Universidade Federal de Ouro Preto
Instituto de Ciências Exatas e Aplicadas
Departamento de Computação e Sistemas

Uma abordagem utilizando geometria computacional para detecção de anomalias

Bruno César Cota Conceição

TRABALHO DE CONCLUSÃO DE CURSO

ORIENTAÇÃO:
Luiz Carlos Bambirra Torres

Setembro, 2021

João Monlevade–MG

Bruno César Cota Conceição

Uma abordagem utilizando geometria computacional para detecção de anomalias

Orientador: Luiz Carlos Bambirra Torres

Monografia apresentada ao curso de Engenharia de Computação do Instituto de Ciências Exatas e Aplicadas, da Universidade Federal de Ouro Preto, como requisito parcial para aprovação na Disciplina "Trabalho de Conclusão de Curso II".

Universidade Federal de Ouro Preto

João Monlevade

Setembro de 2021

SISBIN - SISTEMA DE BIBLIOTECAS E INFORMAÇÃO

C744u Conceição, Bruno César Cota.
Uma abordagem utilizando geometria computacional para detecção de anomalias. [manuscrito] / Bruno César Cota Conceição. - 2021.
74 f.: il.: color..

Orientador: Prof. Dr. Luiz Carlos Bambirra Torres.
Monografia (Bacharelado). Universidade Federal de Ouro Preto. Instituto de Ciências Exatas e Aplicadas. Graduação em Engenharia de Computação .

1. Anomalias - Detecção. 2. Aprendizagem de máquina. 3. Teoria dos grafos. 4. Conjunto de dados - Classificação. I. Torres, Luiz Carlos Bambirra. II. Universidade Federal de Ouro Preto. III. Título.

CDU 004.8:004.62

Bibliotecário(a) Responsável: Sione Galvão Rodrigues - CRB6 / 2526

MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE OURO PRETO
REITORIA
INSTITUTO DE CIENCIAS EXATAS E APLICADAS
DEPARTAMENTO DE COMPUTACAO E SISTEMAS

FOLHA DE APROVAÇÃO

Bruno César Cota Conceição

Uma abordagem utilizando geometria computacional para detecção de anomalias

Monografia apresentada ao Curso de Engenharia de Computação da Universidade Federal
de Ouro Preto como requisito parcial para obtenção do título de Bacharel em Engenharia de Computação

Aprovada em 02 de setembro de 2021

Membros da banca

Doutor - Luiz Carlos Bambirra Torres - Orientador (Universidade Federal de Ouro Preto)
Doutora - Gilda Aparecida de Assis - (Universidade Federal de Ouro Preto)
Mestre - Eduardo da Silva Ribeiro - (Universidade Federal de Ouro Preto)

Luiz Carlos Bambirra Torres, orientador do trabalho, aprovou a versão final e autorizou seu depósito na Biblioteca Digital de Trabalhos de Conclusão de Curso da UFOP em 17/09/2021

Documento assinado eletronicamente por **Luiz Carlos Bambirra Torres**, **PROFESSOR DE MAGISTERIO SUPERIOR**, em 17/09/2021, às 14:28, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

A autenticidade deste documento pode ser conferida no site http://sei.ufop.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0 , informando o código verificador **0222105** e o código CRC **66F16C0C**.

R. Diogo de Vasconcelos, 122, - Bairro Pilar Ouro Preto/MG, CEP 35400-000
Telefone: - www.ufop.br

Este trabalho é dedicado a minha família, em especial a minha mãe, Efigênia Maria Cota da Conceição (In memoriam), que veio a falecer nos primeiros anos de minha graduação.

Agradecimentos

Agradeço primeiramente à Deus pela minha vida, saúde, inteligência, paciência e perseverança para desenvolver este estudo e sobretudo, por não me abandonar em nenhum momento de dificuldade, ao Espírito Santo por iluminar minha mente e conduzir os meus passos, ao meu anjo da guarda e a Santíssima Virgem Maria por passar a frente e conduzir meus passos.

A todos os colegas de graduação, aos profissionais do ICEA-UFOP, porteiros, serviços terceirizados, técnicos administrativos, toda equipe de profissionais da biblioteca e do RU, todos os professores, ao colegiado de Computação, enfim, a todos que fizeram parte da minha vida nestes anos da graduação, se realizasse com tal empenho e dedicação de tempo.

A todas as pessoas envolvidas nos projetos dos quais tive oportunidade de participar durante a graduação.

A todas as empresas e profissionais que tive a oportunidade de contribuir com minhas habilidades, desde antes da graduação, nas quais sempre me incentivaram a seguir minha qualificação profissional, o que não deixa de refletir também na minha postura enquanto ser humano.

Ao meu professor orientador Luiz Bambirra por todo apoio e confiança, toda dedicação nas orientações, por acreditar nas ideias propostas e incentivar o meu desenvolvimento.

A todas as pessoas que intercederam por mim em suas orações. Aos amigos, padrinhos e anjos sem asas que caminharam comigo ao longo destes anos.

Por fim, um agradecimento muito especial aos meus familiares, amigos, e a todos que direta ou indiretamente, contribuíram de alguma forma para que este projeto fosse realizado, incentivando, acreditaram e torceram por mim, o meu muito obrigado e que Deus os abençoe.

"Nunca vou esquecer de um rapaz, estudante, do qual todos sabiam que aspirava com decisão à santidade. Ele tinha este programa de vida. Ele sabia ter sido criado para os grandes ideais. Procurava o belo amor, e o procurava de joelhos, na oração."

— João Paulo II (1920 – 2005),
em: Cruzando o Limiar da Esperança.

Resumo

Este trabalho explora a teoria dos grafos para obter um algoritmo que consiga classificar um conjunto de dados, detectando anomalias, partindo da modelagem dos dados em um grafo e utilizando de operações e conceitos da área de grafos para solução da abordagem proposta, de modo que seja possível utilizar das propriedades próprias deste para análise do mesmo, possibilitando a classificação das classes de dados sem um conhecimento prévio destas. Na prática, após mapear os dados em uma estrutura de grafos, baseada no grafo de Gabriel. Segue-se o cálculo de distância das arestas, e o cálculo da média da distância entre as arestas, desta forma é possível encontrar as arestas que tem uma variação maior do que o limiar definido pelo e assim marcar as arestas de suporte. A partir daí, marcando o ponto médio das arestas de suporte, um hiperplano é traçado sobre o ponto médio. Os vértices da aresta de suporte estão na região de fronteira, na qual o hiperplano passa entre eles, no referido ponto médio. O classificador gera a classificação das classes, traçando a região de divisão das classes sobre os pontos médios, interligando os hiperplanos. Os resultados obtidos foram comparados aos da SVM *one-class*, onde foi observado uma ótima assertividade do método proposto, chegando a ser melhor que o SVM *one-class* para as bases testadas.

Palavras-chaves: Aprendizagem profunda. Detecção profunda de anomalias. Aprendizagem de máquina. Teoria dos grafos. Classificação de dados.

Abstract

This work explores the theory of graphs to obtain an algorithm that can classify a data set, detecting anomalies, starting from modeling the data in a graph and using operations and concepts from the graph area to solve the proposed approach, so that it is It is possible to use its own properties for its analysis, enabling the classification of data classes without prior knowledge of these. In practice, after mapping the data into a graph structure, based on Gabriel's graph. The calculation of the distance between the edges follows, and the calculation of the average of the distance between the edges, in this way it is possible to find the edges that have a variation greater than the threshold defined by and thus mark the support edges. From there, marking the midpoint of the supporting edges, a hyperplane is drawn over the midpoint. The vertices of the supporting edge are in the boundary region, in which the hyperplane passes between them, at said midpoint. The classifier generates the classification of the classes, tracing the region of division of the classes on the midpoints, connecting the hyperplanes. The results obtained were compared to those of the SVM *one-class*, where a great assertiveness of the proposed method was observed, even being better than the SVM *one-class* for the tested bases.

Key-words: Deep learning. Deep Anomaly Detection. Machine learning. Graph theory. Data Classification.

Lista de ilustrações

Lista de abreviaturas e siglas

DL	Deep learning
DAD	Deep Anomaly Detection
IA	Inteligência Artificial
DM	Data Mining
ML	Machine Learning
K-NN	k-nearest neighbors
MVE	Menor Volume Elipsóide
SVMs	Support Vector Machines
SVM	Support Vector Machine
SOMs	Self Organizing Maps
SOM	Self Organizing Map
SVDD	Support Vector Data Description
v-SVM	Variant of the Support Vector Machine
KKT	Karush-Kuhn-Tucker
SVs	Vetores de Suporte
SV	Vetor de Suporte
PERT	Project Evaluation Review Technique
CPM	Critical Path Method
MLP	Multilayer Perceptron
AS	Aresta de Suporte
GG	Grafo de Gabriel
TD	Triangulação de Delaunay
CBG	Classificação baseada em Grafos

Lista de símbolos

$k(\cdot, \cdot)$	Função kernel (semidefinida positiva)
$\Phi(\cdot)$	Transformação não linear do espaço de estados
X	Espaço de entrada
$ł_{sv}$	Conjunto de índices associados a SV
G	Grafo simples
V	Vértices de um grafo simples
E	Arestas de um grafo simples
$\mathcal{S}$	Conjunto de pontos do plano euclidiano
$\ddot{G}$	Grafo de Gabriel
$\mathcal{V}$	Conjunto de Vértices
$\mathcal{E}$	Conjunto de Arestas
$\underset{(dxN)}{X}$	Conjunto de amostras
d	Dimensão
N	Número de amostras
$\underset{(dxN)}{E}$	Matriz de arestas
$\underset{(NxN)}{D_X}$	Matriz de distâncias entre amostras
D_G	Matriz de distância dos vértices de $\ddot{G}$
D_L	Matriz de distâncias
N_L	Tamanho de D_L
L	Limear
A_S	Arestas de suporte
$\mathbf{x}$	Amostra a ser rotulada
l	Ponto médio

$\mathrm{D}(\cdot)$	Função que retorna a distância entre dois vetores
$H_l(\mathbf{x})$	Hiperplano de $\mathbf{x}$ que passa pelo ponto médio l
(b_l, w_l)	Par de vértices da Aresta de suporte
B e W	Parâmetros do hiperplano

Sumário

1 Introdução

O tratamento de dados tem se tornado cada vez mais necessário com a evolução da tecnologia e meios de comunicação. A informação correta no momento certo pode salvar vidas, evitar desastres, gerar economia, maximizar lucros, gerar intervenções adequadas em processos de manutenções de equipamentos e uma série de possibilidades.

Sabemos que a coleta de dados, seja por meio de dispositivos, amostragens, ou quaisquer que sejam as fontes, podem conter anomalias sejam elas geradas pelo meio de obtenção/coleta dos dados, ou ainda uma tendência natural da informação coletada.

Classificar esses dados de forma a diferenciar as normalidades dos dados em questão de alguma eventual discrepância, ou ainda detectar diferentes classes nos mesmos, requer o uso de técnicas de classificação que vem sendo exploradas no decorrer do tempo, sendo algumas bem difundidas nas mais várias áreas de estudo de tratamento de dados.

Obter uma solução precisa que seja capaz de retornar um resultado assertivo das classes para um conjunto de dados, indiferente de saber o seu comportamento ou origem e definir bem as classes que os compõem sem a necessidade de definir parâmetros para redes como os métodos tradicionais implica em um ganho de tempo considerável na análise dos resultados e desta forma, no levantamento das informações sobre os dados analisados.

A exemplo, temos a Figura 1 que mostra um conjunto de dados, onde estes são separados em duas classes distintas na Figura 2.

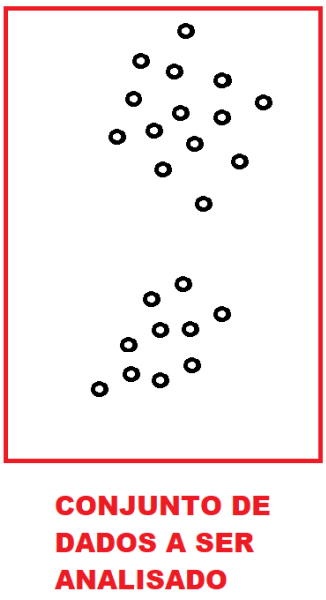

Figura 1 – Exemplo de dados a ser analisado.

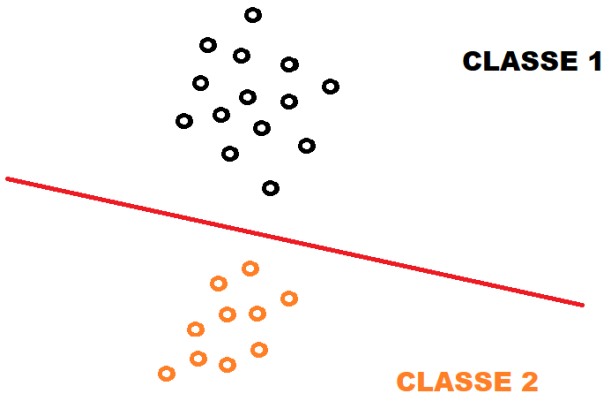

Figura 2 – Exemplo de separação dos dados em duas classes.

1.1 Problema

O problema abordado neste trabalho consiste na detecção de anomalias, também chamadas de *outliers* por alguns autores, em um conjunto de dados. Como o sistema de detecção de *outliers* lida com o *outlier*, depende da área de aplicação. Para entender melhor o problema, alguns contextos em que a detecção de anomalias pode ser utilizada são pontuados.

- Se o *outlier* indicar um erro tipográfico de um funcionário da entrada, então o funcionário da entrada pode ser notificado e simplesmente corrige o erro para que o *outlier* seja restaurado para um registro normal.
- Já um *outlier* resultante de uma leitura de um erro de um instrumento pode simplesmente ser eliminado.
- Uma pesquisa de características da população humana pode incluir anomalias como um punhado de pessoas muito altas. Aqui a anomalia é puramente natural, embora a leitura possa valer apenas para sinalizar a verificação e garantir que não haja erros, deve ser incluído na classificação, uma vez que seja verificada. Um sistema deve usar um algoritmo de classificação que é robusto para *outliers* para modelar dados com pontos como *outlier* de ocorrência natural.
- Por sua vez, um *outlier* em um ambiente crítico de segurança para um sistema de detecção de fraude tem um sistema de análise de imagem ou um sistema de monitoramento de intrusão que deve ser detectado imediatamente (em tempo real)

e um alarme adequado deve soar para alertar o administrador do sistema sobre o problema. Uma vez que a situação foi tratada, esta leitura anômala pode ser armazenada separadamente para comparação com quaisquer novos casos de fraude, mas provavelmente não seria armazenada com os dados do sistema principal, pois essas técnicas tendem a modelar situações de normalidade e usar disso para detectar anomalias.

A detecção de anomalias é uma tarefa crucial em muitos ambientes críticos de segurança, visto que o *outlier* indica condições anormais de execução a partir das quais o sinal detectado pode resultar em uma aeronave com defeito de rotação do motor ou ainda, um problema de fluxo em uma tubulação.

Essa anomalia pode ainda, caracterizar um objeto anômalo em uma imagem, como uma mina terrestre, podendo ainda localizar um intruso dentro de um sistema com intenções maliciosas, exemplos que necessitam de uma detecção rápida. Em suma, a detecção de *outlier* pode detectar uma falha em uma linha de produção de uma fábrica monitorando constantemente as características específicas dos produtos e comparando os dados em tempo real com os recursos e padrões de produtos normais ou de defeitos (HODGE; AUSTIN, 2004).

1.2 Objetivos

Este trabalho tem como objetivo solucionar o problema em estudo, detecção de anomalia, utilizando da teoria de grafos (TORRES, 2012), (TORRES; CASTRO; BRAGA, 2011) para sua detecção em um conjunto de dados, modelando os dados em um grafo o qual seja possível utilizar das propriedades próprias deste para análise do mesmo, possibilitando a classificação das classes de dados sem um conhecimento prévio destas.

1.3 Metodologia

O objeto de pesquisa deste trabalho é desenvolver um método que baseia-se na teoria dos grafos para desenvolver um algoritmo capaz de realizar a detecção de anomalia(s) (HAWKINS, 1980) em conjuntos de dados, utilizando de operações e conceitos da área de grafos (A, 2007) e a aprendizagem de máquina para obter um classificador baseado em grafos capaz de fazer a distinção de classes.

Os passos para execução deste trabalho são assim definidos a seguir:

Figura 3 – Metodologia adotada para este trabalho.

Na Figura 3 temos uma ordem cronológica da disposição abordada para este trabalho. A revisão bibliográfica aborda a detecção de anomalia e os tipos de detecção. Na sequência temos uma relação dos modelos de detecção existentes: modelos estatísticos, redes neurais, aprendizagem de máquina, sistemas híbridos, classificação de classes, cumulando na contextualização da nossa métrica.

A seguir, discorremos a formação teórica que fundamenta nossa abordagem, tratando da definição de grafo, grau de um grafo e em particular do grafo de Gabriel. Um pouco a frente observa-se a metodologia desenvolvida que conta com o método proposto "Classificador Baseado em Grafos" (CBG) e a definição do classificador. Por fim, destacamos os experimentos realizados e a comparação dos resultados obtidos pelo CBG com os resultados provenientes da SVM *one-class* referenciada somente por "SVM" neste trabalho.

1.4 Organização do trabalho

O restante deste trabalho é organizado como se segue. O Capítulo 2 apresenta a revisão bibliográfica contextualizando o estado da arte na detecção de anomalias e conceitos da teoria dos grafos, utilizada em nossa abordagem, seguidos da metodologia proposta no Capítulo 3 e seus respectivos resultados no Capítulo 4. Por fim temos a conclusão e trabalhos futuros dispostos no Capítulo 5.

2 Revisão bibliográfica

Este capítulo trata da detecção de anomalia, que pode ser definida como a atividade de localizar comportamentos inesperados em um conjunto de dados, onde tal incidência diverge dos demais dados segundo alguma característica relevante. D. Hawkins (HAWKINS, 1980) define uma anomalia da seguinte forma: "A anomalia é uma observação que difere tanto de outras observações que suscita suspeita que tenha sido criada por um mecanismo diferente das outras observações".

Na Seção 2.1 contextualizamos a detecção de anomalia, especificando 3 tipos de detecção de *outliers* nas subseções 2.1.1, 2.1.2 e 2.1.3. Em seguida, na Seção 2.2 falamos um pouco sobre alguns modelos de detecção de anomalias, em particular, os modelos estatísticos, redes neurais, aprendizagem de máquina e sistemas híbridos, cada um com suas particularidades. Na Seção 2.3 tratamos da classificação de classes enquanto que a Seção 2.4 tratamos das definições de grafos, em particular, contextualizamos sobre o grau de um grafo, enquanto na Seção 2.5 discorremos sobre o Grafo de Gabriel, utilizado como base em nossa metodologia. Por fim, a Seção 2.6 trata da contextualização deste trabalho.

2.1 Detecção de anomalia

Uma anomalia, também pode ser chamada de *outlier*, evento, surto, alteração, fraude, exceções, aberrações, entre outras (HODGE; AUSTIN, 2004). Sendo este um problema que diversas áreas da computação se propõem a solucionar, utilizando diversas técnicas que se propõem à detecção de anomalias, sendo algumas delas: *Data mining, machine learning* e inteligência artificial; alguns trabalhos mais recentes também utilizam *Deep learning e Deep Anomaly Detection*, no sentido de realizar esta detecção (CHALAPATHY; CHAWLA, 2019).

Quanto à origem dos *outliers*, (HODGE; AUSTIN, 2004) ressalta que algumas ocorrências podem ser devido a erro humano, erro de instrumentos, desvio natural nas populações, comportamento fraudulento, mudanças no comportamento de sistemas ou falhas em sistemas em alguns casos, podendo ocorrer devido a outras circunstâncias ainda não conhecidas. O autor ainda separa em três tipos fundamentais para o problema de detecção de valores discrepantes, que serão descritas nas subseções a seguir.

2.1.1 Tipo 1 - Sem conhecimento prévio dos dados

Esta abordagem de aprendizagem é análoga ao agrupamento supervisionado, em que os *outliers* são determinados sem conhecimento prévio dos dados segundo (HODGE;

AUSTIN, 2004). O autor ainda destaca que a abordagem processa os dados como uma distribuição estática, aponta os pontos mais remotos e os sinaliza como potenciais *outliers*. O Tipo 1 assume que os erros ou falhas são avaliados separadamente a partir dos dados "normais" e, portanto, aparecerá como *outliers*.

Segundo (HODGE; AUSTIN, 2004), a abordagem é predominantemente retrospectiva e é análoga a um sistema de processamento em lote. Requer que todos os dados estejam disponíveis antes do processamento e que os dados sejam estáticos.

Contudo, uma vez que o sistema possui um banco de dados suficientemente grande com boa cobertura, então é possível comparar novos itens com os dados existentes. Existem duas sub técnicas comumente empregadas, diagnóstico e acomodação (ROUSSEEUW; LEROY, 2005).

