UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
DEPARTAMENTO ACADÊMICO DE COMPUTAÇÃO
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

VÍTOR YUDI SHINOHARA

CLASSIFICAÇÃO AUTOMÁTICA DE MÚSICA UTILIZANDO APRENDIZAGEM DE PADRÕES DE VOTAÇÃO

MONOGRAFIA

CAMPO MOURÃO
2018

VÍTOR YUDI SHINOHARA

CLASSIFICAÇÃO AUTOMÁTICA DE MÚSICA UTILIZANDO APRENDIZAGEM DE PADRÕES DE VOTAÇÃO

Trabalho de Conclusão de Curso de graduação apresentado à disciplina de Trabalho de Conclusão de Curso 2, do Curso de Bacharelado em Ciência da Computação do Departamento Acadêmico de Computação da Universidade Tecnológica Federal do Paraná, como requisito parcial para obtenção do título de Bacharel em Ciência da Computação.

Orientador: Prof. Me. Juliano Henrique Foleiss e Prof. Dr. Rodrigo Hübner

CAMPO MOURÃO
2018

ATA DE DEFESA DO TRABALHO DE CONCLUSÃO DE CURSO

Às **17:30** do dia **21 de novembro de 2018** foi realizada na sala **E103** da UTFPR-CM a sessão pública da defesa do Trabalho de Conclusão do Curso de Bacharelado em Ciência da Computação do(a) acadêmico(a) **Vítor Yudi Shinohara** com o título **Classificação automática de música utilizando aprendizagem de padrões de votação**. Estavam presentes, além do(a) acadêmico(a), os membros da banca examinadora composta por: **Prof. Dr. Rodrigo Hübner** (orientador), **Profa. Dra. Aretha Barbosa Alencar** e **Prof. Ms. Luciano Fiorin Junior**. Inicialmente, o(a) acadêmico(a) fez a apresentação do seu trabalho, sendo, em seguida, arguido(a) pela banca examinadora. Após as arguições, sem a presença do(a) acadêmico(a), a banca examinadora o(a) considerou _____ na disciplina de Trabalho de Conclusão de Curso **2** e atribuiu, em consenso, a nota _____ (_____). Este resultado foi comunicado ao(à) acadêmico(a) e aos presentes na sessão pública. A banca examinadora também comunicou ao acadêmico(a) que este resultado fica condicionado à entrega da versão final dentro dos padrões e da documentação exigida pela UTFPR ao professor Responsável do TCC no prazo de **onze dias**. Em seguida foi encerrada a sessão e, para constar, foi lavrada a presente Ata que segue assinada pelos membros da banca examinadora, após lida e considerada conforme.

Observações: _____

_____

_____

_____

_____

_____

Campo Mourão, **21 de novembro de 2018**

Profa. Dra. Aretha Barbosa Alencar
Membro 1

Prof. Ms. Luciano Fiorin Junior
Membro 2

Prof. Dr. Rodrigo Hübner
Orientador

A ata de defesa assinada encontra-se na coordenação do curso.

Resumo

Shinohara, Vítor Yudi. Classificação Automática de Música Utilizando Aprendizagem de Padrões de Votação. 2018. 51. f. Monografia (Curso de Bacharelado em Ciência da Computação), Universidade Tecnológica Federal do Paraná. Campo Mourão, 2018.

Pesquisas na área de *Music Information Retrieval* (MIR) tem proposto métodos de classificação automática de gêneros musicais usando aprendizagem de máquina. Neste contexto surgiram duas abordagens para representação de faixas de áudio: *Single Vector Representation* (SVR), compostas por apenas um vetor de características e *Multiple Vector Representation* (MVR), que usa múltiplos vetores na descrição. Para a classificação usando MVR, a faixa é dividida em trechos, denominados texturas, e todos as texturas são rotuladas com o gênero da faixa e apresentadas ao modelo na fase de treino. Para a classificação, cada textura é classificado independentemente, logo, deve se aplicar uma técnica de votação para atribuir um rótulo à faixa completa.
As técnicas utilizadas para inferir um rótulo à faixa quando se tem apenas os votos de cada textura são limitadas à votação majoritária simples. Para o uso da votação majoritária ponderada é necessário uma distribuição de probabilidades de cada rótulo para cada textura, cujo custo computacional para estimar é usualmente elevado, e portanto nem sempre as probabilidades estão disponíveis.
São propostos dois métodos de combinação de votos alternativos ao voto majoritário simples. Ambos métodos utilizam aprendizagem de padrões de votação onde apenas os votos de texturas são conhecidos, ao invés de distribuições de probabilidade.
Os votos foram combinados de duas maneiras: Composição de um histograma de votos e composição de um vetor de sequência de votos. Os histogramas foram submetidos aos classificadores K-Vizinhos Mais Próximos (K-NN) e Máquina de Vetores de Suporte (SVM). Os vetores de sequência de votos foram submetidos aos classificadores *Hidden Markov Model* (HMM) e duas arquiteturas de redes neurais recorrentes.
Foram computados a acurácia e o desvio padrão da acurácia da classificação em gêneros musicais obtidos pelos dois métodos propostos. O desempenho dos métodos propostos foram comparados com os resultados obtidos pelo voto majoritário simples. O teste estatístico *T de Student* foi usado para avaliar quais foram os ganhos estatisticamente significativos. Os resultados mostram que a aprendizagem de padrões de votação é relevante e pode trazer

ganhos estatisticamente significativos em alguns conjuntos de dados.
Palavras-chaves: *Music Information Retrieval*. Classificação Automática de Música. Aprendizagem de Máquina. Técnicas de Votação. Aprendizagem de Padrões de Votação.

Abstract

Shinohara, Vítor Yudi. Automatic Classification of Music Using Voting Pattern Learning. 2018. 51. f. Monograph (Undergradute Program in Computer Science), Federal University of Technology – Paraná. Campo Mourão, PR, Brazil, 2018.

Research in MIR have proposed many automatic genre classification systems using machine learning. In this context, two main approaches have been used to describe music tracks: *Single Vector Representation* (SVR), which uses a single vector, and *Multiple Vector Representation* (MVR), which uses multiple vectors. For training MVR models, the track is divided into auditory textures, which are all labeled with the ground truth and presented independently during training. When testing, each texture is classified independently, and then some voting scheme must be used to assign a final label to the entire track.
When only the votes are available for each texture, the only option available to assign a final label to the track is majority voting. Other techniques which rely on class probabilities for each texture requires those probabilities to be computed by the classifier during prediction. These probabilities are costly to compute, thus are not always available.
We present two novel voting schemes as alternatives to majority voting. Both methods use voting pattern learning where only the predicted class of each texture is known, not class probability distributions.
The votes were combined in two different ways: a voting histogram and a vote sequence vector. The histograms were used as feature vectors for both K-Vizinhos Mais Próximos (K-NN) and Máquina de Vetores de Suporte (SVM) classifiers. The vote sequence vectors were used as inputs to sequence modelling with *Hidden Markov Model* (HMM) and two recurrent neural network architectures.
The accuracy and accuracy standard deviation of the classification were computed for both proposed methods. The performance of both were compared to the results of the majority voting technique. Student's T-Test was used to evaluate which gains were statistically significant. The results show that voting pattern learning is relevant and can provide statistically significant performance gains in some data sets.
Keywords: Music Information Retrieval. Automatic Music Classification. Machine Learning. Voting Schemes. Voting Pattern Learning.

Lista de figuras

Lista de tabelas

Siglas

CQT: *Constant Q Transform*
EM: *Expect Maximization*
GLCM: *gray level co-occurrence matrix*
HMM: *Hidden Markov Model*
K-NN: K-Vizinhos Mais Próximos
LDA: *Linear Discriminant Analysis*
LMD: *Latin Music Database*
LSTM: *Long Short Term Memory*
MIR: *Music Information Retrieval*
MLP: *Multilayer Perceptron*
MVR: *Multiple Vector Representation*
PSD: *Predictive Sparse Decomposition*
SAPV: Sistema de aprendizagem de padrões de votação
SCTM: Sistema classificador de texturas musicais
SVM: Máquina de Vetores de Suporte
SVR: *Single Vector Representation*

Sumário

Capítulo 1

Introdução

Uma área de pesquisa responsável pela extração de informações de faixas de áudio denominada *Music Information Retrieval* (MIR), tem aplicabilidade nos ramos de psicologia, musicologia e informática em geral. No ramo de tecnologia, o campo vem ganhando maior destaque entre os pesquisadores, acadêmicos e até indústria devido a sua utilidade. Existem diversos usos, tais quais, sistemas de recomendação, categorização em gêneros musicais, geração de música, transcrição automática de áudio, detecção, reconhecimento de instrumentos presentes, entre outros (CASEY et al., 2008).

O problema de classificação de gêneros musicais consiste em determinar o gênero musical de determinada faixa de áudio (e.g. pop, rock, blues, valsa) por meio do sinal de áudio analisado. Este problema vem ganhando importância devido à grande quantidade de conteúdo multimídia e no crescimento das coleções musicais presentes na *Web* (TZANETAKIS; COOK, 2002). As técnicas desenvolvidas facilitam o gerenciamento, organização e rotulação deste conteúdo.

Embora pode se contar com certos padrões, gêneros musicais não são bem definidos. É comum que especialistas em música cheguem em diferentes conclusões na rotulação de uma faixa em um gênero. Uma justificativa é que gêneros distintos compartilham mesmas características, como ritmo ou instrumento musical utilizado. Além disto, não existe um modelo exato de como criar músicas de cada gênero. Como não há um modelo formal, não há como criar algoritmos exatos pra realizar essa classificação.

Uma faixa musical pode ser representada por diferentes conjuntos de características. Espera-se que os conjuntos de características sejam capazes de descrever os aspectos musicais que podem ser usados para agrupar exemplos do mesmo rótulo, ao mesmo tempo que sirvam para separar exemplos de rótulos diferentes. No caso da música, as características musicais como timbre, ritmo, tempo e harmonia são difíceis de modelar matematicamente. A "distância" entre estes conceitos e os modelos matemáticos que tentam retratá-los é conhecida como

abismo semântico, ou *gap* semântico. Além disto, a relação entre os modelos matemáticos para descrever estas características e os rótulos musicais, neste caso gênero musical, também são desconhecidas ou de difícil modelagem. Desta forma, o uso de aprendizagem de máquina é adequado para realização do mapeamento entre as características e os rótulos musicais que se deseja classificar.

Existem duas formas de representação de faixas em um sistema de classificação de gênero baseado em aprendizagem de máquina. A primeira, denominada *Single Vector Representation* (SVR), consiste em obter o vetor de características através da média das características de todo o trecho. Para esta abordagem, é utilizado um vetor para treino, e somente um é usado no teste. Outra maneira é representar a música em diversos vetores de características, sendo titulada *Multiple Vector Representation* (MVR). Neste caso, um conjunto dos vetores da faixa é utilizada durante o treino e um conjunto dos vetores é utilizado no teste. No entanto, como no teste são usados vários vetores, é necessário combinar as predições de todos de uma faixa para obter uma predição final. Vários esquemas de voto são usados atualmente para fazer essa combinação. Trabalhos recentes discutem a importância e as vantagens de representações MVR, que são capazes de atingir resultados estado-da-arte usando características simples (FOLEISS, 2018).

Entretanto, as opções de votação quando se tem apenas uma predição por vetor, e não uma distribuição de probabilidade para as possíveis classes de cada vetor, são limitadas, sendo o voto majoritário empregado nestes cenários (HENAFF et al., 2011). Neste trabalho, são propostas alternativas ao voto majoritário, amplamente utilizado quando apenas os votos de trechos são conhecidos.

Esta monografia está estruturada da seguinte maneira: No Capítulo 2 são abordados conceitos fundamentais para o entendimento da pesquisa e diferentes metodologias para se realizar a classificação automática de gênero. No Capítulo 3 são apresentados trabalhos na área de MIR abordando a classificação automática de gênero utilizando abordagens distintas. No Capítulo 4 são apresentados procedimentos realizados para execução da pesquisa, por fim, no Capítulo 5 são apresentados os resultados dos experimentos realizados.

Capítulo 2

Fundamentação Teórica

Neste capítulo são apresentados conceitos e técnicas utilizadas nesta pesquisa encontradas na literatura de aprendizagem de máquina e processamento de sinais.

2.1. Técnicas de Votação

Utilizadas para os mais diversos fins, as técnicas de votação são empregadas com a finalidade de eleger um opção entre várias com base em uma regra ou mais. Em um cenário real, empregada na eleição de presidente por exemplo, os eleitores escolhem uma dentre, ao menos, duas opções. O candidato com mais votos é eleito. Estas técnicas são empregadas também na computação, como no campo de sistemas distribuídos (PARIS; LONG, 1988), banco de dados (JAJODIA; MUTCHLER, 1990), aprendizagem de máquina, entre outros.

2.1.1. Votação Majoritária Simples

A política de votação majoritária simples ou voto da maioria, consiste em eleger um elemento que obteve a maior quantidade de votos em um cenário, de forma que todos os elementos tenham o mesmo peso. Neste contexto, existem 3 casos que podem ocorrer na votação majoritária, ilustrados na Figura 2.1. O resultado neste cenário para as situações é ■.

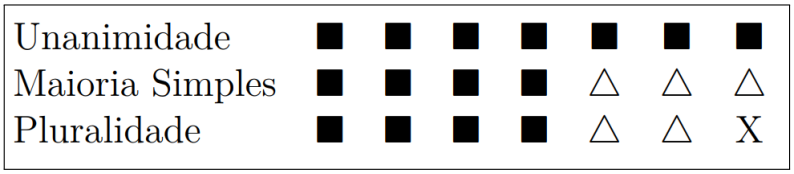


Figura 2.1. Votação majoritária para diversos cenários
Fonte: Kuncheva (2004)

Uma votação unânime, consiste em todos os votos direcionados à unicamente uma

opção. Em outro cenário com apenas duas opções, a escolha é feita através da maioria simples. Neste caso, basta obter pelo menos $50\% + 1$ dos votos. No caso de pluralidade, mais ocorrente em cenários reais, não é necessário se obter a maioria absoluta dos votos, apenas uma quantidade maior do que outras opções.

