出

代码实例

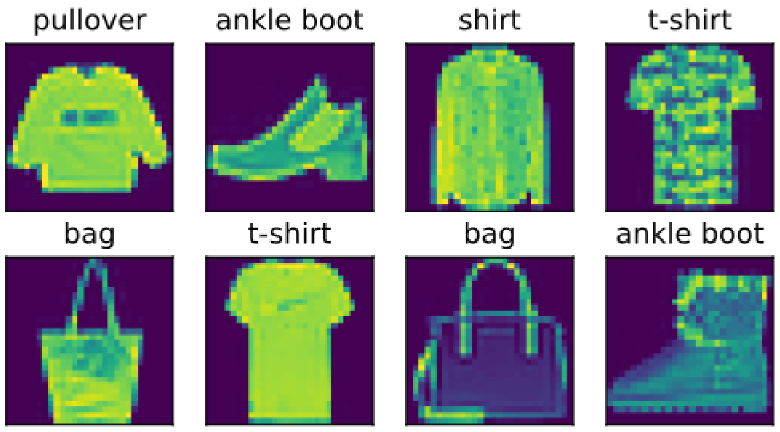
利用Fashion-MNIST服装分类数据集，训练一个多分类模型，数据集包含的10个类别，分别为t-shirt、trouser、pullover、dress、coat、sandal、shirt、sneaker、bag和ankle boot

```python
# 读取训练集和测试集
batch_size = 256
train_iter, test_iter = d2l.load_data_fashion_mnist(batch_size)

# softmax函数
def softmax(X):
    X_exp = torch.exp(X)
    partition = X_exp.sum(1, keepdim=True)  # 矩阵的每一行求和，重新生成新的矩阵
    return X_exp / partition  # 结果中每一行代表一个样本，行中的每个数据代表在该类别的概率，

# 定义模型，
W = torch.normal(0, 0.01, size=(num_inputs, num_outputs), requires_grad=True)
b = torch.zeros(num_outputs, requires_grad=True)
def net(X):
    return softmax(torch.matmul(X.reshape((-1, W.shape[0])), W) + b)

# 交叉熵损失函数
def cross_entropy(y_hat, y):
    return - torch.log(y_hat[range(len(y_hat)), y])
```



代码实例

```python
# 优化算法
def updater(batch_size):
    return d2l.sgd([W, b], lr, batch_size)

# 计算分类精度:正确预测数量与总预测数量之比，作为评价标准
"""..."""
def accuracy(y_hat, y):
    """计算预测正确的数量"""
    if len(y_hat.shape) > 1 and y_hat.shape[1] > 1:
        y_hat = y_hat.argmax(axis=1)  #  每一行中元素最大的那个下标存在
    cmp = y_hat.type(y.dtype) == y  # #将y_hat转换为y的数据类型然后作比
    return float(cmp.type(y.dtype).sum())
```

训练好后，对一些样本的预测结果如下：