(HODGE; AUSTIN, 2004) ainda desta que uma abordagem de um diagnóstico atípico destaca os potenciais pontos periféricos, uma vez detectado, o sistema pode remover esses pontos discrepantes do processamento futuro na distribuição de dados. O autor ainda ressalta que muitas abordagens de diagnóstico podem iterativamente podar os *outliers* e ajustar seu modelo de sistema aos dados restantes até que não sejam detectados mais *outliers*.

Uma metodologia alternativa é a acomodação que incorpora os *outliers* no modelo de distribuição gerado e emprega um método robusto de classificação, destaca (HODGE; AUSTIN, 2004), segundo o autor, essas abordagens robustas podem resistir a presença de *outliers* nos dados e geralmente induzem um limite em torno da maioria dos dados que, portanto, representam o comportamento normal.

Em contrapartida, métodos classificadores não robustos produzem representações que são distorcidas quando os *outliers* são deixados no conjunto de dados (HODGE; AUSTIN, 2004). O autor ainda ressalta que métodos não robustos são mais adequados quando há apenas alguns *outliers* no conjunto de dados, pois são computacionalmente mais baratos do que os métodos robustos, contudo um método robusto deve ser usado se houver um grande número de *outliers* para evitar essa distorção.

2.1.2 Tipo 2 - Aprendendo dados normais e anormais

(HODGE; AUSTIN, 2004) destaca que esta abordagem é análoga à classificação supervisionada e requer dados pré-etiquetados, marcados como normais ou anormais. Toda a área fora da classe normal representa a classe atípica, ressaltando que os pontos normais podem ser classificados como uma única classe ou subdivididos em três classes distintas de acordo com os requisitos do sistema para fornecer uma classificação normal/anormal simples ou para fornecer um classificador anormal e 3 classes de normalidade.

Classificadores são mais adequados para dados estáticos conforme as necessidades

de classificação a ser reconstruído a partir dos primeiros princípios, se a distribuição de dados mudar a menos que o sistema use um classificador incremental, como uma rede neural evolutiva, como descreve (MARSLAND, 2001).

A abordagem tipo 2 pode ser usada para classificação on-line, onde o classificador aprende o modelo de classificação e, em seguida, classifica novos exemplares como e quando necessário em relação ao modelo aprendido. Se o novo exemplar encontra-se em uma região de normalidade é classificado como normal, caso contrário é sinalizado como um *outlier*, (HODGE; AUSTIN, 2004).

(HODGE; AUSTIN, 2004) ainda desta que algoritmos de classificação requerem uma bom dimensionamento de dados normais e anormais, ou seja, os dados devem cobrir toda a distribuição para permitir a generalização pelo classificador, de modo que novos exemplares podem então ser classificados corretamente, como a classificação é limitada a uma distribuição 'conhecida' e um novo exemplar é derivado de uma região da distribuição não vista anteriormente, pode não ser classificada corretamente, a menos que os recursos de generalização do referido algoritmo de classificação sejam bons.

2.1.3 Tipo 3 - Aprendendo somente os dados normais

Este contexto é análogo a um reconhecimento semi-supervisionado ou tarefa de detecção e pode ser considerado semi-supervisionado porque a classe normal é ensinada, mas o algoritmo aprende a reconhecer anormalidades (FAWCETT; PROVOST, 1999), (JAPKOWICZ et al., 1995). A abordagem precisa de dados pré-classificados, mas apenas aprende dados marcado como normal.

Segundo (HODGE; AUSTIN, 2004), é adequado para dados estáticos ou dinâmicos, pois apenas aprende uma classe que fornece o modelo de normalidade e pode aprender o modelo de forma incremental conforme novos dados chegam, ajustando o modelo para melhorar o ajuste à medida que cada novo exemplar se torna disponível. Ele ainda destaca que um sistema tipo 3 reconhece um novo exemplar como normal se estiver dentro da fronteira, do contrário, reconhece o novo exemplar como estranho, ou seja uma anormalidade.

Esta fronteira ou limite, pode ser difícil quando um ponto está totalmente dentro ou totalmente fora do limite, ou suave se o limite é graduado, dependendo do algoritmo de detecção subjacente segundo (HODGE; AUSTIN, 2004). O autor ainda ressalta que um algoritmo de limite flexível pode estimar o grau de "*outlierness*" e isto requer que toda a gama de normalidade esteja disponível para treinamento de modo a permitir a generalização, e que no entanto, não requer dados anormais para treinamento, ao contrário do tipo 2.

Segundo (HODGE; AUSTIN, 2004) outro problema com o tipo 2 é que nem sempre

consegue lidar com *outliers* em regiões inesperados, por exemplo, na detecção de fraude, um novo método de fraude nunca encontrado anteriormente ou falha previamente invisível em uma máquina pode não ser tratada corretamente pelo classificador, a menos que a generalização seja muito boa.

Neste método, contando que a nova fraude esteja fora do limite da normalidade, então o sistema detecta corretamente a fraude diz (HODGE; AUSTIN, 2004), ressaltando ainda que se a normalidade mudar, então a classe normal modelada pelo sistema pode ser deslocada reaprendendo o modelo de dados ou mudando o modelo se a técnica de modelagem referida permitir, como redes neurais evolutivas.

2.2 Modelos de Detecção de Anomalias

Os métodos de detecção de anomalias são derivados de três campos da computação: estatísticas (com base na proximidade, paramétricas, não paramétricas e semi-paramétricas), redes neurais (supervisionadas e não supervisionado) e aprendizado de máquina que é o foco da nosso abordagem.

Segundo (HODGE; AUSTIN, 2004), os *outliers* são determinados a partir da "proximidade" de vetores usando alguma métrica de distância adequada. Diferentes abordagens funcionam melhor para diferentes tipos de dados, para diferentes números de vetores, para diferentes números de atributos, de acordo com a velocidade necessária e de acordo com a precisão exigida.

As duas considerações fundamentais ao selecionar uma metodologia apropriada para um sistemas de detecção de *outliers* segundo (HODGE; AUSTIN, 2004) são:

- Selecionar um algoritmo que pode modelar com precisão a distribuição de dados e destacar com precisão os pontos periféricos para um agrupamento, classificação ou técnica de tipo de reconhecimento. O algoritmo deve também ser escalonável para os conjuntos de dados a serem processados.

- Selecionar uma vizinhança de interesse adequada para um *outlier*. A seleção da vizinhança de interesse não é trivial. Vários algoritmos definem limites em torno da normalidade durante o processamento e induzem um limite autonomamente. No entanto, essas abordagens são frequentemente paramétricas que reforçam um modelo de distribuição específico ou requerem parâmetros especificados pelo usuário, como o número de *cluster*s. Outras técnicas, requerem parâmetros definidos pelo usuário para definir o tamanho ou densidade de vizinhanças para limite de *outlier*. A escolha da vizinhança, seja definida pelo usuário ou induzida autonomamente, deve ser aplicável a todas as densidades de distribuições prováveis de serem encontradas e podem incluir potencialmente aquelas com variações bruscas de densidade.

2.2.1 Modelos estatísticos

São geralmente adequados para conjuntos de dados quantitativos de valor real ou, pelo menos, distribuições de dados ordinais quantitativos onde os dados ordinais podem ser transformados em valores numéricos adequados para processamento estatístico (numérico). Isso limita sua aplicabilidade e aumenta o tempo de processamento se transformações de dados complexas são essencialmente necessárias antes do processamento.

2.2.1.1 Técnicas baseadas em proximidade

São simples de implementar e não priorizam suposições sobre o modelo de distribuição de dados. Eles são adequados para detecção de *outlier* do tipo 1 e tipo 2. No entanto, eles sofrem exposição a um potencial crescimento computacional, uma vez que se baseiam no cálculo das distâncias entre todos os registros. Existem vários variações do algoritmo k-nearest neighbors (k-NN), como foi expandido por Thomas Cover (COVER; HART, 1967). Algoritmo de k vizinhos mais próximos para detecção de *outlier*, no geral, calculam os vizinhos mais próximos de um registro usando uma métrica para cálculo de distância adequada, como distância euclidiana ou distância de Mahalanobis.

2.2.1.2 Métodos Paramétricos

Métodos paramétricos permitem que o modelo seja avaliado muito rapidamente para novas instâncias e são adequadas para grandes conjuntos de dados; o modelo cresce apenas com a complexidade do modelo, não com o tamanho dos dados, ressalta (HODGE; AUSTIN, 2004), no entanto, sua aplicabilidade se limita ao impor um modelo de distribuição pré-selecionado para se adequar aos dados. Se os dados são conhecidos e se enquadram nesse modelo de distribuição, então essas abordagens são altamente precisas, mas muitos conjuntos de dados não cabem em um modelo particular. Uma dessas abordagens é a estimativa de volume mínimo do elipsóide (MVE)(ROUSSEEUW; LEROY, 2005) que se encaixa no menor volume elipsóide permissível em torno da maioria dos modelos de distribuição de dados.

2.2.1.3 Métodos Não Paramétricos

As estatísticas não paramétricas baseiam-se em não ter distribuição ou em ter uma distribuição especificada, mas com os parâmetros da distribuição não especificados.

Dasgupta e Forrest (DASGUPTA; FORREST, 1996) apresentam uma abordagem não paramétrica para detecção de novidade na operação de máquinas. Os autores reconhecem a novidade que contrasta com as outras abordagens do tipo 3, descrita como k-means (NAIRAC et al., 1999).

2.2.1.4 Métodos Semi paramétricos

Métodos semi paramétricos aplicam modelos de kernel locais em vez de um único modelo de distribuição global. Segundo (HODGE; AUSTIN, 2004), eles visam combinar a vantagem da velocidade e do crescimento da complexidade dos métodos paramétricos com a flexibilidade do modelo de métodos não paramétricos. Métodos baseados em kernel estimam a densidade da distribuição do espaço de entrada e identificam *outliers* como situados em regiões de baixa densidade. Tarassenko e Roberts (ROBERTS; TARASSENKO, 1994) e Bishop (BISHOP, 1994) usam modelos de mistura gaussiana para aprender um modelo de dados normais por meio do aprendizado incremental de novos exemplares.

Ainda é importante destacar os chamadas Support Vector Machines SVMs, que segundo (HODGE; AUSTIN, 2004), podem induzir um classificador a partir de um conjunto de dados mal balanceado onde os exemplares anormais/normais são desproporcionais, o que segundo o autor, ocorre em domínios médicos onde dados anormais ou em alguns casos normais são difíceis de ocorrer e caro de se obter. No entanto, SVMs são computacionalmente complexos para determinar heurísticas, então para evitar isso, segundo (DECOSTE; LEVINE, 2000) adaptam o SVM convencional para detecção de de evento em instrumentos espaciais adaptando os pesos dos recursos e o custo de falsos positivos, já que seu conjunto de dados é predominantemente negativo com poucos exemplos positivos de um evento disponível para treinamento.

2.2.2 Redes Neurais

As abordagens de rede neural são geralmente não paramétricas e baseadas em modelos, elas generalizam bem para padrões invisíveis e são capazes de aprender limites de classe complexos. Após o treinamento, a rede neural forma um classificador. No entanto, todo o conjunto de dados deve ser percorrido várias vezes para permitir que a rede estabeleça e modele os dados corretamente. Eles também exigem treinamento e testes para ajustar a rede e determinar as configurações de limite antes de estarem prontos para a classificação de novos dados.

2.2.2.1 Métodos Neurais Supervisionados

As redes neurais supervisionadas usam a classificação dos dados para conduzir o processo de aprendizagem. A rede neural usa a classe para ajustar os pesos e limites para garantir que a rede possa classificar corretamente a entrada. Os dados de entrada são efetivamente modelados por toda a rede com cada ponto distribuído por todos os nós e a saída que representa a classificação. (HODGE; AUSTIN, 2004) ainda ressalta que algumas redes neurais supervisionadas, como o *Multi-Layer Perceptron*, interpolam bem, mas têm um desempenho ruim para extrapolação, portanto, não pode classificar instâncias

invisíveis fora dos limites do conjunto de treinamento. Nairac (NAIRAC et al., 1999) e Bishop (BISHOP, 1994) exploram isso para identificar *outlier*.

2.2.2.2 Métodos Neurais Não Supervisionados

Redes neurais não supervisionadas contém nós que competem para representar partes do conjunto de dados. Tal como acontece com redes neurais baseadas em Perceptron, árvores de decisão ou k-means, eles exigem dados de treinamento definidos para permitir que a rede aprenda.

Self Organizing Maps, são uma abordagem segundo (HULLE, 2012) e (RITTER et al., 1992) bem competitivos em redes neurais não supervisionadas. SOMs realizam quantização vetorial e mapeamento não linear para projetar a distribuição de dados em uma grade de rede dimensional cuja topologia precisa ser pré-especificada pelo usuário. Cada nó na grade tem um vetor de peso associado análogo ao vetor médio que representa cada *cluster* em um sistema *k-means*. A rede aprende lendo iterativamente cada entrada de treinamento do conjunto de dados, encontrando a melhor unidade correspondente, atualizando o peso do vetor vencedor para refletir a nova correspondência com k-means.

2.2.3 Aprendizagem de máquina

Muito da detecção de *outliers* focou-se apenas em atributos de dados contínuos de valor real; houve pouco foco em dados categóricos. A maioria das abordagens estatísticas e neural requerem dados cardinal ou pelo menos ordinal para permitir que distâncias vetoriais sejam calculadas e não têm mecanismo para processar dados categóricos sem ordenação implícita. John (JOHN, 1995) e (SKALAK; RISSLAND, 1990) usam uma árvore de decisão C4.5 para detectar *outliers* em dados categóricos e, assim, identificar erros e entradas inesperadas em bancos de dados.

2.2.4 Sistemas híbridos

O desenvolvimento mais recente em tecnologia de detecção de *outlier* são os sistemas híbridos. Em suma, os sistemas híbridos aqui referidos, incorporam algoritmos de pelo menos duas das categorias citadas anteriormente (estatística, métodos neurais ou de aprendizado de máquina).

Como um bom exemplo, temos (BRODLEY; FRIEDL et al., 1996) que usam uma abordagem em conjunto para identificar instâncias classificadas incorretamente em um conjunto de pixels de imagens de treinamento de satélite de terras, onde cada pixel precisa ser classificado como por exemplo, pastagem, floresta etc. Segundo o autor, eles usam três classificadores: a árvore de decisão, um vizinho mais próximo e aprendizagem de máquina.

2.3 Classificação de Classes

No aprendizado de máquina, uma abordagem para lidar com o problema de detecção de anomalias é a classificação de uma classe, como o Support vector machine (SVM) - uma das técnicas similares ao trabalho proposto. A classificação de uma classe, envolve ajustar um modelo nos dados "normais" e prever se os novos dados são normais ou um *outlier*/anomalia.

Segundo (FERNÁNDEZ et al., 2018), um classificador de uma classe visa capturar características das instâncias de treinamento, a fim de poder distingui-las de potenciais *outliers* que apareçam.

Técnicas de classificação de uma classe podem ser usadas para problemas de classificação desequilibrada binária (duas classes) onde o caso negativo (classe 0) é considerado "normal" e o caso positivo (classe 1) é considerado um *outlier* ou anomalia.

- Caso negativo : Normal ou interno.
- Caso positivo : Anomalia ou *outlier*.

Dada a natureza da abordagem, segundo (FERNÁNDEZ et al., 2018), as classificações de uma classe são mais adequadas para as tarefas em que os casos positivos não têm um padrão ou estrutura consistente no espaço de recursos, tornando difícil para outros algoritmos de classificação aprenderem um limite de classe. Em vez disso, tratando os casos positivos como *outliers*, permite que os classificadores de uma classe ignorem a tarefa de discriminação e, em vez disso, foquem nos desvios do normal ou do que é esperado.

Assim sendo, (FERNÁNDEZ et al., 2018) ainda ressalta, é preciso lembrar que as vantagens dos classificadores de uma classe têm o preço de descartar todas as informações disponíveis sobre a classe majoritária. Portanto, esta solução deve ser usada com cuidado e pode não se adequar a algumas aplicações específicas.

O método usando SVM para detecção de novidades proposto por (SCHÖLKOPF et al., 1999) ao supor que se receba algum conjunto de dados retirado de uma distribuição de probabilidade subjacente P e desejando estimar um subconjunto "simples" S do espaço de entrada de modo que a probabilidade de que um ponto de teste extraído de P esteja fora de S seja igual a algum a princípio especificado entre 0 e 1, propõe deste modo, um método para abordar este problema tentando estimar uma função F que é positiva em S e negativa no complemento.

Deste modo, o autor ainda ressalta que a forma funcional de F é fornecida por uma expansão do kernel em termos de um subconjunto potencialmente pequeno dos dados de treinamento; sendo ele regularizado controlando o comprimento do vetor de peso em um

espaço de recurso associado. Na prática, o algoritmo é uma extensão natural do algoritmo de vetor de suporte para o caso de não rotulado.

Em suma, o método proposto por (SCHÖLKOPF et al., 1999) separa todos os pontos de dados da origem (no espaço de recursos F) e maximiza a distância deste hiperplano à origem. O que resulta em uma função binária que captura regiões no espaço de entrada onde reside a densidade de probabilidade dos dados. Deste modo, a função retorna +1 em uma região "pequena" (capturando os pontos de dados de treinamento) e -1 em outro lugar.

Sendo assim, outro tipo de problema também é de interesse: o problema da descrição de dados ou Classificação de Uma Classe (MOYA; KOCH; HOSTETLER, 1993). Em sua proposta, (TAX; DUIN, 2004), o problema é fazer uma descrição de um conjunto de objetos de treinamento e detectar quais (novos) objetos se assemelham a esse conjunto de treinamento.

Logo, (TAX; DUIN, 2004) utiliza a Support Vector Data Description (SVDD) que é inspirada no Support Vector Classifier. Sendo o limite em formato esférico em torno de um conjunto de dados e, análogo ao Classificador de vetores de suporte, pode ser flexível usando outras funções do kernel. O método é robusto contra *outliers* no conjunto de treinamento e é capaz de restringir a descrição usando exemplos negativos, onde mostra as características das Descrições de Dados do Vetor de Suporte usando dados artificiais e reais.

Graças ao conceito de reprodução de kernels (ARONSZAJN, 1950), uma função kernel (semidefinida positiva) $k(\cdot, \cdot)$ define uma transformação não linear $\Phi(\cdot)$ do espaço de entrada em algum recurso do espaço. Uma esfera definida neste último corresponde (é pré-imaginada (HONEINE; RICHARD, 2011)) a uma característica não linear no espaço de entrada.

Acontece que apenas o produto interno é frequentemente necessário, o que pode ser avaliado usando uma função de kernel, $< \Phi(xi), \Phi(xj) >= k(xi, xj)$ para qualquer xi, xj do espaço de entrada X. O SVM de uma classe foi inicialmente derivado em (SCHÖLKOPF et al., 2001) para uma estimativa do suporte de uma distribuição com o v-SVM, para detecção de novidades com a chamada "descrição de dados vetoriais de suporte" A ideia principal é encontrar uma esfera, de volume mínimo, contendo todas as amostras de treinamento. Esta esfera, descrita por seu centro c e seu raio r, é obtida resolvendo o problema de otimização restrita

$\min_{r,c} \quad r^2$

sujeito a $\quad || \Phi(xi) - c ||^2 \leq r^2 para\ i = 1, 2, ..., n$

Embora a restrição acima possa ser muito restritiva, (NOUMIR; HONEINE; RICHARD, 2012) ainda ressalta que pode-se tolerar uma pequena fração das amostras esteja

fora da esfera. Isso resulta em robustez, no sentido de que é menos sensível à presença de *outliers* no conjunto de dados de treinamento. Segundo o autor, para este propósito, seja v um parâmetro positivo que especifica a compensação entre o volume da esfera e o número de *outliers*. Então. segundo o autor, o problema se torna a estimativa de c, r e um conjunto de variáveis de folga não negativas $\zeta_1, \zeta_2, ..., \zeta_n$:

$\min_{r,c,\zeta} \quad r^2 + \frac{1}{vn} \sum_{i=1}^{n} \zeta_i$

Ao introduzir a otimização de Karush-Kuhn-Tucker (KKT) condições, nós temos

$$c = \sum_{i=1}^{n} \alpha_i \Phi(xi), \tag{2.1}$$

onde os α_i's são a solução para o problema de otimização:

$\max_{\alpha} \sum_{i=1}^{n} \alpha_i k(xi, xj) - \sum_{i,j=1}^{n} \alpha_i \alpha_j k(xi, xj)$

sujeito a

$$\sum_{i=1}^{n} \alpha_i = 1 \, e \, 0 \leq \alpha_i \leq \frac{1}{vn} \; \forall \, i = 1, 2, ..., n. \tag{2.2}$$

De acordo com as condições KKT, cada amostra xi pode ser classificado em três categorias: $\alpha_i = 0$ corresponde a uma amostra situada dentro da esfera, amostras com $0 < \alpha_i \leq \frac{1}{vn}$ encontram-se no limite da esfera, e as amostras com $\alpha_i = \frac{1}{vn}$ ficam fora da esfera, ou seja, são *outliers*. As amostras com α_i diferente de zero são chamadas de vetores de suporte (SVs), pois são suficientes para descrever o centro. Na prática, apenas uma pequena fração dos dados são SV. Seja $ł_{sv}$ o conjunto de índices associados a SV, nomeadamente

$$\begin{cases} \alpha_i \neq 0 \; se \; i \; \epsilon \; ł_{sv}; \\ \alpha_i = 0 \; caso \; contrário. \end{cases}$$

Finalmente, o raio ideal é obtido a partir de qualquer SV situado na fronteira, ou seja, qualquer xi com $0 < \alpha_i < \frac{1}{vn}$ já que neste caso $|| \; \Phi(xi) - c \; ||= r$. Isso é equivalente a

$$r = \min_{i \epsilon ł_{sv}} \left\| \Phi(xi) - c) \right\|.$$

Portanto, a regra de decisão de que qualquer nova amostra X não é um *outlier* é dada como $\|\Phi(x) - c)\| < r$, onde a distância é calculada usando

$$\|\Phi(x) - c)\|^2 = \sum_{i,j \epsilon ł_{sv}} \alpha_i \alpha_j k(xi, xj) - 2 \sum_{i \epsilon ł_{sv}} \alpha_i k(xi, x) + k(x, x). \tag{2.3}$$

Segundo (MOYA; HUSH, 1996), para funcionar como um classificador de uma classe, um classificador deve exibir três tipos de capacidade de generalização. Se o classificador pode reconhecer novos padrões de destino, ele exibe generalização dentro da classe. Se o classificador puder distinguir entre novos padrões de duas classes de destino distintas, ele exibirá uma generalização entre as classes. Se o classificador puder distinguir padrões alvo de padrões não alvo, então, ele exibe generalização fora da classe. Apenas os classificadores que exibem todos os três tipos de generalização podem funcionar como classificadores de uma classe.