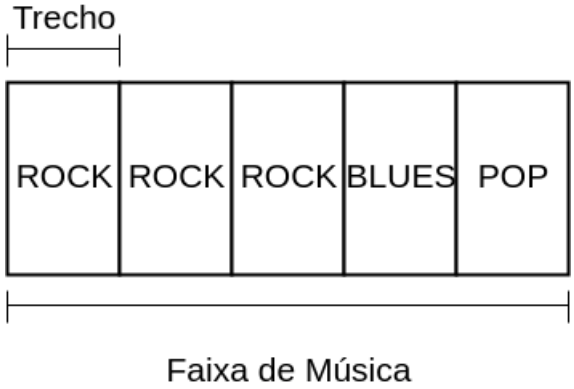


Figura 2.2. Cenário de votação majoritária com pluralidade

No contexto de classificação de música por gênero, diversos trechos de uma faixa de áudio podem ser classificados com gêneros distintos. Suponha que, dentre 5 trechos de uma faixa, 3 sejam classificados como rock, 1 trecho como blues e 1 como pop, ilustrados na Figura 2.2. Por voto majoritário, esta faixa recebe o rótulo rock.

2.1.2. Votação Majoritária Ponderada

Na técnica de votação majoritária ponderada são introduzido pesos aos votos. Os pesos aos votos implica em introduzir a ideia de relevância, dando importância a votos com peso maiores e menos importância à votos com pesos menores.

Em pesquisas que empregam esta técnica, o classificador retorna a probabilidade do trecho pertencer a cada uma das classes ao invés de retornar apenas a classe com maior probabilidade ilustrado na Figura 2.3. Assim, os vetores de probabilidades de cada trecho de uma faixa são somados classe-a-classe, e então a classe com a maior probabilidade acumulada é atribuída como rótulo da faixa (COSTA et al., 2017).

Apesar da acurácia obtida através da votação majoritária ponderada usuamente se mostrar melhor que a acurácia obtida pela votação majoritária simples, nem todos os classificadores emitem probabilidades por classe. O custo computacional de gerar um modelo que compute probabilidades é normalmente mais alto. Logo, a votação majoritária ponderada não pode ser utilizada em todos os casos, desta forma, é necessário desenvolver novas técnicas de votação quando as probabilidades são desconhecidas ou são computacionalmente caras de estimar.

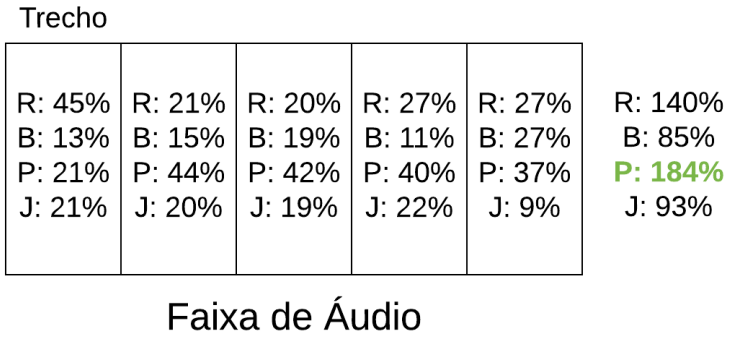


Figura 2.3. Voto majoritário ponderado. Cada letra representa um gênero musical.

2.2. Aprendizagem de Máquina

Aprendizagem de máquina é um campo da ciência da computação o qual dá a capacidade de aprender a um dispositivo. Grandes empresas de tecnologia como Google[1], Amazon[2], Microsoft[3] empregam aprendizagem de máquina para melhorar a experiência do usuário na utilização de seus produtos. No exemplo da empresa Google, aprendizagem de máquina é um dos ramos de pesquisa, abrangendo processamento de áudio (KALCHBRENNER et al., 2018) e análise de imagens (GANIN et al., 2018).

Normalmente os sistemas de aprendizagem de máquina baseados em aprendizagem supervisionada são divididas em três etapas. A primeira é a **extração de características**, que consiste em criar descrições computacionalmente eficientes para os dados cujos padrões devem ser aprendidos. A segunda etapa consiste no **treino** de um modelo a partir de um conjunto de exemplos e seus respectivos rótulos, denominado conjunto de treinamento. Este modelo é responsável por mapear padrões encontrados nos exemplos de treino em seus respectivos rótulos. Por fim, a terceira etapa consiste no **teste dos modelos**. Nesta etapa, exemplos nunca vistos antes são apresentados ao modelo. As predições obtidas são comparadas com os rótulos reais para avaliar a qualidade do modelo obtido. A capacidade de prever corretamente exemplos nunca antes vistos é chamada de **capacidade de generalização**, que é imprescindível para a aplicação em problemas reais (DUDA et al., 2000).

Para a representação de dados é utilizado um conjunto de descritores denominados **características**. Logo, para cada dado, é extraído um vetor de características que o simboliza. Esta fase deve ganhar atenção, visto que, a partir de uma boa representação, a facilidade de distinção dos exemplos de rótulos aumenta (MONARD; BARANAUSKAS, 2003).

Um exemplo simples de característica é a quantidade de *pixels* pretos em um

[1] https://www.google.com/
[2] https://www.amazon.com/
[3] https://www.microsoft.com/

quadrante para a classificação de caracteres escritos à mão. No contexto da música, as características podem descrever uma música através de ritmo, timbre ou tom, como proposto por Tzanetakis e Cook (2002), espectrogramas do sinal de áudio, abordado por Costa et al. (2011), além da extração de características através de sistemas que utilizam aprendizagem de máquina, apresentado por (CHOI et al., 2016). Deve se destacar que características não são necessariamente numéricas. Essas podem tomar diferentes formas, como dados nominais. Um exemplo citado por Kuncheva (2004) é a característica de país de origem, o qual pode ser representado em um vetor, onde cada posição representa um país. O vetor conterá 1 para um determinado país e zeros em outras posições.

A técnica de validação cruzada (*cross-validation*) é usada em aprendizagem de máquina para determinar a acurácia e capacidade do sistema generalizar exemplos não vistos anteriormente (KOHAVI et al., 1995). Um algoritmo de validação cruzada bastante utilizado é o *k-fold*. Para avaliar o modelo, o algoritmo divide o conjunto de dados em k partes, denominadas *folds*, ou pastas. O modelo é treinado com $k-1$ *folds* e testado com 1 *fold*. O processo se repete k vezes alterando o *fold* a ser testado em cada iteração. Por fim, é calculado a média da acurácia e o desvio padrão entre todos os *folds*. Estas estatísticas são usadas para representar a qualidade do modelo.

Validação cruzada também pode ser empregada na escolha de hiperparâmetros, como por exemplo, a técnica *Grid-Search*. A técnica testa todos os parâmetros ou conjunto de parâmetros fornecidos, e retorna o conjunto de valores de parâmetros que obteve a menor taxa de erro utilizando apenas o conjunto de treinamento, dividido em treino e validação.

2.2.1. Máquina de Vetores de Suporte (SVM)

O algoritmo de aprendizagem supervisionada Máquina de Vetores de Suporte (SVM) é um dos mais utilizados atualmente devido a sua eficiência no reconhecimento de padrões (GUO et al., 2000). Existem diversas pesquisas científicas usando SVMs para reconhecimento facial (OSUNA et al., 1997), reconhecimento de caracteres (BAHLMANN et al., 2002), classificação de textos em categorias (CHEN; DUMAIS, 2000), classificação de gênero musical (XU et al., 2003), entre outros.

Proposta para classificações binárias, SVM recebe como entrada um conjunto de vetores de características juntamente com o rótulo correspondente a cada vetor. Em seguida, o SVM busca maximizar uma margem de segurança para separação das duas classes no espaço, a fim de se obter uma distinção das mesmas. Uma vez que a margem é estimada, um hiperplano é traçado, a fim de agrupar exemplos com o mesmo rótulo como mostra a Figura 2.4. Na fase de teste, os vetores também são mapeados no espaço, porém seus rótulos são desconhecidos. A posição do item não rotulado no espaço referente ao hiperplano, especifica sua classe (LORENA; CARVALHO, 2007).

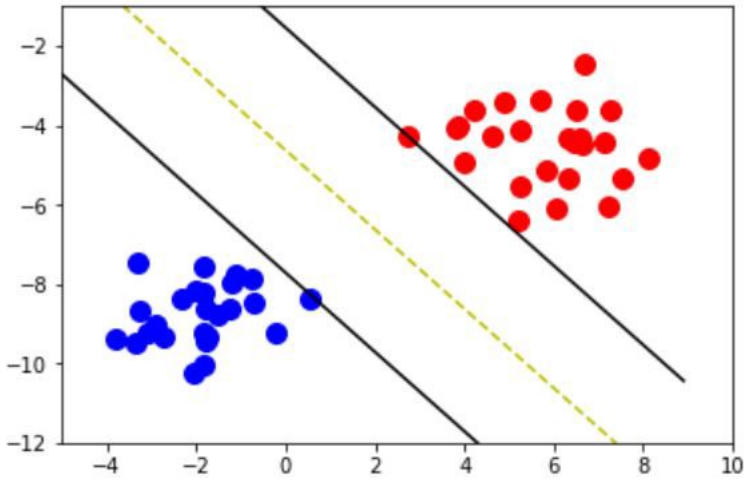


Figura 2.4. Separação de duas classes através de um hiperplano ideal
Fonte: Sanjeevi (2017)

Existe a possibilidade das características de treinamento não serem linearmente separáveis por um hiperplano, como ilustrado na Figura 2.5. Como solução, é realizado o processo de mapeamento dos dados para um espaço de dimensão maior através de uma função *kernel* (LORENA; CARVALHO, 2007). Na Figura 2.5, é possível observar que o conjunto de dados não é separável em 2 dimensões. Quando ocorre a adição de uma terceira dimensão, o problema é solucionado.

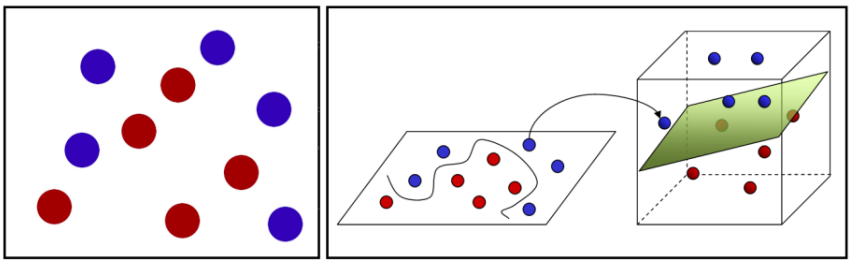


Figura 2.5. Aumento de dimensionalidade para classes linearmente não separáveis.
Fonte: Noronha e Fernandes (2016)

Apesar de SVM's serem propostas inicialmente para lidar com classificação binária, o algoritmo pode ser modificado para lidar com múltiplas classes. Isso é feito através da conversão destes problemas para diversos problemas binários (HEISELE et al., 2001). Existem dois métodos a fim de realizar tal processo, sendo eles um-contra-todos e um-contra-um (RIFKIN; KLAUTAU, 2004).

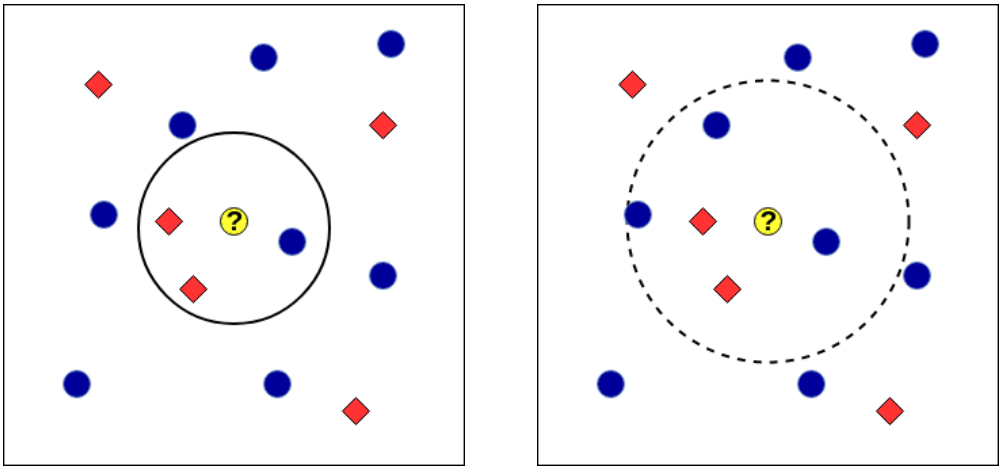
O método de classificação SVM é bastante utilizado devido a sua capacidade de generalização. Isto é, classificar dados que não estão presentes no conjunto de treinamento (LORENA; CARVALHO, 2007). SVMs lidam com grandes dimensões de características, porém a queda de desempenho ao lidar com muitos exemplos no conjunto de treino é significativa

(JOACHIMS, 1998). Logo, quando o conjunto de treino é muito grande, algoritmos escaláveis como *Multilayer Perceptron* (MLP) ou *Random Forest* se tornam mais adequados para estes cenários

2.2.2. K Vizinhos Mais Próximos (K-NN)

Um classificador bastante utilizado para o rotulamento de dados é o K-Vizinhos Mais Próximos (K-NN) (LIU et al., 2003). Pela sua simplicidade, é comum encontrar pesquisas sobre processamento de texto (BIJALWAN et al., 2014), previsões na área da economia (IMANDOUST; BOLANDRAFTAR, 2013) e também de clima para uso na área de agricultura (BANNAYAN; HOOGENBOOM, 2008). Apesar de seu funcionamento simples, este classificador consegue atingir bons resultados dependendo do conjunto de dados.

Para o reconhecimento de padrões, o K-NN classifica dados com base nos exemplos de treino presentes em um espaço n dimensional. Para a rotulação de um dado desconhecido x, as distâncias entre x e os demais exemplos do conjunto de treinamento são calculadas (IMANDOUST; BOLANDRAFTAR, 2013). Existem 3 métricas bastante utilizadas para cálculo da distância, a saber: distância Euclidiana, Manhattan e Minkowski (SINGH et al., 2013). Feito isto, são identificados os k vizinhos mais próximos. A rotulação se dá pelo voto majoritário da classe mais frequente destes k vizinhos.



(a) (b)

Figura 2.6. Parâmetro k do classificador K-NN
(a) Classificação utilizando K-NN com $k = 3$. (b) Classificação utilizando K-NN com $k = 5$.

Um bom desempenho desta técnica de classificação é consequência da escolha de um bom número k de vizinhos próximos (HALL et al., 2008). Grande parte das pesquisas que utilizam o K-NN definem um número ímpar para o parâmetro k com a finalidade de evitar empates no voto majoritário para decisão de um rótulo quando há apenas 2 classes.

Como ilustrado na Figura 2.6, conforme a variação da quantidade de vizinhos mais próximos, é possível obter diferentes resultados. No caso de um k pequeno, o modelo será

sensível a ruídos. Por outro lado, k muito grande, pode ocorrer a inclusão de pontos de diferentes rótulos, degradando o resultado do voto majoritário.

