Universidade Federal do Espírito Santo

Centro de Ciências Exatas

Programa de Pós-Graduação em Matemática

Dissertação de Mestrado em Matemtica

Métodos de Projeção Multidimensional

Alcebíades Dal Col Júnior

Março/2013

Universidade Federal do Espírito Santo

Centro de Ciências Exatas

Programa de Pós-Graduação em Matemática

Métodos de Projeção Multidimensional

Alcebíades Dal Col Júnior

Dissertação apresentada ao Programa de Pós-Graduação em Matemática da Universidade Federal do Espírito Santo como parte dos requisitos necessários para a obtenção de grau de Mestre em Matemática.

Orientador: Fabiano Petronetto do Carmo

Março/2013

Agradecimentos

Agradeço em primeiro lugar a Deus. A minha família que sempre me apoiou e acreditou em mim, em especial agradeço a minha mãe que esteve ao meu lado em todos os momentos de minha formação. Agradeço também aos meus professores, grandes mestres que me instruiram e me deram a formação necessaria para trabalhar com grandes problemas, em especial destaco o Prof. Dr. Fabiano Petronetto do Carmo, sua orientação foi fundamental na minha jornada. Agradeço aos meus amigos pelo companheirismo. E finalmente agradeço o apoio financeiro concedido pela CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior).

Resumo

O problema que estamos interessados em resolver provém de uma área do conhecimento denominada visualização de dados. Nos nossos estudos, grupos de objetos são análisados para produzir os dados de entrada do nosso problema, cada um dos objetos é representado por atributos, temos assim uma lista de atributos para cada objeto. A ideia é representar, através dessas listas de atributos, os objetos através de pontos em $\mathbb{R}^2$ para que possamos realizar um estudo do grupo de objetos.

Como dissemos cada objeto é representado por uma lista de atributos, esta pode ser interpretada como um ponto de um espaço multidimensional. Por exemplo, se são considerados m atributos valorados para todos os objetos podemos interpretá-los como sendo pontos de um espaço de dimensão m, ou m-dimensional. Mas, queremos produzir uma visualização dos dados na tela do computador através de pontos em $\mathbb{R}^2$, realiza-se então um processo conhecido como projeção multidimensional, que é a obtenção de pontos em um espaço de baixa dimensão que represente pontos de um espaço de alta dimensão preservando relações de vizinhaça tanto quanto possível.

Diversos métodos de projeção multidimensional são encontrados na literatura. Neste trabalho, estudamos e implementamos os métodos NNP, Force, LSP, PLP e LAMP. Estes métodos abordam o problema de diferentes formas: geometricamente; sistemas lineares, em particular, sistemas laplacianos; e mapeamentos ortogonais afins.

As listas de atributos associadas aos grupos de objetos recebem o nome de conjuntos de dados. Dois dos conjuntos de dados abordados neste trabalho apresentam tendências de agrupamento conhecidas a priori, portanto foram utilizados para dar credibilidade as nossas implementações dos métodos. Outros dois conjuntos de dados são estudados e esses não eram dotados de tal caracteristica, os métodos de projeção multidimensional são então utilizados para definir tendências de agrupamento para esses dois conjuntos de dados.

Palavras-chave: projeção multidimensional.

Abstract

The problem we are interested in solving comes from a area of knowledge called data visualization. In our studies, groups of objects are analyzed to produce the input data of our problem, each object is represented by attributes, have so a list of attributes for each object. The idea is to represent, through these lists of attributes, objects through points in $\mathbb{R}^2$ so that we can conduct a group of objects.

As we said each object is represented by a list of attributes, this may be interpreted as a point of a multidimensional space. For example, if they are considered m valued attributes for all objects can interpret them as points in a space of dimension m, or m-dimensional. But we want to produce a visualization of the data on the computer screen through points in $\mathbb{R}^2$, it was then performs a process known as multidimensional projection, that is obtaining points in a low dimensional space representing points in a high dimensional space preserving neighborhood relations as much as possible.

Various methods of multidimensional projection are found in the literature. In this work, study and implement methods NNP, Force, LSP, PLP and LAMP. These methods deal with the problem in different ways: geometrically; linear systems, in particular, laplacian systems; and mappings related orthogonal.

The lists of attributes associated with the groups of objects are called dataset. Two sets of data in this paper present trends grouping known a priori, therefore were used to give credibility to our implementations of the methods. Two other data set are studied and these were not provided with such feature, the methods of multidimensional projection are then used to define trends grouping for these two data sets.

Keywords: projection.

Sumário

Introdução

Nosso estudo começa com um conjunto de objetos representado por atributos reais que nos interesse. Cada um desses objetos é então representado por uma lista de números. Por exemplo se estamos interessados em flores, podemos descrever cada uma de nossas flores através de uma lista de números contendo o comprimento e a largura das petálas e das sépalas das flores. Não estamos interessados nessa etapa propriamente dita, estamos interessados em trabalhar com as listas já prontas.

Deseja-se saber então se algum ou alguns dos objetos se assemelham aos demais devido às semelhanças presentes nas listas de números. Essa seria uma tarefa sem duvida complicada se o conjunto de objetos fossem descritos por listas muito grande de números ou até mesmo por uma quantidade de objetos muito grande. Com o intuito de facilitar essa análise foram desenvolvidos os métodos de projeção multidimensional.

Cada método de projeção multidimensional ataca esse problema com uma abordagem particular, mas todos tem um mesmo objetivo: representar essas listas de dados na tela do computador. Os objetos sendo representados por m atributos podem ser interpretados como pontos de um espaço de dimensão m, para representá-los na tela do computador é necessário que se faça uma projeção multidimensional obtendo assim pontos em $\mathbb{R}^2$, o que fazemos então é representar pontos m-dimensionais no plano, ou seja, através apenas de duas diemensões mantendo tanto quanto possível as relações de vizinhança, pois queremos justamente observar as relações de vizinhança dos pontos.

Os métodos de projeção multidimensional realizam a redução dimensional de diferentes maneiras, alguns deles se baseiam em ideias geométricas, trabalhando com interseções de círculos bidimensionais, outros utilizam sistemas laplacianos e um deles até uma família de mapeamentos ortogonais afins. Dentre os métodos de projeção dimensional existentes escolhemos trabalhar com 5 deles: NNP, Force, LSP, LAMP e PLP. Esses métodos são definidos no decorrer deste trabalho. Além das definições dos métodos, também fornecemos, após a definição de cada um dos métodos, o algoritmo utilizado para implementar cada um deles. Com os métodos implementados produzimos algumas projeções, as mesmas foram observadas e comparadas com o intuito de conhecermos melhor o comportamento de cada um desses métodos, a partir dessas comparações destacamos algumas vantagens, bem como, desvantagens dos métodos.

Alguns conceitos matemáticos são importantes na formulação dos métodos. Os métodos LSP, LAMP e PLP fazem uso de um subconjunto do conjunto de dados, chamado de conjunto de pontos de controle, esse conjunto deve representar o conjunto original da melhor forma possível. Para obter os pontos de controle foram aplicadas técnicas de particionamento aos conjuntos de dados.

Nas implementações dos métodos, foi utilizada a técnica de particionamento k-means. O método LAMP faz uso de uma decomposição em valores singulares (SVD) para construir uma família de mapeamentos ortogonais afins, e apartir dela realiza a projeção. A projeção desse método se configura naturalmente através de um somatório, com a utilização da norma de Frobenius ele se torna um problema matricial cuja solução já é conhecida. No método LSP criamos uma nova abordagem para a introdução de informações geometricas através dos pontos de controle, isto é, definimos outro meio de solução para o sistema gerado durante o processo de projeção realizado por esse método, na nossa abordagem acabamos por trabalhar com matrizes especias que são contrações. O método LSP original e o método PLP fazem uso de sistemas laplacianos para realizar a projeção e a solução desses sistemas é obtida através do método de mínimos quadrados, não sendo obtida diretamente.

Um conjunto de dados é constituído por listas de atributos reais. No 1º conjunto de dados estas listas são definidas a partir de algumas plantas do tipo íris, associa-se a cada planta uma lista de atributos, o conjunto de dados assim formado é conhecido como conjunto de dados Íris. Este é bastante conhecido e foi utilizado por nós para exemplificar cada um dos métodos, através de projeções e até mesmo de projeções parciais, possibilitando uma visualização da evolução do método.

Assim como o conjunto de dados Íris existem outros conjuntos de dados construidos de maneira semelhante. Os explorados neste trabalho são Wine, Housing e Abalone, que serão melhores apresentados no decorrer do texto.

Além do conhecimento obtido a cerca dos métodos também possibilitamos um conhecimento mais profundo dos conjuntos de dados, tanto com informações provenientes de seus criadores tanto como informações qualitativas obtidas a partir das projeções.

Em projeção multidimensional, quando estamos interessados em analisar resultados é complicado dizer quando uma projeção é boa ou ruim, existe uma tendência muito forte entre os pesquisadores que trabalham nessa área que é a utilização de uma medida quantitativa chamada stress e, mais recentemente, uma medida visual denominada gráfico de dispersão. Optamos por seguir essa tendência e utilizamos o stress e o gráfico de dirpersão para obtermos medidas comparativas para os métodos, levamos em conta também a qualidade do layout produzido utilizando como referência nosso conhecimento prévio do conjunto de dados, esse conhecimento foi obtido a partir de informações dadas pelo autor.

A dissertação está dividida em quatro capítulos como segue:

Capítulo 1: disponibiliza definições e ferramentas que são de importância fundamental na formulação dos métodos e, portanto, na compreensão da metodologia.

Capítulo 2: introduz o problema que estamos interessados em resolver. Nosso objetivo é obter informações qualitativas dos conjuntos de dados. Também apresenta os métodos de projeção multidimensional que escolhemos para resolver esse problema. Os métodos NNP, Force, LSP, PLP e LAMP são definidos e exemplificados utilizando para tal fim o conjunto de dados Íris.

Capítulo 3: apresenta algumas técnicas utilizadas na comparação de projeções de conjuntos de dados. Os resultados das projeções dos conjuntos de dados

Íris, Wine, Housing e Abalone são então apresentados e analisados para que possamos apontar vantagens e desvantagens de cada um dos métodos.

Capítulo 4: apresenta as conclusões obtidas e aponta os trabalhos futuros.

Capítulo 1

Conceitos Básicos

Alguns dos métodos de projeção apresentados neste trabalho fazem uso de um subconjunto do conjunto de dados para realizar a projeção multidimensional, existe então a necessidade de selecionar pontos a partir dos conjuntos de dados que os representam da melhor forma possível. Para esse fim, utilizamos alguns métodos bem conhecidos como o k-means e o k-medoids que pertencem a uma área conhecida como métodos de particionamento.

Das técnicas de projeção apresentadas uma realiza a redução dimensional resolvendo um problema de minimização. Através da norma de Frobenius esse problema é transformado em um problema de minimização matricial restrito a uma condição de ortogonalidade. O problema de minimização matricial obtido é bastante conhecido e tem uma solução muito simples se consideramos a decomposição em valores singulares de uma matriz dada. Para o leitor não familiarizado com essa decomposição e com essa norma as exploramos resumidadmente nesse capítulo.

Outra técnica de projeção se utiliza de sistemas laplacianos para realizar a projeção multidimensional, as soluções desses sistemas não são obtidas de uma forma direta e sim através do método de mínimos quadrados [1], uma vez que as matrizes associadas aos sistemas podem não ser quadradas ou até mesmo não ser invertíveis.

Com intuito de tornar essa técnica baseada em sistemas laplacianos iterativa, trabalhamos para alterar o sistema laplaciano, de modo que surgisse um sistema cuja matriz tem a propriedade de ser uma contração, com isso a solução do sistema pode ser dada em função de uma série de potências de uma matriz dada. Isso nos permitiu exibir um conjunto de pontos evoluindo para a solução do sistema, além disso permitimos a iteração do usuário enquanto esses pontos convergem para a solução, para darmos embasamento teórico a essa construção a seção matrizes com a propriedade de contração é fundamental.

1.1 Métodos de particionamento

Os métodos de particionamento tem como objetivo construir k partições de n objetos com $k << n$, onde cada partição representa um novo grupo formado. Os grupos juntamente com seus objetos devem estar condicionados aos seguintes princípios básicos:

1. Cada grupo deve conter pelo menos um objeto;
2. Cada objeto deve pertencer a um grupo.
3. Os grupos devem ser dois a dois disjuntos.

Métodos de particionamento agrupam os objetos de tal modo que uma função ou medida de erro seja minimizada, essa medida é calculada a partir dos desvios dos objetos em relação aos pontos que definem os centros de seus grupos, esses desvios podem ser computados por meio de uma medida de similaridade.

Um processo de realocação iterativo é então aplicado para obter o particionamento de dados. Considere um conjunto de n objetos e um número k de partições. São selecionados de maneira aleatória k objetos que definirão k grupos, os objetos restantes são associados a um desses grupos formados. Em cada grupo é feita uma busca por um novo ponto que melhor representem o grupo, isto é, diminua a soma dos desvios dos objetos aos centros de seus respectivos grupos, realoca-se os objetos nos grupos de acordo com esses novos representantes, é calculada então uma medida de erro para o novo agrupamento formado. Esse processo é repetido até que a medida de erro não possa mais ser reduzida.

Dada uma ideia geral do comportamento de um método de particionamento apresentamos a seguir alguns desses métodos.

1.1.1 Método k-means

O método k-means calcula a similaridade entre os objetos através da distância no espaço multidimensional definido pelos atributos dos objetos. Vale a pena destacar também que o número de grupos k é um parâmetro do método devendo assim ser informado pelo usuário. Em geral, definir o valor k para um conjunto de objetos é uma tarefa complicada, como esse valor deve ser informado pelo usuário nos deparamos com uma dificuldade na utilização desse método.

Para um melhor entendimento do que vem a seguir cabe a seguinte

Definição: O centroide ou centro de massa de um grupo de objetos é dado pelo valor médio dos objetos pertencentes a esse grupo.

O método k-means assim como os demais métodos de particionamento inicia seu algoritmo com a escolha aleatória de k objetos para representar os k grupos, onde cada um desses objetos é denominado centro do seu respectivo grupo, os objetos restantes são então associados aos grupos mais próximos. Os centróides dos grupos formados são calculados e definidos como novos centros, daí os objetos são realocados de acordo com os novos centros. Esse processo é repetido até que o critério de agrupamento convirja.

Usa-se em geral como critério de convergência dos grupos a soma dos erros quadráticos, dado por

$$E = \sum_{i=1}^{k} \sum_{p \in C_i} ||p - m_i||^2$$

onde p é o vetor que representa um objeto no espaço multidimensional.

Levando em conta esse critério, o método deixa de realocar os objetos quando encontra um mínimo local para o valor de E, ou seja, quando os objetos não mudam mais de grupos. Obter um valor mínimo para E é equivalente a obter todos os pontos m_i no centro de massa do conjunto C_i. De fato minimizar E significa minimizar

$$\sum_{i=1}^{k} \sum_{p \in C_i} ||p - m_i||^2 = \sum_{p \in C_1} ||p - m_1||^2 + \sum_{p \in C_2} ||p - m_2||^2 + ...$$

como cada $\sum_{p \in C_i} ||p - m_i||^2 \geq 0$, a expressão acima tem valor mínimo quando cada $\sum_{p \in C_i} ||p - m_i||^2$ é mínimo. Derivando esse último termo em relação a m_i e igualando a zero temos

$$\sum_{p \in C_i} (m_i - p) = 0$$

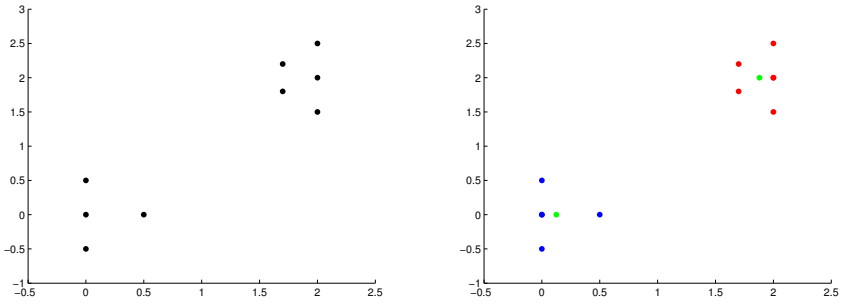
sendo $n_i = \#C_i$ obtemos

$$m_i = \frac{1}{n_i} \sum_{p \in C_i} p,$$

ou seja, m_i é o centro de massa do conjunto C_i.

Os centros de massa por sua vez podem não pertencer ao conjunto de dados, então para não incluir novos objetos ao grupo, é usado o pontos mais próximo do centroide para representar o grupo. Tal ponto é chamado *medoide*.