A capacidade de generalização de um classificador e as características de limite de decisão agrupam-no em uma de duas categorias: discriminadores ou detectores (HUSH; MOYA; CLARK, 1992). Um discriminador pode ter limites de decisão abertos. Ele classifica cada padrão de teste, mesmo aqueles muito diferentes dos exemplos de treinamento, em uma das classes predefinidas. Ele pode fornecer generalização dentro da classe e generalização entre as classes, mas não generalização fora da classe. O discriminador mais familiar é o discriminador Bayes (DUDA; HART et al., 1973), que é projetado para fornecer generalização ideal entre classes e dentro das classes.

Em contraste, um detector decide se o padrão pertence a qualquer uma das classes de destino e inclui a opção de decidir que o padrão não pertence a nenhuma das classes de destino. Um detector tem limites de decisão que circundam completamente as classes de destino e que fornecem generalização fora da classe, bem como generalização dentro e entre classes.

Em um classificador treinado, a qualidade da generalização dentro e entre as classes depende do número de padrões de treinamento e do número de graus de liberdade no classificador (HUSH; MOYA; CLARK, 1992). Com muitos graus de liberdade, um classificador pode ajustar demais os dados de treinamento, o que contribui para uma generalização pobre dentro e entre as classes. Como as redes neurais incorporam funções não lineares, o número de graus de liberdade em uma rede neural depende da ordem da entrada e do número de nós na rede.

Maus resultados de generalização de fora da classe quando limites classificador de grandes dimensões incorporam padrões não-alvo. Para uma boa generalização fora da classe, os limites de decisão devem envolver os padrões de destino de maneira compacta. Segundo (HUSH; MOYA; CLARK, 1992) uma vez que o reconhecimento de padrões com características ausentes está além do escopo de sua abordagem, o que exige que os limites não apenas circundam, mas também se fechem completamente em torno dos padrões de treinamento.

Deste modo, existem duas abordagens possíveis para melhorar a capacidade de generalização fora da classe de um classificador. A primeira abordagem inclui no conjunto de treinamento padrões próximos ao alvo, que são não alvos próximos aos limites da classe

alvo. No entanto, o sucesso dessa abordagem depende da capacidade do projetista de identificar um conjunto completo de alvos próximos que circundam completamente a classe alvo.

O treinamento iterativo (MOYA; HOSTETLER, 1989) pode identificar padrões próximos ao alvo que melhoram a generalização fora da classe para não alvos selecionados, mas não fornece generalização fora da classe em resposta a entradas arbitrárias. Se um projetista tem um conjunto completo de alvos próximos, ele pode treinar várias redes neurais para funcionarem como classificadores de uma classe.

A abordagem alternativa para melhorar a capacidade de generalização de uma classe, segundo (MOYA; HUSH, 1996), incorpora um critério adicional no algoritmo de aprendizagem que busca minimizar o tamanho dos limites de decisão. Essa abordagem incentiva a rede a encontrar os menores limites que também produzem pequenos erros de mapeamento de classificação. Em seu artigo, (MOYA; HUSH, 1996) apresenta um classificador de rede neural restrita que incorpora um critério de tamanho de limite mínimo, bem como um critério de erro mínimo.

Mais detalhes sobre SVMs podem ser encontrados em (NOUMIR; HONEINE; RICHARD, 2012).

2.4 Teoria de grafos

Esta seção trata dos conceitos teóricos de grafos amplamente utilizados para estudar e modelar diversas aplicações, em diferentes áreas. Eles incluem desde estudo de moléculas, construção de ligações em química e o estudo de átomos à medição do prestígio dos atores ou ainda, para explorar os mecanismos de difusão.

Em biologia por exemplo, a teoria dos grafos é utilizada de modo que os esforços de conservação onde um vértice representa regiões onde certas espécies existem e as bordas representam o caminho de migração ou movimento entre as regiões. (RIAZ; ALI, 2011) ressalta que esta informação é importante ao observar os padrões de reprodução ou rastrear a propagação de doenças, parasitas e para estudar o impacto da migração que afeta outras espécies.

Alguns conceitos teóricos de grafos são amplamente utilizados em Pesquisa Operacional. Sendo alguns deles, o problema do caixeiro viajante, a árvore de abrangência mais curta em um grafo ponderado, obtendo uma combinação ótima de empregos e homens e localizando o caminho mais curto entre dois vértices em um grafo. Também é usado na modelagem de redes de transporte, redes de atividades e teoria dos jogos. Onde a atividade de rede é usada para resolver um grande número de problemas combinatórios.

2.4.1 Definição de grafo

Informalmente, um grafo é um monte de pontos com linhas conectando alguns deles. Para muitos propósitos matemáticos, segundo (LEHMAN; LEIGHTON; MEYER, 2010), realmente não importam como os pontos e linhas são dispostos, somente quais pontos são conectados por linhas. O autor ainda ressalta que a definição de grafos simples visa capturar apenas esses dados de conexão. De modo que um grafo simples, G, consiste em um conjunto não vazio, V, chamado de vértices de G, e uma coleção, E, de subconjuntos de dois elementos de V. Os membros de E são chamados de arestas de G. Para exemplificar, um grafo também pode ser representado listando os vértices e arestas de acordo com a definição oficial de grafo simples:

- V = {A, B, C, D, E, F, G, H, I}
- E = {{A, B} , {A, C} , {B, D} , {C, D} , {C, E} , {E, F} , {E, G} , {H, I}} .

Os vértices correspondem aos pontos e as arestas correspondem às linhas, como se pode observar em um outro exemplo na Figura 4.

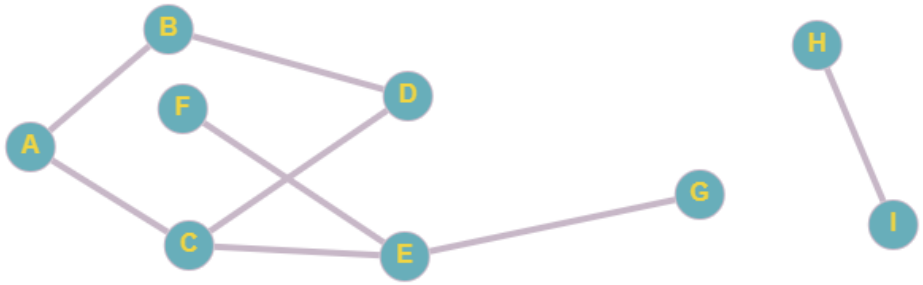

Figura 4 – Representação do grafo segundo (LEHMAN; LEIGHTON; MEYER, 2010).

Desta forma, os vértices correspondem aos pontos e as arestas correspondem às linhas. Será útil usar a notação A—B para a aresta {A, B}. Observe que A—B e B—A são descrições diferentes da mesma aresta, uma vez que os conjuntos não são ordenados. Assim, a definição de grafos simples é a mesma que para grafos direcionados, exceto que em vez de uma aresta direcionada v—>w que começa no vértice v e termina no vértice w, um grafo simples tem apenas uma aresta não direcionada, v—w, que se conecta v e w.

2.4.2 Grau de um grafo

Quanto ao grau de um grafo, (LEHMAN; LEIGHTON; MEYER, 2010) faz as seguintes considerações:

Dois vértices em um grafo simples são considerados adjacentes se forem unidos por uma aresta, e uma aresta é considerada incidente aos vértices que ela une. O número de

arestas incidentes em um vértice é denominado grau do vértice; equivalentemente, o grau de um vértice é igual ao número de vértices adjacentes a ele.

Por exemplo, no grafo simples listado anteriormente, A é adjacente a B e B é adjacente a D, e a aresta A—C incide nos vértices A e C. O vértice H tem grau 1, D tem grau 2 e E tem grau 3, como podemos observar na Figura 5.

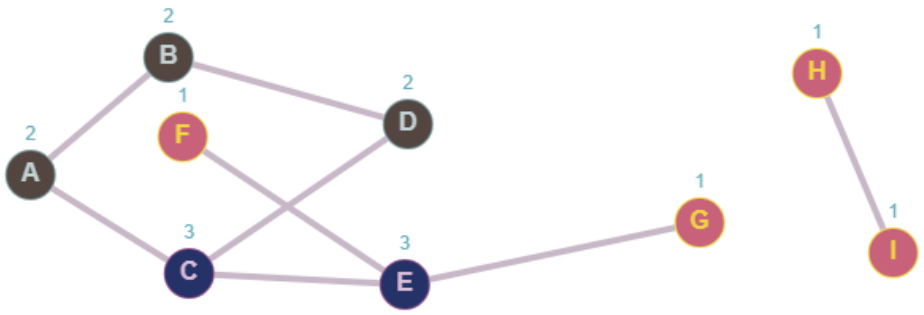

Figura 5 – Representação do grau do grafo segundo (LEHMAN; LEIGHTON; MEYER, 2010).

Um sinônimo para “vértices” é “nós", note que essas palavras podem ser usadas de forma intercambiável. O autor ainda enumera algumas consequências técnicas da definição que devem ser observadas desde o início:

1. Grafos simples não têm *loops* ($\{a, a\}$ não é uma aresta não direcionada porque uma aresta não direcionada é definida como um conjunto de dois vértices).

2. Existe no máximo uma aresta entre dois vértices de um grafo simples.

3. Grafos simples têm pelo menos um vértice, embora possam não ter arestas.

2.5 Grafo de Gabriel

Em geometria computacional, o grafo de Gabriel é um grafo que expressa uma ideia de proximidade de um conjunto $\mathcal{S}$ de pontos do plano euclidiano, seu nome é devido ao matemático K. Ruben Gabriel, que os apresentou em um artigo de Robert Sokal (GABRIEL; SOKAL, 1969), (AURENHAMMER; KLEIN, 2000) em 1969.

O grafo de Gabriel $\ddot{G}$ é um subconjunto de pontos do Diagrama de Voronoi e também um subgrafo da Triangulação de Delaunay (ZHANG; KING, 2002), ou seja, $\ddot{G}$ está contido em G. Segundo (BERG et al., 2008), o grafo de Gabriel $\ddot{G}$ de um conjunto de pontos $\mathcal{S}$ é um grafo cujo conjunto de vértices $\mathcal{V} = \mathcal{S}$ e seu conjunto de arestas $\mathcal{E}$ deve obedecer à seguinte definição:

$$(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{E} \leftrightarrow \delta^2(\mathbf{x}_i, \mathbf{x}_j) \leq \left[\delta^2(\mathbf{v}_i, \mathbf{z}) + \delta^2(\mathbf{v}_j, \mathbf{z})\right] \forall\, \mathbf{z} \in \mathcal{V}, (\mathbf{v}_i, \mathbf{v}_j) \neq z, \tag{2.4}$$

O que implica que, para $(\mathbf{v}_i, \mathbf{v}_j)$ constituir uma aresta de $\ddot{G}$, não pode haver nenhum outro vértice dentro da hiperesfera cujo o diâmetro é a distância euclidiana entre $\mathbf{v}_i$ e $\mathbf{v}_j$. Deste modo, podemos fazer as seguintes considerações, sendo $\ddot{G}$ um grafo de Gabriel:

- $\ddot{G}$ é um grafo plano , ou seja, pode ser desenhado no plano sem cruzamento de arestas.
- $\ddot{G}$ é um subgrafo da triangulação de Delaunay .
- $\ddot{G}$ pode ser calculado em tempo linear a partir da triangulação de Delaunay segundo (MATULA; SOKAL, 1980)
- $\ddot{G}$ contém como subgrafos a árvore de cobertura mínima , o grafo de vizinhança relativa e o grafo de vizinho mais próximo .
- É um caso de esqueleto beta . Igual aos beta-esqueletos, e ao contrário das triangulações de Delaunay, não é um revestimento geométrico, pois existem conjuntos de pontos cujas distâncias medidas em $\ddot{G}$ podem ser muito maiores que as distâncias euclidianas entre os pontos segundo (BOSE et al., 2006).
- Há um limite de percolação para $\ddot{G}$ de conjuntos de pontos finitos segundo (BERTIN; BILLIOT; DROUILHET, 2002) e (NORRENBROCK, 2016).

As Figuras 6(a) a 6(h) mostram a construção do grafo de Gabriel de forma detalhada. É possível observar na Figura 6(f) que a escolha dos dois vértices em questão (círculo pontilhado) não satisfazem Equação 2.4. Logo, eles não possuem uma aresta.

A respeito da ordem de complexidade, é sabido que o algoritmo intuitivo de construção do grafo de Gabriel tem complexidade O(n3) (ZHANG; KING, 2002). Entretanto, se o grafo de Gabriel for construído utilizando a estrutura da triangulação de Delaunay, cuja complexidade dado o pior caso é O(n log n) a ordem de complexidade de construção do grafo se torna O(n) (TOUSSAINT, 1980).

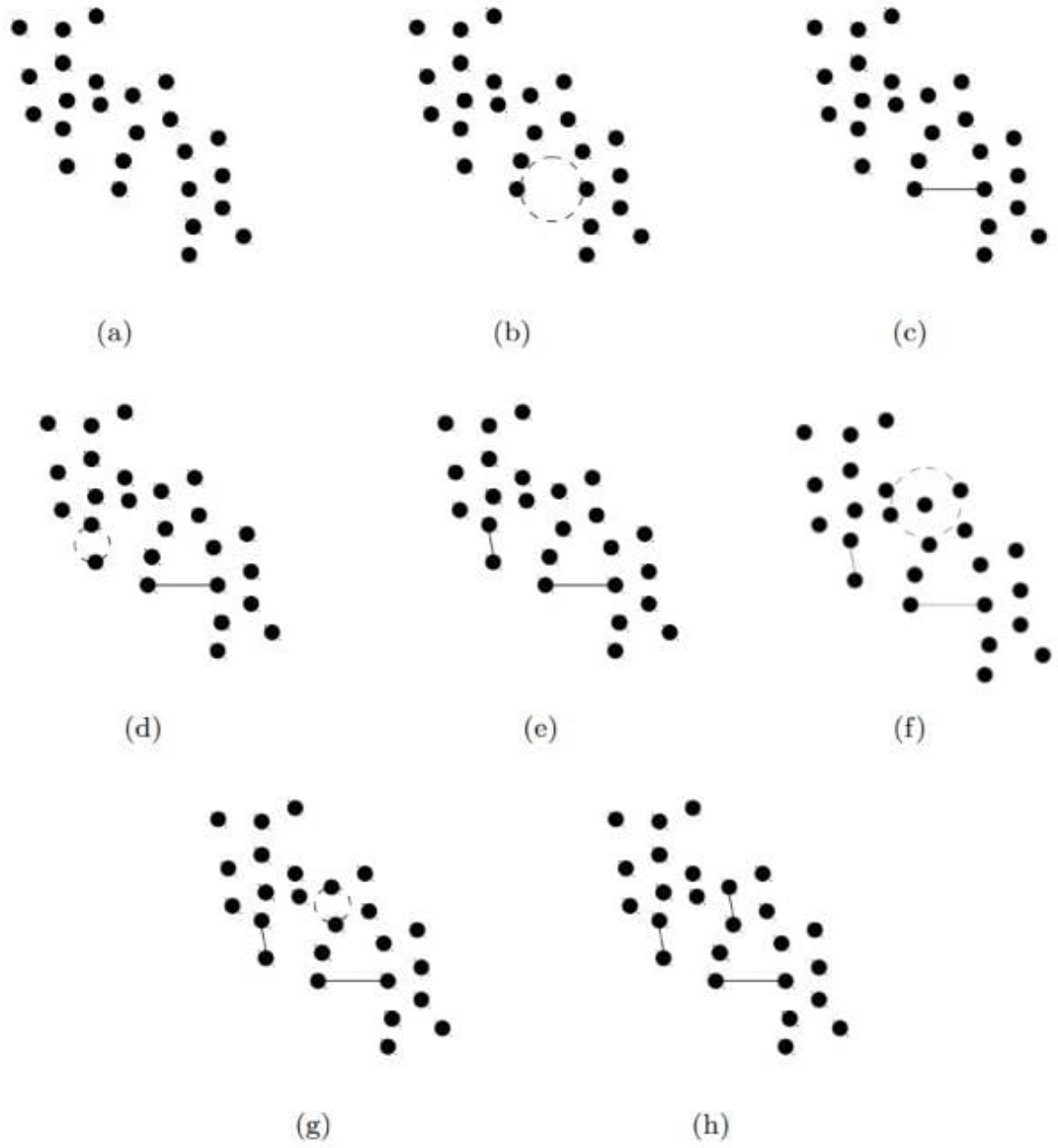

Figura 6 – Construção do Grafo de Gabriel.

Podemos observar um exemplo do Grafo de Gabriel representado na Figura 8 a partir do conjunto de pontos da Figura 7:

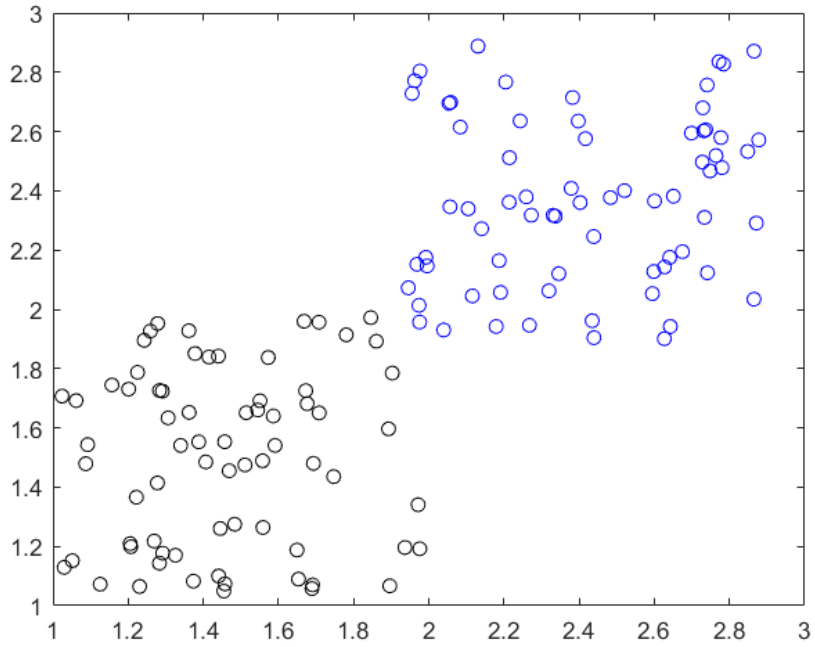

Figura 7 – Representação dos pontos para gerar o Grafo Gabriel.