Em geral, o K-NN é uma abordagem simples de ser implementada e utilizada para se realizar a predição de rótulos obtendo uma boa taxa de acertos. Em contrapartida, o aumento da dimensionalidade das características pode tornar a métrica de distância degenerada, impactando na acurácia do classificador ao lidar com muitas características. Desta forma, uma abordagem comum é realizar a seleção de características e projeções para diminuir a dimensionalidade (SINGH et al., 2013).

2.2.3. Modelos Ocultos de Markov

Um modelo estocástico amplamente utilizado para modelagem de características temporais é o *Hidden Markov Model* (HMM). Sua aplicação está presente não só em trabalhos de processamento de sinais, tais como reconhecimento de fala (LEE et al., 1990) e reconhecimento de acordes (LEE; SLANEY, 2006), mas também no reconhecimento de dígitos manuscritos (RABINER et al., 1985), previsões do tempo (KHIATANI; GHOSE, 2017), entre outros.

Para compreender HMM é necessário entender as Cadeias de Markov.

Cadeias de Markov

Cadeia de Markov é um processo estocástico que apresenta um conjunto de estados, e as probabilidades de, a partir do estado atual X_n, alcançar o estado X_{n+1}, onde X_n é um estado qualquer no tempo n. A probabilidade do sistema alcançar o estado X_{n+1} depende unicamente do estado X_n (BILMES et al., 1998), ou seja, só o estado em que o sistema se encontra é relevante para futuras predições.

A Figura 2.7 ilustra um exemplo de Cadeia de Markov com 2 estados projetada para prever o clima dos próximos dias em ensolarado ou chuvoso.

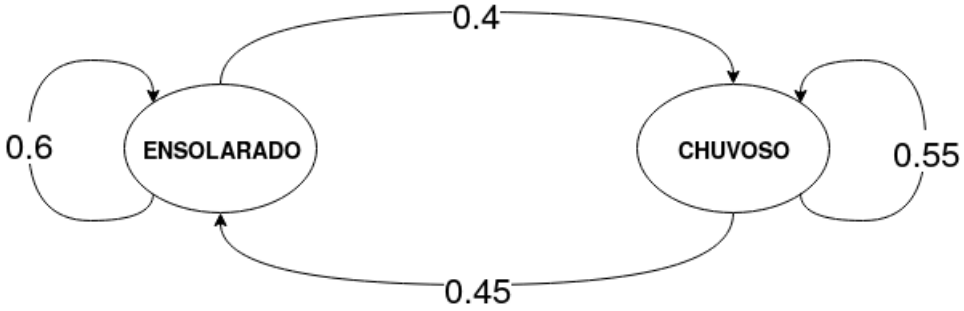


Figura 2.7. Cadeia de Markov com 2 estados para previsão do clima

As cadeias de Markov são compostas por 3 elementos principais:

- Um conjunto de Estado, sendo ensolarado ou chuvoso para o exemplo ilustrado na Figura 2.7;

- Probabilidades de Transição entre estados;
- Distribuição de probabilidade da cadeia de Markov ter como início determinado estado, não ilustrado na Figura 2.7;

As probabilidades de transição de estados são definidas por meio do treinamento do modelo. Durante o treinamento, através de uma sequência de estados, são estipuladas as probabilidades de transição que melhor representam o conjunto de treino (BILMES et al., 1998).

Para representar a probabilidade de transição de estados entre todos os estados do sistema, é utilizada uma Matriz de Probabilidade de Transição de Estados. Nesta matriz, o elemento da n-ésima linha, m-ésima coluna representa a probabilidade de transição do estado n ao estado m.

Vale destacar que os estados podem se repetir ao longo do tempo, logo, é necessário ter uma probabilidade de transição para o mesmo estado.

$$P_{(n)} = P_{(n-1)} \begin{bmatrix} P(Ensolarado|Ensolarado) & P(Chuvoso|Ensolarado) \\ P(Ennsolarado|Chuvoso) & P(Chuvoso|Chuvoso) \end{bmatrix}$$

$$P_{(n)} = P_{(n-1)} \begin{bmatrix} 0.6 & 0.4 \\ 0.45 & 0.55 \end{bmatrix}$$

Supondo que a probabilidade inicial seja 0.3 e 0.7 para um dia ensolarado e chuvoso respectivamente, podemos representar em forma vetorial:

$$P_{(0)} = \begin{bmatrix} 0.3 & 0.7 \end{bmatrix}$$

Usando a matriz de transições é possível calcular a probabilidade do clima estar em qualquer um dos estados no dia n. Para o primeiro dia ($n = 1$), a distribuição de probabilidade do clima ser ensolarado ou chuvoso, se dá pelo seguinte cálculo:

$$P_{(1)} = P_{(0)} \begin{bmatrix} 0.6 & 0.4 \\ 0.45 & 0.55 \end{bmatrix}$$

$$P_{(1)} = \begin{bmatrix} 0.3 & 0.7 \end{bmatrix} \begin{bmatrix} 0.6 & 0.4 \\ 0.45 & 0.55 \end{bmatrix}$$

$$P_{(1)} = \begin{bmatrix} 0.495 & 0.505 \end{bmatrix}$$

Através do resultado, pode-se concluir que a probabilidade do próximo dia fazer sol é de $49,5\%$ e de chover $50,5\%$ com base nas probabilidades definidas anteriormente. O mesmo cálculo pode ser aplicado para revelar estados futuros.

Outro cálculo interessante que pode ser feito é: visto que o clima de hoje está ensolarado, qual a probabilidade dos dias seguintes serem chuvoso, ensolarado e chuvoso respectivamente? A resposta pode ser dada pelos seguintes cálculos:

$$P = P(Chuvoso|Ensolarado) * P(Ensolarado|Chuvoso) * P(Chuvoso|Ensolarado)$$

$$P = 0.4 * 0.45 * 0.4$$

$$P = 0.072$$

Sabendo que $P(Chuvoso|Ensolarado)$ é a probabilidade de transição entre o estado ensolarado para chuvoso e $P(Ensolarado|Chuvoso)$ é a probabilidade de transição entre o estado chuvoso para ensolarado, a probabilidade da sequência de climas ocorrer é 7.2%.

Modelo Oculto de Markov

Um grande problema que impossibilita o uso das cadeias de Markov, é a necessidade de que os estados sejam observáveis, o que nem sempre ocorre em um cenário real. No HMM, esta necessidade é contornada através de observações geradas através de estados não observáveis (RABINER, 1989).

O modelo tem como entrada uma sequência de observações, por fim, emite uma distribuição de probabilidade de uma sequência de estados terem gerado dadas observações.

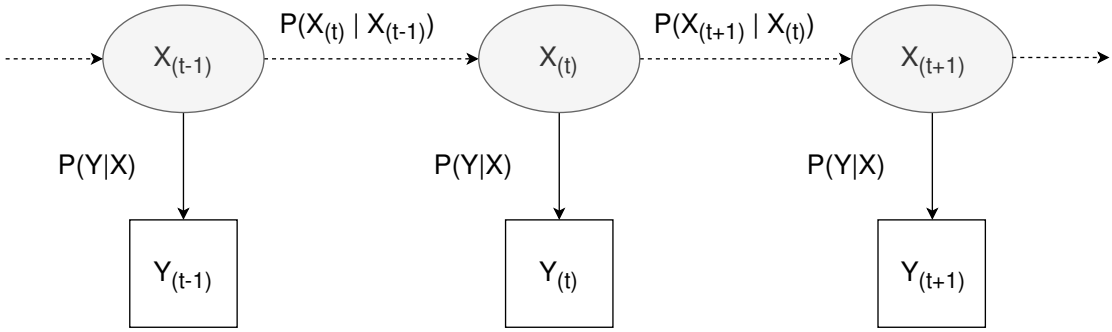


Figura 2.8. Arquitetura de um HMM
Fonte: Adaptado de Ghahramani (2001)

A arquitetura do HMM é ilustrado na Figura 2.8, onde:

- $X(t)$ representa um estado não observável;
- $P(X(t+1)|X(t))$ representa a probabilidade de transição de um estado qualquer para outro estado futuro qualquer. As probabilidades são geradas na fase de treinamento do modelo;
- $Y(t)$ representa uma observação;

- $P(Y|X)$ representa uma probabilidade do estado X emitir uma observação Y, chamada de probabilidade de emissão;

Através da multiplicação das probabilidades de transição de estado com as probabilidades de emissão de observações, é possível obter a probabilidade dos estados $\{X(t-1), X(t), X(t+1)\}$ terem emitido as observações $\{Y(t-1), Y(t), Y(t+1)\}$ (GHAHRAMANI, 2001).

Tomando base o exemplo anterior, onde se deseja prever o clima de dias seguintes. Um funcionário trabalha em um local fechado, onde não é possível saber se o dia está ensolarado ou chuvoso, porém consegue observar outros funcionários ao seu redor. Os funcionários tendem a usar roupas de verão ou camiseta quando o dia está ensolarado, ou costumam trazer um guarda-chuva quando o dia está chuvoso.

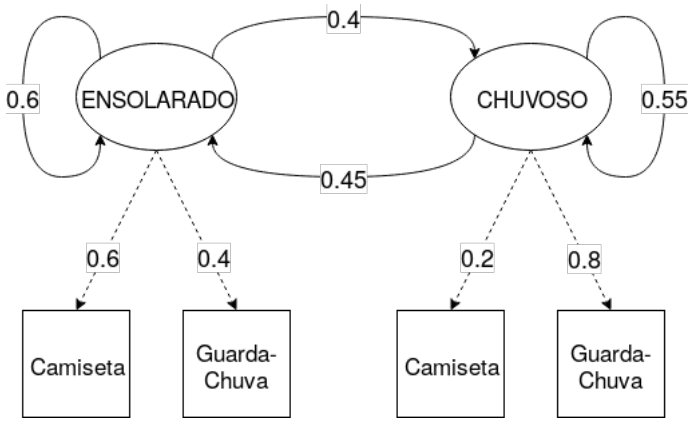


Figura 2.9. Diagrama esquematizando o cenário proposto para modelagem de um HMM

A Figura 2.9 ilustra o esquema do HMM desenvolvido para o cenário proposto. Os estado ensolarado e chuvoso são estados não observáveis. As linhas contínuas juntamente com um número, representam a probabilidade de transição de estado. As observações camiseta e guarda-chuva são observações emitidas por cada estado, juntamente com a probabilidade de emissão da observação, representado pelo número da linha tracejada. A distribuição de probabilidades do sistema iniciar em cada estado é de 30% e 70% para ensolarado e chuvoso respectivamente.

Suponha que foi observado que em 2 dias, os funcionários levaram guarda-chuvas. A maior probabilidade do clima dos dois dias é chuvoso, como ilustra a Figura 2.10.

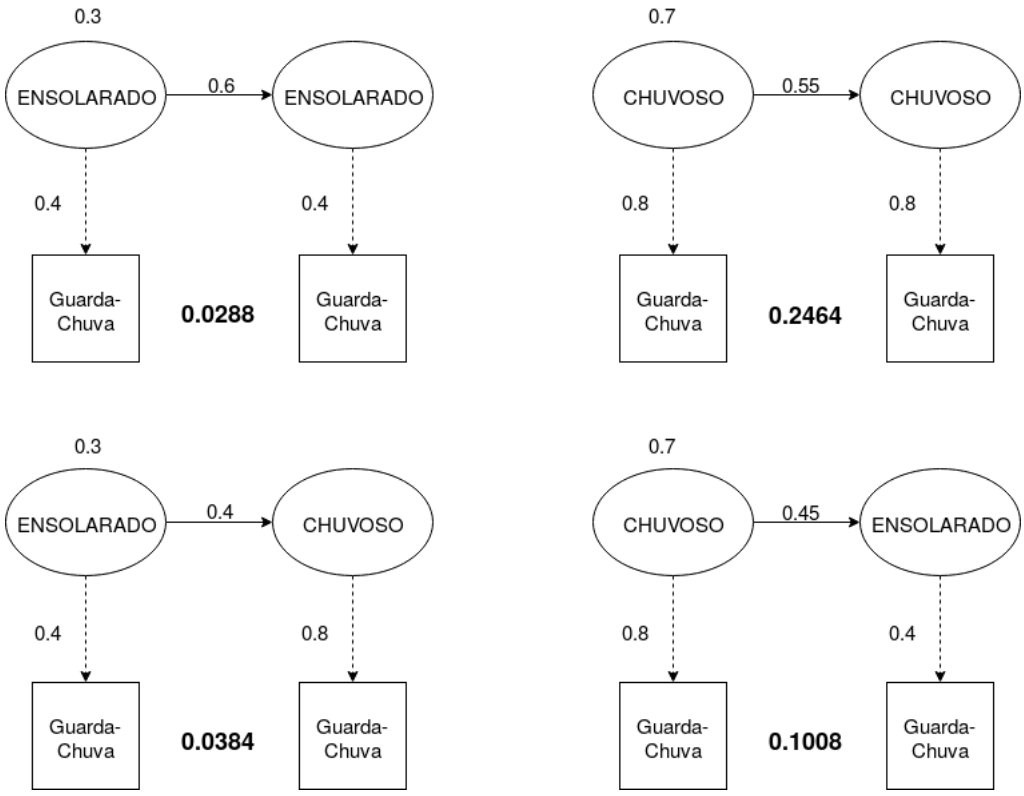


Figura 2.10. Combinação de observações e estados do cenário proposto

Os mesmos cálculos são válidos para sequências de observações maiores, sendo possível inferir os estados a partir das próprias observações.

Para a estimativa das probabilidades de transição de estados e probabilidade de emissões de observação é utilizado o algoritmo *Expect Maximization* (EM) com o objetivo de maximizar a verossimilhança a partir de uma sequência tomada como entrada (RABINER, 1989).

2.2.4. Redes Neurais

Redes neurais ou também chamadas de redes neurais artificiais, são modelos matemáticos bio-inspirados que são capazes de aproximar qualquer função computável. Um tipo de função bastante interessante na área de aprendizagem de máquina é a função de classificação de padrões, que mapeia objetos em classes de objetos (HAYKIN; HAYKIN, 2009).

Redes neurais são amplamente utilizadas na área de *Music Information Retrieval* (MIR) devido ao grande poder de reconhecimento de padrões. Medhat et al. (2018) em sua pesquisa sobre reconhecimento de gêneros atingiu um resultado de 92.12% utilizando um tipo de rede neural artificial, um resultado estado-da-arte.