O método k-means apresenta bons resultados quando os grupos são compactos e bem separados uns dos outros. A Figura 1.1 dá uma demonstração dessa afirmação aplicando o método a um conjunto bidimensional e bastante simples (Figura 1.1(a)).



(a) Conjunto de dados

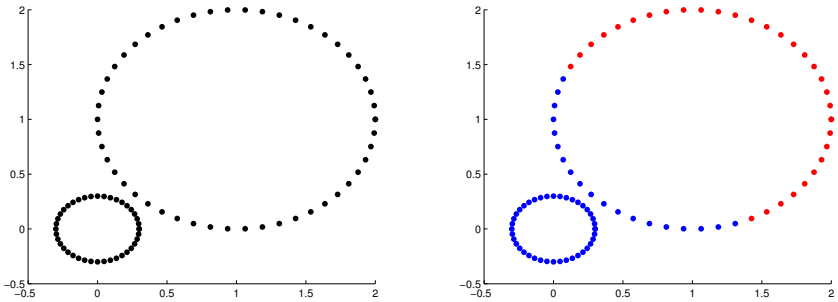
(b) k-means ($k = 2$)

Figura 1.1: Conjunto de dados com grupos compactos e bem separados

Os pontos foram coloridos conforme os grupos fornecidos pelo método, os pontos de um dos aglomerados receberam a cor azul e os pontos do outro a cor vermelha. Os pontos verdes representam os centroides dos algomerados, e os pontos do conjunto de dados mais póximos dos centroides são os medoides.

Utilizamos um conjunto de dados bidimensional para um melhor entendimento do método k-means. Alguns métodos de projeção multidimensional, porém, particionam conjuntos de dados para realizar a redução dimensional, aplica-se então os métodos de particionamento a esses conjuntos, que são em geral, m-dimensionais, com m consideravelmente grande.

Uma dificuldade enfrentada pelo método é lidar com descoberta de grupos com formato côncavo ou com grupos de tamanhos diferentes. Ao conjunto de dados Círculo-círculo, que aparece na Figura 1.2(a), aplicamos o método k-means com $k = 2$, a Figura 1.2(b) ilustra o particionado fornecido pelo método, esperavamos que um grupo fosse constituido por um dos círculos e o outro grupo pelo outro círculo, no entanto, isso não ocorre devido a diferença de tamanho dos grupos.



(a) Conjunto de dados: Círculo-círculo

(b) k-means ($k = 2$)

Figura 1.2: Conjunto de dados com grupos de tamanhos diferentes.

Outra desvantagem desse método é a sensibilidade à valores extremos, pois estes influenciam de maneira significativa a média. Afetando assim a posição dos centróides e a distribuição homogênea dos grupos.

Apesar das desvantagens, esse método é relativamente escalável e eficiente para o processamento de grandes conjuntos de dados. Podem ser encontradas algumas variações do método k-means, estas podem diferir na seleção das k partições iniciais, na medida de similaridade ou na estratégia do cálculo da média.

1.1.2 Método k-medoids

O principal objetivo do método k-medoids é apresentar uma menor sensibilidade à valores extremos, uma das desvantagens mais relevantes do método k-means. Com essa finalidade, esse método escolhe objetos representantes de cada grupo como sendo objetos presentes no próprio conjunto de dados, chamados de medoides.

Ao invés de utilizar a média dos objetos (como o método k-means) o método k-medoids utiliza esses medoides, em seguida os $n-k$ objetos restantes nos dados

são associados aos grupos de acordo com a sua similaridade com os medoids.

O restante do processo decorre de maneira semelhante ao k-means, diferindo apenas no critério de agrupamento que no caso do k-medoids consiste em minimizar a soma dos erros absolutos e não mais quadráticos. A soma dos erros absolutos é dada por

$$E = \sum_{i=1}^{k} \sum_{p \in C_i} ||p - o_i||$$

onde o_i é o objeto representativo ou medoide do grupo C_i e p é o vetor que representa um objeto no espaço multidimensional.

Os algoritmos PAM, CLARA e CLARANS são bastante conhecidos na literatura e também são bons exemplos de métodos que utilizam a abordagem k-medoids para encontrar k grupos através de objetos representativos (medoides).

Método PAM. O método PAM [4] (Partinioning Around Medoids) utiliza a abordagem do método k-medoids para definir k grupos. Damos agora uma descrição do método.

O algoritmo PAM inicia com a escolha aleatória de k medoides. Os objetos restantes são associados aos grupos de acordo com a sua similaridade com os medoides definidos na etapa anterior. Para todos pares de objetos formados por um medoide e por um objeto não-representativo é verificada a troca do medoide pelo objeto não-representativo. Cada objeto não-representativo, denotado por o_r, selecionado de um grupo i é confrontado com o medoide o_i de seu grupo, os valores E_r e E_i são calculados (os erros absolutos com o_r e o_i como medoides do grupo i), caso o valor de E_r seja menor que o valor de E_i, então o_i deixa de ser o medoide do grupo i e o_r assume seu lugar. Obtemos assim um novo conjunto de k medoides, os objetos não-representativos são realocados de acordo com a sua similaridade entre os novos medoides. O processo descrito acima é repetido até que nenhum medoide seja alterado.

Todos os passos dados durante a execução do método PAM podem ser vistos no Algoritmo 1.

Algorithm 1 Algoritmo PAM

Require: Conjunto de n objetos e número de partições k.
  Escolha aleatória de k medoides: $o_1, ..., o_k$
  while os medoids forem alterados **do**
    Definição dos grupos $C_i = \{o_i\}$
    Associar os elementos restantes aos grupos C_i
    for all grupo C_i **do**
      for all par de medoid o_i e objeto não-representativo o_r **do**
        Cálculo dos erros absolutos E_r e E_i
        Avaliar a troca do medoide pelo objeto não-representarivos
      end for
    end for
  end while

O PAM funciona muito bem para pequenos conjuntos de dados. Entretanto quando conjuntos de dados médios e grandes são levados em conta sua eficiência

é baixa. Justamente para suprir essa deficiência do método PAM foi criado o método CLARA que em vez de procurar medoides em todo o conjunto de dados investiga amostras selecionadas do conjunto de dados, a busca por medoides nas amostras é feita através do algoritmo PAM. Finalmente, assim como o CLARA o método CLARANS utiliza a abordagem do método PAM em amostras do conjunto de dados, entretanto as amostras são escolhidas dinamicamente a partir do conjunto de dados durante a execução desse método, isto é, as amostras podem ser alteradas durante a execução do método, o que não acontece no método CLARA onde essas amostras são definidas a priori.

Os métodos PAM, CLARA e CLARANS apresentam um grande potencial para a seleção de pontos a partir de um conjunto de dados. No entanto, não os utilizamos, optamos pelo método k-means, pois estamos interessados em comparar os métodos de projeção multidimensional e o método k-means está presente na rotina original de um dos métodos.

1.2 Soluções de sistemas lineares no sentido de mínimos quadrados

É comum que em problemas concretos ocorram "erros de medida" nas entradas de uma matriz A e de um vetor b que definem um sistema linear $Ax = b$, perturbando-o a ponto de torná-lo inconsistente. Em tais situações procuramos um valor de x que seja tão "próximo" de uma solução para o sistema linear quanto possível, no sentido que minimize a expressão $||Ax - b||$ em relação ao produto interno euclidiano. O valor $||Ax - b||$ pode ser encarado como uma medida de "erro" que resulta do fato de considerar x como uma solução aproximada para o sistema $Ax = b$. Se estamos lidando com um sistema consistente e x é uma solução exata, então o erro é zero, pois $||Ax - b|| = 0$. Na verdade, em boa parte dos casos quanto maior o valor de $||Ax - b||$, mais a aproximação x se distancia de uma solução do sistema.

Estamos preocupados então em resolver o seguinte

Problema: Dado um sistema linear $Ax = b$ encontrar, se possível, um vetor x que minimize $||Ax-b||$ em relação ao produto interno euclidiano. Um tal vetor é chamado uma solução de mínimos quadrados de $Ax = b$.

Considerando que o sistema $Ax = b$ é constituido de m equações e n variáveis se torna bem simples entender a origem do termo mínimos quadrados. Seja $e = Ax - b$ (o vetor erro), sendo $e = (e_1, e_2, ..., e_m)$ temos que minimizar $||e|| = (e_1^2 + e_2^2 + ... + e_m^2)^{1/2}$ o que é equivalente a minimizar

$$||e||^2 = e_1^2 + e_2^2 + ... + e_m^2,$$

daí o termo mínimos quadrados.

Para resolver esse problema começa-se considerando W como sendo o espaço coluna da matriz A. O produto Ax é uma combinação linear dos vetores coluna de A, para cada matriz x de tamanho $n \times 1$. Portando à medida que o vetor x varia no espaço R^n, o vetor Ax varia no espaço coluna W. Resolver o nosso problema nada mais é do que encontrar um vetor x de R^n tal que Ax é o vetor

de W que mais se aproxima de b. Uma ideia geométrica dessa descrição é dada na Figura 1.3.

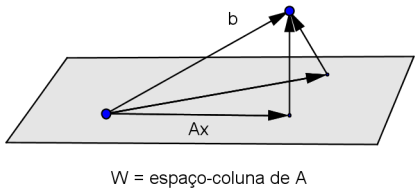


Figura 1.3: Interpretação geométrica do problema de mínimos quadrados

Nos deparamos assim com um problema bastante conhecido, encontrar um vetor Ax em um espaço W que mais se aproxima de um vetor dado b, cuja solução é a projeção ortogonal de b sobre W, ou seja,

$$Ax = \text{proj}_W(b)$$

Sabemos também que $b - Ax = b - \text{proj}_W(b)$ é ortogonal ao espaço W, além disso como W é o espaço coluna de A, temos que $b - Ax$ está no espaço nulo de A^T. Assim x deve satisfazer

$$A^T(b - Ax) = 0$$

ou, equivalentemente,

$$A^T Ax = A^T b$$

Este sistema é chamado sistema normal associado a $Ax = b$ e as equações que o compõem são chamadas equações normais associadas a $Ax = b$. O problema de encontrar uma solução aproximada para $Ax = b$ foi reduzido a encontrar uma solução exata do sistema normal associado $A^T Ax = A^T b$. Note que

1. A matriz $A^T A$ é uma matriz quadrada $n \times n$.
2. O sistema normal é consistente, pois é satisfeito por uma solução de mínimos quadrados de $Ax = b$.

Finalmente, se a matriz $A^T A$ é invertível o sistema linear $Ax = b$ terá uma única solução de mínimos quadrados, dada por

$$x = (A^T A)^{-1} A^T b.$$

Um resultado que simplifica a tarefa de determinar se $A^T A$ é invertível, é o Teorema 1.2.1.

Teorema 1.2.1 *Se A é uma matriz $m \times n$, então as seguintes afirmações são equivalentes.*

1. *Os vetores colunas de A são linearmente independentes.*
2. *$A^T A$ é invertível*

Demonstração: Vamos mostrar apenas que $(a) \Rightarrow (b)$. Suponha que os vetores colunas de A sejam linearmente independentes. A matriz A^TA é uma matriz quadrada $n \times n$, podemos assim provar que ela é invertível mostrando que o sistema linear $A^TAx = 0$ tem somente a solução trivial. De fato, seja x uma solução qualquer desse sistema, então $A^T(Ax) = 0$, logo o vetor Ax está no espaço nulo de A^T, mas por outro lado Ax é uma combinação linear dos vetores coluna de A, daí pertence ao espaço coluna de A. Estes dois espaço por sua vez são ortogonais, e contém o vetor Ax, portanto necessariamente $Ax = 0$, sendo as colunas de A linearmente independentes resulta em $x = 0$. ■

1.3 Decomposição em Valores Singulares

A decomposição em valores singulares ou SVD (singular value decomposition) é uma decomposição de uma matriz real ou complexa $A_{m\times n}$ que tem a forma:

$$A = U\Sigma V^*,$$

onde U é uma matriz unitária $m \times m$, Σ é uma matriz retangular $m \times n$ com números reais não-negativos na diagonal, V é uma matriz unitária $n \times n$ e V^* denota a matriz conjugada transposta de V.

As entradas diagonais σ_{ii} da matriz $\Sigma = (\sigma_{ij})_{m\times n}$ são chamados valores singulares de A. As colunas de U são chamados vetores singulares à esquerda e as colunas de V vetores singulares à direita de A. Esses apresentam as seguintes propriedades:

1. Os vetores singulares à direita de A são autovetores de AA^*.
2. Os vetores singulares à esquerda de A são autovetores de A^*A.
3. Os valores singulares não-nulos de A (σ_{ii}) são as raízes dos autovalores não-nulos (σ_{ii}^2) das matrizes A^*A e AA^*.

Muitas aplicações na algebra linear involvem a SVD como a determinação do posto, imagem e núcleo de uma matriz, cálculo da pseudoinversa e aproximação de matrizes aplicações em outras áreas também são comuns como o ajuste de dados por mínimos quadrados.

1.4 Matrizes com a propriedade de contração

Dedicamos esta seção ao estudo de uma propriedade bastante interessante que algumas matrizes possuem. Uma matriz é uma contração quando satisfaz a seguinte

Definição: W é dita ser uma contração quando $||W|| < 1$.

O Teorema 1.4.1 mostra como podemos calcular $(I - W)^{-1}$ em função das pontências de W $(I, W, W^1, W^2, ...)$, quando W é uma matriz invertível.

Teorema 1.4.1 *Seja W uma matriz $n \times n$ invertível. Se W é uma contração então*

$$(I - W)^{-1} = \sum_{i=0}^{\infty} W^i$$

Demonstração: Observe inicialmente que

$$(I - W)\sum_{i=0}^{n} W^i = I - W^{n+1}$$

considerando o limite quando $n \to \infty$, temos

$$\begin{aligned} \lim_{n\to\infty}((I - W)\sum_{i=0}^{n} W^i) &= \lim_{n\to\infty}(I - W^{n+1}) \\ \Rightarrow (I - W)\lim_{n\to\infty}(\sum_{i=0}^{n} W^i) &= I - \lim_{n\to\infty} W^{n+1} \end{aligned} \tag{1.1}$$

Agora sendo W uma contração a sequencia $(W^n)_{n\in N}$ tende a matriz nula quando $n \to \infty$, ou seja,

$$\lim_{n\to\infty} W^n = 0$$

utilizando esse fato na equação 1.1, obtemos

$$(I - W)\lim_{n\to\infty}(\sum_{i=0}^{n} W^i) = I.$$

Finalmente,

$$(I - W)^{-1} = \lim_{n\to\infty}\sum_{i=0}^{n} W^i = \sum_{i=0}^{\infty} W^i$$

■

1.5 Matrizes especiais e norma de Frobenius

Matriz unitária. Dada uma matriz complexa $U_{n\times n}$, dizemos que U é uma matriz unitária se satisfaz a seguinte condição

$$U^*U = UU^* = I_n$$

onde I_n é a matriz identidade $n \times n$ e U^* é o conjugado transposto de U.

Note que esta condição garante que a matriz U é unitária se, e somente se, tem uma inversa e essa inversa é igual a seu conjugado transposto, ou seja,

$$U^{-1} = U^*$$

Matriz unitária acaba por ser uma generalização de matriz ortogonal para os números complexos, pois se todos os valores de uma matriz unitária são reais ela nada mais é do que uma matriz ortogonal. Além disso, assim como as matrizes ortogonais, as matrizes unitárias tem a propriedade de preservar o produto interno,

$$< Ux, Uy >=< x, y >$$

para todos os vetores complexos x e y e onde $< ., . >$ é o produto interno sobre C^n.

Considerando que U é uma matriz unitária $n \times n$ algumas proprieades são conhecidas, as que merecem destaque nesse texto estão presentes na equivalência das seguintes condições:

1. U é unitária
2. U^* é unitária
3. as colunas de U formam uma base ortogonal de $\mathbb{C}^n$

Conjugada transposta de uma matriz. Seja $A = (a_{ij})$ uma matriz complexa $m \times n$, o conjugado transposto de A é a matriz A^* de tamanho $n \times m$ obtida de A tomando a transposta e então considerando o conjugado complexo de cada entrada. Formalmente a definição fica sendo

$$(A^*)_{ij} = \overline{a_{ji}}$$

onde $i = 1, 2, ..., n$, $j = 1, 2, ..., m$ e a barra denota o conjugado complexo escalar.