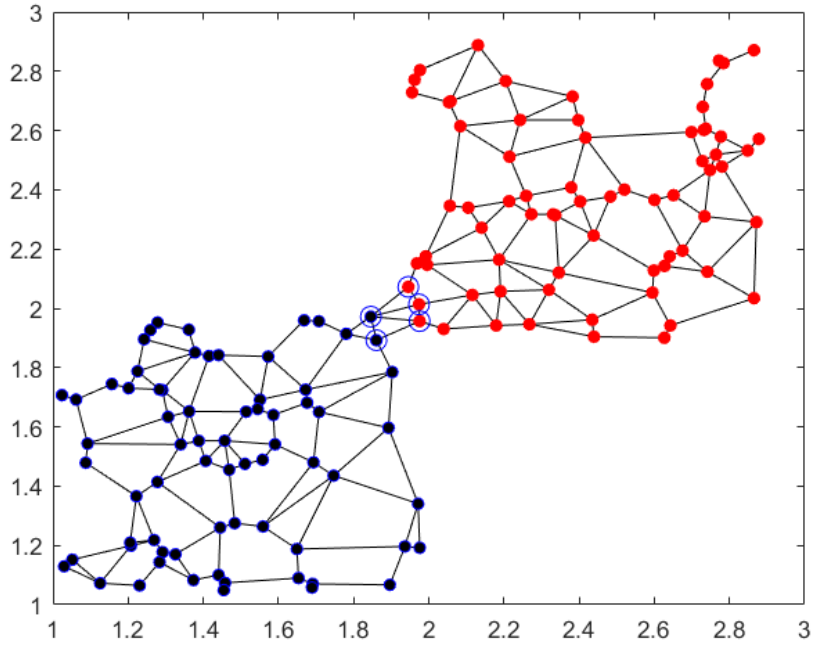

Figura 8 – Representação do Grafo de Gabriel.

2.6 Contextualização do trabalho

Os estudos realizados nestas áreas, em suma, se baseiam em explorar conjuntos de dados e/ou treinamento de algoritmos, de modo a separar os dados normais de anomalias. De modo que a técnica a ser utilizada varia conforme o tipo de problema e também da

natureza dos dados a serem analisados.

Nas seções do Capítulo 2, destacamos três campos que compreendem uma coleção de técnicas que vão desde modelos individuais até sistemas híbridos que utilizam algoritmos de vários campos. A abordagem proposta aqui se estende em específico, a uma análise geométrica do conjunto de dados como pontos do plano euclidiano, com base em modelos de algoritmos geométricos numa abordagem direcionada na teoria dos grafos para detecção de *outliers*.

Numa visão mais detalhada, é importante destacarmos que os métodos baseados em geometria computacional (SALGADO; TORRES; BRAGA,), buscam estudar problemas geométricos sob o ponto de vista algorítmico. Onde os problemas, por sua vez, são tratados em termos de objetos elementares, por exemplo: pontos, retas, segmentos de reta, polígonos, entre outros. Tendo como intuito resolver os problemas geométricos de forma eficiente, utilizando-se da menor quantidade de operações simples sobre os elementos de forma a resolver o problema.

Em suma, esta abordagem se baseia na estimativa geométrica na condição de fronteira entre os conjuntos de dados, mensurando as classes de dados, tendo em mente obter um separador para a(s) classe(s) ao analisar as condições de fronteira entre o(s) conjunto(s) de dados (que podem conter anomalias ou não), dando sequência da modelagem dos dados de entrada como um grafo (TORRES, 2012), (TORRES; CASTRO; BRAGA, 2011), ao qual é possível aplicar as métricas e conceitos já existentes, definidos pela teoria de grafos (A, 2007), para solução do problema nesta abordagem propriamente dita, em específico, direcionada ao grafo de Gabriel (TORRES, 2016).

3 Metodologia

Este capítulo descreve a abordagem proposta para este trabalho, em que utilizamos uma representação baseada em grafo para construção de um classificador, denominada (CBG). O método proposto foi implementado usando a ferramenta MATLAB. A formulação do método proposto é apresentado na Seção 3.1. Já a Seção 3.2 define o classificador com base no Algoritmo de Classificação - CHIP-clas Reduzido.

Sendo assim, na Figura 9 podemos observar o conjunto de dados, que na sequência, deram origem ao grafo contendo uma classe com pontos em azul e outra classe com um ponto em vermelho, que seria uma nova classe, ou ainda, ser denotado como uma anomalia ou *outlier* no conjunto da classe em azul como se observa na Figura 10.

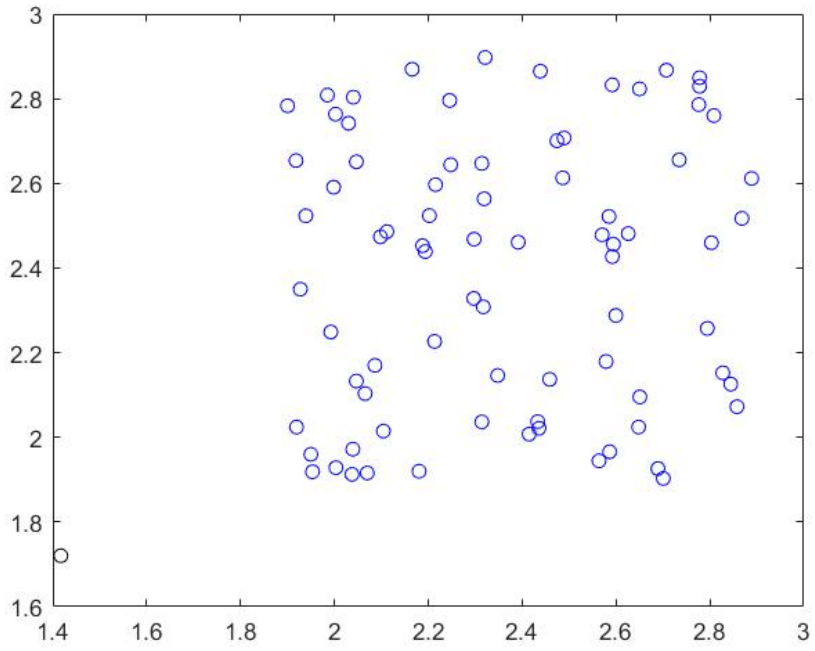

Figura 9 – Exemplos de modelos segundo (TORRES; CASTRO; BRAGA, 2011)

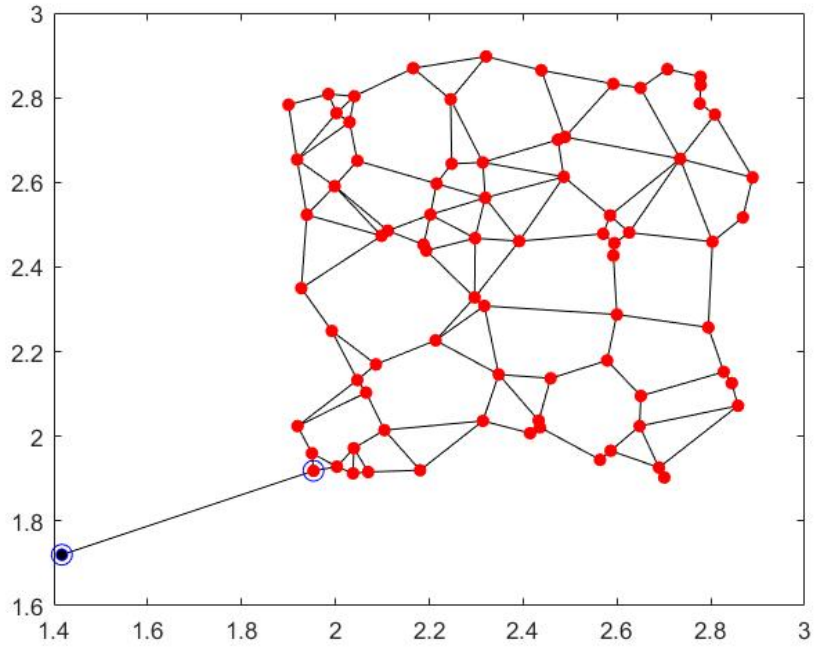

Figura 10 – Exemplos de grafo gerado a partir do modelo segundo (TORRES; CASTRO; BRAGA, 2011)

Como efeito, podemos destacar que até gerar a saída do classificador, passamos por algumas etapas que contribuem para a saída do CBG, podemos enumerá-las na seguinte ordem:

1 - Gerar/carregar a base de dados.

2 - Obter grafo a partir da base gerada/carregada.

3 - Pegar o valor médio de distâncias das arestas e marcar as arestas de suporte.

4 - Determinar borda da região pelas arestas de suporte.

5 - Definir ponto médio na aresta de suporte.

6 - Traçar hiperplano no ponto médio.

7 - Gerar separador com base no hiperplano definido.

3.1 Método proposto: Classificação Baseada em Grafos (CBG)

Seja o grafo $\ddot{G}$ o grafo de Gabriel, do conjunto de amostras $\underset{(dxN)}{X}$, onde d define a dimensão e N o número de amostras respectivamente, para X. De modo que cada linha uma dimensão e cada coluna uma amostra.

Sendo $\underset{(dxN)}{E}$ a matriz de arestas de $\ddot{G}$, temos que:

$$E(\mathbf{x}_i, \mathbf{x}_j) = 1, \forall (\mathbf{x}_i, \mathbf{x}_j) \in \ddot{G}, \tag{3.1}$$

Onde E, é a matriz de adjacência das arestas de $\ddot{G}$, de modo que se a posição $(\mathbf{x}_i, \mathbf{x}_j) = 1$ se existe uma aresta entre $(\mathbf{x}_i, \mathbf{x}_j)$, de modo que $\mathbf{x}_i$ e $\mathbf{x}_j$ são vértices do grafo $\ddot{G}$.

Sendo $\underset{(NxN)}{D_X}$ é a matriz que contém a relação de distância entre todas as amostras, na prática, a distância de cada amostra para todas as demais.

Desta forma, temos que a matriz de distâncias D_G é a multiplicação ponto a ponto que obtém a distância somente dos vértices do grafo $\ddot{G}$. Funcionando como um filtro obtido da matriz D_X, somente as distâncias que existem em E, as demais são zeradas. Sendo assim, D_G é calculada como:

$$D_G = E * D_X, \tag{3.2}$$

Sendo

$$D_L = D_G(\mathbf{x}_i, \mathbf{x}_j), \forall\, E(\mathbf{x}_i, \mathbf{x}_j) \neq 0, \tag{3.3}$$

Onde N_L é o tamanho de D_L. Deste modo, podemos fazer:

$$\beta = \frac{\sum D_L}{N_L}, \tag{3.4}$$

Sendo assim, podemos definir um limiar L, tal que:

$$L = \beta + (3 * \delta(D_2)), \tag{3.5}$$

onde,

$$\delta(D_L) = \sqrt{\frac{\sum (\mathbf{x}_i - D_L(\mathbf{i}))^2}{N_L}}, \forall\, X_{\mathbf{i}} \epsilon D_L, \tag{3.6}$$

Definimos neste momento as arestas de suporte A_S, de modo que:

$$A_S = D_G(\mathbf{i}, \mathbf{j}) \geq L, \forall\, (\mathbf{i}, \mathbf{j}), \tag{3.7}$$

Para ilustrar nossa abordagem (CBG), temos algumas imagens que contextualizam as formulações propostas pelo algoritmo, executadas na base gaussiana. Na Figura 11, é gerado um conjunto de amostras utilizando a base gaussiana. Em seguida na Figura 12, é gerado um grafo em cima dos pontos da amostra.

Na sequência, após pegar o valor médio das distâncias das arestas do grafo gerado, é calculado o limiar a partir das médias das arestas, para por fim determinar as arestas de suporte, o que pode ser observado na Figura 13.

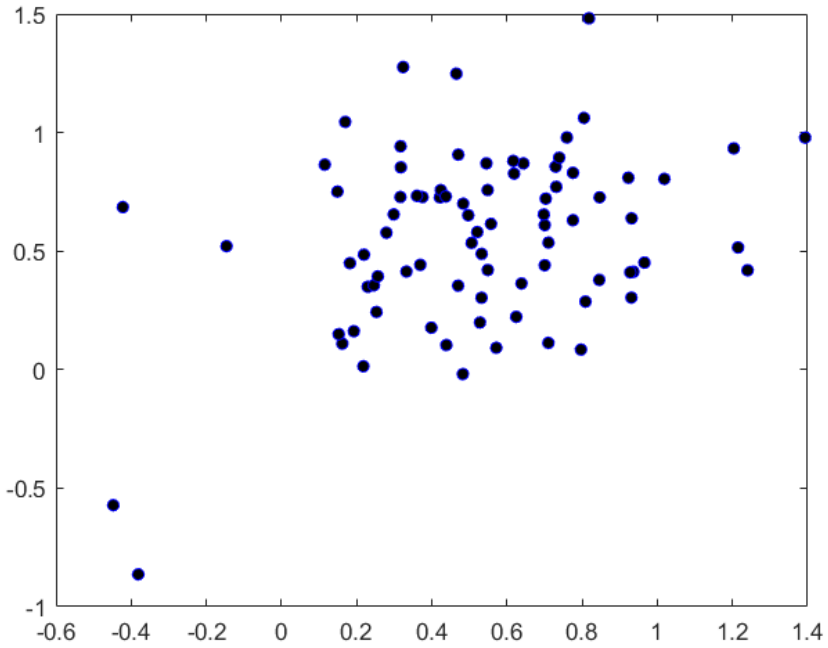

Figura 11 – CBG - Construção da Base Gaussiana.

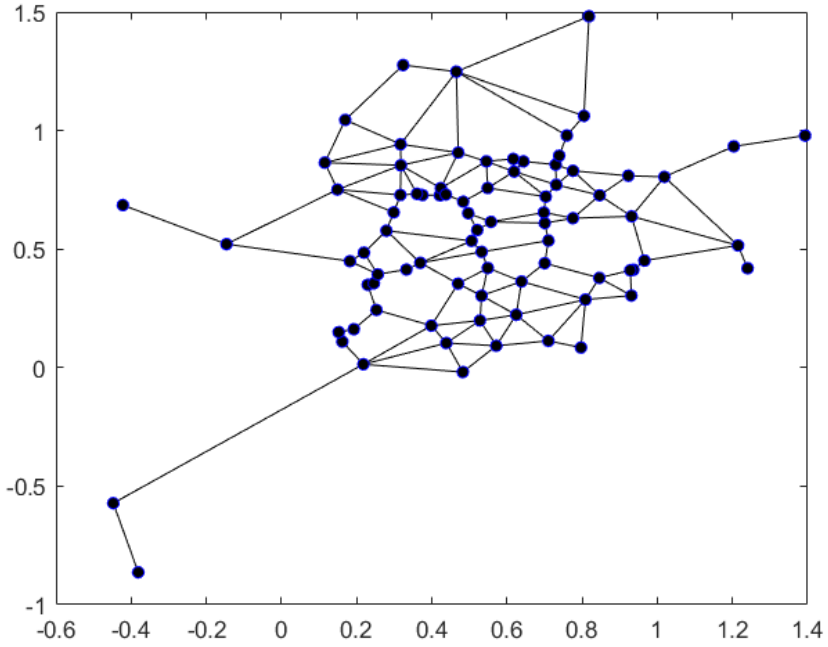

Figura 12 – CBG - Construção do Grafo a partir da Base Gaussiana gerada.

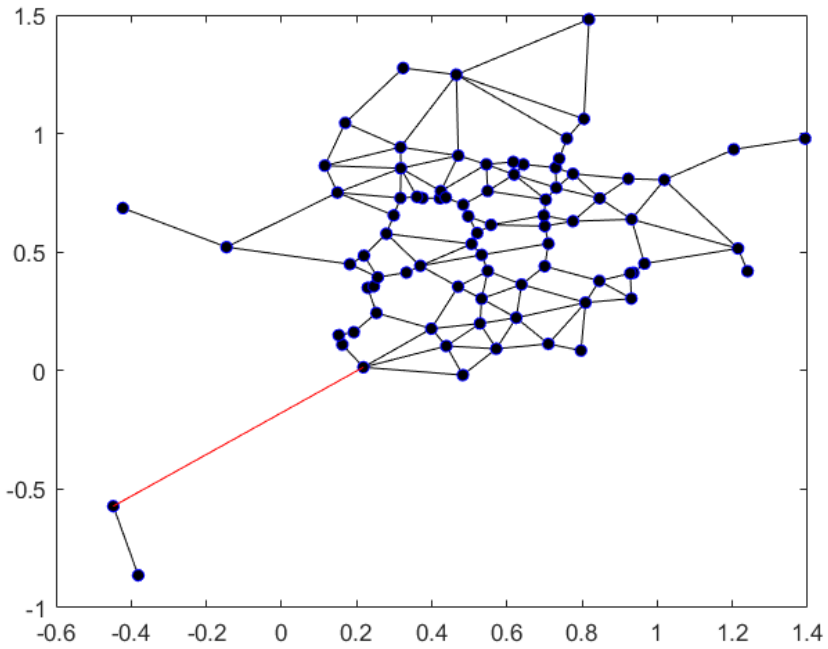

Figura 13 – CBG - Base Gaussiana - Definição do valor médio e marcação das Arestas de Suporte.

3.2 Definição do Classificador

Em suma, a classificação é realizada pela técnica do CHIP-clas reduzido, que pode ser encontrado em (GADE et al.,), e é obtida da seguinte forma:

Sendo $\mathbf{x}$ uma amostra a ser rotulada, encontra-se o ponto médio l mais próximo de x através de:

$$arg \min_{l} D(\mathbf{x}, Pl), \quad l = 1, ..., m, \tag{3.8}$$

onde D($\cdot$) é a função que retorna o valor da distância entre dois vetores. O hiperplano de separação que passa pelo ponto médio l mais próximo de $\mathbf{x}$ é definido como $H_l(\mathbf{x}) = (x^T w_l - b_l)$, onde $b_l = [(\frac{1}{2})(\mathbf{x}_i + \mathbf{x}_j)]w_l^T$ e $w_l = (\mathbf{x}_i + \mathbf{x}_j)$, sendo $(\mathbf{x}_i + \mathbf{x}_j)$ o par de vértices que formam a Aresta de Suporte associada ao ponto médio l. O rótulo de x é dado pela função de classificação $f(\mathbf{x})$, descrita por,

$$f(x) = \begin{cases} +1 \;\; se \;\; H_l(\mathbf{x}) > 0 \\ -1 \;\; se \;\; H_l(\mathbf{x}) \leq 0 \end{cases} \tag{3.9}$$

Esse processo pode ser visto de maneira mais detalhada pelo pseudocódigo do algoritmo a seguir:

Algorithm 1 Classificação - CHIP-clas Reduzido

▷ **Entradas:** conjunto de parâmetros B e W dos hiperplanos, conjunto de pontos médios das arestas de suporte P e conjunto de teste X_t.

▷ **Saída:** classes do conjunto de teste CT_e

for j in X_t **do**
  for i in P **do**
    $d(i) = D(X_t(j), P(i))$ ▷ Calcula a distância entre o novo padrão e os pontos médios
  end for
  $h_l = sign(x^T W_l - b_l)$ ▷ Hiperplano h_l que passa pelo ponto médio P_l mais próximo de $X_t(j)$
  if $h_l > 0$ **then**
    $CT_e(j) = +1$
  else
    $CT_e(j) = -1$
  end if
end for
return CT_e

Na Figura 14, podemos observar as bordas encontradas pela aresta de suporte. Na sequência, na Figura 15, temos o hiperplano traçado no ponto médio. Por fim, na Figura 16, temos o classificador que separa as classes, como base no hiperplano traçado sobre o ponto médio definido na aresta de suporte que se encontra entre as regiões de borda.

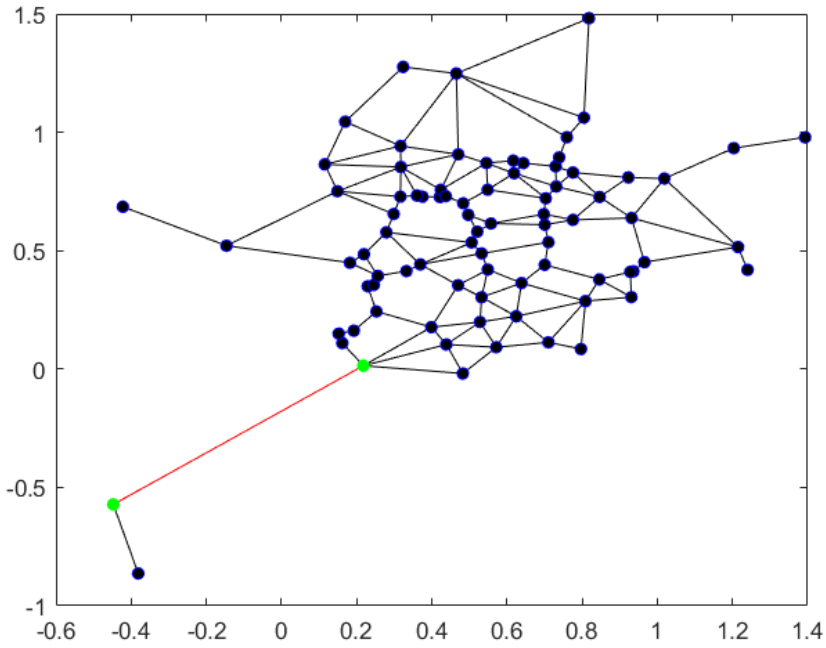

Figura 14 – CBG - Base Gaussiana - Localização das bordas pela aresta de suporte.