A arquitetura de uma rede neural artificial é composta por um conjunto de nós, denominados **neurônios**, os quais são agrupados em camadas. Cada neurônio de determinada camada é conectado aos neurônios da próxima camada por uma estrutura denominada aresta,

a qual possui um determinado peso denominado **peso sináptico**. Os neurônios da mesma camada não são conectados entre si (GROSSBERG, 1988).

A Figura 2.11 exemplifica uma rede neural com 6 neurônios, 3 camadas e arestas conectando cada neurônio com neurônios da camada seguinte.

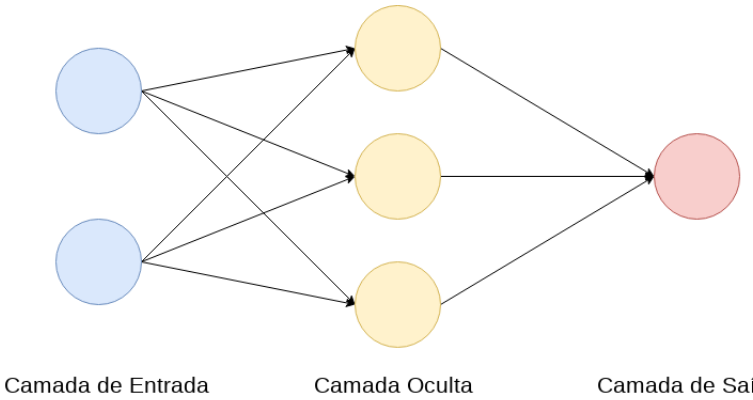


Figura 2.11. Diagrama de uma rede neural com 3 camadas

Na primeira camada, denominada camada de entrada, os neurônios recebem os dados de entrada e repassam para a camada a seguir. Para cada aresta conectando um determinado neurônio com o neurônio da próxima camada, o valor transmitido é ponderado por seu peso sináptico. Portanto, valor recebido por um neurônio, chamado de **campo induzido**, é a combinação linear de suas entradas e seus respectivos pesos sinápticos. (FAUSETT et al., 1994).

Os pesos de uma rede neural artificial são inicializados com valores aleatórios. No processo de treinamento, os pesos são modificados com base no resultado da rede. A função de erro é utilizada para avaliar o desempenho da rede, calculando o custo, que seria a diferença do resultado emitido com o resultado esperado. O objetivo é reduzir ao máximo o custo através de modificações nos pesos das arestas e *bias* (FAUSETT et al., 1994).

Na Figura 2.12, é ilustrado um cenário com um neurônio N conectado por 3 entradas com pesos $\{w1, w2, w3\}$. Estas arestas estão transmitindo os sinais $\{x1, x2, x3\}$ pelos neurônios da camada anterior. O valor de entrada do neurônio N é:

$$N_{(entrada)} = (x1 * w1) + (x2 * w2) + (x3 * w3) + bias$$

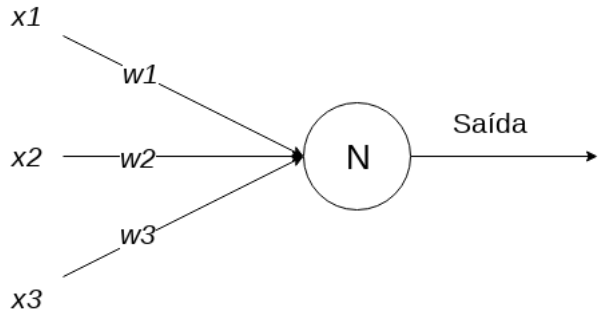


Figura 2.12. Neurônio de uma rede neural conectado por 3 arestas

O sinal de entrada é submetido a uma função de ativação. O papel da função de ativação é tornar a rede não-linear, permitindo a solução de problemas mais complexos. Outro objetivo da função de ativação é filtrar dados não relevantes para a rede, determinando quais sinais devem ser passados adiante e quais sinais devem ser ignorados através de uma função matemática não-linear (FAUSETT et al., 1994). Desta forma, a saída de um neurônio pode ser escrita como

$$Y = \sigma(v)$$

tal que σ é uma função de ativação e v é o campo induzido. Exemplos de funções bastante utilizadas são a família de funções sigmoidais como *sigmoid*, *tanh* (tangente hiperbólica) e as funções da família de retificadores lineares como ELU *(exponential linear unit)* e ReLU *(rectified linear unit)*.

Uma função de ativação bastante usada para classificação e presente neste trabalho, é a função *softmax*. Esta função aplicada na última camada da rede neural, retorna um valor que corresponde à probabilidade do dado de entrada ser classificado em determinada classe dentro das classes definidas.

Redes Neurais Recorrentes

As redes neurais recorrentes se diferem das redes neurais artificiais pelo fato de que as arestas que ligam um neurônio à próxima camada, também podem ligar aos neurônios da própria camada, formando um ciclo, cujo processo é denominado retro-alimentação. Isto permite que a rede leve em consideração informações que ocorreram anteriormente, através de uma espécie de memória, influenciando no processamento atual.

As redes neurais recorrentes são utilizadas para classificação de séries temporais, como reconhecimento de fala (GRAVES et al., 2013), reconhecimento de padrões no mercado de ações (KAMIJO; TANIGAWA, 1990), entre outros.

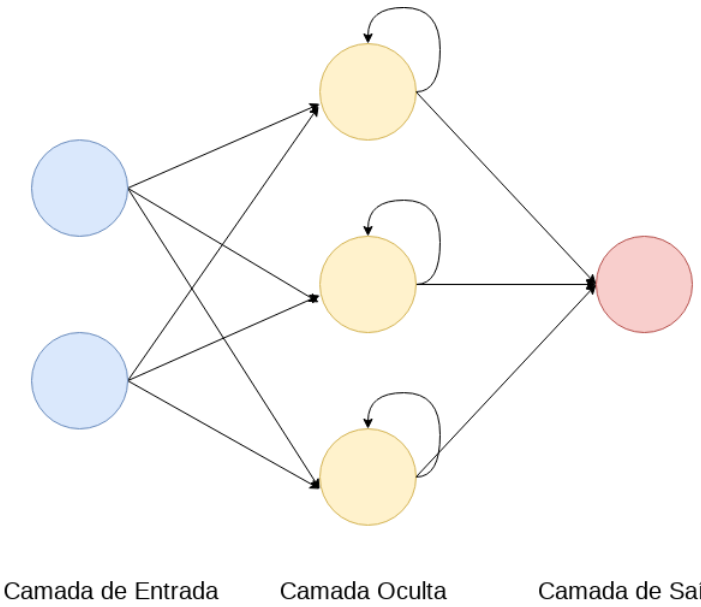


Figura 2.13. Arquitetura de uma rede neural recorrente

A Figura 2.13 mostra uma rede neural recorrente com 3 camadas, sendo a camada oculta alimentando ela mesma. Esta camada é responsável por armazenar o histórico dos dados relevantes para rede.

Na Figura 2.14 é ilustrado um neurônio da camada recorrente o qual sua saída é tomada como entrada. O diagrama ilustra o neurônio de forma sequencial para o melhor entendimento.

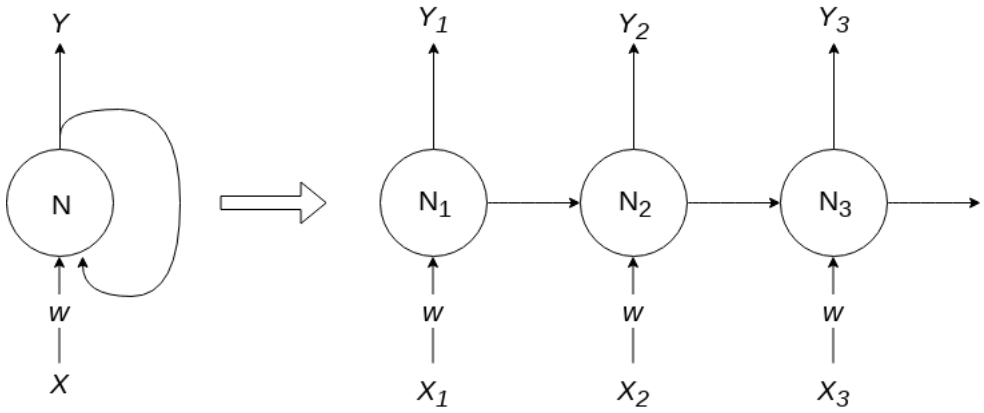


Figura 2.14. Representação de um neurônio recorrente

Através do diagrama, é possível notar os seguintes elementos:

- X: Dado de entrada do neurônio em um determinado tempo.
- w: Peso da aresta conectada ao neurônio.
- N: Valor do neurônio em um determinado tempo.
- Y: Saída do neurônio após a realização dos cálculos;

Um grande problema em treinar redes neurais recorrentes, denominado *vanishing gradient*, é lidar com sequências muito grandes. Este problema reflete na atualização de pesos das camadas iniciais de forma muito lenta através da propagação do gradiente do erro da camada mais externa em direção à camada de entrada. Este processo implica na diminuição do gradiente conforme é retropropagado (BENGIO et al., 1994).

Redes *Long Short Term Memory* (LSTM)

A rede LSTM contorna o problema de *vanishing gradient* através da adição de células de estado que funcionam como uma memória, armazenando dados relevantes, além da mudança na arquitetura dos neurônios recorrentes (HOCHREITER; SCHMIDHUBER, 1997). Em uma rede neural recorrente comum, as informações temporais não são tratadas com relevância, sendo modificada a cada iteração da camada recorrente.

A célula de estado consiste em armazenar e permitir a leitura de dados temporais relevantes para rede, solucionando o problema do *vanishing gradient*. O acesso a esta estrutura é feita através de 3 portões (do inglês: *gates*): *forget gate*, *input gate* e *output gate* (GERS et al., 1999).

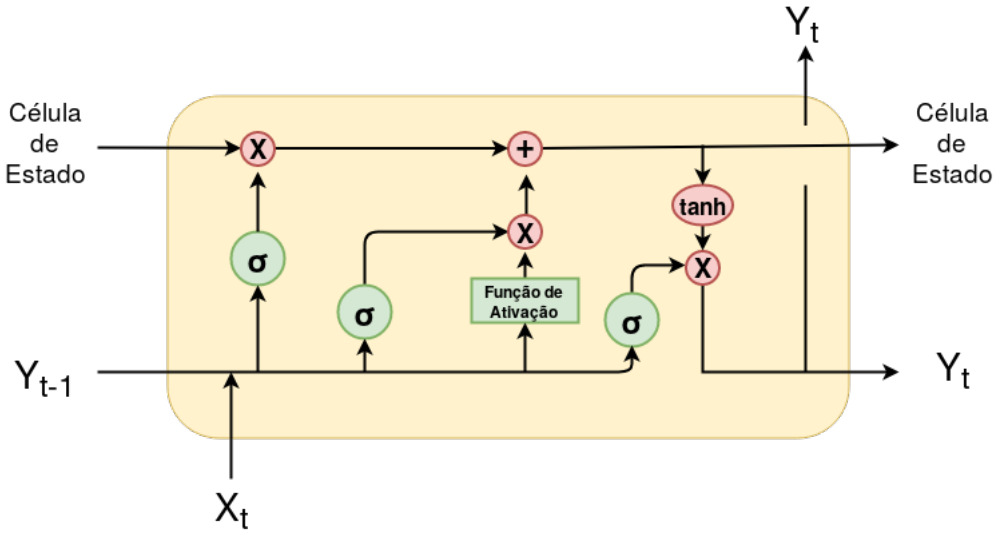


Figura 2.15. Arquitetura de um neurônio da rede LSTM

Fonte: Adaptado de Olah (2015)

A Figura 2.15 ilustra a arquitetura de um neurônio recorrente, onde:

- σ: Representa uma função sigmóide;
- tanh: Representa uma função tangente hiperbólica;
- X: Representa uma multiplicação;
- +: Representa uma soma;
- Y_t: Representa a saída do neurônio recorrente;
- X_t: Representa o dado de entrada do neurônio;

Forget Gate

O *forget gate* é funcionar como uma espécie de ponderador sobre as informações da célula de estado, dando maior peso aos dados mais relevantes e menor peso para os dados menos relevantes armazenados até o momento. O principal objetivo do *forget gate* é manter somente informações importantes na célula de estado (GERS et al., 1999).

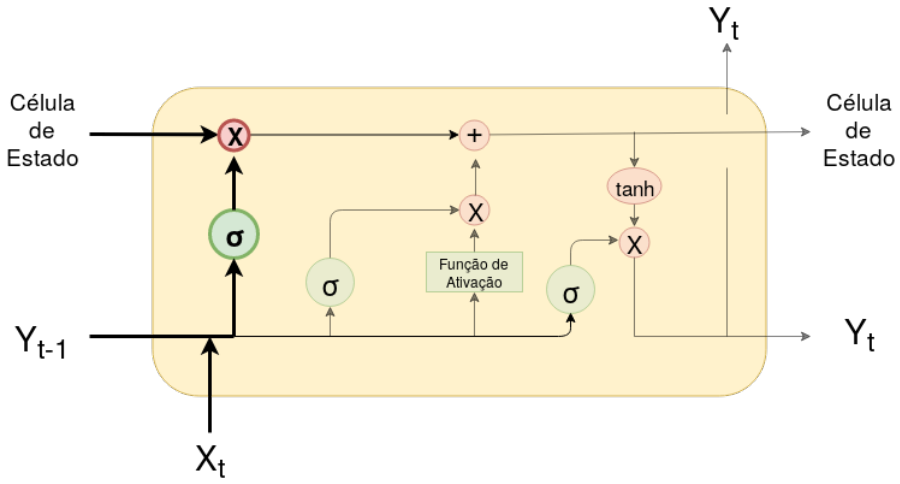


Figura 2.16. *Forget Gate*
Fonte: Adaptado de Olah (2015)

Este processo toma como entrada a saída do neurônio recorrente da iteração passada $Y_{(t-1)}$ e a entrada da sequência no tempo t, X_t. A entrada é multiplicada pelo peso da aresta e adicionado ao *bias*. Após isso é submetida à uma função sigmóide e tem como saída um valor no intervalo de 0 a 1 para cada elemento da célula de estado. Um valor 0 representa que a informação não é importante e deve ser esquecida, por outro lado, o valor 1 representa que a informação é muito importante e deve ser mantida (OLAH, 2015).