O conjugado transposto também é conhecido como transposto Hermitiano ou matriz adjunta, que com a definição acima pode ser escrita como

$$A^* = (\overline{A})^T = \overline{A^T}$$

onde A^T denota a transposta de A e $\overline{A}$ é a matriz obtida de A por conjugar cada entrada.

A notação A^H também é comum para designar o conjugado transposto da matriz A, o termo H aparece em função do outro nome que a operação leva transposto Hermitiano.

Norma de Frobenius. Uma norma matricial $||.||$ é uma função que associa a cada matriz um número real não negativo, satisfazendo as seguintes propriedades

1. $||A|| = 0 \Leftrightarrow A = 0$
2. $||\lambda A|| = |\lambda||A||, \lambda \in \mathbb{R}^2$
3. $||A + B|| \leq ||A|| + ||B||$

A norma de Frobenius, também conhecida como norma Euclidiana (podendo causar confusão com a norma vetorial L^2 que às vezes também é chamada de norma Euclidiana), é uma norma matricial definida como a raiz quadrada da soma dos quadrados absolutos de seus elementos (veja em [8]).

Seja $A = (a_{ij})$ uma matriz $m \times n$ a norma de Frobenius de A é definida por

$$||A||_F = \sqrt{\sum_{i=1}^{m} \sum_{j=1}^{n} a_{ij}^2}$$

A norma de Frobenius de A é também igual à raiz quadrada do traço de AA^*, onde A^* é a conjugada transposta de A, isto é,

$$||A||_F = \sqrt{Tr(AA^*)}.$$

Capítulo 2

Métodos de Projeção Multidimensional

Antes de passarmos para os métodos de projeção multidimensional, damos uma idea do problema que estamos interessados em resolver, a formulação do nosso problema é bastante simples.

2.1 Formulação do problema

Hoje em dia é cada vez mais comum lidarmos com fenômenos complexos descritos por várias variáveis, um exemplo bem simples é a previsão do tempo e o entendimento dos dados envolvidos em uma tarefa deste tipo é complicada na maior parte das vezes. Dados, em geral, são objetos em um espaço m-dimensional, onde m é o número de variáveis envolvidas no problema se $m \in \{1, 2\}$, então muitos métodos estão disponíveis para a compreenção dos dados, no entanto, quando m é significativamente grande a compreensão do fenômeno a partir das inúmeras variáveis, que o descrevem, torna-se muito difícil.

Para enfrentar tal desafio desenvolveu-se uma série de técnicas, denominadas projeções multidimensionais (MP), cujo principal objetivo é representar os dados m-dimensionais em um espaço de baixa dimensão (em geral $\mathbb{R}^2$) de modo a preservar tanto quanto possível a relação de vizinhança presente no espaço original. Com isso, um pesquisador pode ganhar alguma intuição sobre os dados, como por exemplo tendências de agrupamento.

Uma questão que surge naturalmente é como definir então relações de vizinhanças entre os dados, os métodos MP provém de uma classe maior de técnicas chamada escalonamento multidimensional (MDS [2]). Os métodos MDS usam medidas de distâncias (chamadas também de dissimilaridade) entres os dados para definir essas tais relações e com isso incorporar os dados no espaço de baixa dimensão.

É muito comum que exista apenas coordenadas cartesianas para os conjutos de dados fornecidos, desse modo pode-se usar a distância euclidiana para gerar uma medida de distância entre os dados, conhecida como dissimilaridades, recurso que utilizamos em todos os conjuntos de dados, contudo vale a pena ressaltar que existem situações onde a distância euclidiana faz pouco sentido,

levando em conta o conjunto de dados ao qual o método está sendo aplicado. Um bom exemplo desse tipo de conjunto de dados é o conjunto de dados Housing que é melhor apresentado posteriormente.

Em termos mais numéricos temos a seguinte formulação: Dado um conjunto de dados $S = \{p_1, ..., p_n\}$ em R^m, com uma medida de dissimilaridade $\delta(p_i, p_j)$ entre dois dados quaisquer p_i e p_j do espaço m-dimensional, e seja P um conjunto de pontos no espaço de baixa dimensão (ou espaço visual). Uma técnica de projeção multidimensional é uma função bijetiva $f : S \to P$ que procura tornar $|\delta(p_i, p_j) - d(f(p_i), f(p_j))|$ tão próximo de zero quanto possível, $\forall p_i, p_j \in S$, onde d se refere a distância euclidiana. O ponto $f(p_i)$ é chamado de projeção do ponto p_i. Às vezes representamos a projeção $f(p_i)$ de um ponto p_i simplesmente por p'_i.

Com o objetivo de tornar a situação acima mais familiar ao leitor, apresentamos um conjunto de dados bem simples e razoavelmente pequeno, denomidado Íris. O conjunto de dados Íris é constituido de 150 íris. Íris é um genero de plantas com flores, muito apreciado por suas diversas espécies, que ostentam flores de cores muito vivas.

Figura 2.1: Íris

O criador R.A.Fisher fornece uma lista com características das 150 plantas, para cada planta é informado as seguintes medidas da flor, em centímetros

1. Comprimento da sépala
2. Largura da sépala
3. Comprimento da pétala
4. Largura da pétala

Por exemplo a primeira planta é representada pela lista

$$p_1 \;=\; 5.1 \;\; 3.5 \;\; 1.4 \;\; 0.2$$

ou seja, a sépala da primeira planta tem 5.1 cm de comprimento e 3.5 cm de largura e sua pétala tem 1.4 cm de comprimento e 0.2 cm de largura.

Nessas circunstâncias, temos o número de dados $n = 150$, a dimensão dos dados (ou número de atributos envolvidas) $m = 4$. E ainda podemos considerar a medida de dissimilaridade δ como sendo a distância euclidiana. Isto é, sendo

$$p_2 \;=\; 4.9 \;\; 3.0 \;\; 1.4 \;\; 0.2$$

teriamos por exemplo

$$\delta(p_1, p_2) = \sqrt{(5.1 - 4.9)^2 + (3.5 - 3.0)^2 + (1.4 - 1.4)^2 + (0.2 - 0.2)^2} \approx 0.5385.$$

Uma característica bem conhecida desse conjunto de dados é a presença de dados de três espécies diferentes de plantas do tipo íris (Figura 2.2), são elas

1. Íris Versicolor
2. Íris Virginica
3. Íris Setosa

Figura 2.2: As três espécies de plantas do tipo íris presentes no conjunto de dados Íris. Da esquerda para a direita Íris-versicolor, Íris-virginica e Íris-setosa.

Fisher afirma que uma espécie é linearmente separada das outras 2, e estas últimas não são separadas linearmente de cada outra. Nos preocupamos então em resolver o seguinte

Problema: apartir da lista com as características das plantas produzir uma representação em $\mathbb{R}^2$ (layout), que revele os três grupos com suas tendências de agrupamento.

Com isso o nosso problema fica bem definido, e ilustrado. Passamos agora as técnicas de projeção multidimensional, que transformam dados m-dimensionais em pontos do espaço $\mathbb{R}^2$, para possibilitar a sua visualização.

2.2 Método NNP

O método de projeção de vizinhos mais próximos, do inglês Nearest Neighbor Projection (NNP) [20] tem como objetivo atacar o problema formulado anteriormente, realizar a redução dimensional e possibilitar uma visualização dos dados na tela do computador.

Para determinar a posição de um novo ponto no espaço visual o esquema NNP usa as posições (em $\mathbb{R}^2$) dos dois vizinhos mais próximos desse ponto (em R^m) dentre todos os pontos que já foram projetados nas etapas anteriores. É necessário uma etapa inicial, que consiste em projetar dois pontos de tal forma que a distância entre eles e a distância entre suas projeções sejam iguais, isto é, a distância entre as projeções desses dois pontos no espaço visual deve ser igual a distância entre os mesmos no espaço m-dimensional original.

Método NNP: seja $\tilde{S} \subset S$ o conjunto de pontos em S que já foram projetados, ou seja, já possuem suas coordenadas em $\mathbb{R}^2$. Seja $p \in S$ um novo ponto a ser projetado e q e r os dois pontos em $\tilde{S}$ mais próximos de p, consideramos então dois círculos C_q e C_r com centros em $q' = f(q)$ e $r' = f(r)$ e raios iguas a $\delta(p, q)$ e $\delta(p, r)$, respectivamente.

Os círculos C_q e C_r definidos na projeção de cada novo ponto podem ser tangentes, podem apresentar dois pontos de interseção, ou até mesmo não possuirem pontos de interseção. A posição da projeção p' de p no espaço visual depende de qual das três possibilidades ocorrem:

1ºcaso: Círculos tangentes (Figura 2.3).

A posição de $p' = f(p)$ no espaço visual é dada pelo ponto de tangência.

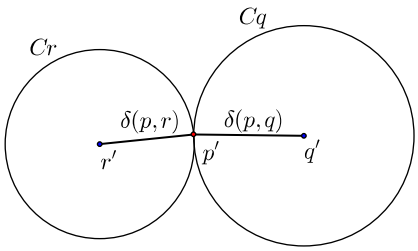


Figura 2.3: Os círculos C_q e C_r são tangentes. C_q e C_r são círculos com centros em q' e r' e raios $\delta(p, q)$ e $\delta(p, r)$, respectivamente. O ponto $p' = f(p)$ é a projeção do ponto p.

2ºcaso: Os círculos se intersectam em dois pontos (Figura 2.4).

A posição de $p' = f(p)$ no espaço visual é dada por um dos pontos de interseção, escolhido aleatoriamente.

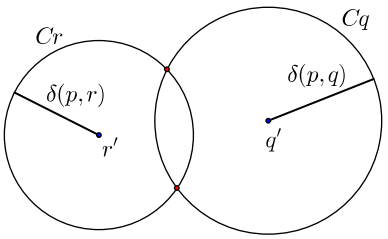


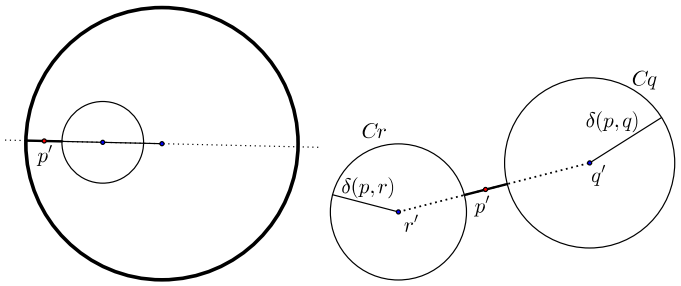
Figura 2.4: Os círculos C_q e C_r se intersectam em dois pontos. C_q e C_r são círculos com centros em q' e r' e raios $\delta(p,q)$ e $\delta(p,r)$, respectivamente. O ponto $p' = f(p)$ é a projeção do ponto p.

3ºcaso: Não há interseção entre os círculos (Figura 2.5).

A posição de $p' = f(p)$ no espaço visual é dada por um ponto intermediário determinado pelos raios dos dois círculos.

(a) Um círculo está contido no outro (Figura 2.5(a)).
Considera-se o segmento de reta partindo do centro do círculo maior passando pelo centro do círculo menor e de comprimento igual ao raio do círculo maior, subtrai-se a partir do centro do círculo maior a distância entre os centros e o raio do círculo menor. O ponto p' é tomado como sendo o ponto médio do segmento restante.

(b) Nenhum dos círculos contém o outro (Figura 2.5(b)).
Considera-se o segmento de reta ligando os centro, e a partir de cada um deles subtrai-se, o respectivo raio. O ponto p' é tomado como sendo o ponto médio do segmento de reta restante.



(a) Um dos círculos, C_q ou C_r, está contido no outro.

(b) Nenhum dos círculos contém o outro.

Figura 2.5: Não há interseção entre os círculos C_q e C_r. C_q e C_r são círculos com centros em q' e r' e raios $\delta(p,q)$ e $\delta(p,r)$, respectivamente. O ponto $p' = f(p)$ é a projeção do ponto p.

No primeiro caso os círculos C_q e C_r apresentam somente um ponto de interseção, nesse caso a tarefa de definir a projeção p' de p é bem simples. No entando, dificilmente observa-se o caso de círculos tangentes durante o processo de projeção.

Já no segundo caso os círculos C_q e C_r apresentam dois pontos de interseção, precisamos então optar por um deles, a escolha é feita aleatoriamente, isso faz com que o resultado de uma projeção seja imprevisível. Na escolha de um ponto de interseção para os círculos, dois pontos geometricamente distintos estão disponíveis, então a projeção pode seguir duas orientações espaciais diferentes durante a projeção do dado em questão, de forma que um mesmo conjuto de dados pode apresentar diferentes projeções, onde cada uma respeita bem as relações de vizinhança presentes no espaço multidimensional. Este tipo de interseção entre o círculos C_q e C_r é o que predomina no processo de projeção. Uma alternativa diferente para a escolha do ponto de interseção é abordada por Pekalska [18], conhecido como Triangulação [13], este método utiliza um terceiro ponto para decidir qual dos pontos de interseção será utilizado.

Por fim, no terceiro caso nenhum ponto de interseção é obtido, o novo ponto projetado p' é então escolhido de forma que sua distância a q' e r' seja ponderada pelas distâncias $\delta(p,q)$ e $\delta(p,r)$, respectivamente. Esse tipo de situação ocorre em geral quando os pontos q e r estão muito próximos no espaço visual.

Um bom resumo do esquema de projeção de vizinhos mais próximos (NNP) é apresentado no Algoritmo 2. Ao rodar o Algoritmo NNP várias vezes obtemos projeções muito distintas uma das outras, isso é devido a aleatoriedade introduzida ao se escolher um dos pontos de interseção quando os círculos C_q e C_r se intersectam em dois pontos.

Algorithm 2 Algoritmo NNP

```
Projetar os dois primeiros pontos tal que a distância entre eles no espaço
visual é igual a sua distância no espaço m-dimensional
for all ponto de dados p do
  Realizar uma pergunta k vizinhos mais próximos (kNN) no subconjunto
  dos pontos projetados S̃, retornando os dois vizinhos mais próximos q e r
  Encontrar os pontos de intersação dos circulos C_q e C_r com centro nos
  pontos q' = f(q) e r' = f(r) e raios iguais a d(p,q) e d(p,r), respectivamente
  if não existir ponto de interseção then
    if um circulo contém o outro then
      Coloque o ponto de interseção como na Figura 2.5(a)
    else
      Coloque o ponto de interseção como na Figura 2.5(b)
    end if
  end if
  if existir um ponto de interseção then
    Coloque o novo ponto projetado como sendo o ponto de interseção
  end if
  if existem dois pontos de interseção then
    Escolha aleatoriamente um dos pontos de interseção e coloque-o como
    sendo o novo ponto projetado
  end if
end for
```

A fim de tornar o funcionamento do método NNP um pouco mais claro, dedicamos o próximo tópico a análise de uma projeção particular do conjunto de dados Íris.

Exemplificando o método NNP

Começamos a projeção a partir dos dois primeiros dados presentes no conjunto de dados. O primeiro ponto $p'_1 = f(p_1)$ é colocado na origem, em seguida o segundo ponto $p'_2 = f(p_2)$ é posto sobre o eixo-y, de modo que a distância entre p'_1 e p'_2 seja igual à distância original (isto é, $d(p'_1, p'_2) = \delta(p_1, p_2)$). Sabemos que $p_1 = (5.1, 3.5, 1.4, 0.2)$ e $p_2 = (4.9, 3.0, 1.4, 0.2)$, logo

$$\delta(p_1, p_2) = \sqrt{(5.1 - 4.9)^2 + (3.5 - 3.0)^2 + (1.4 - 1.4)^2 + (0.2 - 0.2)^2} \approx 0.54.$$

Como $p'_1 = (0, 0)$ e $p'_2 = (0, y_2)$, com $y_2 > 0$ temos que

$$d(p'_1, p'_2) = \sqrt{(0 - 0)^2 + (0 - y_2)^2} = y_2.$$

Então teremos $y_2 = d(p'_1, p'_2) = \delta(p_1, p_2) \approx 0.54$, ou seja, $p'_2 = (0, 0.54)$. A Figura 2.6 mostra a projeção dos pontos p_1 e p_2, o ponto p'_1 conhecidindo com a origem e o ponto $p'_2 = (0, 0.54)$ sobre o eixo-y, note que a primeira coordenada de p'_2 é nula.

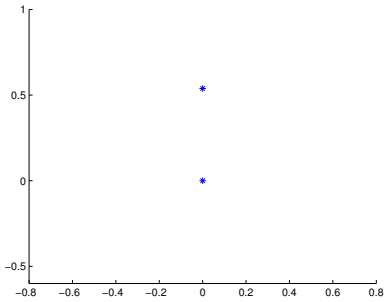


Figura 2.6: Projeção dos pontos p_1 e p_2 do conjunto de dados Íris. O ponto p'_1 é colocado na origem, enquanto que o ponto p'_2 é colocado sobre o eixo-y, com ordenada $\delta(p_1, p_2)$.