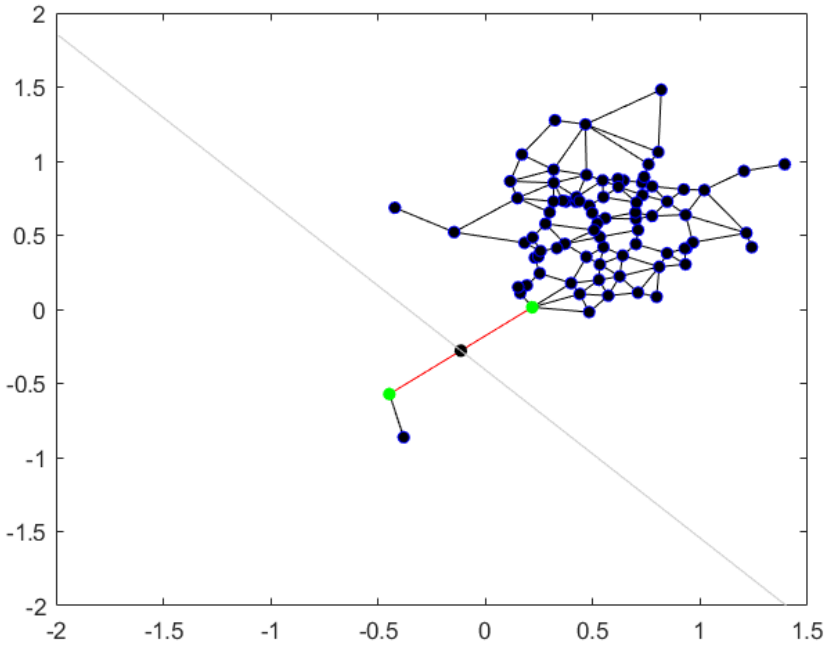

Figura 15 – CBG - Base Gaussiana - Marcação do ponto médio na aresta de suporte.

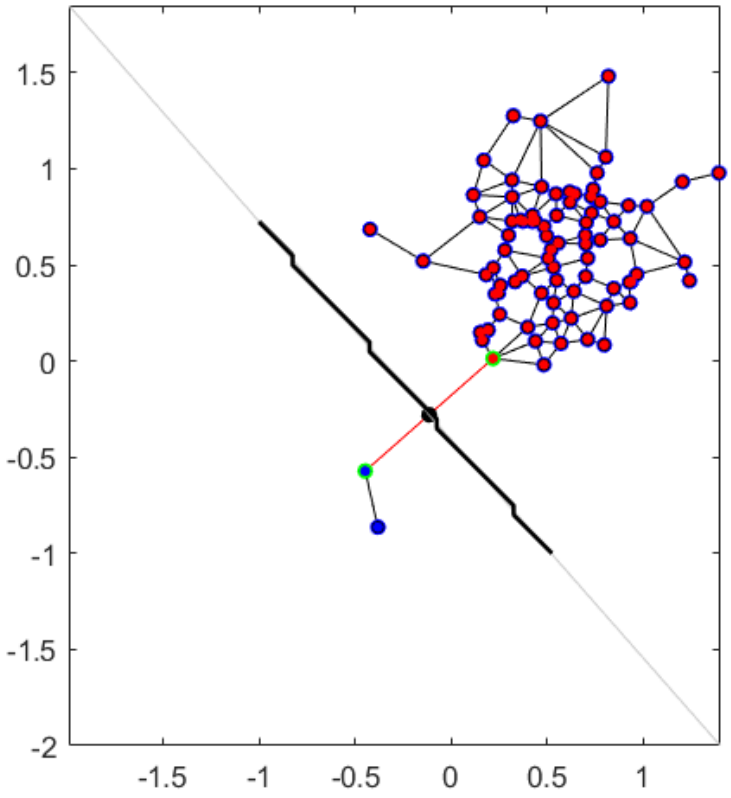

Figura 16 – CBG - Base Gaussiana - Construção do separador no hiperplano.

4 Resultados

Este capítulo trata da comparação, descrita na Seção 4.1, dos resultados obtidos usando o método proposto (CBG) implementado com a ferramenta MATLAB, que por sinal já possui em sua biblioteca o pacote para a SVM *one-class*, de modo que os nossos resultados serão comparados com os da SVM de uma classe, a qual vamos referenciar somente por SVM, mas que trata em específico da SVM *one-class*. As saídas são para as mesmas bases (*cluster, corners,* duas Luas*, fullmoon, halfkernel*) que são *benchmarks* da área de aprendizagem de máquina.

No Apêndice A, temos os resultados parciais que mostram cada etapa antes da saída do classificador, dispostas da Figura 27 à Figura 51. A seguir, temos as imagens com os resultados do método proposto (CBG), para as bases (*cluster, corners,* duas Luas*, fullmoon, halfkernel*) versos a saída da SVM para as respectivas bases.

A Figura 17 demonstra o resultado do CBG para a base Cluster, enquanto a Figura 18 tem o resultado da SVM para a mesma base. Percebe-se a marcação das classes, onde o CBG define melhor as mesmas, este foi o melhor resultado da SVM, onde sua detecção de classes se aproxima bem do resultado obtido utilizando o CBG.

Enquanto que, a Figura 19 demonstra o resultado do CBG para a base Corners, e a Figura 20 tem o resultado da SVM para a mesma base. É possível notar que a SVM não detecta bem as classes para esta base, e que o CBG não teve 100% de assertividade neste caso, contudo, seu desempenho ainda mostra uma distinção bem melhor comparado ao da SVM.

Já Figura 21 demonstra o resultado do CBG para a base duas Luas, enquanto a Figura 22 tem o resultado da SVM para a mesma base. É possivel observar que na saída da SVM não separou as classes, enquanto o CBG consegue distinguir ambas as classes.

Na Figura 23 demonstra o resultado do CBG para a base Fullmoon, enquanto a Figura 24 tem o resultado da SVM para a mesma base. Nesse caso a SVM começa a detectar bordas das classes, mas não as define por total, enquanto o CBG novamente deixa bem definida as duas classes.

Por fim, a Figura 25 demonstra o resultado do CBG para a base Halfkernel, enquanto a Figura 26 tem o resultado da SVM para a mesma base. Novamente a SVM faz uma marcação de borda das clases, mas não as difere em tipos diferentes de classes, enquanto o CBG separa bem as duas classes.

4.1 CBG vs SVM *one-class*

Para a saída SVM, seu resultado mais preciso, se deu para a base *Cluster*, como se pode observar no comparativo visual, em específico na Figura 18, por isso, seguimos nossa comparação tratando do melhor resultado do SVM com o respectivo para o CBG. Quanto à sua disposição, na saída da base *Cluster* para a classificação da SVM, observa-se que na Figura 18, às bolinhas vermelhas são as que têm os vetores de suporte; as bolinhas pretas, são as amostras; já as bolinhas em azul, são a outra classe.

Percebe-se que ele distingue a classe do meio, mais interna, para a outra classe de fora, para a imagem em questão. Porém, neste círculo de fora, mais interno, tem algumas bolinhas que só estão vermelhas, elas não estão marcadas como as demais da classe daquela região.

Esse método da SVM *one-class* em específico, gera scores para cada amostra desta. Quem tem *score* menor do que zero, é classificado como outra classe; Aqueles que têm o score igual a zero, são os vetores de suporte.

Desta forma, o SVM funciona baseado em parâmetros que se passam para ele, no caso, ele recebe um parâmetro que é a quantidade de dados de anomalia que a função deveria ter, como se fosse os *outliers*. Como já se tem a priori, o número de elementos da outra classe, passa-se o número de elementos da outra classe. Por exemplo, para a classe *cluster*, supondo-se que tendo 100 elementos da classe 1 e 200 elementos da classe 2, especifica-se como parâmetro da SVM, que existem 100 elementos que seriam *outliers*. E o método tenta encontrar estes 100 elementos, sendo a linha amarela na Figura 18, a fronteira que o método define.

Para a saída da SVM em específico, existem casos em que ela não consegue distinguir as duas classes, ela pega somente a borda, de modo que não encontra a outra classe, em contrapartida, o método proposto neste trabalho consegue definir bem as classes como se pode observar na Figura 17, que é a saída do método proposto, abordando grafos em sua construção, para a mesma base cluster.

Uma outra imagem que podemos observar bem a diferença entre a saída para a SVM e a saída do nosso método (CBG), é para a base *Fullmoon*, na Figura 24, temos a saída da SVM para a *Fullmoon*, onde ela não consegue distinguir direito as duas classes, e mostram que tem-se em azul uma classe do lado de dentro, e na envoltória é como se tivesse uma outra classe.

Logo os pontos pretos sem nenhum círculo são uma classe, e os pontos que estão com círculo em azul é a outra classe. Deste modo, ele reconhece o contorno e não consegue identificar que existe outra classe.

4.1.1 Resultados do método proposto utlizando grafos - comparativo CBG com SVM

CBG vs SVM - Resultados para a base *Cluster*:

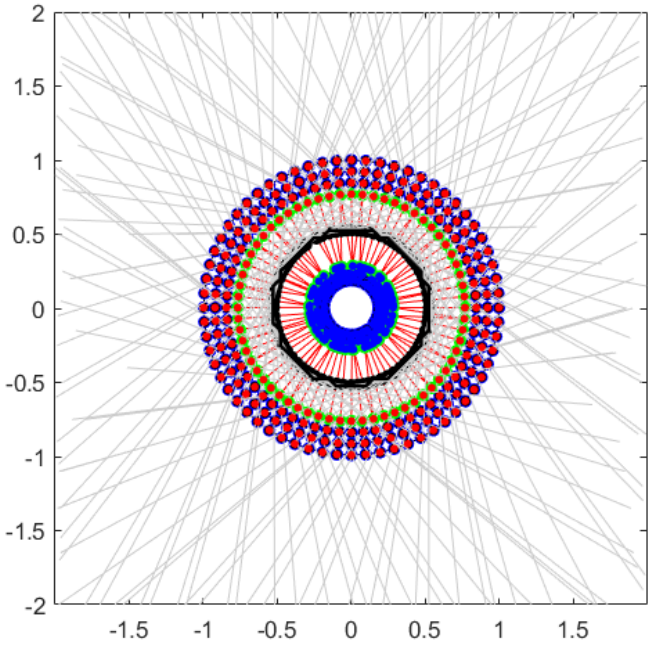

Figura 17 – CBG - Base Cluster - Construção do separador no hiperplano.

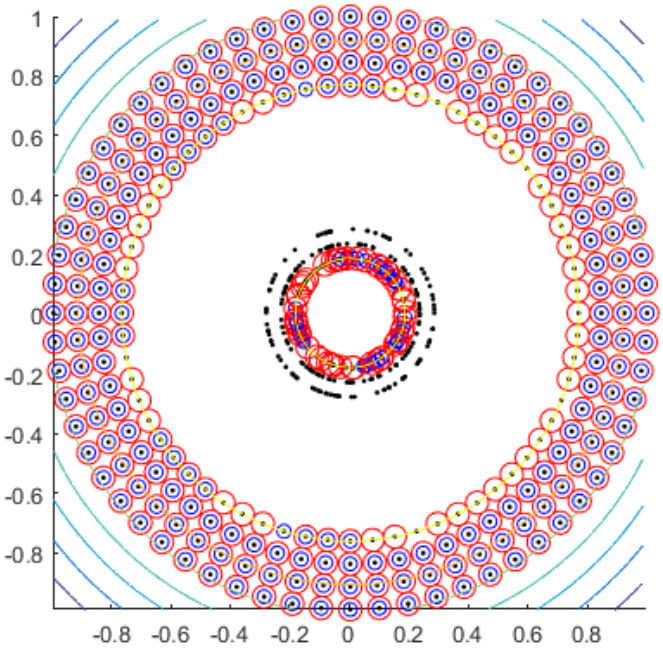

Figura 18 – SVM - Base cluster.

CBG vs SVM - Resultados para a base *Corners*:

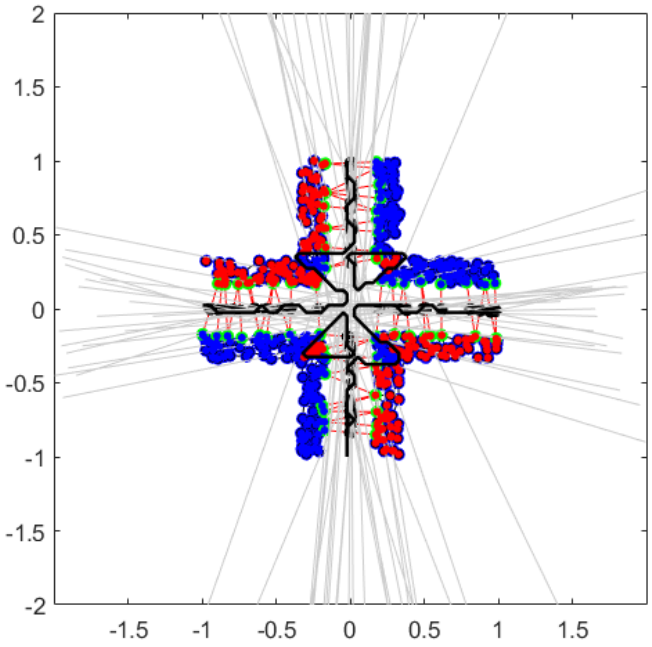

Figura 19 – CBG - Base Corners - Construção do separador no hiperplano.

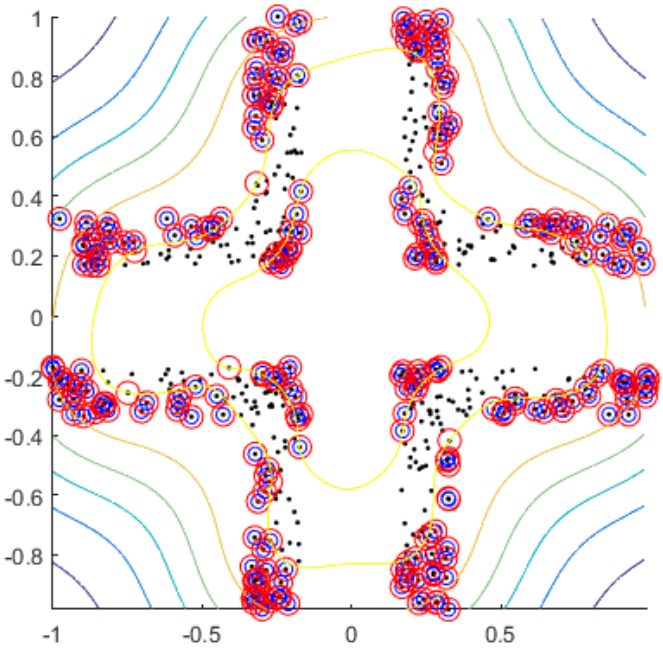

Figura 20 – SVM - Base corners.

CBG vs SVM - Resultados para a base duas Luas:

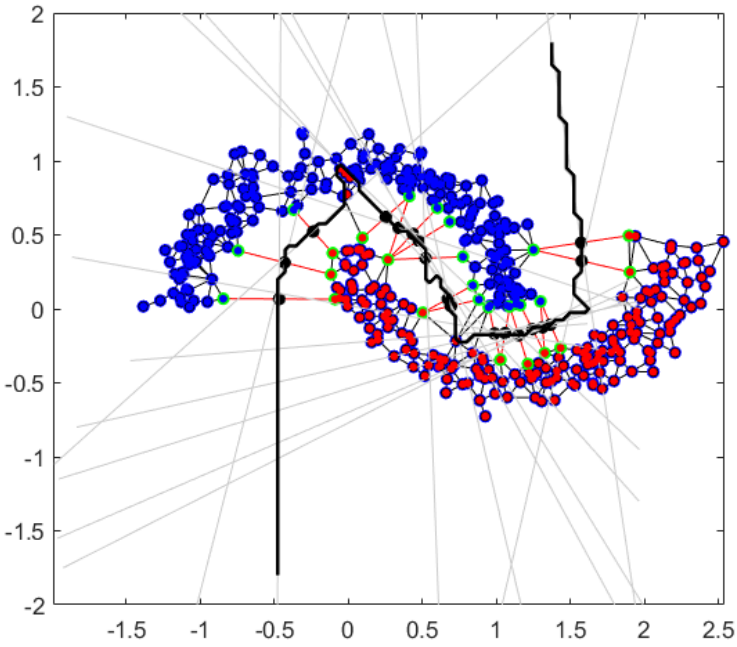

Figura 21 – CBG - Base duas Luas - Construção do separador no hiperplano.

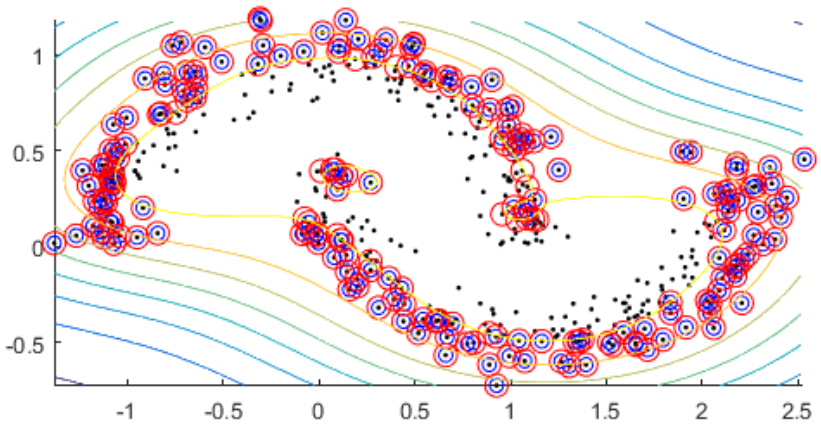

Figura 22 – SVM - Base duas luas.

CBG vs SVM - Resultados para a base *Fullmoon*:

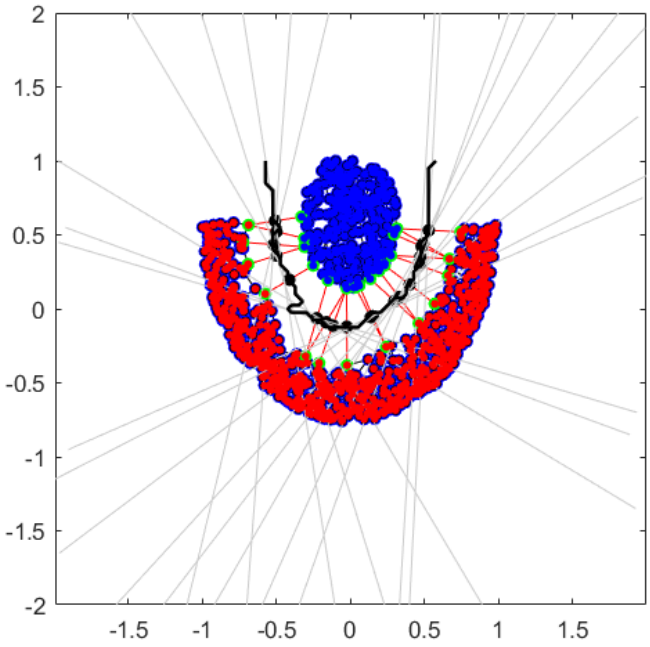

Figura 23 – CBG - Base Fullmoon - Construção do separador no hiperplano.

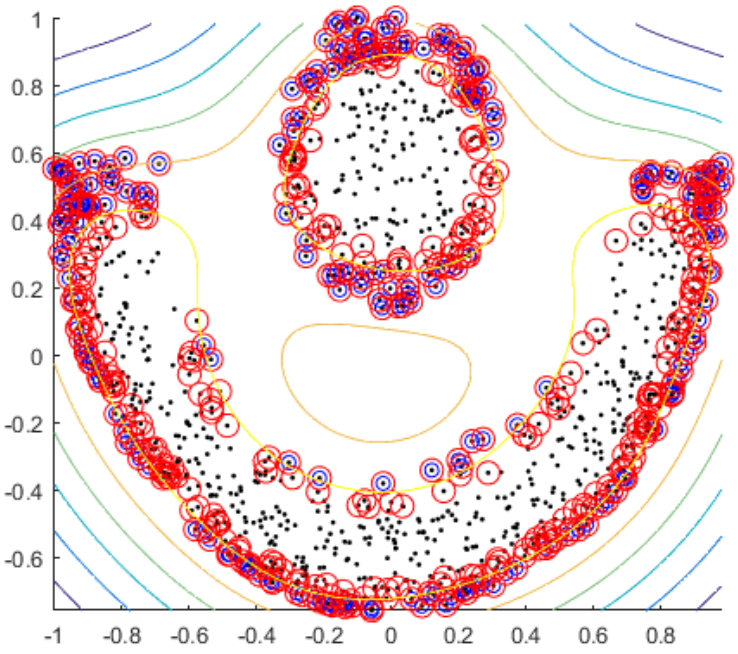

Figura 24 – SVM - Base fullmoon.