Input Gate

O *input gate* tem a responsabilidade de adicionar uma nova informação correspondente ao tempo atual à célula de estado a partir dos dados de entrada do neurônio. Primeiramente, deve-se calcular a relevância da informação de entrada através de outra função sigmóide. Em seguida, é aplicado uma função de ativação para obter os candidatos a partir da entrada. Logo, basta multiplicar o resultado das funções e adicionar o produto à célula de estado (HOCHREITER; SCHMIDHUBER, 1997).

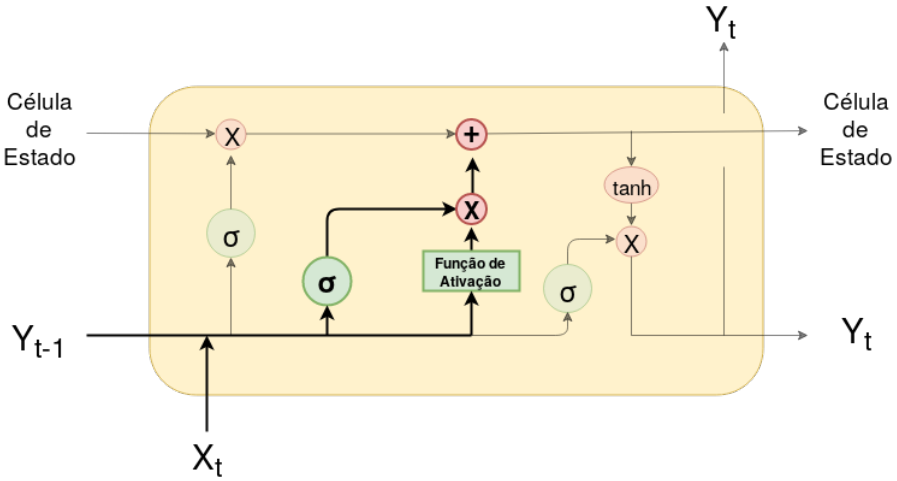


Figura 2.17. *Input Gate*
Fonte: Adaptado de Olah (2015)

Output Gate

Por fim, o *output gate* tem o objetivo de retornar informações úteis da célula de estado para a saída do neurônio. Para isto é aplicado uma função tangente hiperbólica nos valores da célula de estado, escalando os valores em um intervalo de -1 a 1. Feito isso, o próximo passo é aplicar uma função sigmóide nos valores de entrada do neurônio e multiplicar o resultado pelo resultado da função tangente hiperbólica (SAK et al., 2014).

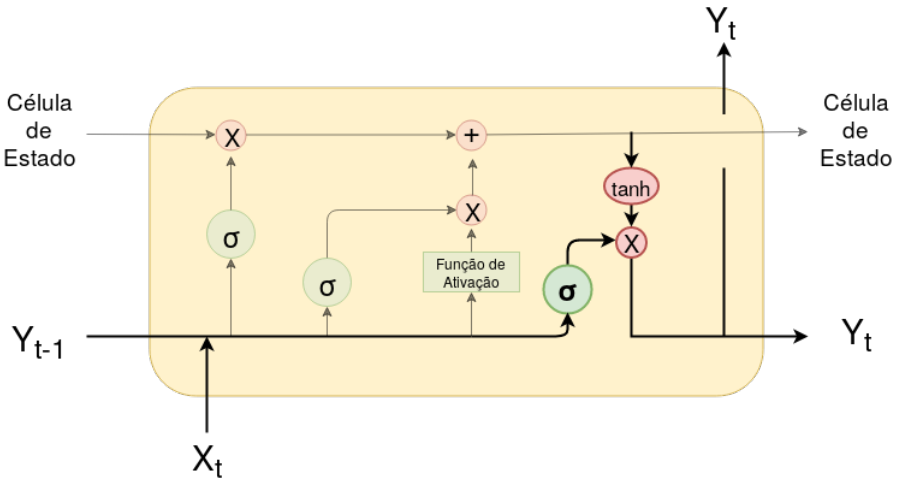


Figura 2.18. *Output Gate*
Fonte: Adaptado de Olah (2015)

Neste trabalho, o objetivo do uso de redes neurais recorrentes é tomar proveito da sequência temporal dos votos para inferir um gênero à faixa.

2.3. Considerações Finais

Neste capítulo foram apresentados os principais conceitos para compreender o problema de votação em aprendizagem de máquina. Na Seção 2.1, foram apresentadas técnicas de votação para a inferência de rótulos quando há várias predições disponíveis por objeto. Já na Seção 2.2 foram esclarecidos conceitos básicos sobre Aprendizagem de Máquina. Foram apresentados dois algoritmos de aprendizagem de máquina amplamente utilizados (SVM e K-NN) para modelarem dados não-temporais. Para dados temporais, foram apresentados os conceitos de HMM e redes neurais artificiais.

Por fim, quando os votos de cada seção da faixa não possuem probabilidades por classe, o uso de métodos para inferir um rótulo à faixa se limita ao voto majoritário. A ideia abordada nesta pesquisa é explorar outras possibilidades além do voto majoritário. Foram propostas 2 abordagens: A primeira proposta é utilizar a contagem de votos e um classificador tradicional para mapear a contagem de votos em classes. A segunda abordagem é utilizar um classificador capaz de modelar sequências de votos ordenados em relação ao tempo da faixa para inferir uma classe à uma faixa.

Capítulo 3

Trabalhos Relacionados

É comum encontrar pesquisas voltadas a classificação automática de música utilizando *Multiple Vector Representation* (MVR). Estas pesquisas usam alguma técnica de votação para inferir um gênero à faixa através do resultado da classificação de cada trecho. A seguir são apresentadas diferentes abordagens juntamente com as técnicas de votação usadas.

3.1. *Music Genre Classification Using Novel Features and a Weighted Voting Method*

Na pesquisa realizada por Jang et al. (2008), foram propostas duas técnica de votação para classificação automática de música: votação majoritária e votação majoritária ponderada. Os autores extraíram características de trechos de áudio de 3 segundos para definir o gênero da faixa de áudio. A acurácia do sistema foi de 76% dentre 10 gêneros.

As faixas presentes no conjunto de dados foram divididas em segmentos de 93 milissegundos, denominados janelas de análise e em trechos de 3 segundos, denominados janelas de textura. Foram extraídas características das janelas de análise tais quais *Mel-Frequency Cepstral Coefficients* (MFCC), *Spectral Centroid*, *Spectral Flux*, *Spectral Roll-off*, *Zero-Crossing rate* e *low-energy*. As janelas de textura também foram utilizadas como características, representando a média e a variância de uma sequência de janelas de análise.

Na fase de treino, todas as janelas de textura são apresentadas ao modelo rotuladas com o rótulo da faixa toda. No teste, as janelas de textura de uma faixa são classificadas afim de receber um gênero para a mesma. Logo, vários resultados de classificação são obtidos para um clipe de música. Por fim, o gênero final é determinado pelo voto majoritário simples e voto majoritário ponderado através dos resultados da classificação.

Na técnica de votação majoritária ponderada, abordada na Seção 2.1.2, os autores obtiveram a distribuição de probabilidade da classificação de cada janela de textura pertencer

a cada um dos gêneros. O gênero inferido ao rótulo da faixa é aquele que obteve a maior probabilidade acumulada entre todos os trechos da faixa.

O sistema foi avaliado em um conjunto de 1000 trechos de músicas com duração de 30 segundos cada, dividido em 10 gêneros, todas com 100 exemplos. Os gêneros da base de dados são: clássico, jazz, R&B, country, rock, hip hop, metal, dance, nova era e eletrônica. A base não está disponível ao público.

O processo de classificação consistiu na utilização dos métodos Máquina de Vetores de Suporte (SVM), K-Vizinhos Mais Próximos (K-NN) e *Linear Discriminant Analysis* (LDA). A partir disto, a melhor taxa de acerto utilizando votação majoritária resulta em 70.4% utilizando o classificador SVM. Em contrapartida, o resultado com votação majoritária ponderada foi 76% de acurácia.

3.2. *Music Genre Recognition Using Spectrograms*

Costa et al. (2011) sugeriram a extração de características a partir da imagem de espectrogramas de áudio. A ideia dos autores foi dividir cada faixa em partes. Descritores de imagens são computados de cada parte e usados como característica. Cada parte é classificada independentemente e o voto majoritário é usado para inferir o gênero da faixa toda.

Primeiramente, o espectrograma de cada faixa foi computado. A partir disto, foram extraídos trechos de 30 segundos do início, meio e fim do espectrograma para uma melhor representação. Cada trecho foi dividido em 10 segmentos, o método de extração de características titulado *gray level co-occurrence matrix* (GLCM) foi aplicado em cada segmento. Através de análises de níveis de cinza, são obtidos a entropia, correlação, homogeneidade, momento estatístico de 3º ordem, máxima verossimilhança, contraste e energia.

Na fase de treino, os vetores de características obtidos dos 30 segmentos de cada faixa foram apresentados ao modelo junto com seu gênero. Na classificação, foi usado o SVM para determinar o gênero de cada segmento. É computado os votos de todos os segmentos e esses são transformados em porcentagem. Isto é, a quantidade de votos de determinado gênero é dividido na quantidade total de votos. As técnicas de regra máxima e regra mínima (KITTLER et al., 1998) foram aplicados a estes percentuais, atingindo 67.2% de acurácia.

A base *Latin Music Database* (LMD) foi usada para avaliar o sistema proposto. Os gêneros que compõem a base são: axé, bachata, bolero, forró, gaucha, merengue, pagode, salsa, sertanejo e tango. Para evitar o sobreajuste, um filtro de artistas (*artist filter*) (PAMPALK et al., 2005) foi usado para garantir que o mesmo artista não apareça no treino e teste ao mesmo tempo.

3.3. *Unsupervised Learning of Sparse Features for Scalable Audio Classification*

Henaff et al. (2011) propôs uma abordagem de aprendizagem não supervisionada de características de áudio para descrever faixas. A metodologia consiste na divisão das faixas em trechos de 46.4 milissegundos. Aplicar a técnica de *Predictive Sparse Decomposition* (PSD) para aprendizagem das características em cada trecho. O algoritmo SVM foi usado para classificar os trechos, e então é aplicado voto majoritário simples para decisão do gênero da faixa. A acurácia obtida pelo sistema foi de 83.4% usando a base de dados GTZAN.

Para o treino, primeiramente o sinal de áudio foi pré-processado e através da técnica *Constant Q Transform* (CQT), é gerado o espectrograma para cada faixa. Os espectrogramas são divididos em pequenos trechos, e a partir destes trechos, dicionários esparsos são aprendidos. Para ganho de performance, os dicionários são apresentados para aprendizagem a um codificador. E então, o codificador é apresentado ao modelo para treinamento.

Na fase de teste, o processo é o mesmo. É feito um pré-processamento no sinal, o espectrograma é dividido em trechos, dicionários são aprendidos a partir dos trechos e o codificador é gerado. Os codificadores são apresentados ao classificador já treinado para obter o gênero de cada trecho. O voto majoritário simples é aplicado nos trechos da faixa inferindo o gênero à faixa completa.

Um problema abordado pelos autores é o tempo de processamento de outros sistemas na área de *Music Information Retrieval* (MIR). Por fim, o sistema foi desenvolvido para obter resultados rapidamente, o que leva ao uso do codificador.

3.4. Considerações Finais

Foram apresentados apenas algumas pesquisas que empregam a técnica de votação. Vários outros trabalhos que também usam descritores MVR utilizam algum esquema de votação. No trabalho de Wülfing e Riedmiller (2012) e Lippens et al. (2004) é constado o uso de voto majoritário para definição dos gêneros. Cataltepe et al. (2007) usa votação majoritária ponderada em sua pesquisa.

Os trabalhos apresentados nesta Seção, mostram que votação majoritária é uma técnica fundamental em sistemas de aprendizagem de máquina que utilizam MVR. Embora existam diversas formas de combinação de votos quando há predição de probabilidade para classe-alvo, não existem muitas opções quando apenas o voto do trecho é conhecido. Desta forma, o objetivo deste trabalho é desenvolver e testar alternativas ao voto da maioria para os casos onde apenas os votos de trechos são conhecidos.

Capítulo 4

Metodologia

Sistemas de classificação de música em gêneros e afins normalmente utilizam a votação majoritária quando a probabilidade de cada trecho pertencer a cada classe é desconhecida. Neste capítulo são apresentadas duas abordagens alternativas para a votação majoritária. Além disto, o sistema classificador de texturas musicais, responsável por atribuir os votos a cada trecho, também é apresentado.

4.1. Sistema Classificador de Texturas Musicais

O objetivo do Sistema classificador de texturas musicais (SCTM), proposto por Foleiss (2018), é classificar as texturas musicais em classes, que posteriormente, alimentam o sistema proposto nesta monografia. O SCTM foi utilizado para obtenção dos votos de todos os trechos, também chamados de texturas, das músicas das bases de dados. O processo de classificação das texturas é ilustrado pela Figura 4.1.

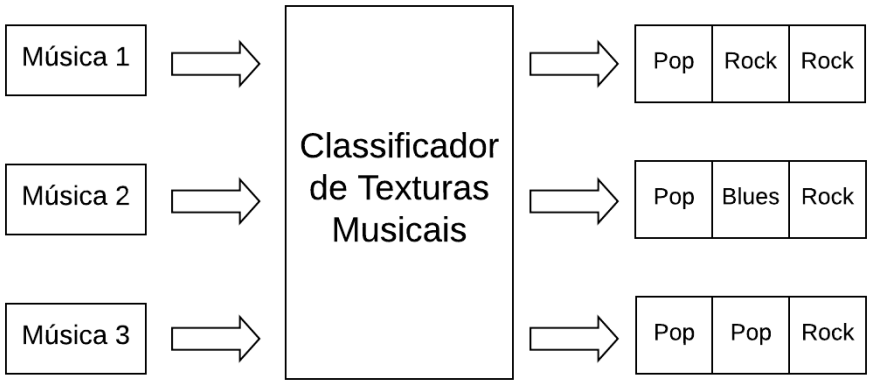


Figura 4.1. Diagrama de funcionamento do classificador de texturas musicais

O SCTM utiliza quatro conjuntos de características diferentes, cada uma em um nível de abstração diferente. Estes conjuntos são: características projetadas à mão (MARSYAS), espectrogramas em escala Mel (MEL-SPEC), projeções aleatórias obtidas de espectrogramas em escala Mel (RP), e características obtidas de uma rede de *autoencoder* (AE).

Na fase de treinamento, ao invés de usar todas as texturas de uma faixa, são utilizados apenas um subconjunto. Este subconjunto de texturas é extraído através dos algoritmos *K-Means* e *Linearly Spaced Vector* (*linspace*). *K-Means* tem o objetivo de tentar identificar as texturas típicas da faixa, enquanto *linspace* simplesmente pega trechos linearmente espaçados.