Seguindo a ordem dos dados passamos ao ponto $p_3 = (4.7, 3.2, 1.3, 0.2)$, nesse momento o conjunto dos pontos já projetados é $\tilde{S} = \{p_1, p_2\}$, daí os dois pontos em $\tilde{S}$ mais próximos de p_3 são

$$q = p_1 \text{ e } r = p_2$$

Agora precisamos encontrar a interseção dos círculos bidimensionais C_q e C_r com centro nos pontos $p'_1 = q'$ e $p'_2 = r'$ e raios iguais a $\delta(p_3, p_1)$ e $\delta(p_3, p_2)$, respectivamente. Efetuando alguns cálculos obtemos

$$\delta(p_3, p_1) \approx 0.51 \text{ e } \delta(p_3, p_2) = 0.3$$

São encontrados dois pontos de interseção, um dos dois pontos de interseção é escolhido aleatoriamente como sendo o ponto p'_3 (Figura 2.7). Nesse momento,

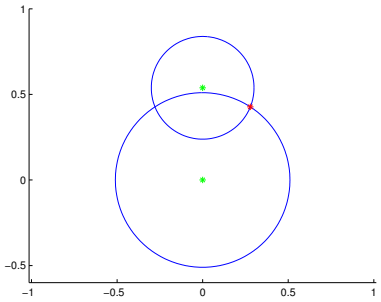


Figura 2.7: Projeção do ponto p_3 do conjunto de dados Íris. Os dois pontos mais próximos de p_3 pertencentes ao conjunto $\tilde{S} = \{p_1, p_2\}$ são trivialmente determinados, sendo $q = p_1$ e $r = p_2$. O ponto p'_3 é escolhido entre os dois pontos de interseção dos cíırculos C_q e C_r.

o outro ponto de interseção poderia ter sido escolhido, isso resultaria em uma projeção diferente.

Passamos ao quarto ponto $p_4 = \{4.6, 3.1, 1.5, 0.2\}$, o conjunto de pontos projetados é $\tilde{S} = \{p_1, p_2, p_3\}$, precisamos definir nesse caso quais os dois dados mais próximos a p_4 no espaço m-dimensional. Calculamos então as distâncias desses pontos p_1, p_2, p_3 ao ponto p_4,

$$\delta(p_4, p_1) \approx 0.65,\ \delta(p_4, p_2) \approx 0.33 \text{ e } \delta(p_4, p_3) = 0.2$$

assim,

$$q = p_2 \text{ e } r = p_3$$

Novamente precisamos encontrar a interseção dos círculos C_q e C_r, dessa vez com centro nos pontos $p'_2 = q'$ e $p'_3 = r'$ e raios iguais a $\delta(p_4, p_2)$ e $\delta(p_4, p_3)$, respectivamente. Então um dos dois pontos de interção encontrados são definidos aleatoriamente como sendo o ponto p'_4 (Figura 2.8).

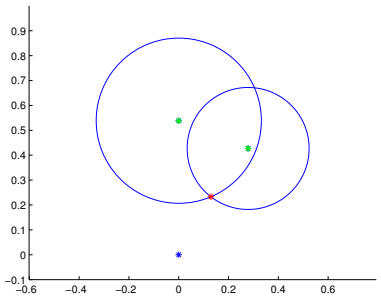


Figura 2.8: Projeção do ponto p_4 do conjunto de dados Íris. Os dois pontos mais próximos de p_4 pertencentes ao conjunto $\tilde{S} = \{p_1, p_2, p_3\}$ são $q = p_2$ e $r = p_3$. O ponto p'_4 é escolhido entre os dois pontos de interseção dos cíırculos C_q e C_r.

Na busca pelos pontos de interção dos círculos bidimensionais C_q e C_r podem aparecer também tanto um como nenhum ponto de interseção. O caso em que C_q e C_r são tangentes é raro e nem aparece nessa projeção. Contudo círculos disjuntos aparecem ao se projetar alguns pontos, por exemplo o ponto p_{12}, veja na Figura 2.9 o instante em que esse ponto é projetado.

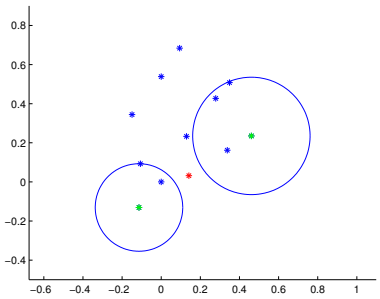


Figura 2.9: Projeção do ponto p_{12}. Os círculos C_q e C_r são disjuntos.

Durante o processo de projeção também encontramos círculos contidos uns nos outros, com uma frequência razoável (Figura 2.10). Note que os pontos q' e r', representados pela cor verde, estão próximos possibilitando esse tipo de situação. Os valores $\delta(p, q)$ e $\delta(p, r)$ calculados na projeção de cada novo ponto p são tipicamente próximos, portanto, essa situação tende a ocorrer somente quando os pontos q' e r' estão próximos.

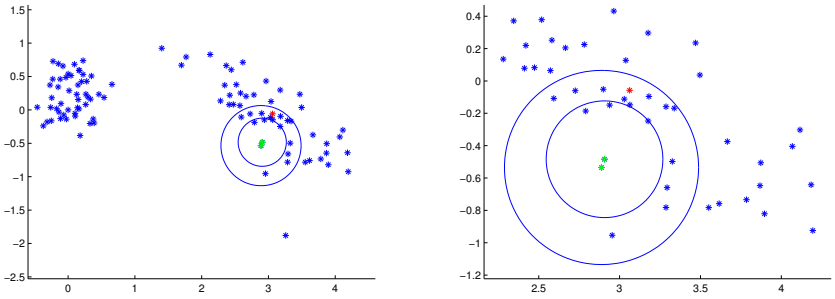
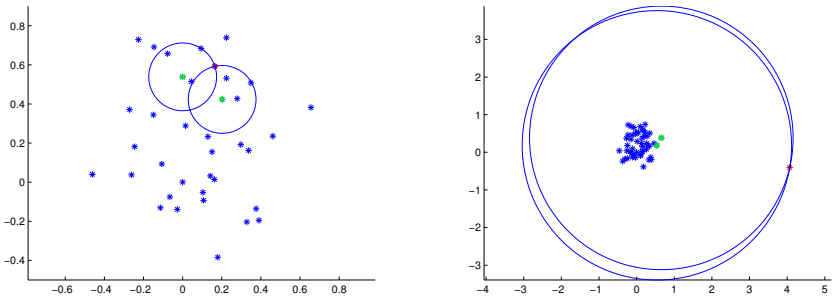


Figura 2.10: Projeção do ponto p_{102}. Um dos círculos, C_q ou C_r, está contido no outro. A imagem da direita é um zoom da imagem da esquerda.

Quando os pontos da primeira classe são projetados são comuns raios de valores similares, no entanto ao se projetar o primeiro ponto pertencente a segunda classe os valores dos raios considerados nessa etapa se tornam consideravelmente maiores a fim de separar bem essa segunda classe da primeira (Figura 2.11). Uma análise do conjunto de dados revela que esses valores de distância maiores são devido a uma mudança significativa nos valores correspondentes a medida do comprimento e largura das sépalas nas flores que pertencem a segunda e terceira classes. Os valores de comprimento da sépala para as flores íris-setosa são próximos à 1.5, enquando que o valores para as flores íris-versicolor e íris-

virginica são próximos à 4.5 e 5.5, respectivamente. Além disso os valores de largura da sépala para as flores íris-setosa são próximos à 0.2, enquando que o valores para as flores íris-versicolor e íris-virginica são próximos à 1.5 e 2.0, respectivamente.



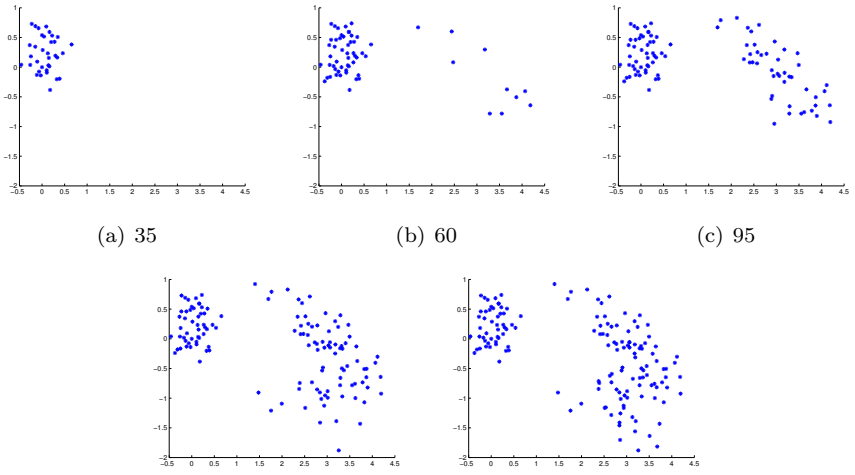
(a) Projeção do ponto p_{38}.

(b) Projeção do ponto p_{51}.

Figura 2.11: Projeção dos pontos p_{38} e p_{51}. Separação da segunda classe durante a projeção.

Na Figura 2.11 são esperados 38 pontos, no entando somente 37 podem ser observados isso ocorre pois as projeções dos pontos p_{35} e p_{38} coincidem, isto é, $p'_{35} = p'_{38}$.

O método segue com esses procedimentos até que os 150 pontos pertencentes ao conjunto de dados Íris sejam projetados. Na Figura 2.12 ilustramos algumas projeções parciais, com 35, 60, 95, 130 e 150 pontos já projetados.



(a) 35

(b) 60

(c) 95

(d) 130

(e) 150

Figura 2.12: Projeção NNP do conjunto de dados Íris. Projeções parciais.

O criador do conjunto de dados Íris fornece uma separação para o conjunto de dados de acordo com a espécie das plantas, são três as espécies, uma delas se

separa linearmente das outras duas e estas, no entanto, não se separam linearmente. Com a projeção do conjunto de dados concluída, demos destaque as três classes definidas por ele colorindo os pontos projetados de acordo com as classes a que pertencem, a Figura 2.13 ilustra esse nosso experimento. Atribuimos a cor azul a primeira classe constituida dos primeiros 50 pontos (íris-setosa), a cor vermelha a segunda classe constituida dos pontos 51 ao 100 (íris-versicolor) e a cor verde a última classe composta pelos pontos 101 ao 150 (íris-virginica).

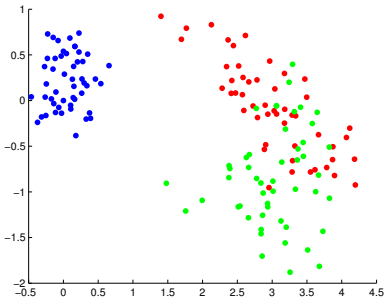
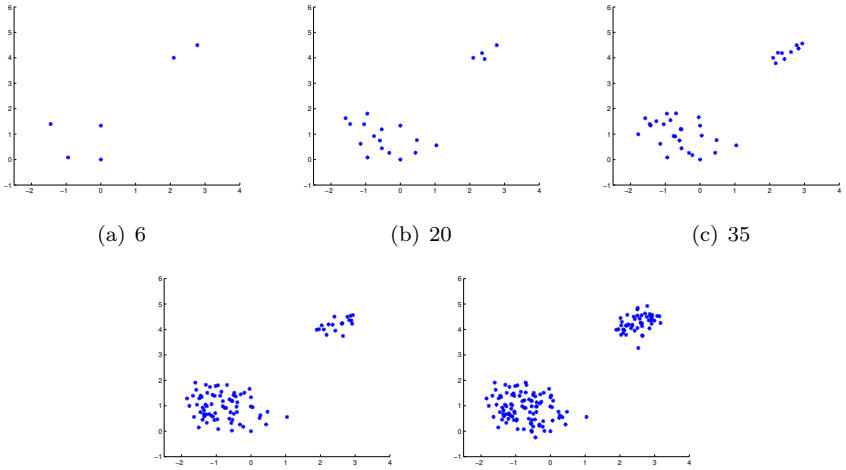


Figura 2.13: Projeção do conjunto de dados Íris através do método NNP com destaque das classes.

A projeção realizada pelo método NNP é sensível a presença dessas três classes como mostra a Figura 2.13, isto é possível pois a primeira espécie de planta do tipo íris apresenta características da pétala bem distintas das outras duas espécies separando bem essa classe das outras duas, estas por apresentarem todos os atributos próximos não podem ser separadas, mas ainda sim essas duas classes tem seus pontos projetados bem agrupadas

Essa projeção foi realizada seguindo a ordem do conjunto de dados, isto é, primeiro projetamos os pontos que representam as flores de íres-setosa, depois íris-versicolor e por fim íris-virginica. A questão é: o processo de projeção não seria afetado ao projetarmos os pontos numa ordem diferente? Aplicamos um algoritmo para reordenar o conjunto de dados Íris e aplicamos novamente o método NNP a ele e os resultados obtidos foram exatamente como os esperados (Figura 2.14).



(a) 6 (b) 20 (c) 35

(d) 90 (e) 150

Figura 2.14: Projeções do conjunto de dados Iris, após reordenação, através do método NNP. São mostradas projeções parciais imediatamente após a projeção dos pontos 6, 20, 35, 90, 150

2.3 Método Force

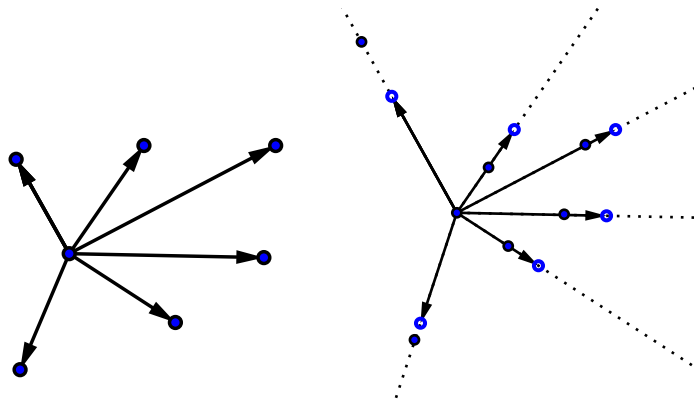
O método Force [20] é na verdade um esquema de melhoria de projeção, isto é, uma vez realizada uma projeção, a posição dos pontos é então redefinida de modo a tentar diminuir as perdas naturais de informações decorrentes do processo de projeção.

A ideia é tentar separar pontos que foram projetados muito próximos, e aproximar pontos que foram projetados muito distantes. O método deixa a impressão de que pontos estão sendo "atraidos" ou "repelidos" uns dos outros, por esse motivo é chamado de aproximação Force.

Esquema de melhoria Force: Considere uma projeção $P = \{p'_1, ..., p'_n\}$ de um certo conjunto de dados $S = \{p_1, ..., p_n\}$, para cada instância $p'_i \in P$, calcula-se o vetor $\vec{v_{ij}} = p'_j - p'_i$, $\forall p'_j \neq p'_i$, então aplica-se uma perturbação ao ponto p'_j na direção de $\vec{v_{ij}}$, que depende da diferença das distâncias $\delta(p_i, p_j)$ e $d(p'_i, p'_j)$.

A diferença entre as distâncias $\delta(p_i, p_j)$ e $d(p'_i, p'_j)$ é denotada por Δ_{ij} e a perturbação utilizada sobre o ponto p'_j é apenas uma fração desse Δ_{ij}. Esta fração tem bastante influência sobre o resultado da projeção, por isso pode variar entre os conjuntos de dados a fim de melhorar o resultado da projeção.

De modo mais ilustrativo o que fazemos pode ser visto na Figura 2.15. A Figura 2.15(a) mostra um ponto em destaque no centro que faz o papel do ponto p'_i e os vetores $\vec{v_{ij}}$ que o liga aos pontos $p'_j \neq p'_i$, enquanto que a Figura 2.15(b) retrata a perturbação experimentada por esses pontos p'_j. Note que alguns pontos p'_j se afastam do ponto p'_i ($\Delta_{ij} > 0$) e outros se aproximam ($\Delta_{ij} < 0$), além disso a perturbação aplicada muda de acordo com o ponto p'_j, o que é de se esperar uma vez que Δ_{ij} depende do par de pontos levado em consideração.



(a) Disposição dos pontos antes da aplicação do esquema Force.

(b) Disposição dos pontos após a aplicação do esquema Force.