CBG vs SVM - Resultados para a base *Halfkernel*:

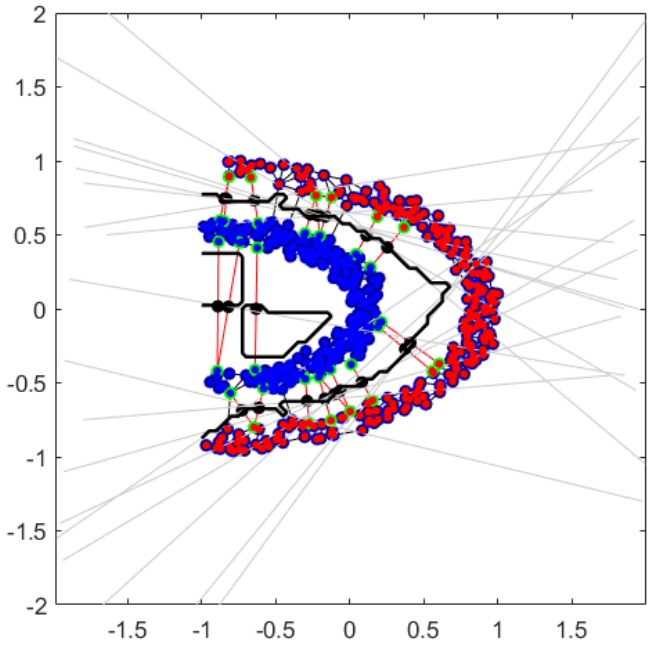

Figura 25 – CBG - Base Halfkernel - Construção do separador no hiperplano.

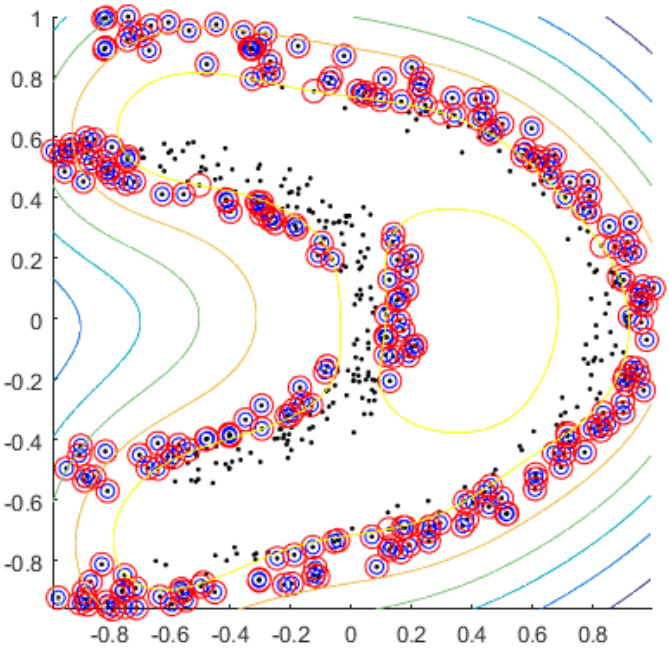

Figura 26 – SVM - Base halfkernel.

5 Conclusão

Este trabalho apresentou uma abordagem que se baseia na estimativa geométrica na condição de fronteira entre os conjuntos de dados de pontos do plano euclidiano, mensurando as classes de dados, de modo a obter um separador para a(s) classe(s) ao analisar as condições de fronteira entre o(s) conjunto(s) de dados a partir de uma modelagem destes como grafos, em particular o grafo de Gabriel, o qual foi nomeada (CBG) Classificação Baseada em Grafos.

Deste modo, ao comparar nossa abordagem com a SVM *one-class*, observamos que o melhor resultado para SVM nas 5 bases testadas, foi somente para 1, a base cluster, na qual encontrou as duas classes, contudo, ainda não foi 100% assertivo, se comparado com o método proposto neste trabalho, de modo que, destacamos que a abordagem usando grafo teve grande sucesso em sua saída, superando em qualidade de detecção das classes o SVM já existente. As demais bases, processadas na SVM, tiveram resultados piores ao tentar encontrar as classes, sendo que para as mesmas bases no nosso método, o resultado se mostrou bem superior.

Ainda destacamos que para utilizar a SVM, temos que especificar parâmetros, que precisam ser definidos para sua execução. Já a nossa abordagem utilizando grafos (CBG), não necessita de parâmetros, contudo, para uma base de dados muito grande, existe a tendência de um custo maior.

Para as bases que foram testadas em nossos experimentos, percebe-se que o resultado do método usando grafo, foi bem melhor do que o da SVM de uma classe, e por se tratar de bases que não são tão grandes, os resultados foram bem rápidos.

É importante ressaltar, que para todas as bases testadas, a única que utilizando a SVM, chegou com um resultado próximo da CBG, foi para a base *cluster*, onde ainda é perceptível que o melhor resultado se deu para o nosso método, devido ao sucesso na detecção das classes, que é notável visualmente ao se comparar as imagens. Assim, a SVM teve sucesso em 20% dos testes, 1 das 5 bases testadas.

Em contrapartida, em nossa abordagem, 4 das 5 bases tiveram 100% de acerto, o que totaliza 80% dos testes. Ainda ressaltando que a base *corners*, a qual não teve um resultado totalmente assertivo, ainda se encontra com uma classificação bem mais definida visualmente do que a mesma pela SVM.

Desta forma, é possível observar que nossa abordagem foi 4 vezes melhor que a SVM de uma classe para as bases testadas (*cluster, corners,* duas Luas*, fullmoon, halfkernel*).

5.1 Propostas de trabalhos futuros

Como trabalhos futuros, podemos listar algumas opções que seriam interessantes:

- Realizar experimentos com o método proposto para bases de dados com maiores dimensões, com um número maior de dados. De modo a observar o seu desempenho para a detecção de classes neste contexto.
- Aplicar o método proposto em base de dados reais, como bases médicas de tamanhos menores, visto da assertividade do método para as bases testadas.
- Desenvolver um levantamento de custo vs tamanho de base, por meio de experimentos variados, para ter uma expectativa de custo vs tamanho de base vs assertividade na classificação.
- Realizar estudo da viabilidade de adaptação para explorar a abordagem em problemas reais da indústria, de modo a fazer uma classificação em tempo real.
- Utilizar o método proposto para classificar a base de dados da coluna cervical obtida nos estudos da iniciação científica disponível em https://github.com/brunocesarti/Iniciacao-Cientifica, obtida por meio da ferramenta DeepLabCut, podendo ser aplicado a demais contextos.

Referências

A, B. M. M. *Operações elementares em grafos*. [S.l.]: Emarc, Sbmac, 2007. v. 7. Citado 2 vezes nas páginas 18 e 38.

ARONSZAJN, N. Theory of reproducing kernels. *Transactions of the American mathematical society*, v. 68, n. 3, p. 337–404, 1950. Citado na página 29.

AURENHAMMER, F.; KLEIN, R. Voronoi diagrams. *Handbook of computational geometry*, v. 5, n. 10, p. 201–290, 2000. Citado na página 34.

BERG, M. et al. Computational geometry: Algorithms and applications, springer-verlag telos. *Santa Clara, CA, USA*, 2008. Citado na página 34.

BERTIN, E.; BILLIOT, J.-M.; DROUILHET, R. Continuum percolation in the gabriel graph. *Advances in Applied Probability*, Cambridge University Press, v. 34, n. 4, p. 689–701, 2002. Citado na página 35.

BISHOP, C. M. Novelty detection and neural network validation. *IEE Proceedings-Vision, Image and Signal processing*, IET, v. 141, n. 4, p. 217–222, 1994. Citado 2 vezes nas páginas 26 e 27.

BOSE, P. et al. On the spanning ratio of gabriel graphs and beta-skeletons. *SIAM Journal on Discrete Mathematics*, SIAM, v. 20, n. 2, p. 412–427, 2006. Citado na página 35.

BRODLEY, C. E.; FRIEDL, M. A. et al. Identifying and eliminating mislabeled training instances. In: CITESEER. *Proceedings of the National Conference on Artificial Intelligence*. [S.l.], 1996. p. 799–805. Citado na página 27.

CHALAPATHY, R.; CHAWLA, S. Deep learning for anomaly detection: A survey. *arXiv preprint arXiv:1901.03407*, 2019. Citado na página 21.

COVER, T.; HART, P. Nearest neighbor pattern classification. *IEEE transactions on information theory*, IEEE, v. 13, n. 1, p. 21–27, 1967. Citado na página 25.

DASGUPTA, D.; FORREST, S. Novelty detection in time series data using ideas from immunology. In: CITESEER. *Proceedings of the international conference on intelligent systems*. [S.l.], 1996. p. 82–87. Citado na página 25.

DECOSTE, D.; LEVINE, M. B. Automated event detection in space instruments: a case study using ipex-2 data and support vector machines. Citeseer, 2000. Citado na página 26.

DUDA, R. O.; HART, P. E. et al. *Pattern classification and scene analysis*. [S.l.]: Wiley New York, 1973. v. 3. Citado na página 31.

FAWCETT, T.; PROVOST, F. Activity monitoring: Noticing interesting changes in behavior. In: *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*. [S.l.: s.n.], 1999. p. 53–62. Citado na página 23.

FERNÁNDEZ, A. et al. *Learning from imbalanced data sets*. [S.l.]: Springer, 2018. v. 10. Citado na página 28.

GABRIEL, K. R.; SOKAL, R. R. A new statistical approach to geographic variation analysis. *Systematic zoology*, Society of Systematic Zoology, v. 18, n. 3, p. 259–278, 1969. Citado na página 34.

GADE, L. dos R. et al. Nn-clas: classificador geométrico de margem larga baseado na regra do vizinho mais próximo. Citado na página 43.

HAWKINS, D. M. *Identification of outliers*. [S.l.]: Springer, 1980. v. 11. Citado 2 vezes nas páginas 18 e 21.

HODGE, V.; AUSTIN, J. A survey of outlier detection methodologies. *Artificial intelligence review*, Springer, v. 22, n. 2, p. 85–126, 2004. Citado 7 vezes nas páginas 18, 21, 22, 23, 24, 25 e 26.

HONEINE, P.; RICHARD, C. Preimage problem in kernel-based machine learning. *IEEE Signal Processing Magazine*, IEEE, v. 28, n. 2, p. 77–88, 2011. Citado na página 29.

HULLE, M. M. V. *Self-organizing Maps*. 2012. Citado na página 27.

HUSH, D. R.; MOYA, M. M.; CLARK, S.-Y. Constrained neural network architectures for target recognition. In: INTERNATIONAL SOCIETY FOR OPTICS AND PHOTONICS. *Applications of Artificial Neural Networks III*. [S.l.], 1992. v. 1709, p. 96–106. Citado na página 31.

JAPKOWICZ, N. et al. A novelty detection approach to classification. In: CITESEER. *IJCAI*. [S.l.], 1995. v. 1, p. 518–523. Citado na página 23.

JOHN, G. H. Robust decision trees: Removing outliers from databases. In: *KDD*. [S.l.: s.n.], 1995. v. 95, p. 174–179. Citado na página 27.

LEHMAN, E.; LEIGHTON, T.; MEYER, A. R. *Mathematics for computer science*. [S.l.], 2010. Citado 3 vezes nas páginas 9, 33 e 34.

MARSLAND, S. *On-line novelty detection through self-organisation, with application to inspection robotics*. [S.l.]: The University of Manchester (United Kingdom), 2001. Citado na página 23.

MATULA, D. W.; SOKAL, R. R. Properties of gabriel graphs relevant to geographic variation research and the clustering of points in the plane. *Geographical analysis*, Wiley Online Library, v. 12, n. 3, p. 205–222, 1980. Citado na página 35.

MOYA, M. M.; HOSTETLER, L. D. *One-class generalization in second-order backpropagation networks for image classification*. [S.l.], 1989. Citado na página 32.

MOYA, M. M.; HUSH, D. R. Network constraints and multi-objective optimization for one-class classification. *Neural networks*, Elsevier, v. 9, n. 3, p. 463–474, 1996. Citado 2 vezes nas páginas 31 e 32.

MOYA, M. M.; KOCH, M. W.; HOSTETLER, L. D. One-class classifier networks for target recognition applications. *NASA STI/Recon Technical Report N*, v. 93, p. 24043, 1993. Citado na página 29.

NAIRAC, A. et al. A system for the analysis of jet engine vibration data. *Integrated Computer-Aided Engineering*, IOS Press, v. 6, n. 1, p. 53–66, 1999. Citado 2 vezes nas páginas 25 e 27.

NORRENBROCK, C. Percolation threshold on planar euclidean gabriel graphs. *The European Physical Journal B*, Springer, v. 89, n. 5, p. 1–6, 2016. Citado na página 35.

NOUMIR, Z.; HONEINE, P.; RICHARD, C. On simple one-class classification methods. In: IEEE. *2012 IEEE International Symposium on Information Theory Proceedings*. [S.l.], 2012. p. 2022–2026. Citado 2 vezes nas páginas 29 e 32.

RIAZ, F.; ALI, K. M. Applications of graph theory in computer science. In: IEEE. *2011 Third International Conference on Computational Intelligence, Communication Systems and Networks*. [S.l.], 2011. p. 142–145. Citado na página 32.

RITTER, H. et al. *Neural computation and self-organizing maps: an introduction*. [S.l.]: Addison-Wesley Reading, MA, 1992. Citado na página 27.

ROBERTS, S.; TARASSENKO, L. A probabilistic resource allocating network for novelty detection. *Neural Computation*, MIT Press, v. 6, n. 2, p. 270–284, 1994. Citado na página 26.

ROUSSEEUW, P. J.; LEROY, A. M. *Robust regression and outlier detection*. [S.l.]: John wiley & sons, 2005. v. 589. Citado 2 vezes nas páginas 22 e 25.

SALGADO, M. N.; TORRES, L. C.; BRAGA, A. P. Modelo geométrico de margem larga baseado em propriedades estruturais de grafos de gabriel e em distância geodésica. Citado na página 38.

SCHÖLKOPF, B. et al. Estimating the support of a high-dimensional distribution. *Neural computation*, MIT Press One Rogers Street, Cambridge, MA 02142-1209, USA journals-info . . . , v. 13, n. 7, p. 1443–1471, 2001. Citado na página 29.

SCHÖLKOPF, B. et al. Support vector method for novelty detection. In: CITESEER. *NIPS*. [S.l.], 1999. v. 12, p. 582–588. Citado 2 vezes nas páginas 28 e 29.

SKALAK, D. B.; RISSLAND, E. L. Inductive learning in a mixed paradigm setting. In: *AAAI*. [S.l.: s.n.], 1990. p. 840–847. Citado na página 27.

TAX, D. M.; DUIN, R. P. Support vector data description. *Machine learning*, Springer, v. 54, n. 1, p. 45–66, 2004. Citado na página 29.

TORRES, L. C.; CASTRO, C. L.; BRAGA, A. P. Estratégia de decisão baseada em margem para o aprendizado multiobjetivo de redes neurais. 2011. Citado 5 vezes nas páginas 9, 18, 38, 39 e 40.

TORRES, L. C. B. Uma nova abordagem baseada em margem para seleção de modelos neurais. Universidade Federal de Minas Gerais, 2012. Citado 2 vezes nas páginas 18 e 38.

TORRES, L. C. B. Classificador por arestas de suporte (clas): Métodos de aprendizado baseados em grafos de gabriel. Universidade Federal de Minas Gerais, 2016. Citado na página 38.

TOUSSAINT, G. T. The relative neighbourhood graph of a finite planar set. *Pattern recognition*, Elsevier, v. 12, n. 4, p. 261–268, 1980. Citado na página 35.

ZHANG, W.; KING, I. A study of the relationship between support vector machine and gabriel graph. In: IEEE. *Proceedings of the 2002 International Joint Conference on Neural Networks. IJCNN'02 (Cat. No. 02CH37290)*. [S.l.], 2002. v. 1, p. 239–244. Citado 2 vezes nas páginas 34 e 35.

Apêndices

APÊNDICE A – Materiais elaborados pelo autor

CBG - Resultados passo a passo das etapas até antes de gerar a saída da classificação para base Cluster:

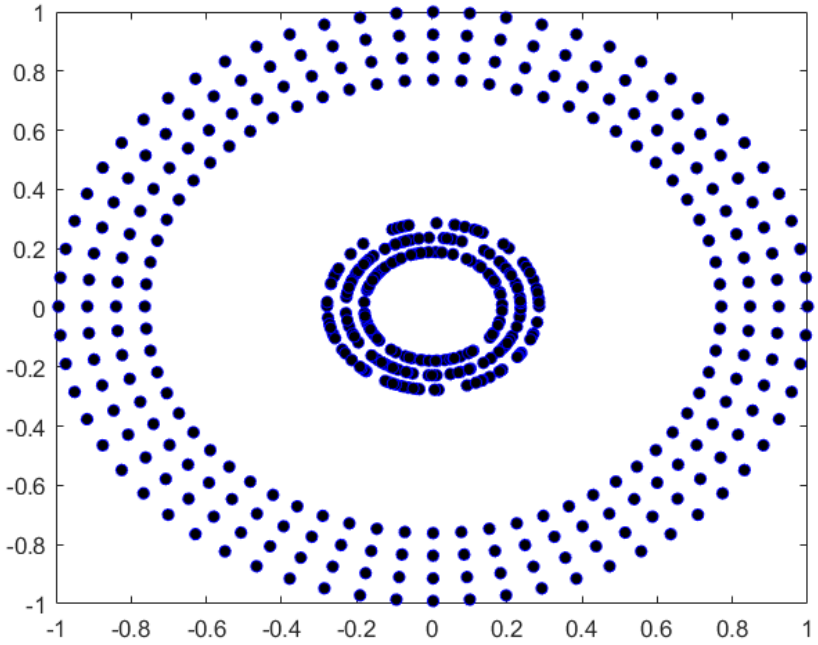

Figura 27 – CBG - Construção da Base Cluster.

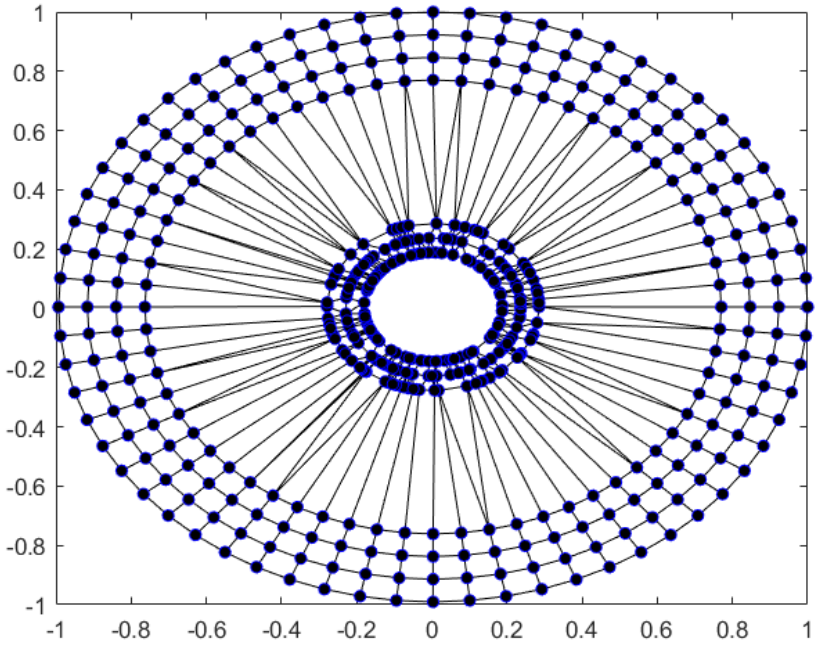

Figura 28 – CBG - Construção do Grafo a partir da Base Cluster gerada.

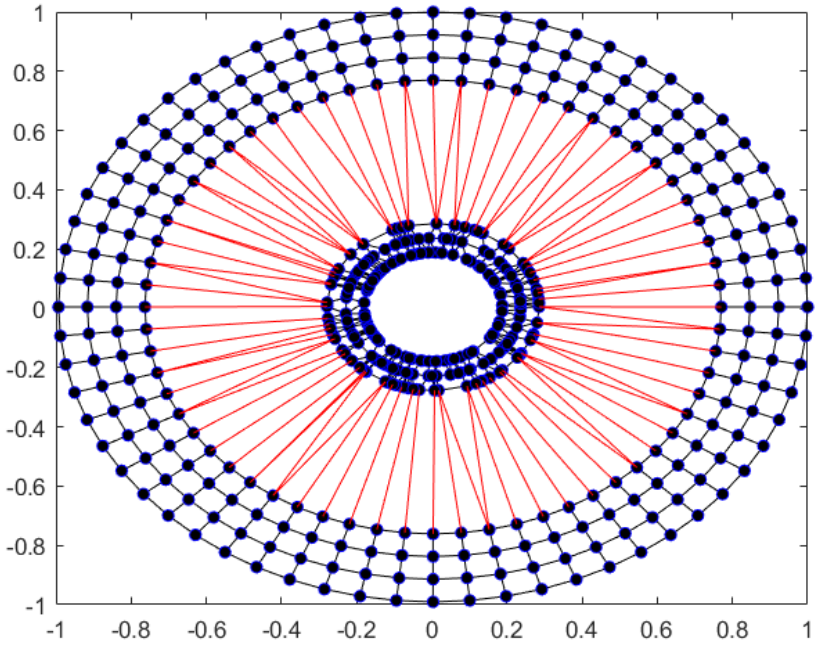

Figura 29 – CBG - Base Cluster - Definição do valor médio e marcação das Arestas de Suporte.