Por fim, para classificar a faixa em uma classe, é computado o voto majoritário simples dentre as classes das texturas, ou seja, o gênero mais frequente dentre as texturas é atribuído à faixa.

Os parâmetros testados para o conjunto de características MARSYAS e Mel-Spec estão ilustrados na Tabela 4.1, para o conjunto obtido pela rede *autoencoder* (AE) na Tabela 4.2 e por fim, para projeções aleatórias (RP) na Tabela 4.3.

Tabela 4.1. Parâmetros usados para características MARSYAS e Mel-Spec

	Seletor de Texturas	**No. Trechos**
MARSYAS	*KMeans* e *Linspace*	[5, 20, 40]
MEL-Spec	*KMeans* e *Linspace*	[5, 20, 40]

Tabela 4.2. Parâmetros usados para características AE

	Seletor de Texturas	**No. Trechos**	**No. Características Aprendidas**
AE	*KMeans* e *Linspace*	[5, 20, 40]	[16, 32. 64, 128, 256]

Tabela 4.3. Parâmetros usados para características RP

	Seletor de Texturas	**No. Trechos**	**Tam. Dimensão Alvo**
RP	*KMeans* e *Linspace*	[5, 20, 40]	[9, 26, 51, 75, 100]

Para cada conjunto de características, foram obtidos os votos gerados pela saída do SCTM. Este processo se repetiu para todos as bases de dados.

4.2. Sistema de Aprendizagem de Padrões de Votação

A principal contribuição deste trabalho foi a elaboração do Sistema de aprendizagem de padrões de votação (SAPV). A ideia é obter a classificação de cada textura do SCTM e explorar como estes votos podem ser combinados para cada faixa. A principal restrição imposta é que não há conhecimento das probabilidades de cada textura pertencer a cada classe.

A Figura 4.2 ilustra o sistema proposto, recebendo a classificação de texturas do

SCTM como entrada. A partir da combinação dos votos é inferido uma classe à faixa a qual originou os votos.

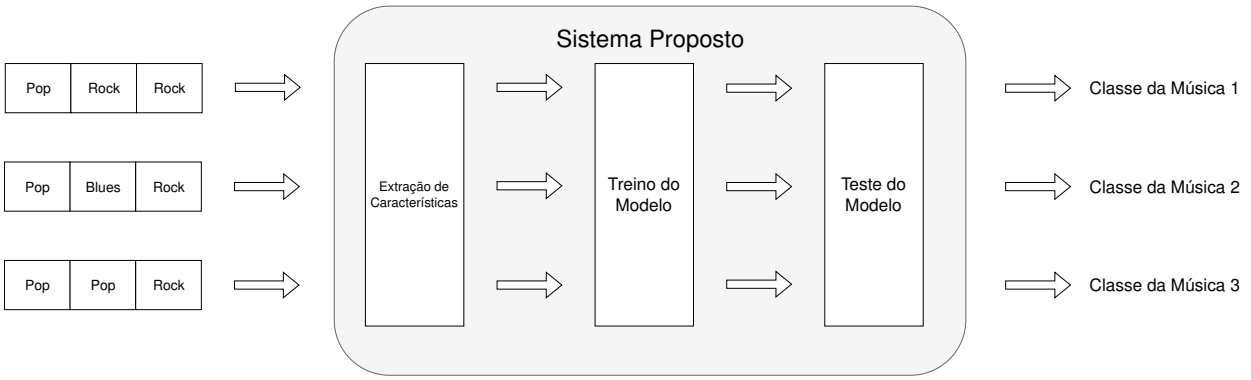


Figura 4.2. Diagrama de funcionamento do sistema proposto

Quatro classificadores distintos são utilizados para realizar a classificação dos votos atribuídos a cada trecho e o rótulo da faixa correspondente, sendo: K-Vizinhos Mais Próximos (K-NN), Máquina de Vetores de Suporte (SVM), *Hidden Markov Model* (HMM) e redes neurais recorrentes. É importante destacar que não houve fusão de classificadores, apenas foi comparado a acurácia resultante de cada um.

A ideia do uso dos classificadores K-NN e SVM é avaliar se é possível realizar a combinação da contagem de votos de maneira a inferir a classe correta à faixa a qual os votos pertencem. O objetivo do uso dos classificadores HMM e redes neurais recorrentes é avaliar se a ordem temporal dos votos é relevante para a classificação do gênero de uma faixa através da sequência de votos. Os classificadores serão abordados com mais detalhes na Seção 4.4.

4.3. Extração de Características

4.3.1. Extração dos Histogramas de Votação

Tendo como objetivo a aprendizagem de padrões de votação para inferir o rótulo desejado às classes, as distribuições de frequência (histogramas) dos votos foram obtidas através da predição do gênero de texturas de faixas pelo SCTM.

Os histogramas possuem n posições, sendo n a quantidade de classes da base de dados. Vale ressaltar que cada posição $1...n$ do histograma deve representar unicamente um gênero. Por exemplo, a primeira posição do histograma deve representar o gênero blues, a segunda posição representa o gênero clássico, assim até todas as n classes serem mapeadas à uma posição fixa no histograma.

Figura 4.3. Classificador de texturas musicais

Suponha o resultado de uma faixa classificada pelo SCTM da Figura 4.3. A contagem de aparições de cada gênero nos trechos é realizada. Este número é designado a determinada posição no histograma. Por fim, os histogramas são utilizados como característica do modelo desenvolvido por esta monografia.

Histograma Resultante	[	**2**	**1**	**0**	**0**	**1**	]
		Blues	Disco	Pop	Rock	Jazz	

4.3.2. Extração das Sequências de Votação

Para fornecer dados aos classificadores HMM e rede neurais recorrente, que levam em consideração a ordem temporal da entrada, foram obtidas as sequências de votos através da saída do SCTM.

Os dados submetidos aos classificadores temporais no sistema proposto se restringiu a saída do SCTM somente para o algoritmo de seleção de texturas *linspace*. O uso de votos obtidos somente pelo *linspace* se dá pois o algoritmo *K-Means* não fornece informações temporais.

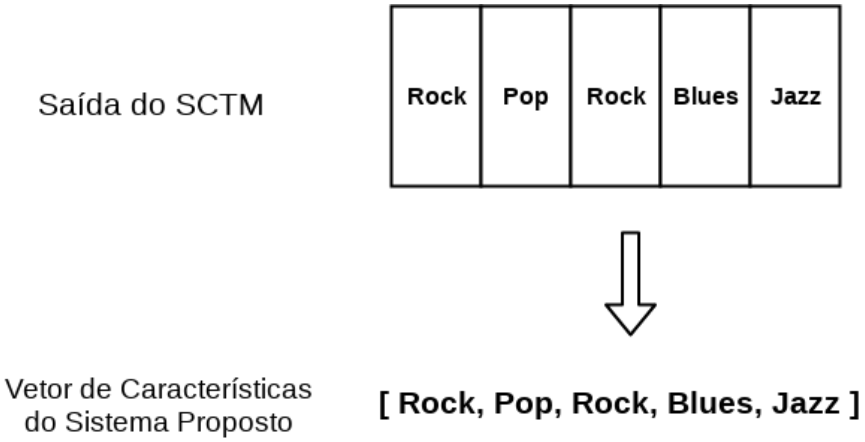


Figura 4.4. Procedimento de extração das sequências de votação

A Figura 4.4 ilustra o processo de obtenção das características de sequências de votação. O gênero da textura n é colocado na posição n do vetor de características.

4.4. Classificação

Cada base de dados foi testada separadamente para cada conjunto de características usadas pelo SCTM. Em cada base, foi verificada a influência da aprendizagem dos padrões de votação com os parâmetros do sistema classificador de texturas musicais.

Afim de se obter a confiabilidade no resultado final, foi utilizada a técnica de validação cruzada denominada *k-fold*, abordada na Seção 2.2. Para cada classificador, foram utilizados os mesmos *folds* em cada base de dados. Estes valores estão descritos na coluna *Folds* da Tabela 4.7.

4.4.1. Classificação dos Histogramas

O objetivo de usar os histogramas de votação é verificar se os erros de rotulação do SCTM podem ser compensados pela detecção de padrões na distribuição dos votos entre os gêneros.

Na fase de treinamento dos classificadores dos histogramas (SVM e K-NN), foram utilizados os mesmos *folds* que o SCTM utilizou para treino e teste. Os classificadores foram treinados com os histogramas de votação obtidos do SCTM. Os rótulos utilizados foram os rótulos verdadeiros de cada faixa, não o rótulo atribuído pela votação majoritária realizada pelo SCTM.

Validação cruzada foi usada também para escolher hiperparâmetros que otimizam o resultado da classificação. Para o algoritmo SVM, os parâmetros C e $Gamma$ foram testados. Para o K-NN, apenas o parâmetro k foi testado. Os valores para cada parâmetro testado está inserido na Tabela 4.4.

Tabela 4.4. Hiperparâmetros testados por *Grid-search*

Classificador	Parâmetro	Intervalo
SVM	*C*	{ 0.001, 0.01, 0.1, 1, 10, 100 }
SVM	*Gamma*	{ 0.001, 0.01, 0.1, 1 }
K-NN	*k*	{ 1, 3, 5, 7, 9, 11, 13, 15, 17, 19 }

4.4.2. Classificação das Sequências de Votação

Uma das possibilidades que investigamos foi se a sequencia dos votos emitidos pelo SCTM pode ser explorada para inferir as classes das faixas correspondentes. Os classificadores HMM e redes neurais recorrentes foram utilizados para classificar as sequências de votações.

De forma análoga a classificação dos histogramas, na fase de treinamento dos classificadores capazes de modelarem sequências de votos usaram os mesmos *folds* para treino e teste que o SCTM utilizou estas fases. O rótulo apresentado juntamente com as sequências de votos foi o rótulo verdadeiro da faixa.

Para a classificação das sequências de votos utilizando o classificador HMM foram testados diferentes parâmetros, sendo eles: número de estados ocultos, tipo de matriz de covariância e número de iterações. Os valores dos parâmetros testados estão descritos na Tabela 4.5.

Tabela 4.5. Parâmetros utilizados para o classificador HMM

Parâmetro	Valores Testados
No. de Estados Ocultos	{3, 5, 7, 9}
Tipo de Matriz de Covariância	{full, diag}
No. de Iterações	{50, 100, 150}

Nas redes neurais recorrentes foi proposto uma arquitetura com 4 camadas, sendo uma camada recorrente. A representação da arquitetura está ilustrada na Figura 4.5. Os parâmetros utilizados, tal como a quantidade de neurônios de cada camada é informado na Tabela 4.6.

Tabela 4.6. Parâmetros da rede neural proposta

Camada	Tipo	Ativação	Quantidade de Neurônios
1ª	Densa	Linear	1
2ª	RNN Simples / LSTM	Tangente Hiperbólica	{20, 30, 40, 50}
3ª	Densa	ReLu	{5, 10, 15}
4ª	Densa	Softmax	{9, 10, 13}

Na camada recorrente (camada 2), foram testados dois tipos de neurônios: neurônios RNN tradicional e neurônios LSTM. Na camada de saída, a quantidade de neurônios se dá pela base de dados a ser classificada, logo, a quantidade de neurônios corresponde a quantidade de classes disponíveis. O modelo foi treinado por 1000 épocas. Foram testadas

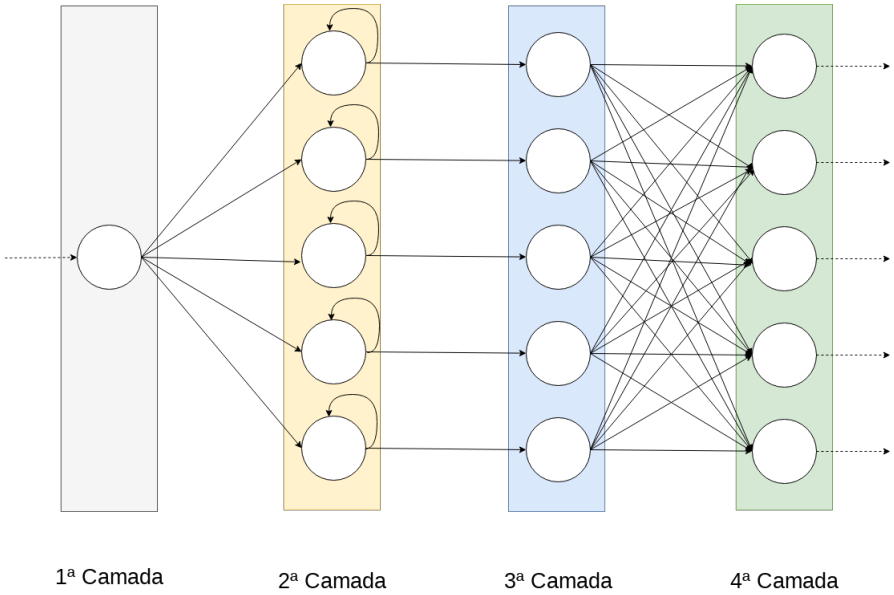


Figura 4.5. Esquema da arquitetura da rede neural proposta

2 métricas para salvar o modelo após cada época, sendo erro na validação ou acurácia na validação. Em um experimento inicial foram testadas ambas as métricas mas optamos por usar a métrica de acurácia na validação para escolher o melhor modelo nos demais conjuntos de dados. No final do treino o modelo com a melhor acurácia na validação foi restaurado para a classificação do conjunto de teste. O otimizador Adam foi usado com os parâmetros padrão para treinar os modelos.

4.5. Bases de Dados

Foram escolhidas 4 base de dados distintas com diferentes quantidades de faixas e tamanhos, além da distinção de gêneros e quantidade de *folds* representada na Tabela 4.7.

Tabela 4.7. Base de Dados Utilizada

Base de Dados	Faixas	Classes	Balanceado	Tamanho da Faixa	*Folds*
GTZAN	1000	10	Sim	30 s	10
LMD	1300	10	Sim	Completa	3
HOMBURG	1886	9	Não	10 s	10
EXBALLROOM	4180	13	Não	30 s	10

A base de dados GTZAN (TZANETAKIS; COOK, 2002), utilizada na maioria das pesquisas relacionados à *Music Information Retrieval* (MIR) para comparação da eficiência dos sistemas propostos, é composta por 10 gêneros. Cada gênero é composto por 100 faixas. Os gêneros são: blues, clássico, country, disco, hip hop, jazz, metal, pop, reggae e rock.