Figura 2.15: Perturbação realizada pelo esquema Force. O ponto central simboliza p'_i, as setas os vetores $\vec{v_{ij}}$ e os demais pontos os $p'_j \neq p'_i$.

Para evitar inconsistências no cálculo dessas diferenças de distâncias multidimensionais e bidimensionais trabalha-se sempre com distâncias normalizadas. O processo de normalização é necessário apenas uma vez para as distâncias multidimensionais e, para cada iteração, aplica-se uma normalização às coordenadas dos dados projetados.

O Algoritmo 3 apresenta o processo realizado em cada iteração do esquema de melhoria Force.

Algorithm 3 Esquema de melhoria Force

for all ponto projetado p'_i **do**
  for all ponto projetado $p'_j \neq p'_i$ **do**
    Cálcule $\vec{v_{ij}}$ como sendo o vetor de p'_i a p'_j
    Mova p'_j na direção de $\vec{v_{ij}}$ uma fração de Δ_{ij}
  end for
end for
Normalize as coordenadas de projeção no intervalo $[0, 1]$ em ambas as dimensões

O esquema de melhoria Force pode ser aplicado sobre os resultados de qualquer método de projeção multidimensional, contudo Tajeda, Minghim e Nonato comparam o método NNP com um outro método de projeção multidimensional bastante conhecido, chamado Fastmap [6], e mostram que melhores resultados são obtidos quando o esquema Force tem como projeção inicial o resultado do método NNP [20]. Além disso o número de iterações necessárias para estabilizar a melhoria da projeção (geralmente custosa) é menor se comparado com o método Fastmap. Por esse motivo se tornou comum na literatura denominar o método NNP mais o esquema de melhoria Force, apenas por método Force (Algoritmo 4).

Algorithm 4 Algoritmo Force

P = NNP(Conjunto de dados);
Calculo das distâncias m-dimensionais
Normalização das distâncias m-dimensionais
for all $i = 1 : numIte$ **do**
  Esquema de melhoria Force(P)
end for

Exemplificando o método Force

Como já mencionado o método Force não realiza projeções, mas quando aplicado ao resultado de uma projeção tende a melhorá-la. Consideremos então uma projeção especifica do conjunto de dados Iris obtida através do método NNP que é apresentada na Figura 2.16.

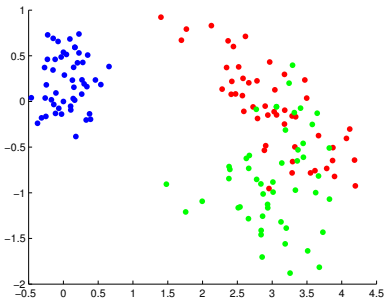
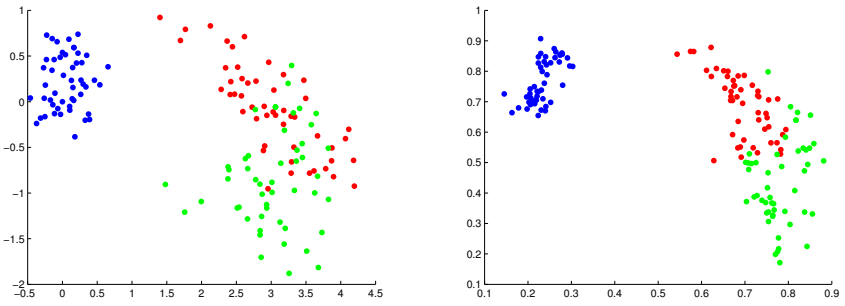


Figura 2.16: Projeção do conjunto de dados Íris com destaque das classes

Para cada ponto presente na projeção vamos realizar uma pertubação nos demais pontos, movendo-os para mais perto ou mais longe, na direção do vetor que liga os pontos.



(a) Método NNP (b) Método Force

Figura 2.17: Projeção do conjunto de dados Íris através dos métodos NNP e Force com destaque das classes. O esquema de melhoria Force, componente do método Force, é aplicado sobre a projeção NNP que aparece em (a).

Após aplicar o esquema de melhoria Force obtemos a projeção ilustrada na Figura 2.17. Pode-se constatar, visualmente, uma melhora significativa no agrupamento dos dados. As classes vermelha e verde são mais bem distinguíveis nessa projeção.

Nas projeções do conjunto de dados Íris é utilizado como fração de Δ_{ij} um centésimo (0.01) de seu valor, para realizar as perturbações dos pontos p'_j.

Os métodos NNP e Force dão ótimos resultados em termos de qualidade de projeção como veremos na seção 4, no entanto eles são computacionalmente caros tornando seu uso em conjuntos de dados muito grandes inviável.

Passamos agora a uma nova modalidade de métodos, cujo principal objetivo é lidar com conjuntos de dados significativamente maiores do que os já apresentados até aqui. Esses métodos fazem uso de um subconjunto dos dados, chamado conjunto dos pontos de controle, esse subconjunto é interpretado como um novo conjunto de dados, razoavelmente menor e aplica-se sobre ele um método de projeção multidimensional (mais geral, um método MDS [2]) bastante preciso, mas computacionalmente caro para obter um bom layout inicial. Em seguida, levamos em conta as informações fornecidas por esses pontos para realizarmos a projeção de todo o conjunto de dados.

2.4 Método LSP

O método LSP [16] (Least Square Projections) é o primeiro método abordado neste trabalho que faz uso de pontos de controle para realizar a projeção, os pontos de controle são projetados e sua posição no espaço visual orienta a projeção de todo o conjunto de dados. Para a projeção dos dados restantes define-se uma relação de vizinhos para cada dado, relação essa que se baseia em uma ideia geométrica, em seguida constroi-se alguns sistemas lineares (o número de sistemas depende da dimensão do espaço de projeção), onde a solução desses sistemas são as coordenadas dos pontos de $S = \{p_1, ..., p_n\}$ no espaço visual.

Começemos com a definição do subconjunto S_c dos pontos de controle para o conjunto de dados S em questão. Atualmente existem várias técnicas de seleção dos pontos de controle, obviamente essas técnicas tentam representar da melhor forma possível a distribuição dos dados no espaço m-dimensional.

No LSP utilizamos o método k-means para dividir o conjunto de dados em $n_c = \sqrt{\#S}$ aglomerados, então escolhemos um ponto de controle em cada aglomerado como sendo o medóide, isto é, o ponto do conjunto de dados mais próximo ao centróide do aglomerado. Os pontos de controle poderiam ser escolhidos de outras formas, aleatoriamente por exemplo, no entanto escolhemos essa técnica baseada no k-means pois os aglomerados definidos também são usados em uma próxima etapa para definir a relação de vizinhança dos pontos.

Com os pontos de controle já definidos, ainda precisamos defenir uma relação de vizinhança entre os dados para que possamos montar alguns sistemas lineares necessários a esse método para realizar a projeção multidimensional. A cada ponto $p_i \in S$ é associada uma lista de pontos $V_i \subset S$, onde esta lista deve refletir a vizinhança de p_i em R^m, isto é, deve indicar como p_i se relaciona com os pontos de S.

Dado o conjunto $V = \{V_1, V_2, ..., V_n\}$, é fundamental que V satizfaça uma restrição, denominada condição de sobreposição. O conjunto V serve de base para gerar um sistema homogêneo. Uma vez que a condição de sobreposição é satisfeita, o sistema gerado tem posto igual a $n - 1$, como afirma Paulovich em [16], garantindo assim infinitas soluções e, consequentemente, uma solução não trivial.

Definição (Condição de Sobreposição): Seja $S = \{p_1, p_2, ..., p_n\}$ um conjunto de dados e seja V o conjunto de relações de vizinhança dos pontos em S. O conjunto V satisfaz a condição de sobreposição se para cada dois pontos p_i e p_j existe uma sequência de vizinhanças V_1^{ij},V_2^{ij},...,V_q^{ij} de modo que $V_1^{ij} = V_i$, $V_q^{ij} = V_j$ e $V_k^{ij} \cap V_{k+1}^{ij} \neq \emptyset$, para $k = 1, 2, ..., q - 1$.

A condição de sobreposição também implica que no cálculo da projeção de um ponto todos os pontos de S influenciam (implicitamente). Agora precisamos esclarecer como a lista $V_i \subset S$ associada a p_i é definida:

Métodos de definição de vizinhança

Como veremos na seção 4, a forma como se define a vizinhança de um ponto influência fortemente no resultado da projeção. Nosso trabalho aborda dois métodos de definição de vizinhaça: pelo número de grupos e raio de influência,

e pelo número de vizinhos.

Pelo número de aglomerados e raio de influência. Para definir os pontos de controle associados a um conjunto de dados S, o método k-means divide o conjunto de dados em vários aglomerados. A ideia é se utilizar desses aglomerados para definir os vizinhos de um ponto p_i.

A cada ponto p_i vamos associar n_a aglomerados, os aglomerados associados a p_i são: o aglomerado que p_i pertence e os $n_a - 1$ aglomerados mais próximos do aglomerado que contém p_i, a ordenação dos aglomerados é dada pela distância dos medoides que os representam. O número de aglomerados n_a associados a cada ponto de S é um parametro do método podendo ser ajustado de acordo com o conjunto de dados em questão.

Os vizinhos de um ponto p_i são todos os pontos que pertencem aos n_a aglomerados e que estão dentro de um raio de influência. Assim a pesquisa por vizinhos é realizada apenas no aglomerado que p_i pertence e nos aglomerados mais próximos. O raio de influência é definido como a distância média entre os pontos pertencentes ao conjunto de dados S.

Pelo número de vizinhos. Os vizinhos de um ponto podem ser tomados sem levar em consideração os aglomerados definidos pelo método k-means. A ideia é associar a cada ponto p_i de S um mesmo número de vizinhos (n_{viz}). O número de vizinhos é um parametro do método e pode ser escolhido de acordo com o conjunto de dados.

Os vizinhos de um ponto p_i são os n_{viz} pontos de S que mais se aproximam desse ponto. Com essa definição de vizinhos cada ponto de S tem um mesmo número de vizinhos, igual a n_{viz}.

A definição de vizinhos baseado no número de vizinhos, define uma quantidade fixa de vizinhos para cada ponto, enquanto que a baseada no número de aglomerados e raio de influência permite uma variação no número de vizinhos associado a cada ponto. Contudo, ambos procuram definir como vizinhos de um ponto p_i aqueles pontos de S que estão mais próximos de p_i em $\mathbb{R}^m$.

Com esses dados em mão podemos passar a construção dos sistemas. Considere um dado p_i e sua vizinhança $V_i = \{p_{i_1}, p_{i_2}, ..., p_{i_k}\}$ com $k = k(i)$ pontos e seja p'_i sua projeção no espaço visual. O método LSP trabalha com uma ideia fundamental para realizar a projeção, posicionar cada ponto após projetado no fecho convexo de seus vizinhos (dos pontos em Vi). Isto é queremos impor a seguinte condição a p_i:

$$p'_i - \sum_{j=1,...,k} \alpha_{ij} p'_{i_j} = 0$$

$$0 \leq \alpha_{ij} \leq 1; \quad \sum \alpha_{ij} = 1$$

Supondo que a equação anterior é satisfeita por todos os pontos de S, podemos considerar um conjunto de sistemas lineares a partir dos quais se determinam as coordenadas dos pontos p'_i no espaço de projeção, são eles

$$LX_1 = 0, LX_2 = 0, ..., LX_d = 0$$

onde $X_1, X_2, ..., X_d$ são vetores contendo as coordenadas cartesianas dos pontos

$$X_1 = \begin{pmatrix} x_{11} \\ x_{12} \\ \vdots \\ x_{1n} \end{pmatrix}, \; X_2 = \begin{pmatrix} x_{21} \\ x_{22} \\ \vdots \\ x_{1n} \end{pmatrix}, \cdots, X_d = \begin{pmatrix} x_{d1} \\ x_{d2} \\ \vdots \\ x_{dn} \end{pmatrix}$$

daí, o ponto p_i' é obtido por

$$p_i' = (x_{1i}, x_{2i}, ..., x_{di})$$

e $L = (l_{ij})$ é a matriz $n \times n$ cujas entradas são dadas por:

$$l_{ij} = \begin{cases} 1, & i = j \\ -\alpha_{ij}, & p_j \in V_i \\ 0, & \text{caso contrario} \end{cases}$$

Quando escolhemos $\alpha_{ij} = 1/k$ (lembre que $k = k(i)$ se trata do número vizinhos do ponto p_i) a soma dos elementos de cada linha da matriz L se torna igual a zero, à matrizes com essa propriedade dá-se o nome de matrizes laplacianas, consequentemente, os sistemas da forma $Lx = b$ passam a ser denominados sistemas laplacianos. Daqui em diante mantemos essa escolha $\alpha_{ij} = 1/k$.

Vejamos as contas que confirmam essas afirmações para o caso em que $S = \{p_1, ..., p_6\}$ e as relações de vizinhança são dadas como a seguir:

$$\begin{aligned} V_1 &= \{p_3 \; p_4 \; p_6\} \\ V_2 &= \{p_5 \; p_4 \; p_6\} \\ V_3 &= \{p_1 \; p_5 \; p_6\} \\ V_4 &= \{p_1 \; p_6\} \\ V_5 &= \{p_3 \; p_2 \; p_6\} \\ V_6 &= \{p_1 \; p_4 \; p_2 \; p_5\} \end{aligned}$$

Consideramos nesse momento $d = 2$ apenas para exemplificar. Esse caso é o que damos maior atenção em todo o trabalho, note contudo que com o mesmo argumento pode-se tratar o caso geral onde $d \in \{1, 2, 3\}$:

$$\begin{gathered} p_1' - 1/3p_3' - 1/3p_4' - 1/3p_6' = 0 \\ p_2' - 1/3p_4' - 1/3p_5' - 1/3p_6' = 0 \\ p_3' - 1/3p_1' - 1/3p_5' - 1/3p_6' = 0 \\ p_4' - 1/2p_1' - 1/2p_6' = 0 \\ p_5' - 1/3p_2' - 1/3p_3' - 1/3p_6' = 0 \\ p_6' - 1/4p_1' - 1/4p_2' - 1/4p_4' - 1/4p_5' = 0 \end{gathered}$$

Escrevendo as esquações acima em coordenadas, ficamos com:

$$\begin{gathered} (x_{11}, x_{21}) - 1/3(x_{13}, x_{23}) - 1/3(x_{14}, x_{24}) - 1/3(x_{16}, x_{26}) = 0 \\ (x_{12}, x_{22}) - 1/3(x_{14}, x_{24}) - 1/3(x_{15}, x_{25}) - 1/3(x_{16}, x_{26}) = 0 \\ (x_{13}, x_{23}) - 1/3(x_{11}, x_{21}) - 1/3(x_{15}, x_{25}) - 1/3(x_{16}, x_{26}) = 0 \\ (x_{14}, x_{24}) - 1/2(x_{11}, x_{21}) - 1/2(x_{16}, x_{26}) = 0 \\ (x_{15}, x_{25}) - 1/3(x_{12}, x_{22}) - 1/3(x_{13}, x_{23}) - 1/3(x_{16}, x_{26}) = 0 \\ (x_{16}, x_{26}) - 1/4(x_{11}, x_{21}) - 1/4(x_{12}, x_{22}) - 1/4(x_{14}, x_{24}) - 1/4(x_{15}, x_{25}) = 0 \end{gathered}$$

Olhando agora somente para a primeira coordenada, obtemos:

$$\begin{gathered}
x_{11} - 1/3x_{13} - 1/3x_{14} - 1/3x_{16} = 0 \\
x_{12} - 1/3x_{14} - 1/3x_{15} - 1/3x_{16} = 0 \\
x_{13} - 1/3x_{11} - 1/3x_{15} - 1/3x_{16} = 0 \\
x_{14} - 1/2x_{11} - 1/2x_{16} = 0 \\
x_{15} - 1/3x_{12} - 1/3x_{13} - 1/3x_{16} = 0 \\
x_{16} - 1/4x_{11} - 1/4x_{12} - 1/4x_{14} - 1/4x_{15} = 0
\end{gathered}$$

Finalmente montamos o primeiro sistema:

$$\begin{pmatrix}
1 & 0 & -1/3 & -1/3 & 0 & -1/3 \\
0 & 1 & 0 & -1/3 & -1/3 & -1/3 \\
-1/3 & 0 & 1 & 0 & -1/3 & -1/3 \\
-1/3 & 0 & 0 & 1 & 0 & -1/3 \\
0 & -1/3 & -1/3 & 0 & 1 & -1/3 \\
-1/3 & -1/3 & 0 & -1/3 & -1/3 & 1
\end{pmatrix}
\begin{pmatrix} x_{11} \\ x_{12} \\ x_{13} \\ x_{14} \\ x_{15} \\ x_{16} \end{pmatrix} =
\begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

De maneira análoga, montamos o segundo:

$$\begin{pmatrix}
1 & 0 & -1/3 & -1/3 & 0 & -1/3 \\
0 & 1 & 0 & -1/3 & -1/3 & -1/3 \\
-1/3 & 0 & 1 & 0 & -1/3 & -1/3 \\
-1/3 & 0 & 0 & 1 & 0 & -1/3 \\
0 & -1/3 & -1/3 & 0 & 1 & -1/3 \\
-1/3 & -1/3 & 0 & -1/3 & -1/3 & 1
\end{pmatrix}
\begin{pmatrix} x_{21} \\ x_{22} \\ x_{23} \\ x_{24} \\ x_{25} \\ x_{26} \end{pmatrix} =
\begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

Obtemos com isso dois sistemas lapacianos (já que $d = 2$), no lado esquerdo de cada equação acima temos a matriz L seguida de vetores coordenadas, X_1 na 1ªequação e X_2 na 2ªequação.

$$LX_1 = 0 \text{ e } LX_2 = 0$$

a matriz laplaciana L como definida acima não carrega nenhuma informação geométrica a cerca do problema, essa matriz apenas guarda as relações de vizinhança de cada ponto, por esse motivo se resolvermos os sistemas $LX_1 = 0$ e $LX_2 = 0$ obteremos soluções que não são atraentes. Para resolver esse prolema adicionamos algumas informações geométricas aos sistemas por meio dos pontos de controle. No tópico a seguir, estão disponíveis três possíveis formas de se adicionar ao sistema essas informações geométricas.