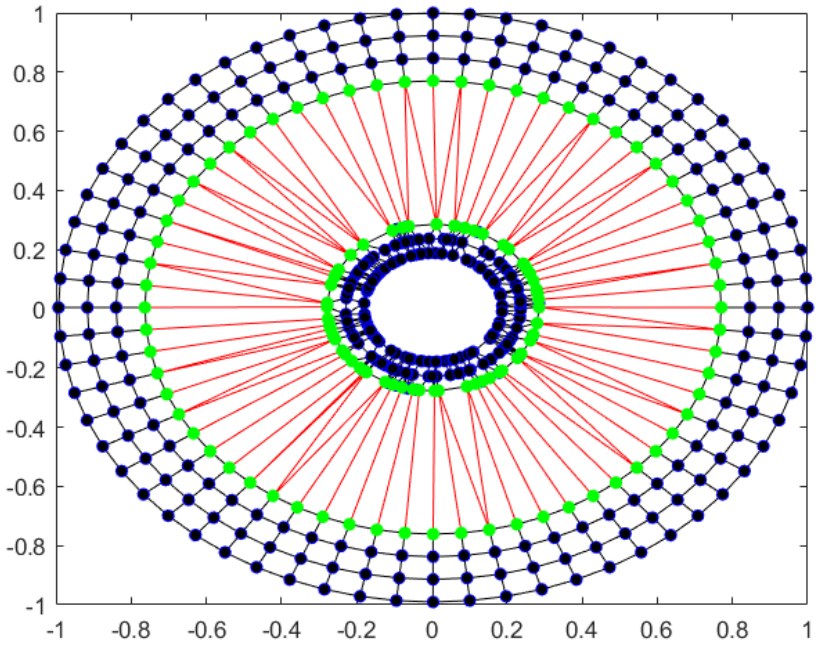

Figura 30 – CBG - Base Cluster - Localização das bordas pela aresta de suporte.

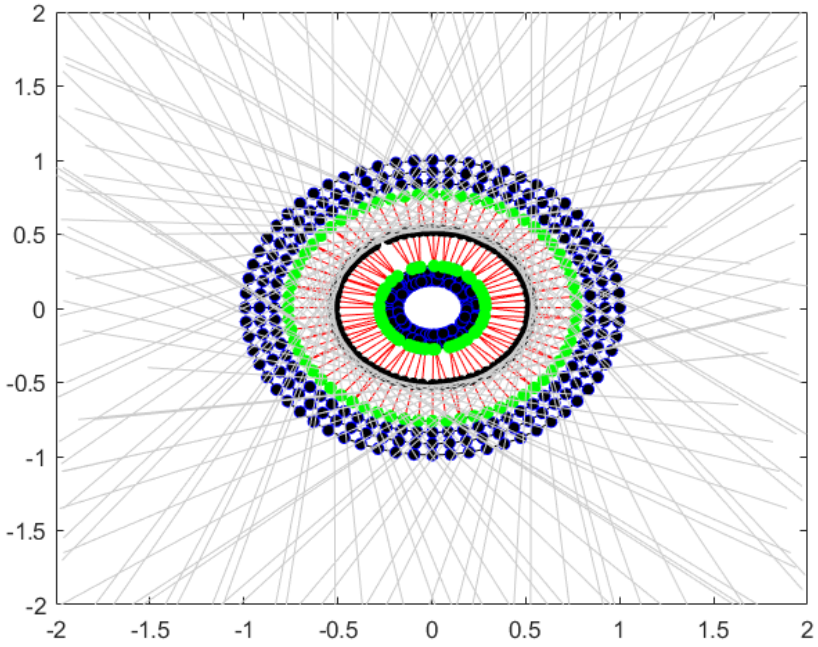

Figura 31 – CBG - Base Cluster - Marcação do ponto médio na aresta de suporte.

CBG - Resultados passo a passo das etapas até antes de gerar a saída da classificação para base Corners:

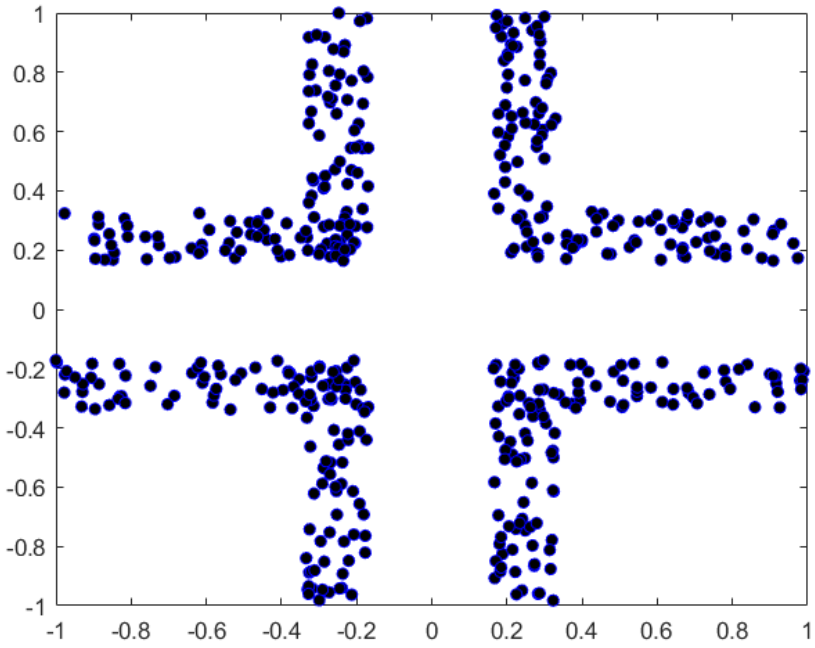

Figura 32 – CBG - Construção da Base Corners.

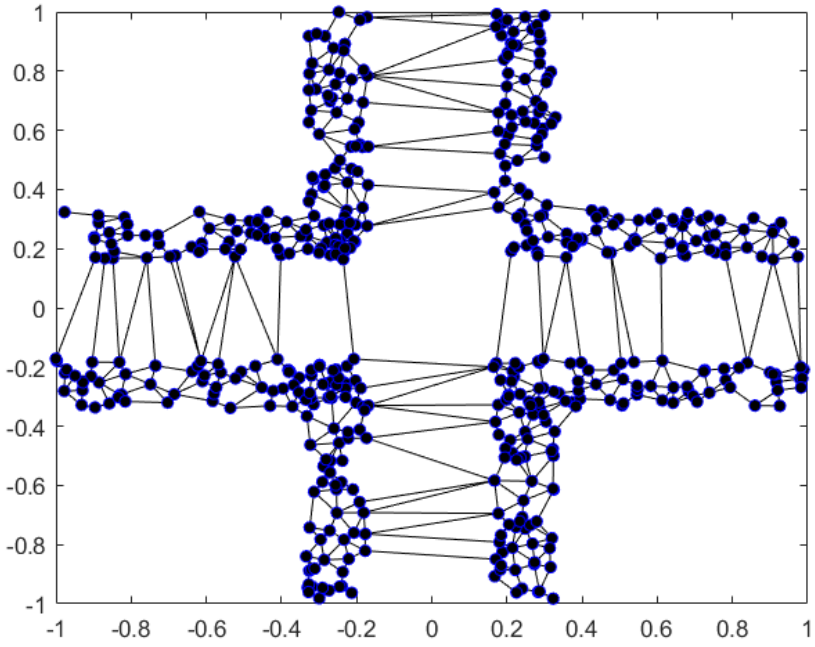

Figura 33 – CBG - Construção do Grafo a partir da Base Corners gerada.

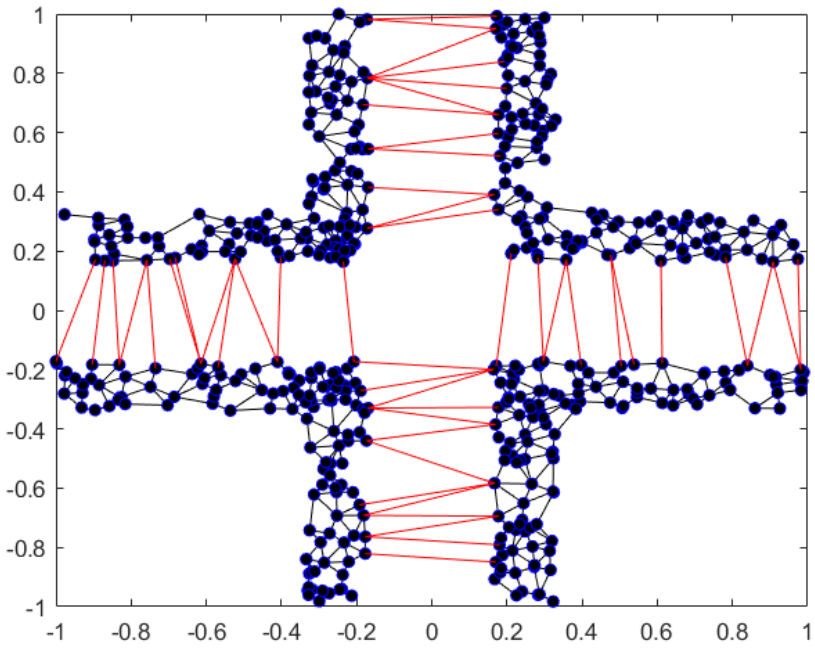

Figura 34 – CBG - Base Corners - Definição do valor médio e marcação das Arestas de Suporte.

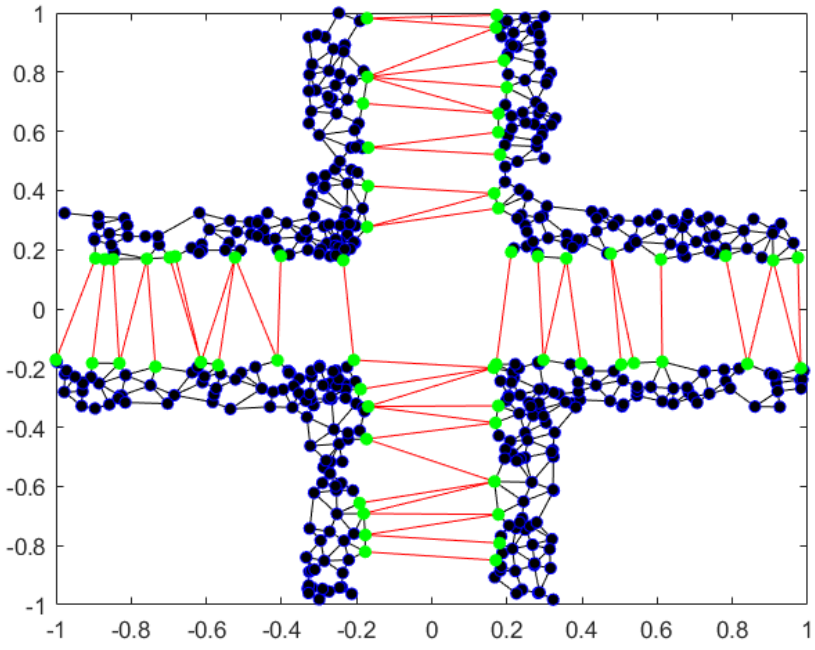

Figura 35 – CBG - Base Corners - Localização das bordas pela aresta de suporte.

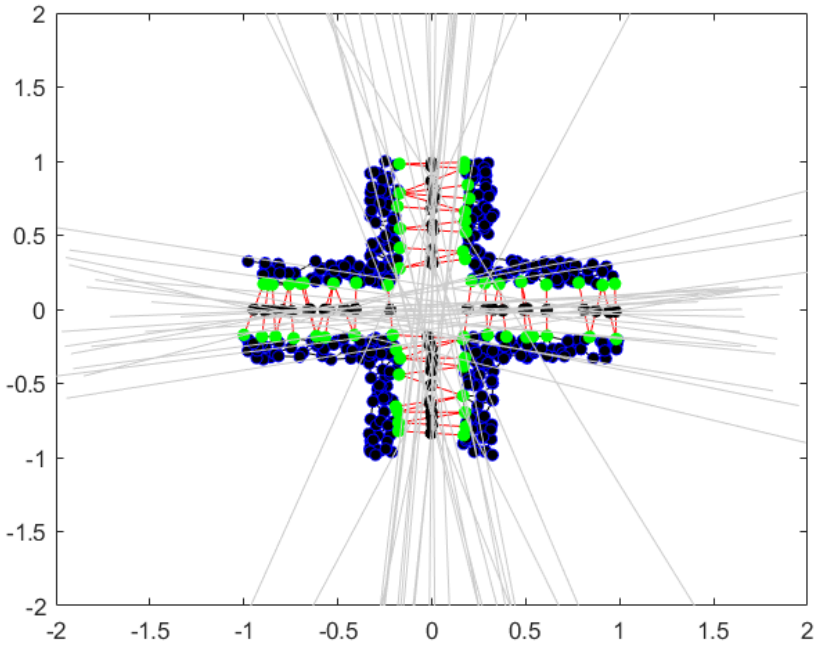

Figura 36 – CBG - Base Corners - Marcação do ponto médio na aresta de suporte.

CBG - Resultados passo a passo das etapas até antes de gerar a saída da classificação para base duas Luas:

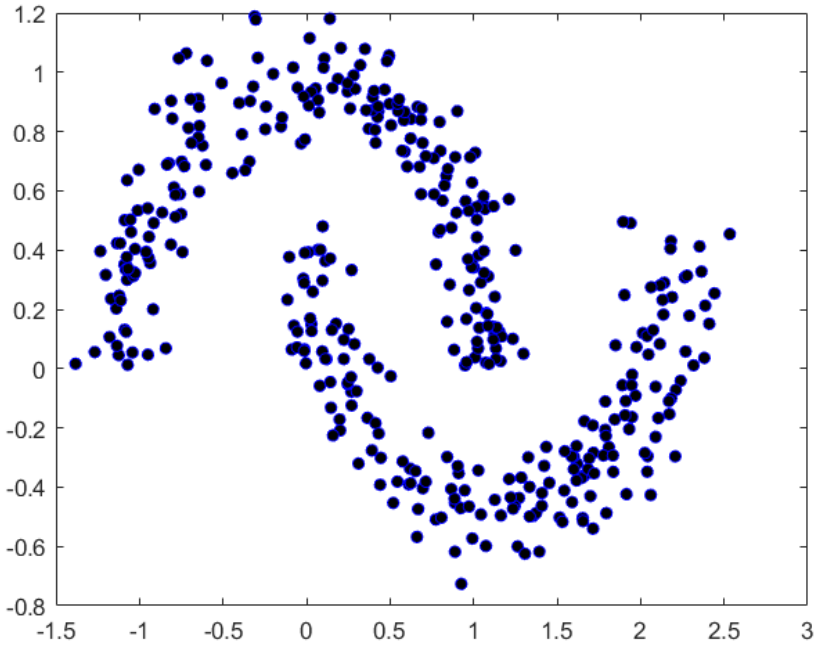

Figura 37 – CBG - Construção da Base duas Luas.

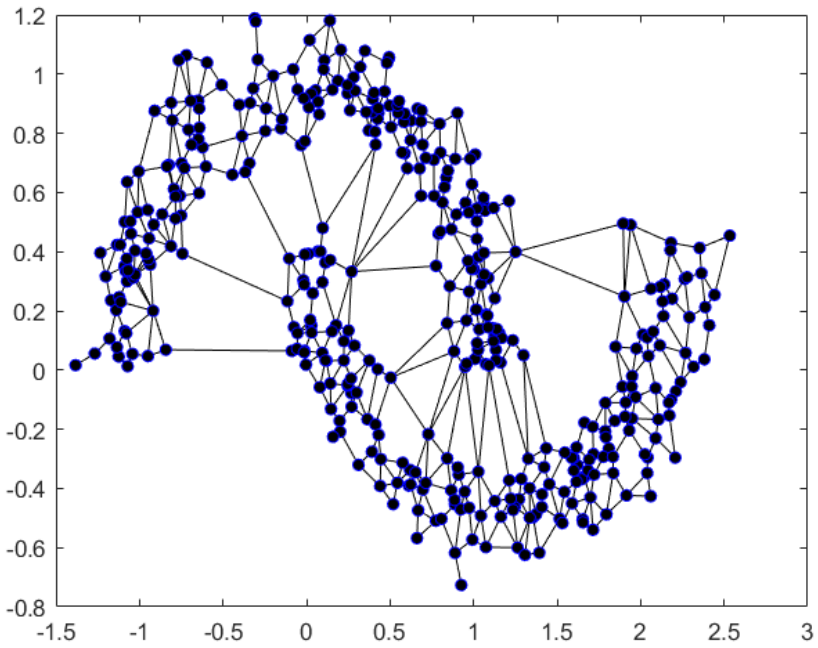

Figura 38 – CBG - Construção do Grafo a partir da Base duas Luas gerada.

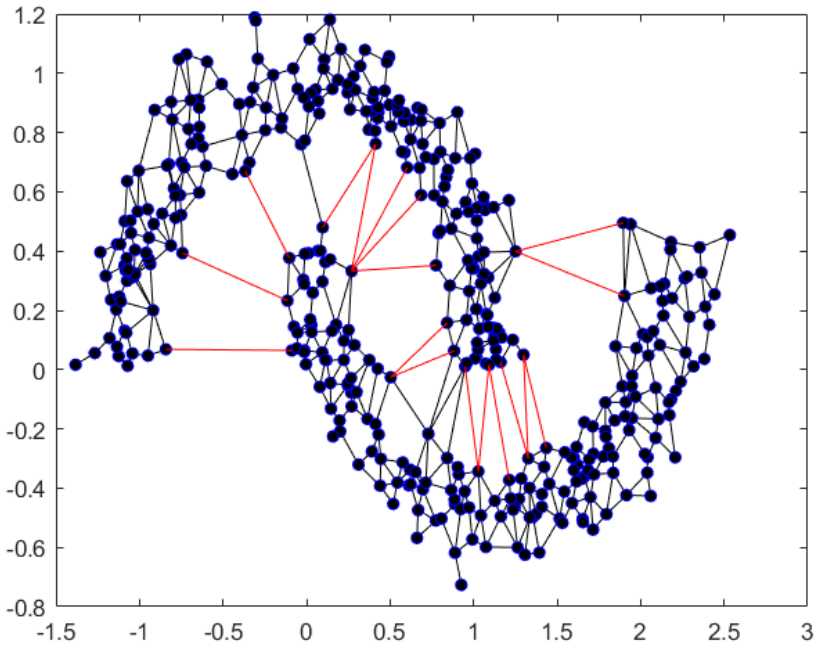

Figura 39 – CBG - Base duas Luas - Definição do valor médio e marcação das Arestas de Suporte.

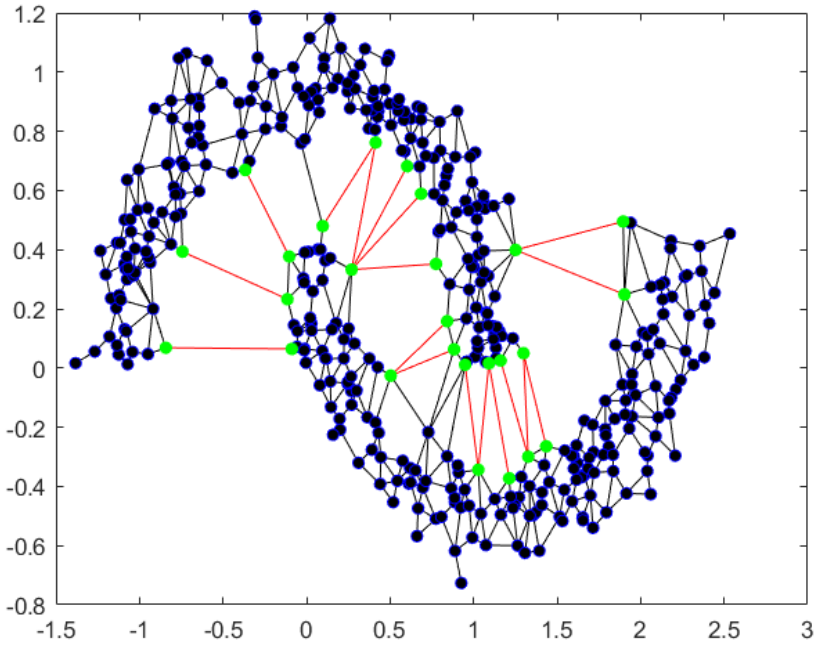

Figura 40 – CBG - Base duas Luas - Localização das bordas pela aresta de suporte.