Um subconjunto da base *Latin Music Database* (LMD) (SILLA et al., 2008) também foi usado. Composta por 1300 faixas divididas igualmente entre os gêneros axé, bachata, bolero, forró, gaúcha, merengue, pagode, salsa, sertanejo e tango. Um filtro de artista foi

aplicado à este subconjunto, evitando músicas com mesmo artista estar presente no mesmo *fold*. Desta forma, o modelo não tem a possibilidade de detectar o artista ao invés de generalizar os padrões de gênero musical. Logo, é garantido que o sistema não acertará o gênero devido ao reconhecimento das faixas do artista.

As 1886 faixas distribuídas de forma desbalanceada entre 9 gêneros da base HOMBURG (HOMBURG et al., 2005) foram utilizadas. Estes gêneros são: alternativo, blues, eletrônica, country/folk, funk/soul/R&B, jazz, pop, rap/hiphop e rock. A principal característica desta base de dados é que todas as faixas são curtas, com apenas 10 segundos de duração.

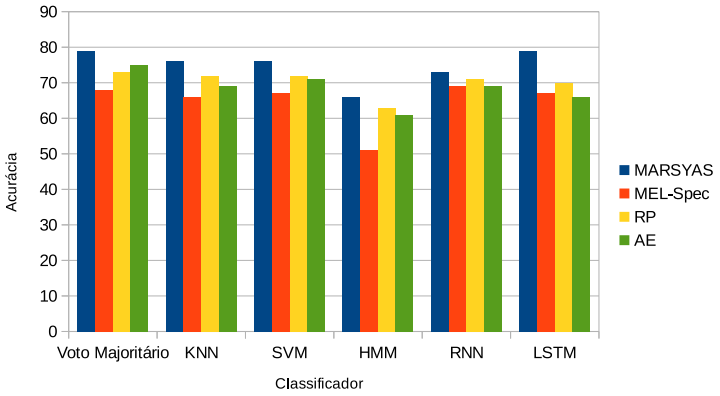
A base EXBALLROOM (MARCHAND; PEETERS, 2016) foi escolhida para avaliar o sistema com situações com um número mais elevado de faixas. Esta base é composta por 4180 faixas desbalanceadas entre os gêneros chacha, foxtrot, jive, pasodoble, quickstep, rumba, salsa, samba, valsa lenta, tango, valsa vienense, valsa e west coast swing. Outra característica notável é que EXBALLROOM é composta de vários subconjuntos de gêneros com timbres muito parecidos, cuja diferença principal está em atributos rítmicos. Estas características são desafiadoras para sistemas que dependem principalmente de características invariantes a ritmo.

Capítulo 5

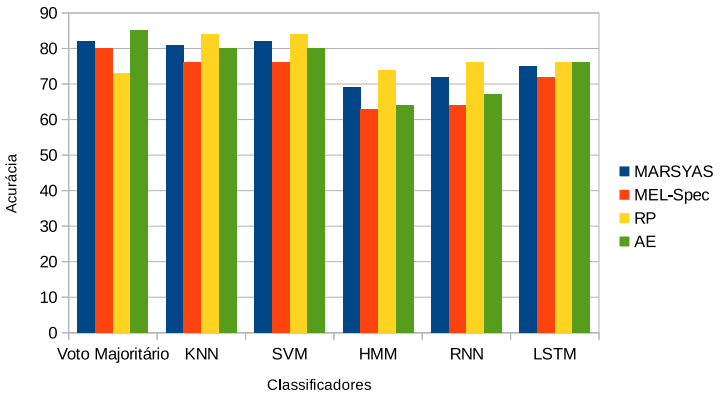
Resultados

Neste capítulo são apresentados os resultados obtidos através dos 4 classificadores: K-Vizinhos Mais Próximos (K-NN), Máquina de Vetores de Suporte (SVM), *Hidden Markov Model* (HMM) e rede neural recorrente. Para a rede neural recorrente, são informados os resultados obtidos através da arquitetura utilizando neurônios recorrentes tradicionais e neurônios *Long Short Term Memory* (LSTM) separadamente. Todos os resultados dos classificadores abordados neste trabalho são comparados com o resultado obtido pelo voto majoritário com o objetivo de avaliar o ganho obtido.

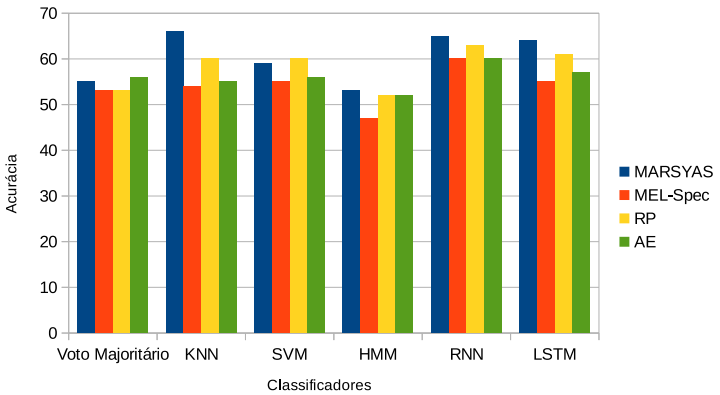
A Figura 5.1 apresenta os melhores resultados obtidos com cada um dos classificadores em todos cenários avaliados. Os resultados para RNN + RP e LSTM + RP para a base EXBALLROOM foram omitidos pois não tivemos recursos computacionais para executar todos os experimentos. Para grande parte das bases de dados e extratores de características, os classificadores K-NN e SVM apresentaram resultados inferiores ou iguais ao voto majoritário. Entretanto, houve ganho estatisticamente significativo para a base de dados HOMBURG.



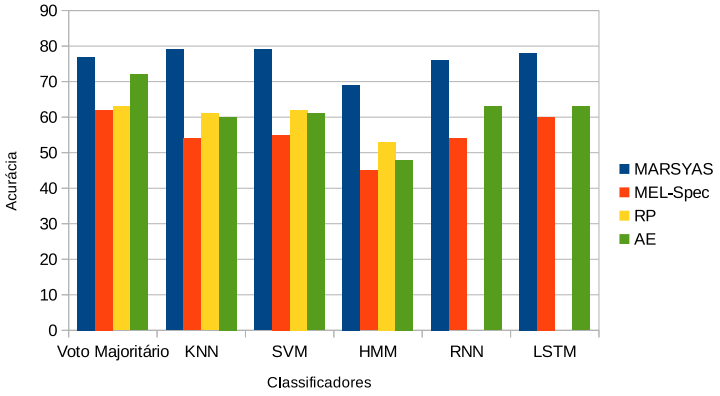
(a) GTZAN



(b) LMD



(c) HOMBURG



(d) EXBALLROOM

Figura 5.1. Melhores resultados

As Tabelas 5.1 e 5.2 apresentam resultados que foram superiores ao voto majoritário da classificação dos histogramas de votação com K-NN e SVM respectivamente, com grau de confiabilidade de 95%. Resultados diferentes estatisticamente com aumento na acurácia foram obtidos de 3 métodos de extração de características no SCTM nas bases de dados LMD e HOMBURG utilizando os classificadores que tomaram como entrada os histogramas.

Tabela 5.1. Resultados significativamente melhores utilizando K-NN comparados ao voto majoritário. O resultado apresentado é média da acurácia entre os *folds* da validação cruzada.

	MARSYAS	**MEL-Spec**	**RP**	**AE**
GTZAN	*	*	*	*
LMD	*	*	0.84 ± 0.01	*
HOMBURG	0.66 ± 0.07	*	0.51 ± 0.07	0.55 ± 0.04
EXBALLROOM	*	*	*	*

Tabela 5.2. Resultados significativamente melhores utilizando SVM comparados ao voto majoritário. O resultado apresentado é média da acurácia entre os *folds* da validação cruzada.

	MARSYAS	**MEL-Spec**	**RP**	**AE**
GTZAN	*	*	*	*
LMD	*	*	0.84 ± 0.01	*
HOMBURG	0.59 ± 0.04	*	0.57 ± 0.05	0.56 ± 0.04
EXBALLROOM	*	*	*	*

De forma geral, os classificadores K-NN e SVM mostraram resultados semelhantes. Na maioria das bases o SVM apresentou acurácia 1% maior que do K-NN. Vale lembrar que a simplicidade do algoritmo K-NN reflete no tempo de treinamento dos dados, sendo assim, mais rápido que o SVM. Isto se deve provavelmente à baixa dimensionalidade dos histogramas. Neste cenário, o KNN tende a funcionar muito bem, com o benefício de ser muito mais eficiente computacionalmente que SVM.

Os resultados obtidos via a classificação de histogramas de votação mostraram que, embora não tenham sido muito superiores ao voto da maioria, há certa lógica em aprender padrões de votação. Informalmente, o Sistema classificador de texturas musicais (SCTM) tem um erro consistente nas predições. Isto faz com que haja um padrão nestes erros. Desta forma faz sentido usar classificadores para compensar os erros de atribuição de rótulos pelo SCTM. A partir disso, também testamos se a sequência dos rótulos atribuídos pelo SCTM também podia ser explorada para compensar os erros de predição de rótulos. Para isso, testamos os três classificadores de séries temporais: HMM, RNN tradicional e LSTM.

O HMM obteve apenas um resultado superior ao voto majoritário. Para a base de dados *Latin Music Database* (LMD) utilizando características RP, o classificador de sequência de votos obteve 74% de taxa de acerto, sendo apenas 1% superior ao sistema base. O HMM não obteve nenhum resultado superior aos outros classificadores utilizados, independente das bases de dados.

As duas arquiteturas de rede neural recorrente não obteveram ganho significativo na maioria das bases. A maioria dos resultados obtidos pelas redes neurais diferiram pouco dos resultados obtidos pelos classificadores que tomavam como entrada histogramas de votos. Entretanto, houve ganho estatisticamente significativo para a base de dados HOMBURG utilizando as redes neurais com neurônios recorrentes tradicionais e neurônios LSTM. As Tabelas 5.3 e 5.4 ilustram os resultados estatisticamente diferentes com grau de confiabilidade de 95%.

Tabela 5.3. Resultados significativamente melhores utilizando RNN comparados ao voto majoritário. O resultado apresentado é média da acurácia entre os *folds* da validação cruzada.

	MARSYAS	**MEL-Spec**	**RP**	**AE**
GTZAN	*	*	*	*
LMD	*	*	*	*
HOMBURG	0.65 ± 0.08	*	0.63 ± 0.08	*
EXBALLROOM	*	*	*	*

Tabela 5.4. Resultados significativamente melhores utilizando LSTM comparados ao voto majoritário. O resultado apresentado é média da acurácia entre os *folds* da validação cruzada.

	MARSYAS	**MEL-Spec**	**RP**	**AE**
GTZAN	*	*	*	*
LMD	*	*	*	*
HOMBURG	0.64 ± 0.07	*	0.61 ± 0.08	*
EXBALLROOM	*	*	*	*

De forma geral, os resultados dos classificadores temporais foram muito semelhantes aos resultados dos classificadores de histogramas de votos. Com os recursos computacionais disponíveis não foi possível testar outras arquiteturas para as redes neurais. Desta forma, não há como concluir que resultados melhores não são possíveis com o uso de arquiteturas mais poderosas.

Diferente das outras bases de dados, houveram ganhos significativos na base de dados HOMBURG. Este é um resultado interessante pois esta base representa um grande desafio para sistemas de classificação de gêneros musicais. Isto se deve ao fato que os áudios tem apenas 10 segundos. Assim, a textura musical não varia tanto. Desta forma, os modelos gerados a partir desta base não são tão ricos quanto os obtidos em outras bases de dados. O melhor resultado reportado para esta base na literatura é $64.3\% \pm 2.5$ de acurácia (PANAGAKIS et al., 2014). Portanto, o resultado $66\% \pm 7$ é estatisticamente estado-da-arte. Isto representa um ganho médio de 17.86% em relação ao voto majoritário.

Embora os resultados apresentados não mostrem ganhos significativos na maioria dos cenários testados, eles mostram que a ideia de aprender padrões de votação tem fundamento. Em geral os resultados dos classificadores de histogramas foram muito parecidos com o voto majoritário. Com a representação de histograma a estrutura de voto majoritário está

embutida nos casos onde o histograma foi corretamente classificado pelo voto majoritário. Ao apresentar um histograma cujo voto majoritário está correto, o classificador tende a associar o comportamento do voto majoritário. Portanto, o voto majoritário pode ser visto como um limite inferior para classificação de histogramas. Os resultados dos classificadores de sequencia mostram que a sequência dos votos está correlacionada com os rótulos, devido à proximidade com os resultados obtidos no voto majoritário. No entanto, é necessário testar outras arquiteturas de redes neurais para tirar conclusões sobre seus limites.

Capítulo 6

Conclusão e Trabalhos Futuros

Vários trabalhos de classificação automática de gêneros musicais dividem as faixas em trechos menores e atribuem um rótulo para cada trecho. No final, é necessário usar alguma política de combinação dos rótulos atribuídos aos trechos de uma faixa para decidir um rótulo final a ela.

Tradicionalmente, quando não se conhece a distribuição de probabilidade de cada trecho pertencer a cada uma das classes, a decisão final de uma faixa assumir um rótulo de saída se restringe ao voto majoritário. Esta pesquisa avaliou dois métodos alternativos novos de combinação de votos quando a distribuição de probabilidades por textura não está disponível. O primeiro método consiste em representar a votação obtida de cada faixa por histogramas com a contagem de quantos votos cada classe recebeu. O segundo método consiste em utilizar sequências de votos ordenados em relação ao tempo da faixa. Estes votos são provenientes de um sistema de classificação de texturas musicais.

Os resultados mostraram que ambas as abordagens obtiveram resultados semelhantes ao voto majoritário. No entanto, na base HOMBURG houve ganho significativo na acurácia obtida em praticamente todos os cenários testados em relação ao voto majoritário. No melhor caso da base HOMBURG, houve um ganho de 17.86% em relação ao voto majoritário, chegando a $66\% \pm 7$ de acurácia. Este é um resultado estado-da-arte. Este resultado é interessante pois esta base é conhecida por ser desafiadora no problema de classificação automática de gêneros musicais. Os resultados indicam que o a aprendizagem de padrões de votação pode trazer ganhos em alguns casos e que é uma tarefa de pós-processamento promissora para pesquisas futuras.

Referências

BAHLMANN, C.; HAASDONK, B.; BURKHARDT, H. Online handwriting recognition with support vector machines - a kernel approach. In: *Proceedings Eighth International Workshop on Frontiers in Handwriting Recognition*. Niagara on the Lake, Ontario, Canada, Canada: IEEE, 2002. p. 49–54.