Métodos de solução dos sistemas

Neste ponto os sistemas laplacianos já estão formados, ao resolver esses sistemas obteremos a projeção dos dados, três formas para a solução desses sistemas são apresentadas a seguir.

Novas Linhas. A solução do sistema pelo método Novas Linhas começa com a projeção do conjunto Sc de pontos de controle, através de um método MDS, em seguida os pontos de controle serão adicionados ao sistema laplaciano como

novas linhas na matriz. Suas coordenadas cartesianas serão adicionadas no lado direito da equação, dando origem a um vetor não nulo.

Experimentamos assim a seguinte mudança de estrutura dos sistemas:

$$LX_i = 0 \longrightarrow AX_i = b_i,\ i = 1, ..., d.$$

Onde A é uma matriz retangular $(n + n_c) \times n$, dada por

$$A = \begin{pmatrix} L \\ C \end{pmatrix},$$

a matriz $C = (c_{ij})$ é uma matriz retangular $n_c \times n$ e a cada linha dessa matriz associamos um dos pontos de controle, assim c_{ij} fica definida como segue:

$$c_{ij} = \begin{cases} 1, & p_j \text{ é ponto de controle associado a linha } i \\ 0, & \text{caso contrário} \end{cases}$$

e o vetor $b_i = (b_{i1}, b_{i2}, ..., b_{i(n+nc)})'$ é definido por:

$$b_{i1}, ..., b_{in} = 0$$

e, para $n < j \leq n + n_c$, b_{ij} é i-ésima coordenada do ponto de controle que corresponde a linha j de A.

Com intuito de tornar esses conceitos mais próximos ao leitor voltamos ao exemplo, considere a configuração presente na Figura 2.18, onde os pontos em negrito representam pontos de controle.

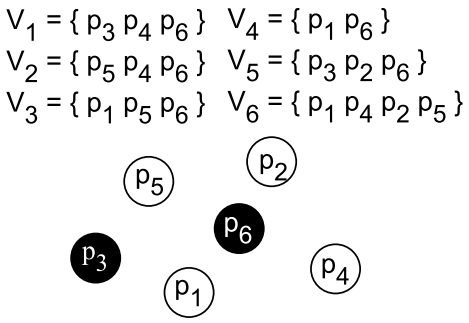


Figura 2.18: Exemplo dos 6 pontos. Os pontos em negrito representam pontos de controle e as listas $V_1, ..., V_6$ as relações de vizinhança de cada ponto.

Nosso objetivo é construir a matriz A e os vetores b_i para esse caso. Anteriormente já haviamos cálculado a matriz L que corrresponde a essa configuração:

$$L = \begin{pmatrix} 1 & 0 & -1/3 & -1/3 & 0 & -1/3 \\ 0 & 1 & 0 & -1/3 & -1/3 & -1/3 \\ -1/3 & 0 & 1 & 0 & -1/3 & -1/3 \\ -1/3 & 0 & 0 & 1 & 0 & -1/3 \\ 0 & -1/3 & -1/3 & 0 & 1 & -1/3 \\ -1/3 & -1/3 & 0 & -1/3 & -1/3 & 1 \end{pmatrix}$$

Como os pontos p_3 e p_6 são os pontos de controle, a matriz C é dada como segue:

$$C = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Apartir das matrizes L e C podemos então construir a matriz A:

$$A = \begin{pmatrix} 1 & 0 & -1/3 & -1/3 & 0 & -1/3 \\ 0 & 1 & 0 & -1/3 & -1/3 & -1/3 \\ -1/3 & 0 & 1 & 0 & -1/3 & -1/3 \\ -1/3 & 0 & 0 & 1 & 0 & -1/3 \\ 0 & -1/3 & -1/3 & 0 & 1 & -1/3 \\ -1/3 & -1/3 & 0 & -1/3 & -1/3 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Para a construção dos vetores b_1 e b_2 precisamos supor que os pontos de controle foram projetados no espaço visual, apenas para exemplificar consideramos $p'_3 = (0.5, 0.6)$ e $p'_6 = (2, 1.1)$. Daí

$$b_1 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0.5 \\ 2 \end{pmatrix}, \; b_2 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0.6 \\ 1.1 \end{pmatrix}$$

Com isso temos os sistemas $AX_1 = b_1$ e $AX_2 = b_2$. Por fim, resolvemos os sistemas no sentido de mínimos quadrados, o que significa que nós devemos encontrar x e y que minimize $||AX - b_1||$ e $||AX - b_2||$, respectivamente, temos então que $x = (A^T A)^{-1} A^T b_1$ e $y = (A^T A)^{-1} A^T b_2$ (seção 1.2). Uma boa motivação para utilização desse tipo de solução é que a matriz A não é quadrada.

Penalidade. O método de solução via penalidade difere do método novas linhas basicamente na forma como as condições de contorno são impostas ao sistema, o objetivo é impor as condições de contorno sem que a dimensão da matriz L mude, assim poderemos trabalhar com uma matriz que é ainda uma matriz quadrada.

Nesse caso ao definirmos a matriz A, em vez de introduzirmos novas linhas na matriz L, alterando sua dimensão, as linhas de L correspondentes aos pontos de controle tem o elemento pertencente a diagonal somado a um valor bastante grande, como por exemplo 10^8. Tal valor recebe o nome de penalidade.

O vetor nulo que aparece no lado direito da equação $LX_i = 0$ é substituido por um vetor b_i de tamanho $n \times 1$, onde as entradas de b_i nas linhas correspondentes aos pontos de controle (lembre-se que cada linha da matriz L e, consequentemente do vetor b_i, representa um ponto) são dadas pelas i-ésimas coordenadas dos pontos de controle multiplicadas pela penalidade, as demais entradas são nulos. Novamente temos uma mudança de estrutura do nosso sistema

$$LX_i = 0 \longrightarrow AX_i = b_i, \; i = 1, ..., d$$

sem alterar a dimensão da matriz do sistema.

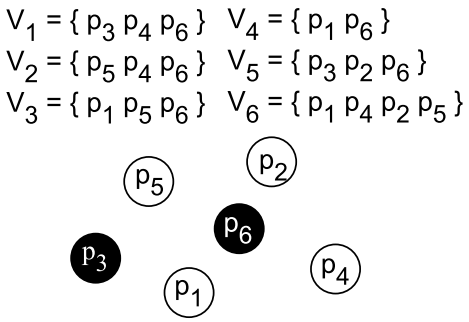


Figura 2.19: Exemplo dos 6 pontos. Os pontos em negrito representam pontos de controle e as listas $V_1, ..., V_6$ as relações de vizinhança de cada ponto.

Trabalhamos novamente com o exemplo dos 6 pontos (Figura 2.19) a fim de que se torne clara a diferença entre os dois métodos: Novas Linhas e Penalidade. Já vimos que para essa configuração a matriz L fica sendo

$$L = \begin{pmatrix} 1 & 0 & -1/3 & -1/3 & 0 & -1/3 \\ 0 & 1 & 0 & -1/3 & -1/3 & -1/3 \\ -1/3 & 0 & 1 & 0 & -1/3 & -1/3 \\ -1/3 & 0 & 0 & 1 & 0 & -1/3 \\ 0 & -1/3 & -1/3 & 0 & 1 & -1/3 \\ -1/3 & -1/3 & 0 & -1/3 & -1/3 & 1 \end{pmatrix}$$

Como os pontos p_3 e p_6 são os pontos de controle, a matriz L é alterada nas linhas 3 e 6 para obtermos a matriz A, somando ao elemento da diagonal um valor razoavelmente grande, utilizamos nesse exemplo 10^8:

$$A = \begin{pmatrix} 1 & 0 & -1/3 & -1/3 & 0 & -1/3 \\ 0 & 1 & 0 & -1/3 & -1/3 & -1/3 \\ -1/3 & 0 & 1+\mathbf{10^8} & 0 & -1/3 & -1/3 \\ -1/3 & 0 & 0 & 1 & 0 & -1/3 \\ 0 & -1/3 & -1/3 & 0 & 1 & -1/3 \\ -1/3 & -1/3 & 0 & -1/3 & -1/3 & 1+\mathbf{10^8} \end{pmatrix}$$

Além disso considerando $p'_3 = (0.5, 0.6)$ e $p'_6 = (2, 1.1)$, os vetores b_1 e b_2 são dados como a seguir

$$b_1 = \begin{pmatrix} 0 \\ 0 \\ 0.5 * \mathbf{10^8} \\ 0 \\ 0 \\ 2 * \mathbf{10^8} \end{pmatrix}, \; b_2 = \begin{pmatrix} 0 \\ 0 \\ 0.6 * \mathbf{10^8} \\ 0 \\ 0 \\ 1.1 * \mathbf{10^8} \end{pmatrix}$$

Formando assim os novos sistemas

$$AX_1 = b_1 \text{ e } AX_2 = b_2,$$

esses sistemas também são resolvidos no sentido de mínimos quadrados, uma vez que mais de uma solução pode ser encontrada para cada um deles.

Para exemplificar a ideia por trás deste método, considere a multiplicação da 3ª linha da matriz A pelo vetor $X_1 = [x_1, ..., x_6]'$

$$\begin{aligned} A[3,:][x_1,...,x_6]' &= -\frac{1}{3}x_1 + 0x_2 + (1+10^8)x_3 + 0X_4 - \frac{1}{3}x_5 - \frac{1}{3}x_6 \\ &= (-\frac{1}{3}x_1 + 0x_2 + x_3 + 0X_4 - \frac{1}{3}x_5 - \frac{1}{3}x_6) + x_3 10^8 \end{aligned}$$

igualar esta expressão com $0.5 * 10^8$ significa na verdade impor que

$$-\frac{1}{3}x_1 + 0x_2 + x_3 + 0X_4 - \frac{1}{3}x_5 - \frac{1}{3}x_6 = 0 \text{ e } x_3 = 0.5$$

com isso estamos adicionando ao sistema a informação de que $x_3 = 0.5$ e ainda mantemos a equação original dada pelo sistema homogêneo $LX_1 = 0$.

Evolução. O método de solução evolução tem como objetivo permitir a interação do usuário de modo que o mesmo possa melhorar a qualidade (visual) do layout.

Como as coordenadas dos pontos de controle já são conhecidas não as consideramos como variáveis dos sistemas, com isso eliminamos as linhas correspondentes aos pontos de controle e substituimos as variaveis introduzidas pelos pontos de controle por seus valores conhecidos. Subtraimos esses valores de ambos os lados da equação obtendo um vetor b não nulo no lado direito da equação. Esse procedimento é apenas um processo de substituição.

Retornemos a configuração que temos usado como exemplo para esses métodos de soluções dos sistemas (Figura 2.20).

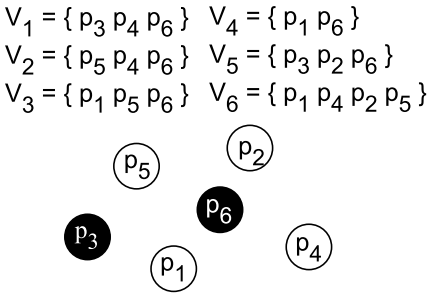


Figura 2.20: Exemplo dos 6 pontos. Os pontos em negrito representam pontos de controle e as listas $V_1, ..., V_6$ as relações de vizinhança de cada ponto.

Para esse exemplo a matriz L é

$$L = \begin{pmatrix} 1 & 0 & -1/3 & -1/3 & 0 & -1/3 \\ 0 & 1 & 0 & -1/3 & -1/3 & -1/3 \\ -1/3 & 0 & 1 & 0 & -1/3 & -1/3 \\ -1/3 & 0 & 0 & 1 & 0 & -1/3 \\ 0 & -1/3 & -1/3 & 0 & 1 & -1/3 \\ -1/3 & -1/3 & 0 & -1/3 & -1/3 & 1 \end{pmatrix}$$

Como os pontos p_3 e p_6 são pontos de controle digamos com $p'_3 = (0.5, 0.6)$ e $p'_6 = (2, 1.1)$, então teriamos que eliminar as linhas 3 e 6.

$$L' = \begin{pmatrix} 1 & 0 & -1/3 & -1/3 & 0 & -1/3 \\ 0 & 1 & 0 & -1/3 & -1/3 & -1/3 \\ -1/3 & 0 & 0 & 1 & 0 & -1/3 \\ 0 & -1/3 & -1/3 & 0 & 1 & -1/3 \end{pmatrix}$$

Agora precisamos introduzir as coordenadas dos pontos de controle que são conhecidas, isto é, $x_3 = 0.5$, $x_6 = 2$, $y_3 = 0.6$ e $y_6 = 1.1$, daí

$$L' \begin{pmatrix} x_1 \\ x_2 \\ 0.5 \\ x_4 \\ x_5 \\ 2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \text{ e } L' \begin{pmatrix} y_1 \\ y_2 \\ 0.6 \\ y_4 \\ y_5 \\ 1.1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

Nesse momento apenas subtraimos de ambos os lados os valores conhecidos. Definindo a matriz A por

$$A = \begin{pmatrix} 1 & 0 & -1/3 & 0 \\ 0 & 1 & -1/3 & -1/3 \\ -1/3 & 0 & 1 & 0 \\ 0 & -1/3 & 0 & 1 \end{pmatrix}$$

ficamos com

$$A \begin{pmatrix} x_1 \\ x_2 \\ x_4 \\ x_5 \end{pmatrix} = \begin{pmatrix} 1/3 * 0.5 + 1/3 * 2 \\ 0 * 0.5 + 1/3 * 2 \\ 0 * 0.5 + 1/3 * 2 \\ 1/3 * 0.5 + 1/3 * 2 \end{pmatrix}$$

e

$$A \begin{pmatrix} x_1 \\ x_2 \\ x_4 \\ x_5 \end{pmatrix} = \begin{pmatrix} 1/3 * 0.6 + 1/3 * 1.1 \\ 0 * 0.6 + 1/3 * 1.1 \\ 0 * 0.6 + 1/3 * 1.1 \\ 1/3 * 0.6 + 1/3 * 1.1 \end{pmatrix}$$

Isto é, ficamos com os seguintes sistemas

$$AX_1 = b_1 \text{ e } AX_2 = b_2.$$

Note como esse método de solução é vantajoso: introduzimos os pontos de controle aos sistemas sem que a matriz L deixasse de ser uma matriz quadrada, além disso obtivemos um vetor não nulo no lado direito da equação.