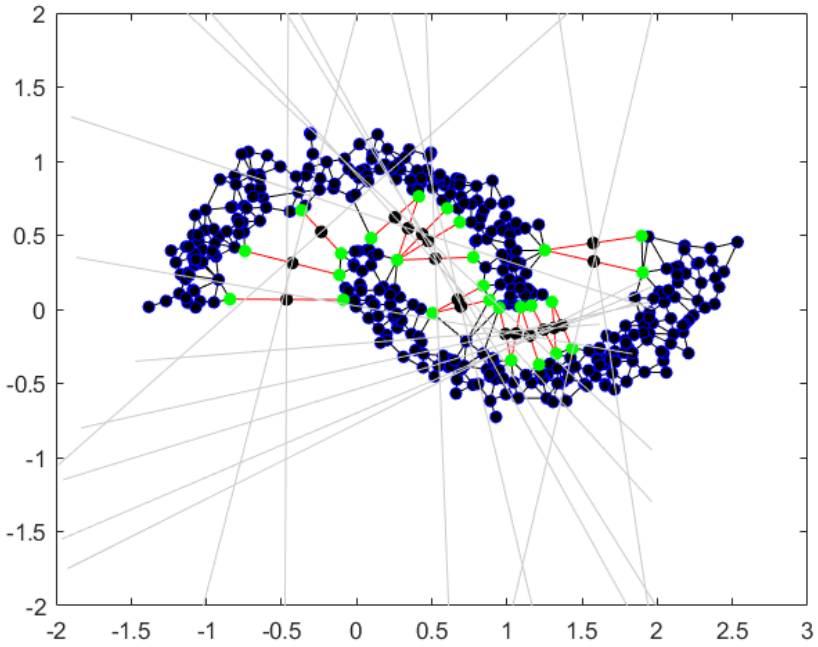

Figura 41 – CBG - Base duas Luas - Marcação do ponto médio na aresta de suporte.

CBG - Resultados passo a passo das etapas até antes de gerar a saída da classificação para base Fullmoon:

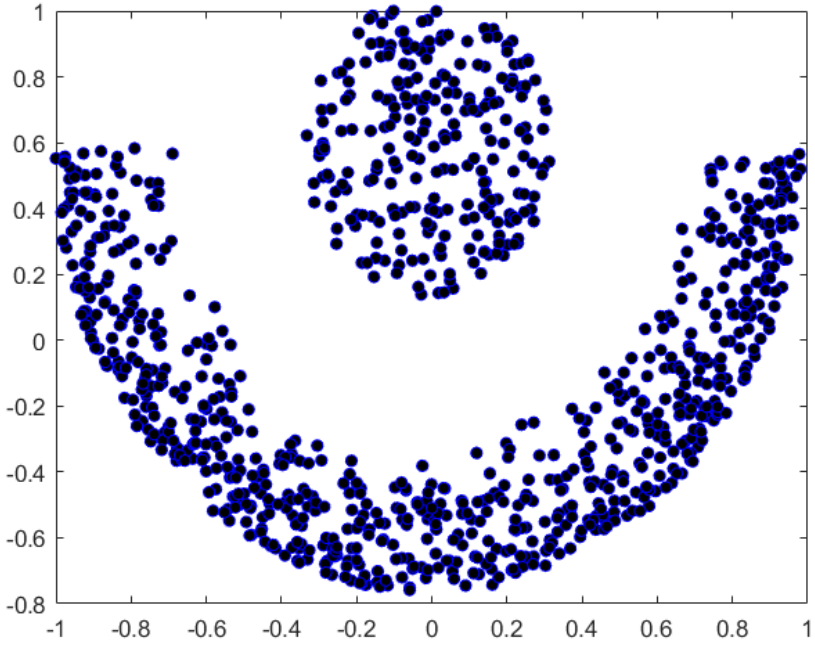

Figura 42 – CBG - Construção da Base Fullmoon.

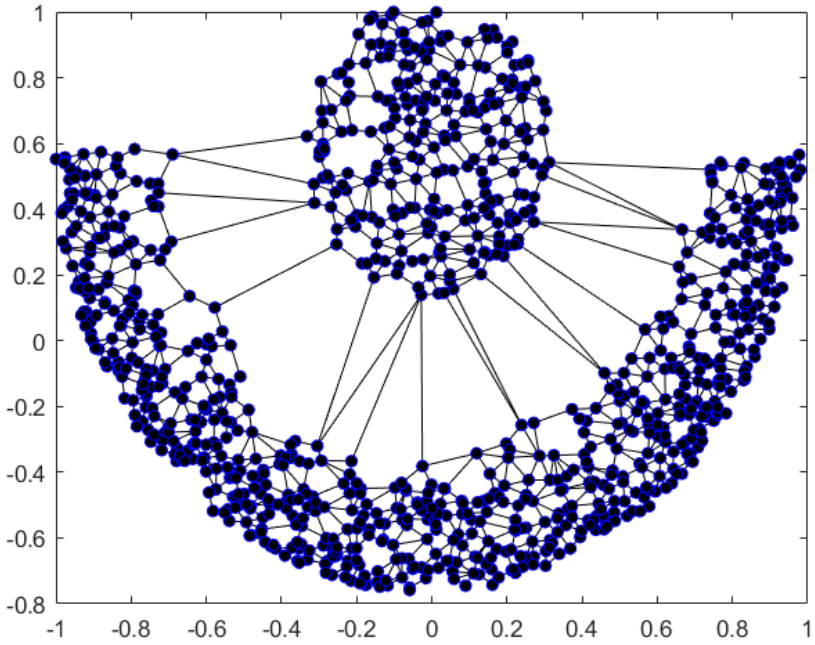

Figura 43 – CBG - Construção do Grafo a partir da Base Fullmoon gerada.

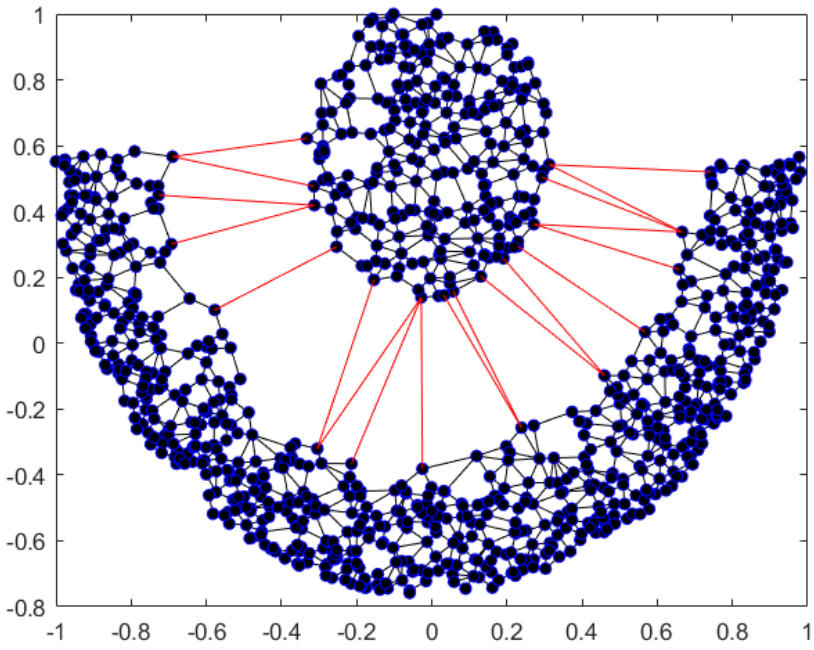

Figura 44 – CBG - Base Fullmoon - Definição do valor médio e marcação das Arestas de Suporte.

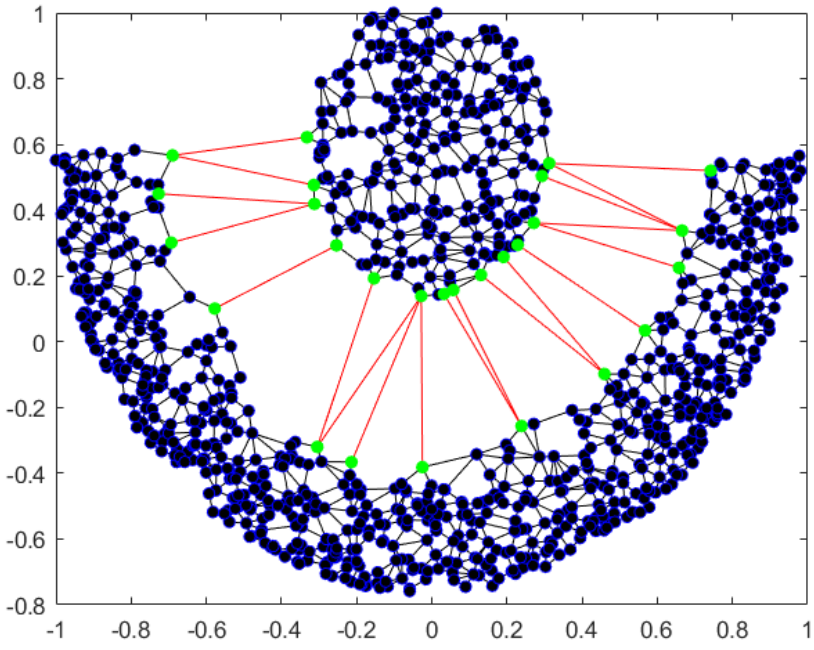

Figura 45 – CBG - Base Fullmoon - Localização das bordas pela aresta de suporte.

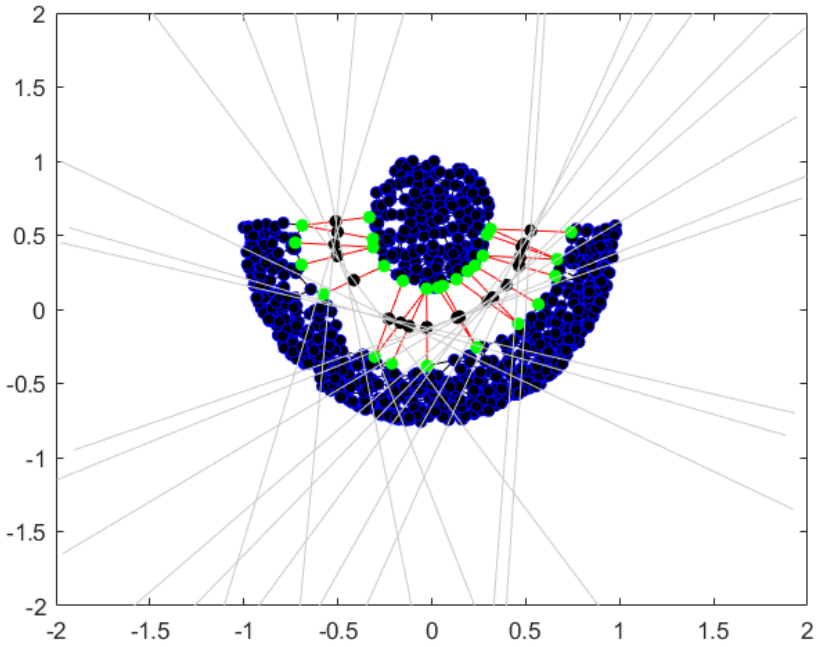

Figura 46 – CBG - Base Fullmoon - Marcação do ponto médio na aresta de suporte.

CBG - Resultados passo a passo das etapas até antes de gerar a saída da classificação para base Halfkernel:

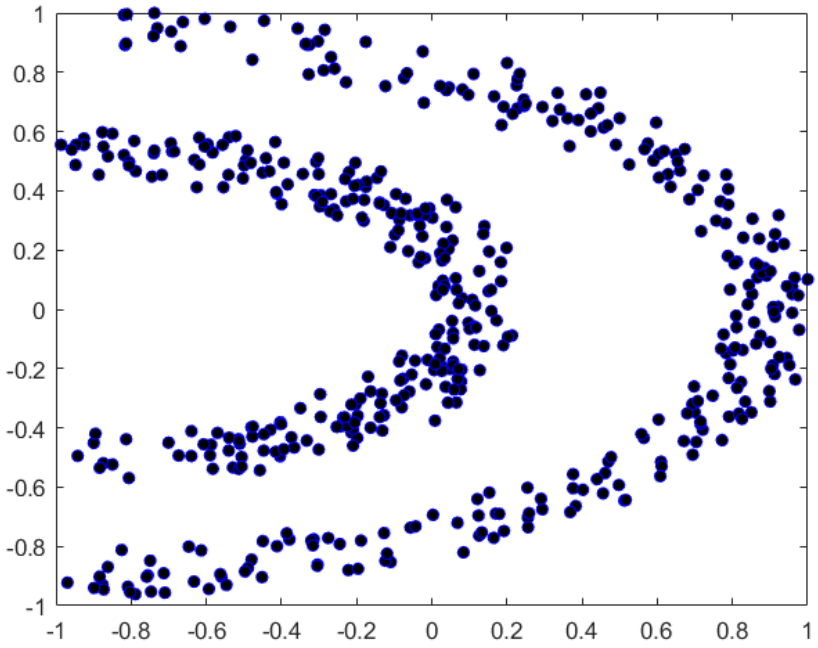

Figura 47 – CBG - Construção da Base Halfkernel.

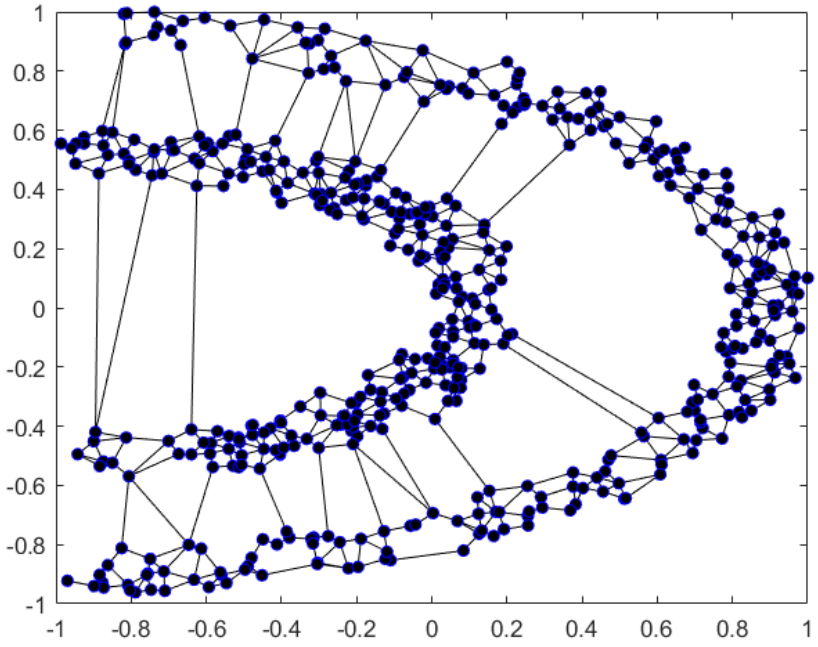

Figura 48 – CBG - Construção do Grafo a partir da Base Halfkernel gerada.

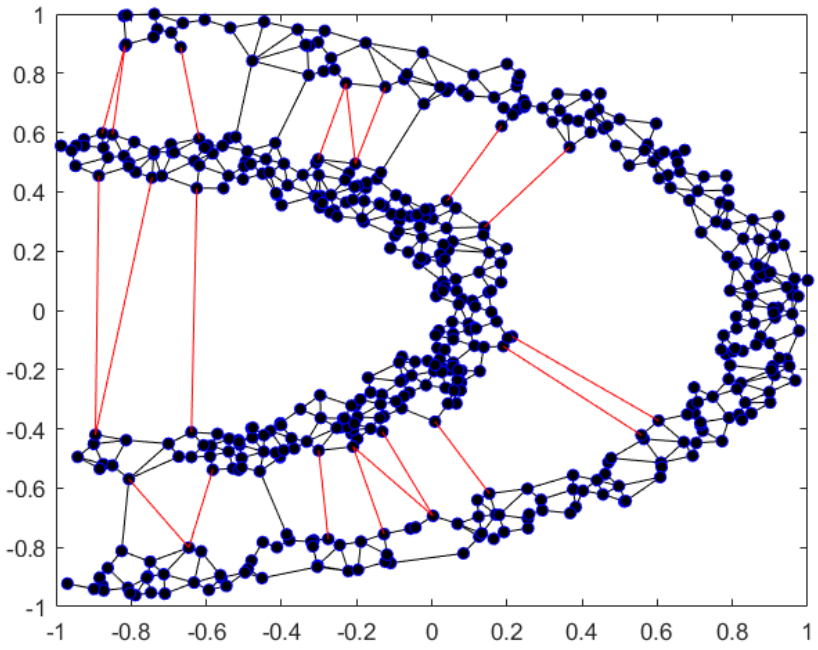

Figura 49 – CBG - Base Halfkernel - Definição do valor médio e marcação das Arestas de Suporte.

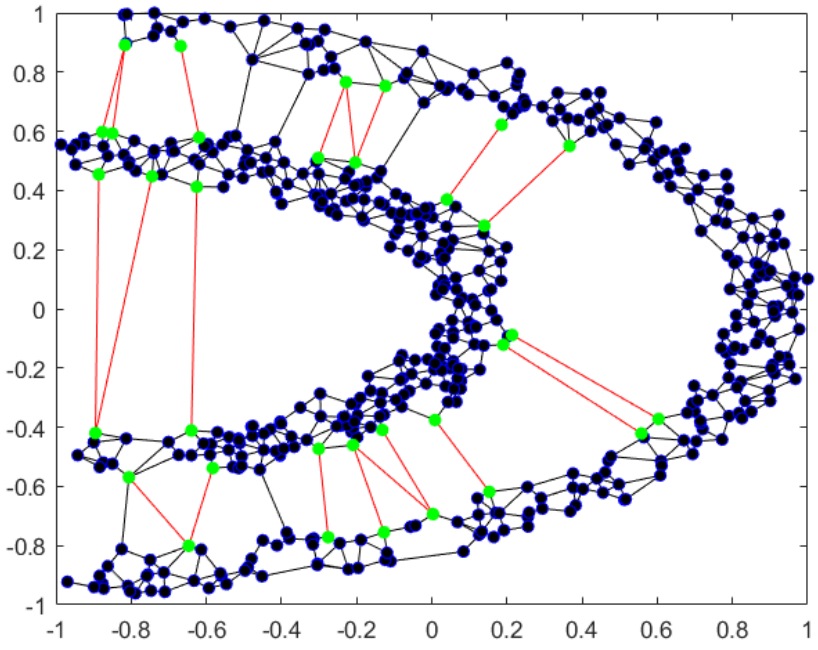

Figura 50 – CBG - Base Halfkernel - Localização das bordas pela aresta de suporte.

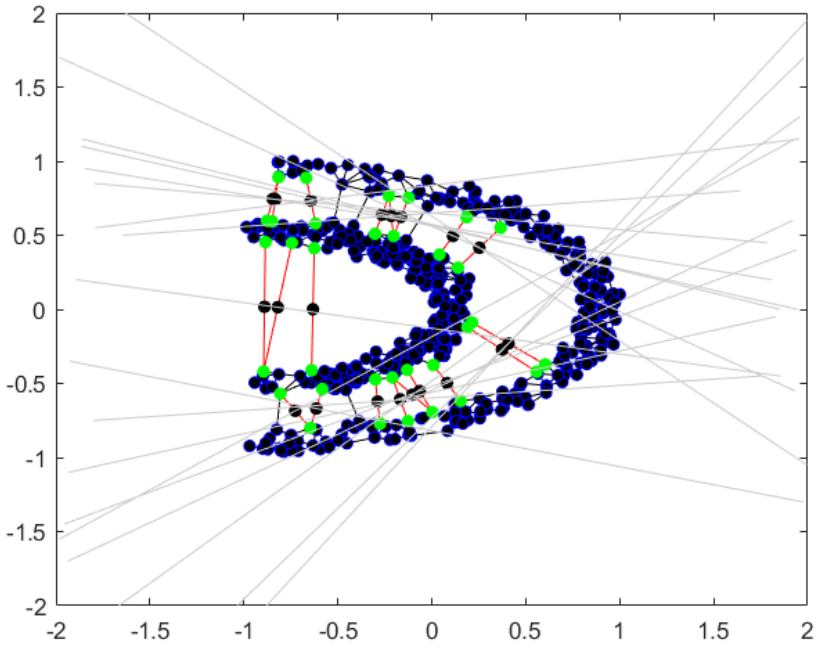

Figura 51 – CBG - Base Halfkernel - Marcação do ponto médio na aresta de suporte.