BANNAYAN, Mohammad; HOOGENBOOM, Gerrit. Weather analogue: a tool for real-time prediction of daily weather data realizations based on a modified k-nearest neighbor approach. *Environmental Modelling & Software*, Elsevier, v. 23, n. 6, p. 703–713, 2008.

BENGIO, Yoshua; SIMARD, Patrice; FRASCONI, Paolo. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, v. 5, n. 2, p. 157–166, 1994.

BIJALWAN, Vishwanath; KUMAR, Vinay; KUMARI, Pinki; PASCUAL, Jordan. Knn based machine learning approach for text and document mining. *International Journal of Database Theory and Application*, v. 7, n. 1, p. 61–70, 2014.

BILMES, Jeff A et al. A gentle tutorial of the em algorithm and its application to parameter estimation for gaussian mixture and hidden markov models. *International Computer Science Institute*, v. 4, n. 510, p. 126, 1998.

CASEY, M. A.; VELTKAMP, R.; GOTO, M.; LEMAN, M.; RHODES, C.; SLANEY, M. Content-based music information retrieval: Current directions and future challenges. *Proceedings of the IEEE*, v. 96, n. 4, p. 668–696, April 2008. ISSN 0018-9219.

CATALTEPE, Zehra; YASLAN, Yusuf; SONMEZ, Abdullah. Music genre classification using midi and audio features. *EURASIP Journal on Advances in Signal Processing*, Springer, v. 2007, n. 1, 2007.

CHEN, Hao; DUMAIS, Susan. Bringing order to the web: Automatically categorizing search results. In: ACM. *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*. The Hauge, South Holland, Netherlands, 2000. p. 145–152.

CHOI, Keunwoo; FAZEKAS, George; SANDLER, Mark. Automatic tagging using deep convolutional neural networks. *arXiv preprint arXiv:1606.00298*, 2016.

COSTA, Yandre MG; OLIVEIRA, Luiz S; JR, Carlos N Silla. An evaluation of convolutional neural networks for music classification using spectrograms. *Applied soft computing*, Elsevier, v. 52, p. 28–38, 2017.

COSTA, Yandre MG; OLIVEIRA, Luiz S; KOERICB, Alessandro L; GOUYON, Fabien. Music genre recognition using spectrograms. In: IEEE. *Systems, Signals and Image Processing (IWSSIP), 2011 18th International Conference on*. Sarajevo, Bosnia-Herzegovina, 2011. p. 1–4.

DUDA, Richard O.; HART, Peter E.; STORK, David G. *Pattern Classification (2nd Edition)*. New York, NY, USA: Wiley-Interscience, 2000. ISBN 0471056693.

FAUSETT, Laurene V et al. *Fundamentals of neural networks: architectures, algorithms, and applications*. Prentice-Hall, Inc. Upper Saddle River, NJ, USA: Pearson, 1994. v. 3.

FOLEISS, Juliano Henrique. Automatic genre classification by representing tracks with multiple vectors. 2018.

GANIN, Yaroslav; KULKARNI, Tejas; BABUSCHKIN, Igor; ESLAMI, S. M. Ali; VINYALS, Oriol. Synthesizing programs for images using reinforced adversarial learning. *CoRR*, abs/1804.01118, 2018. Disponível em: <http://arxiv.org/abs/1804.01118>.

GERS, F. A.; SCHMIDHUBER, J.; CUMMINS, F. Learning to forget: continual prediction with lstm. In: *1999 Ninth International Conference on Artificial Neural Networks ICANN 99. (Conf. Publ. No. 470)*. Lugano,Ticino,Switzerland: IET, 1999. v. 2, p. 850–855 vol.2. ISSN 0537-9989.

GHAHRAMANI, Zoubin. An introduction to hidden markov models and bayesian networks. *International journal of pattern recognition and artificial intelligence*, World Scientific, v. 15, n. 01, p. 9–42, 2001.

GRAVES, Alex; MOHAMED, Abdel-rahman; HINTON, Geoffrey. Speech recognition with deep recurrent neural networks. In: IEEE. *Acoustics, speech and signal processing (icassp), 2013 ieee international conference on*. Vancouver,Canada, 2013. p. 6645–6649.

GROSSBERG, Stephen. Nonlinear neural networks: Principles, mechanisms, and architectures. *Neural Networks*, v. 1, n. 1, p. 17 – 61, 1988. ISSN 0893-6080. Disponível em: <http://www.sciencedirect.com/science/article/pii/0893608088900214>.

GUO, Guodong; LI, Stan Z; CHAN, Kapluk. Face recognition by support vector machines. In: IEEE. *Automatic Face and Gesture Recognition, 2000. Proceedings. Fourth IEEE International Conference on*. Face recognition by support vector machines: IEEE, 2000. p. 196–201.

HALL, Peter; PARK, Byeong U; SAMWORTH, Richard J. Choice of neighbor order in nearest-neighbor classification. *The Annals of Statistics*, JSTOR, p. 2135–2152, 2008.

HAYKIN, S.; HAYKIN, S.S. *Neural Networks and Learning Machines*. Prentice Hall, 2009. (Neural networks and learning machines, v. 10). ISBN 9780131471399. Disponível em: <https://books.google.com.br/books?id=K7P36lKzI_QC>.

HEISELE, B.; HO, P.; POGGIO, T. Face recognition with support vector machines: global versus component-based approach. In: *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*. Vancouver, BC, Canada: IEEE, 2001. v. 2, p. 688–694.

HENAFF, Mikael; JARRETT, Kevin; KAVUKCUOGLU, Koray; LECUN, Yann. Unsupervised learning of sparse features for scalable audio classification. In: CITESEER. *ISMIR*. Miami,Florida,USA, 2011. v. 11, n. 445, p. 2011.

HOCHREITER, Sepp; SCHMIDHUBER, Jürgen. Long short-term memory. *Neural computation*, MIT Press, v. 9, n. 8, p. 1735–1780, 1997.

HOMBURG, Helge; MIERSWA, Ingo; MÖLLER, Bülent; MORIK, Katharina; WURST, Michael. A benchmark dataset for audio classification and clustering. In: . Malaga, Spain: ISMIR, 2005. v. 2005, p. 528–31.

IMANDOUST, Sadegh Bafandeh; BOLANDRAFTAR, Mohammad. Application of k-nearest neighbor (knn) approach for predicting economic events: Theoretical background. *International Journal of Engineering Research and Applications*, v. 3, n. 5, p. 605–610, 2013.

JAJODIA, Sushil; MUTCHLER, David. Dynamic voting algorithms for maintaining the consistency of a replicated database. *ACM Transactions on Database Systems (TODS)*, ACM, v. 15, n. 2, p. 230–280, 1990.

JANG, Dalwon; JIN, Minho; YOO, C. D. Music genre classification using novel features and a weighted voting method. In: *2008 IEEE International Conference on Multimedia and Expo*. Hannover, Germany: IEEE, 2008. p. 1377–1380. ISSN 1945-7871.

JOACHIMS, Thorsten. Text categorization with support vector machines: Learning with many relevant features. In: SPRINGER. *European conference on machine learning*. Chemnitz, Germany, 1998. p. 137–142.

KALCHBRENNER, Nal; ELSEN, Erich; SIMONYAN, Karen; NOURY, Seb; CASAGRANDE, Norman; LOCKHART, Edward; STIMBERG, Florian; OORD, Aäron van den; DIELEMAN, Sander; KAVUKCUOGLU, Koray. Efficient neural audio synthesis. *CoRR*, abs/1802.08435, 2018. Disponível em: <http://arxiv.org/abs/1802.08435>.

KAMIJO, Ken-ichi; TANIGAWA, Tetsuji. Stock price pattern recognition-a recurrent neural network approach. In: IEEE. *Neural Networks, 1990., 1990 IJCNN International Joint Conference on*. San Diego, CA, USA, 1990. p. 215–221.

KHIATANI, D.; GHOSE, U. Weather forecasting using hidden markov model. In: *2017 International Conference on Computing and Communication Technologies for Smart Nation (IC3TSN)*. Gurgaon, India: IC3TSN, 2017. p. 220–225.

KITTLER, J.; HATEF, M.; DUIN, R. P. W.; MATAS, J. On combining classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 20, n. 3, p. 226–239, Mar 1998. ISSN 0162-8828.

KOHAVI, Ron et al. A study of cross-validation and bootstrap for accuracy estimation and model selection. In: *Ijcai*. Montreal, Canada: Ijcai, 1995. v. 14, n. 2, p. 1137–1145.

KUNCHEVA, Ludmila I. *Combining pattern classifiers: methods and algorithms*. New Jersey,USA: John Wiley & Sons, 2004. 113-127 p.

LEE, Kyogu; SLANEY, Malcolm. Automatic chord recognition from audio using a hmm with supervised learning. In: *ISMIR*. Victoria, Canada: ISMIR, 2006. p. 133–137.

LEE, Kai-Fu; HON, Hsiao-Wuen; HWANG, Mei-Yuh; HUANG, Xuedong. Speech recognition using hidden markov models: A cmu perspective. *Speech Communication*, v. 9, n. 5, p. 497 – 508, 1990. ISSN 0167-6393. Neuropeech '89. Disponível em: <http://www.sciencedirect.com/science/article/pii/0167639390900255>.

LIPPENS, S.; MARTENS, J. P.; MULDER, T. De. A comparison of human and automatic musical genre classification. In: *2004 IEEE International Conference on Acoustics, Speech, and Signal Processing*. Montreal, Que., Canada: IEEE, 2004. v. 4, p. iv–233–iv–236 vol.4. ISSN 1520-6149.

LIU, Cheng-Lin; NAKASHIMA, Kazuki; SAKO, Hiroshi; FUJISAWA, Hiromichi. Handwritten digit recognition: benchmarking of state-of-the-art techniques. *Pattern recognition*, Elsevier, v. 36, n. 10, p. 2271–2285, 2003.

LORENA, Ana Carolina; CARVALHO, André CPLF de. Uma introdução às support vector machines. *Revista de Informática Teórica e Aplicada*, v. 14, n. 2, p. 43–67, 2007.

MARCHAND, Ugo; PEETERS, Geoffroy. Scale and shift invariant time/frequency representation using auditory statistics: Application to rhythm description. In: IEEE. *Machine Learning for Signal Processing (MLSP), 2016 IEEE 26th International Workshop on*. Vietri sul Mare, Salerno, Italy, 2016. p. 1–6.

MEDHAT, Fady; CHESMORE, David; ROBINSON, John. Automatic classification of music genre using masked conditional neural networks. *CoRR*, abs/1801.05504, 2018. Disponível em: <http://arxiv.org/abs/1801.05504>.

MONARD, Maria Carolina; BARANAUSKAS, José Augusto. Conceitos sobre aprendizado de máquina. *Sistemas inteligentes-Fundamentos e aplicações*, v. 1, n. 1, p. 32, 2003.

NORONHA, Daniel H; FERNANDES, Marcelo Augusto Costa. Implementação em fpga de máquina de vetores de suporte (svm) para classificação e regressão. *Encontro Nacional de Inteligencia Artificial e Computacional*, 2016.

OLAH, Christopher. *Understanding LSTM Networks*. 2015. <http://colah.github.io/posts/2015-08-Understanding-LSTMs>. Acessado em: 10/11/2018.

OSUNA, Edgar; FREUND, Robert; GIROSIT, Federico. Training support vector machines: an application to face detection. In: IEEE. *Computer vision and pattern recognition, 1997. Proceedings., 1997 IEEE computer society conference on*. San Juan, Puerto Rico, USA, USA, 1997. p. 130–136.

PAMPALK, Elias; FLEXER, Arthur; WIDMER, Gerhard et al. Improvements of audio-based music similarity and genre classificaton. In: LONDON, UK. *ISMIR*. London, UK, 2005. v. 5, p. 634–637.

PANAGAKIS, Y.; KOTROPOULOS, C. L.; ARCE, G. R. Music genre classification via joint sparse low-rank representation of audio features. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, v. 22, n. 12, p. 1905–1917, Dec. 2014. ISSN 2329-9290.

PARIS, J. F.; LONG, D. D. E. Efficient dynamic voting algorithms. In: *Proceedings. Fourth International Conference on Data Engineering*. New York, NY, USA: ACM Transactions on Database Systems (TODS), 1988. p. 268–275.

RABINER, L. R. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, v. 77, n. 2, p. 257–286, Feb 1989. ISSN 0018-9219.

RABINER, L. R.; JUANG, B. .; LEVINSON, S. E.; SONDHI, M. M. Recognition of isolated digits using hidden markov models with continuous mixture densities. *AT T Technical Journal*, v. 64, n. 6, p. 1211–1234, July 1985. ISSN 8756-2324.

RIFKIN, Ryan; KLAUTAU, Aldebaro. In defense of one-vs-all classification. *Journal of machine learning research*, v. 5, n. Jan, p. 101–141, 2004.

SAK, Haşim; SENIOR, Andrew; BEAUFAYS, Françoise. Long short-term memory recurrent neural network architectures for large scale acoustic modeling. In: *Fifteenth annual conference of the international speech communication association*. Singapore: INTERSPEECH, 2014.

SANJEEVI, Madhu. *Support Vector machine with Math*. 2017. <https://medium.com/deep-math-machine-learning-ai/chapter-3-support-vector-machine-with-math-47d6193c82be>. Acessado em: 09/06/2018.

SILLA, Carlos Nascimento; KOERICH, Alessandro L; KAESTNER, Celso AA. The latin music database. In: . Philadelphia, PA: ISMIR, 2008. p. 451–456.

SINGH, Archana; YADAV, Avantika; RANA, Ajay. K-means with three different distance metrics. *International Journal of Computer Applications*, Foundation of Computer Science, v. 67, n. 10, 2013.

TZANETAKIS, G.; COOK, P. Musical genre classification of audio signals. *IEEE Transactions on Speech and Audio Processing*, v. 10, n. 5, p. 293–302, Jul 2002. ISSN 1063-6676.

WÜLFING, Jan; RIEDMILLER, Martin A. Unsupervised learning of local features for music classification. In: *3th International Society for Music Information Retrieval Conference*. Porto,Portugal: ISMIR, 2012. p. 139–144.

XU, Changsheng; MADDAGE, N. C.; SHAO, Xi; CAO, Fang; TIAN, Qi. Musical genre classification using support vector machines. In: *2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP '03)*. Hong Kong: ICASSP, 2003. v. 5, p. V–429. ISSN 1520-6149.