Agora veja que a matriz A pode ser escrita na forma $I - W$, onde I é a matriz identidade e $W = (w_{ij})$ é uma matriz com a propriedade $\sum_{j=1}^{n-nc} w_{ij} < 1$, para algum i. Isso assegura que W é uma contração e que

$$A^{-1} = (I - W)^{-1} = \sum_{k=0}^{\infty} W^k.$$

Daí, as soluções dos sistemas são

$$X_1 = (\sum_{k=0}^{\infty} W^k)b_1 \text{ e } X_2 = (\sum_{k=0}^{\infty} W^k)b_2.$$

Assim podemos criar um processo iterativo para obtenção dessas soluções. O esquema de iteração que nós utilizamos é dado por

$$u^{k+1} = Wu^k + b,$$

onde $u^0 = b$ e $b \in \{b_1, b_2\}$. Note com essa definição que

$$\begin{aligned} u^1 &= Wu^0 + b \\ &= Wb + b \\ &= (W + I)b \end{aligned}$$

mais ainda,

$$\begin{aligned} u^2 &= Wu^1 + b \\ &= W(W + I)b + b \\ &= (W^2 + W + I)b \end{aligned}$$

conjecturamos então que

$$u^k = (\sum_{i=0}^{k} W^i)b$$

verificamos então essa afirmação por indução: o caso $k = 0$ é trivial, suponhamos a afirmação verdadeira para k e vamos verificá-la para $k + 1$:

$$\begin{aligned} u^{k+1} &= Wu^k + b \\ &= W((\sum_{i=0}^{k} W^i)b) + b \\ &= \sum_{i=1}^{k+1} W^i b + b \\ &= (\sum_{i=1}^{k+1} W^i + I)b \\ &= (\sum_{i=0}^{k+1} W^i)b \end{aligned}$$

o que comprova nossa afirmação, isto é,

$$u^k = (\sum_{i=0}^{k} W^i)b, \forall k$$

Finalmente,

$$\lim_{k\to\infty} u^k = \lim_{k\to\infty} (\sum_{i=0}^{k} W^i)b = (\sum_{i=0}^{\infty} W^i)b = X$$

onde X representa X_1 (caso $b = b_1$) ou X_2 (caso $b = b_2$).

Exemplificando o método LSP

Para ilustrar esse método o aplicamos ao conjunto de dados Íris, veja na Figura 2.21 o resultado de uma projeção. Podemos observar uma certa linearidade nessa projeção em relação as projeções obtidas pelos métodos NNP e Force, isto é os pontos projetados parecem se aproximar de uma curva. Uma forma encontrada para reduzir esse aspecto linear é aumentar o número de pontos de controle utilizado pelo método.

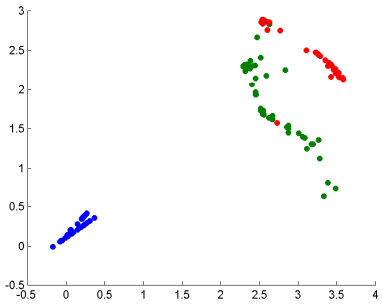
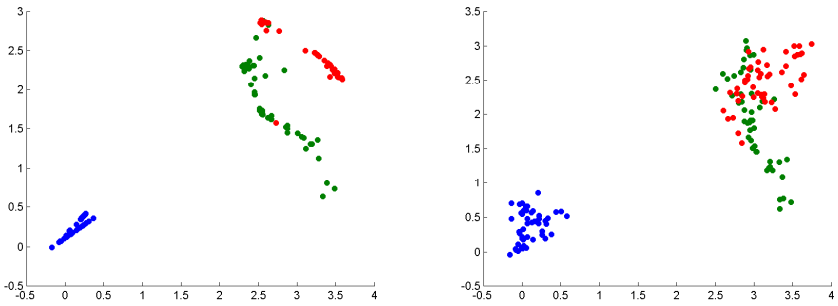


Figura 2.21: Projeção do conjunto de dados Íris através do método LSP.

O método LSP realiza a projeção apresentada na Figura 2.21 com um número de pontos de controle n_c igual a $\sqrt{n}$, lembrando que n é o número de dados que o conjunto de dados contém. O número de pontos de controle utilizado pelo método LSP é então alterado para $n_c = 0.5{*}n$, isto é, são utilizados como pontos de controle a metada dos dados presentes no conjunto de dados, a Figura 2.22 apresenta a projeção do conjunto de dados Íris com $n_c = \sqrt{n}$ (Figura 2.22(a)) e uma outra projeção com a modificação do número de pontos de controle para $n_c = 0.5 * n$ (Figura 2.22(b)).

Essa alteração no número de pontos de controle causa um aumento no esforço computacional, no entanto o layout produzido pela projeção é de melhor qualidade.



(a) $n_c = \sqrt{n}$ (b) $n_c = 0.5 * n$

Figura 2.22: Modificação do número de pontos de controle utilizado pelo método LSP para projetar o conjunto de dados Íris.

No método LSP cada ponto projetado p'_i fica no fecho convexo de seus vizinhos $p_{i_1}, p_{i_2}, ..., p_{i_k}$ devido as seguintes condições

$$p'_i - \sum_{j=1,...,k} \alpha_{ij} p'_{i_j} = 0$$

$$0 \leq \alpha_{ij} \leq 1; \sum \alpha_{ij} = 1$$

com essas condições os elementos l_{ii} da matriz $L = (l_{ij})$ são todos iguais a 1 e, para cada i fixado, a soma de todos os elementos $l_{ij}, j \neq i$ é igual a -1. Na projeção do conjunto de dados Íris apresentada na Figura 2.23 modificamos as condições na segunda linha para

$$\alpha_{ij} = 1; \sum \alpha_{ij} = k$$

com essas novas condições os elementos l_{ii} devem ser iguas a $k = k(i)$ e para cada i fixado a soma de todos os elementos $l_{ij}, j \neq i$ é igual a $-k$, observe que com as novas condições a matriz L continua sendo uma matriz laplaciana. Note também que essas condições não impactam negativamente no resultado da projeção (Figura 2.23).

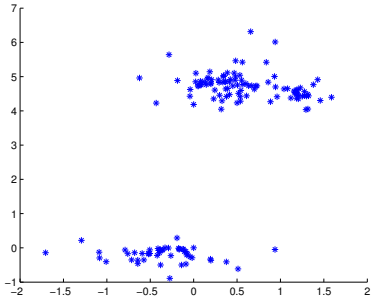


Figura 2.23: Projeção do conjunto de dados Íris através do método LSP com alteração das condições $0 \leq \alpha_{ij} \leq 1; \sum \alpha_{ij} = 1$ para $\alpha_{ij} = 1; \sum \alpha_{ij} = k$.

2.5 Método PLP

O método PLP [15] (Piecewise Laplacian-based Projection), em contraste com a maioria dos métodos existentes, tem uma natureza local e é constituido por três etapas principais:

- amostragem;
- construção do grafo de vizinhança e seleção dos pontos de controle;
- construção e solução do sistema.

A amostragem refere-se a seleção de um subconjunto $\{s_1, ..., s_{n_a}\}$ do conjunto de dados S, veremos que esse conjunto tem um papel diferente do conjunto S_c de pontos de controle, definido nos métodos anteriores. Para cada uma dessas amostras s_i é definido um grafo de vizinhança, utilizado para definir um sistema, e um conjunto de pontos de controle que tem como função restringir o sistema gerado.

Amostragem. Seja $S = \{p_1, ..., p_n\}$ o conjunto de dados e $\{s_1, ..., s_{n_a}\}$ um subconjunto de amostras retirado de S, como nosso objetivo inicial é apenas projetar o conjunto de dados, essas amostras são escolhidas utilizando uma aproximação de agrupamento (em nossos experimentos utilizamos o método k-means com $k = n_a$).

Com intuito de dividir o conjunto de dados usamos as amostras para definir n_a subconjuntos $S_1, S_2, ..., S_{n_a} \subset S$ condicionados a

1. Cada conjunto S_i compreendem os dados em S mais próximos da amostra s_i que qualquer outra amostra s_j, com $j \neq i$.
2. $S = S_1 \cup S_2 \cup ... \cup S_{n_a}$.

Construção do grafo de vizinhança e seleção dos pontos de controle. Cria-se para cada um desses subconjuntos S_i um grafo de vizinhança, denotado por G_i. Cada nó em G_i representa um dado em S_i, e dois nós são conectados por uma aresta se pelo menos um deles está entre os vizinhos k-próximos do outro, por esse motivo G_i é conhecido também como grafo de vizinhos k-próximos (k-NNG). O parametro k é escolhido de forma a obter um bom negócio entre custo computacional e conexão de gráfico, uma vez que grandes valores de k representam maior custo para a construção do gráfico e pequenos valores representam S_i fracamente conectado, fornecendo projeções pobres.

Com o grafo de vizinhança G_i em mãos podemos definir um sistema laplaciano para projetar os dados de S_i para o $\mathbb{R}^2$, a fim de garantir uma única solução para a projeção, devemos impor restrições ao sistema que são dadas através das coordenadas cartesianas dos pontos de controle. Esse conjunto de pontos de controle é definido pela escolha aleatória de $\sqrt{m_i}$ dados de S_i, onde m_i é o número de dados em S_i.

Note que com essa construção cada S_i será projetado independentemente, e portanto nenhuma garantia pode ser dada de que subconjuntos vizinhos serão projetados próximos. Uma ideia bem simples para reaver a relação global existente é considerar o conjunto $C = C_1 \cup C_2 \cup ... \cup C_{n_a}$, onde C_i é o conjunto de pontos de controle correspondente a S_i, como um novo conjunto de dados e projetá-lo através de um método MDS.

Construção e solução do sistema. Para a construção do sistema laplaciano associado a S_i consideremos p^i um dado qualquer de $S_i = \{p^1, p^2, ..., p^{m_i}\}$ e seja $V_i = \{p_{i_1}, p_{i_2}, ..., p_{i_k}\}$ o conjunto de nós conectados a p^i em G_i. Considere (x_j, y_j) como sendo as coordenadas cartesianas dos pontos $p_{i_j} \in V_i$, assumindo que cada elemento de um conjunto de dados pode ser escrito como uma combinação convexa de seus vizinhos no espaço visual, podemos escrever a projeção de p^i como

$$(x_{p^i}, y_{p^i}) = \sum_{j=1}^{k} \alpha_{ij}(x_j, y_j) \tag{2.1}$$

onde (x_{p^i}, y_{p^i}) é a projeção de p^i em $\mathbb{R}^2$.

Fazendo i variar no conjunto $\{1, ..., m_i\}$, obtemos m_i equações vetoriais como a equação 2.1, olhando para a primeira e segunda coordenadas de todas as equações elas acabam por dar origem a dois sistemas lineares

$$LX = 0;\ LY = 0$$

onde X e Y são vetores contendo as coordenadas x e y dos dados projetados e $L = (l_{ij})$ é a matriz dada por

$$l_{ij} = \begin{cases} 1, & i = j \\ -\alpha_{ij}/\alpha_i^*, & i \neq j \ e \ p^j \in V_i \\ 0, & \text{caso contrário} \end{cases}$$

onde $\alpha_i^* = \sum_{j=1}^{k} \alpha_{ij}$. O peso α_{ij} pode ser definido como o inverso da distância entre os pontos p^i e p^j, no entanto, escolhemos α_{ij} simplesmente igual a 1, note que nesse caso $\alpha_{ij}/\alpha_i^* = 1/k$, onde k é o número de vizinhos de cada ponto, assim a soma dos elementos de cada linha da matriz L se torna igual a zero, justificando a nomeclatura laplaciano para os sistemas $LX = 0$ e $LY = 0$ conforme explicado no método LSP.

Pode ser mostrado que se o grafo de vizinhança G_i tem apenas uma componente conectada então o posto da matriz L é $m_i - 1$. Então assumindo G_i assim conectada, os sistemas lineares admitem uma solução não trivial.

Novamente nos deparamos com o problema da falta de informações geométricas, o que produz soluções pobre, como anteriormente, solucionamos esse problema através dos pontos de controle. Os pontos de controle fornecem ao sistema informações geométricas por meio de um esquema de penalidade.

Note que o método PLP nada mais é do que n_a aplicações do método LSP aos conjuntos $S_1, S_2, ..., S_{n_a}$, diferindo apenas no modo como os pontos de controle adicionam suas informações geométricas ao sistema. Em outras palavras, PLP com $n_a = 1$ resulta no LSP.

Apresentamos um esboço do algoritmo PLP (Algoritmo 5), sem as modificações que visam recuperar as relações globais.

Expomos também o algoritmo PLP onde nos preocupamos com a recuperação das relações globais (Algoritmo 6), existentes no conjunto de dados original.

Algorithm 5 Algoritmo PLP (com perda das relações globais)

Require: Conjunto de dados S.
  Obtenha o conjunto de amostras $\{s_1, s_2, ..., s_{n_a}\}$ de S
  Dividida o conjunto $S = S_1 \cup S_2 \cup ... \cup S_{n_a}$ a partir das amostras s_i's
  for all $S_i \subset S$ **do**
    Defina o grafo de vizinhança NS_i associado a S_i
    Construa os sistemas laplacianos $LX = 0$ e $LY = 0$
    Defina o conjunto de pontos de controle C_i associado a S_i
    Projete o conjunto C_i através de um método MDS
    Restrinja os sistemas laplacianos através de um método de penalidade
    Resolva os sistemas
  end for

Algorithm 6 Algoritmo PLP

Require: Conjunto de dados S.
  Obtenha o conjunto de amostras $\{s_1, s_2, ..., s_{n_a}\}$ de S
  Dividida o conjunto $S = S_1 \cup S_2 \cup ... \cup S_{n_a}$ a partir das amostras s_i's
  for all $S_i \subset S$ **do**
    Defina o conjunto de pontos de controle C_i associado a S_i
  end for
  Defina o conjunto $C = C_1 \cup C_2 \cup ... \cup C_{n_a}$
  Projete o conjunto C através de um método MDS
  for all $S_i \subset S$ **do**
    Defina o grafo de vizinhança NS_i associado a S_i
    Construa os sistemas laplacianos $LX = 0$ e $LY = 0$
    Restrinja os sistemas laplacianos através de um método de penalidade
    Resolva os sistemas
  end for

Exemplificando o método PLP

Aplicamos o método PLP ao conjunto de dados Íris para obtermos uma projeção do mesmo, a Figura 2.24 mostra o resultado dessa projeção. Podemos observar alguns aglomerados de pontos na projeção, cada um desses aglomerados são o resultado da projeção de um dos conjuntos S_i's.

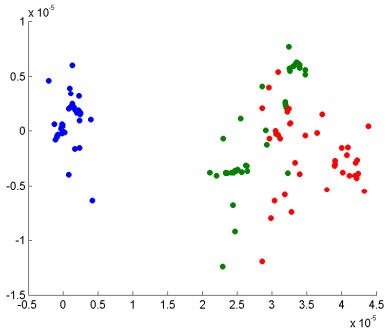
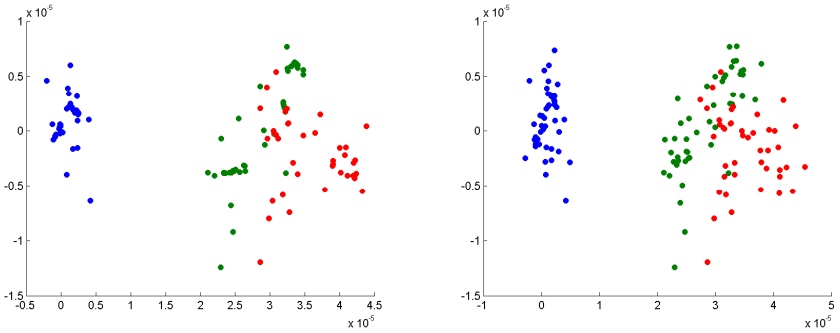


Figura 2.24: Projeção do conjunto de dados Íris através do método PLP .

Para eliminar a presença desses algomerados utiliza-se a mesma ideia que ajudou a eliminar a linearidade no método LSP. É considerado um aumento no número de pontos de controle, assim em vez de utilizar $\sqrt{m_i}$ pontos de controle para restringir o sistema associado a S_i utiliza-se $0.5 * m_i$. A Figura 2.25 nos mostra uma projeção do conjunto de dados Íris, onde o número de pontos de controle m_i associado a cada conjunto S_i igual a $\sqrt{m_i}$ (Figura 2.25(a)) e $0.5 * m_i$ (Figura 2.25(b)).



(a) $\sqrt{m_i}$ (b) $0.5 * m_i$

Figura 2.25: Modificação do número de pontos de controle utilizado para restringir os sistemas associados aos conjuntos S_i's .

2.6 Método LAMP

O método Local Affine Multidimensional Projection [10], também denominado LAMP, se baseia em uma fórmula matemática derivada da teoria de mapeamento ortogonal para construir uma família de mapeamentos afins ortogonais, um para cada dado a ser projetado.

Veremos que a formulação do método LAMP pode ser ligeiramente modificada para torná-lo um método mais rápido. Além disso esse método é muito robusto com respeito ao número de pontos de controle, métodos mais famosos na literatura possuem uma limitação no número mínimo de pontos de controle, na prática usam $n_c = \sqrt{n}$, enquanto que LAMP apresenta baixa distorção mesmo quando poucos pontos de controle são usados. Sendo essa uma vantagem do método LAMP.

Cálculo dos mapeamentos afins. Considere um conjunto de dados $S = \{p_1, ..., p_n\} \subset R^m$, um conjunto de pontos de controle $S_c = \{a_1, a_2, ..., a_{n_c}\} \subset S$, e a projeção $Pc = \{a'_1, a'_2, ..., a'_{n_c}\} \subset \mathbb{R}^2$ dos pontos de controle.

Seja $p \in S$ um dado qualquer, o esquema LAMP visa encontrar a melhor transformação afim $f_p(x) = xM + t$ que minimiza o somatório

$$\sum_{i=1}^{n_c} \alpha_i ||f_p(a_i) - b_i||^2, \tag{2.2}$$

onde a matriz $M_{m\times 2}$ ortogonal e o vetor $t_{1\times 2}$ são parâmetros a ser determinados e α_i são pesos escalares definidos por

$$\alpha_i = \frac{1}{||a_i - p||^2},$$

com essa definição dos pesos α_i's um ponto de controle distante do ponto de avaliação p tem pouca influência na projeção desse ponto. De fato, dado um ponto de controle a_i distante de p temos um valor grande de $||a_i - p||$ o que implicam em α_i pequeno, logo a expressão $\alpha_i||f_p(a_i) - b_i||^2$ já é próxima a zero. Portanto o método LAMP é de certa forma um método local.

Alguns aspectos da construção acima merecem destaque, a restrição $M^T M = I$ por exemplo assegura uma certa rigidez as transformações afins resultantes, isto é, os dados só podem ser rodados e transladados durante o processo de projeção, característica bastante desejável quando se quer preservar distâncias tanto quanto possível, como é o nosso caso. Além disso a restrição de ortogonalidade garante que erros introduzidos ao se projetar os pontos de controle não sejam drasticamente propagados.

Outro aspecto importante é a dependência que os pesos $\alpha_i = \alpha_i(p)$ tem do ponto de avaliação p, com isso uma transformação afim é obtida para cada dado.

Por fim, o somatório na equação 2.2 percorre todos os pontos de controle, podemos no entanto restringi-lo para levar em conta apenas os pontos de controle em uma vizinhança de p, tornando o processo realmente local.

O problema de minimização do somatório possui uma solução não tão imediata, o próximo tópico é destinado a apresentá-la.

Solução do problema de minimização. Utilizando a expressão para f_p, podemos reescrever a equação 2.2 como

$$\sum_i \alpha_i ||a_i M + t - b_i||^2, \tag{2.3}$$

escrevemos então $t = (t_1, t_2)$, já que $t \in \mathbb{R}^2$, em seguida consideramos a derivada parcial em relação a t_1 da expressão

$$||a_i M - b_i + t||^2 =< a_i M - b_i + t, a_i M - b_i + t >,$$

onde obtemos

$$2 < (1,0), a_i M - b_i + t >$$

derivando-se então o somatório em 2.3 com relação a t_1 e igualando a zero chegamos a

$$\begin{aligned}
&\sum_i \alpha_i < (1,0), a_i M - b_i + t >= 0 \\
&\sum_i \alpha_i [(a_i M)_1 - (b_i)_1 + t_1] = 0 \\
&t_1 \sum_i \alpha_i = \sum_i \alpha_i [(b_i)_1 - (a_i M)_1],
\end{aligned}$$

considerando $\alpha = \sum_i \alpha_i$ e que as contas são inteiramente analogas para a segunda coordenada, temos:

$$\begin{aligned}
t_1 \alpha &= \sum_i \alpha_i [(b_i)_1 - (a_i M)_1] \\
t_2 \alpha &= \sum_i \alpha_i [(b_i)_2 - (a_i M)_2]
\end{aligned}$$

Finalmente,

$$\begin{aligned}
t &= (t_1, t_2) \\
&= (\frac{1}{\alpha} \sum_i \alpha_i (b_i)_1 - \frac{1}{\alpha} \sum_i \alpha_i (a_i M)_1, \frac{1}{\alpha} \sum_i \alpha_i (b_i)_2 - \frac{1}{\alpha} \sum_i \alpha_i (a_i M)_2)
\end{aligned}$$

escrevendo $\tilde{a} = \dfrac{\sum_i \alpha_i a_i}{\alpha}$ e $\tilde{b} = \dfrac{\sum_i \alpha_i b_i}{\alpha}$, obtemos:

$$t = \tilde{b} - \tilde{a} M$$

Substituindo essa expressão de t no somatório em 2.3 o problema de minimização pode ser escrito na forma

$$\sum_i \alpha_i ||(a_i - \tilde{a}) M - (b_i - \tilde{b})||^2 \text{ , desde que } M^T M = I$$

e logo

$$\sum_i \alpha_i ||\hat{a}_i M - \hat{b}_i||^2, \text{ desde que } M^T M = I \tag{2.4}$$

onde $\hat{a}_i = a_i - \tilde{a}$ e $\hat{b}_i = b_i - \tilde{b}$.

Considerando as matrizes A e B dadas por

$$A = \begin{bmatrix} \sqrt{\alpha_1}\hat{a}_1 \\ \sqrt{\alpha_2}\hat{a}_2 \\ \vdots \\ \sqrt{\alpha_{n_c}}\hat{a}_{n_c} \end{bmatrix}, B = \begin{bmatrix} \sqrt{\alpha_1}\hat{b}_1 \\ \sqrt{\alpha_2}\hat{b}_2 \\ \vdots \\ \sqrt{\alpha_{n_c}}\hat{b}_{n_c} \end{bmatrix}$$

podemos reescrever nosso problema na forma matricial

$$||AM - B||_F, \text{ desde que } M^T M = I,$$

onde $||.||_F$ é norma de Frobenius. De fato,

$$\begin{aligned} ||AM - B||_F^2 &= || \begin{bmatrix} \sqrt{\alpha_1}\hat{a}_1 M \\ \sqrt{\alpha_2}\hat{a}_2 M \\ \vdots \\ \sqrt{\alpha_{n_c}}\hat{a}_{n_c} M \end{bmatrix} - \begin{bmatrix} \sqrt{\alpha_1}\hat{b}_1 \\ \sqrt{\alpha_2}\hat{b}_2 \\ \vdots \\ \sqrt{\alpha_{n_c}}\hat{b}_{n_c} \end{bmatrix} ||_F^2 \\ &= \sum_{i=1}^{n_c}\sum_{j=1}^{m} |\sqrt{\alpha_i}(\hat{a}_i M - \hat{b}_i)_j|^2 \\ &= \sum_{i=1}^{n_c} \alpha_i \sum_{j=1}^{m} ((\hat{a}_i M - \hat{b}_i)_j)^2 \\ &= \sum_{i=1}^{n_c} \alpha_i ||\hat{a}_i M - \hat{b}_i||^2 \end{aligned}$$

Assim mostramos que minimizar o somatório em 2.4 é equivalente a minimizar $||AM - B||_F^2$, mas é claro que minimizando $||AM - B||_F$ minimizamos também $||AM - B||_F^2$ e vice e versa. Logo

$$\text{minimizar } ||AM - B||_F \Leftrightarrow \text{ minimizar } \sum_{i=1}^{n_c} \alpha_i ||\hat{a}_i M - \hat{b}_i||^2$$

O problema de minimização $||AM - B||_F$ restrito a $M^T M = I$ já é bem conhecido e sua solução é dada por

$$M = UV,$$

onde UDV é a decomposição em valores singulares (SVD) da matriz $A^T B$ (seção 1.3). Tendo a matriz M em mãos podemos calcular p', a projeção de p, como

$$\begin{aligned} p' &= f_p(p) \\ &= (p - \tilde{a})M + \tilde{b}. \end{aligned}$$

Note que a matriz $A^T B$ é uma matriz $m \times 2$ (com apenas duas colunas), pois a matriz A^T é $m \times n_c$ e a matriz B é $n_c \times 2$, lembre-se que os elementos b_i, estão no espaço visual (bidimensional no nosso caso), consequentemente $\hat{b}_i$ está em $\mathbb{R}^2$, o que assegura o fato de B ser $n_c \times 2$. Por esse motivo a matriz $A^T B$ pode ser decomposto muito rapidamente com pacotes SVD. Os pacotes de decomposição em valores singulares utilizados são rotinas do programa MATLAB.

Apresentamos o método LAMP também em formato de algoritmo (Algoritmo 7). O algoritmo LAMP apresenta uma qualidade muito significativa que é a faciliade em ser implementado, característica dificilmente encontrada em outros métodos.

Algorithm 7 Algoritmo LAMP

Require: Conjunto de dados S, pontos de controle Sc, e a projeção Pc de Sc.
  for all $p \in S$ **do**
    Cálcule os pesos α_i
    Cálcule $\tilde{a}$ e $\tilde{b}$
    Construa as matrizes A e B
    Cálcule a decomposição em valores singulares UDV para a matriz $A^T B$
    Faça $M = UV$
    Cálcule o mapeamento $p' = (p - \tilde{a})M + \tilde{b}$ de p
  end for

Exemplificando o método LAMP

Mostraremos as principais etapas da projeção do conjunto de dados Íris via método LAMP. Temos S = conjunto de dados Íris, $Sc = \{a_1, ..., a_{12}\} = \{p_4, p_8, p_{34}, p_{49}, p_{70}, p_{74}, p_{87}, p_{94}, p_{95}, p_{102}, p_{106}, p_{113}\}$ obtido de S através da técnica k-means e $Pc = \{b_1, ..., b_{12}\} = \{a'_1, ..., a'_{12}\}$ que é uma projeção dos pontos em Sc através do esquema Force. Assim temos os dados de entrada necessários para rodar o algoritmo LAMP.

Considere o ponto $p_1 = (5.1, 3.5, 1.4, 0.2) \in S$, calculamos os pesos α_i que nesse caso são dados por

$$\alpha_i = \frac{1}{||a_i - p_1||^2}, \ i = 1, ..., 12$$

levando em conta que $\alpha = \sum_{i=1}^{12} \alpha_i = 14.1099$, computamos $\tilde{a}$ e $\tilde{b}$

$$\begin{aligned} \tilde{a} &= \frac{\sum_{i=1}^{12} \alpha_i a_i}{\alpha} \\ &= (5.2214, 3.3985, 1.9627, 0.3920) \\ \tilde{b} &= \frac{\sum_{i=1}^{12} \alpha_i b_i}{\alpha} \\ &= (0.3604, 2.3359) \end{aligned}$$

Assim as matrizes $A_{12,4}$ e $B_{12,2}$ são facilmente definidas. Aplicando uma

rotina do Matlab, obtemos a decomposição em valores singulares da matriz

$$A^T B = \begin{bmatrix} 0.9545 & -6.2761 \\ -1.8137 & 5.0091 \\ 4.5163 & -20.1135 \\ 1.8014 & -8.1897 \end{bmatrix}$$

que é

$$U = \begin{bmatrix} -0.2668 & -0.5565 \\ 0.2227 & -0.8289 \\ -0.8686 & -0.0198 \\ -0.3533 & -0.0534 \end{bmatrix}, D = \begin{bmatrix} 23.7339 & 0 \\ 0 & 0.8061 \end{bmatrix} \text{ e } V = \begin{bmatrix} -0.2198 & 0.9755 \\ 0.9755 & 0.2198 \end{bmatrix}$$

logo a matriz M dada por UV fica sendo

$$M = \begin{bmatrix} -0.4843 & -0.3826 \\ -0.8576 & 0.0350 \\ 0.1716 & -0.8517 \\ 0.0255 & -0.3564 \end{bmatrix}$$

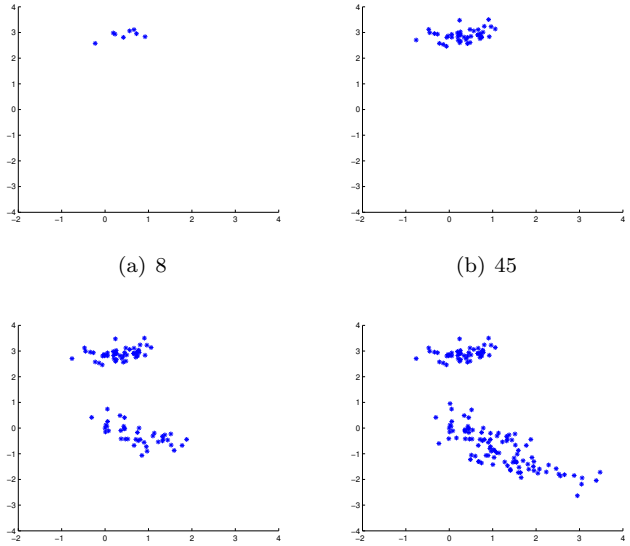
Finalmente a projeção p_1' do ponto p_1 é

$$\begin{aligned} p_1' &= (p_1 - \tilde{a})M + \tilde{b} \\ &= (0.2307, 2.9336). \end{aligned}$$

Esse processo é repitido para todos os pontos $p_i,\ i = 2, ..., 150$, sendo desnecessário a sua aplicação aos pontos de controle uma vez que as coordenadas desses pontos no espaço visual já são conhecidas.

Na Figura 2.26 exibimos algumas etapas da projeção do conjunto de dados Íris através do método LAMP, monstramos a projeção imediatamente após projetarmos os pontos p_8, p_{45}, p_{90} e p_{150}.

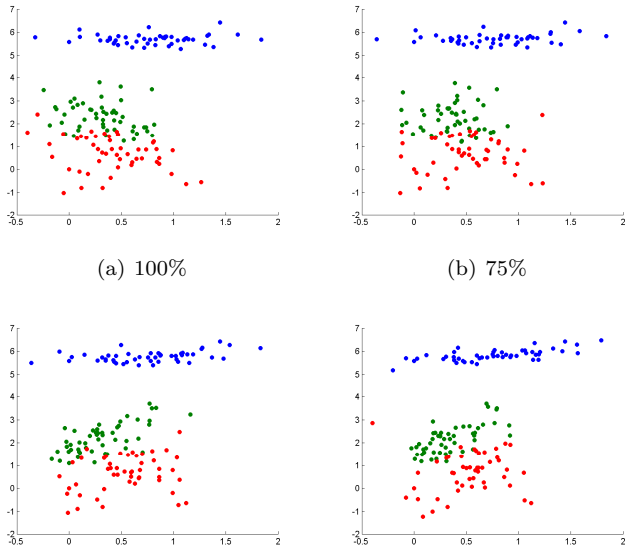
No método LAMP podemos restringir o somatório em (2.2) para levar em conta apenas os pontos de controle em uma vizinhança de p, com isso o método se torna verdadeiramente local, pois quanto maior o número de pontos de controle considerados no somatório menos local é a projeção de p. A Figura 2.27 nos mostra uma projeção do conjunto de dados Íris levando em conta 100%, 75%, 50% e 25% dos pontos de controle mais próximos a p. Podemos perceber que a qualidade da projeção não se altera drasticamente quando a porcentagem de pontos mais próximos a p diminui de 100% para 25%, vale a pena destacar que uma pequena alteração começa a ser observada quando a porcentagem chega a 25%, um dado pertencente a classe vermelha é projetado consideravelmente longe da classe vermelha.



(a) 8 (b) 45

(c) 90 (d) 150

Figura 2.26: Projeção do conjunto de dados Íris através do método LAMP. Projeções parciais com 8, 45, 90 e 150 pontos projetados.



(a) 100% (b) 75%

(c) 50% (d) 25%

Figura 2.27: Projeção do conjunto de dados Íris através do método LAMP. Levando em conta 100%, 75%, 50% e 25% dos pontos de controle mais próximos a p.

Capítulo 3

Resultados

A finalidade dessa seção é mostrar e comparar os resultados produzidos pelos métodos de projeção apresentados na seção anterior ao projetar alguns conjuntos de dados. Os conjuntos de dados utilizados para produzir as comparações, juntamente com suas principais características: tamanho e dimensionalidade, são apresentados na Tabela 3.1. Os conjuntos de dados Íris, Wine, Housing e Abalone foram fornecidos pelo grupo UCI Machine Learning Repository [7], que disponibililiza em seu site não só esses, mas muitos outros conjuntos de dados.

Nome	Tamanho	Dimensionalidade
Iris	150	4
Wine	178	13
Housing	506	14
Abalone	4177	8

Tabela 3.1: Conjuntos de dados

O primeiro conjunto de dados utilizado nos testes foi o conjunto de dados Íris, em seguida nos preocupamos em lidar com conjuntos mais complexos em termos de tamanho e dimensionalidade, o conjunto de dados Wine nos foi útil para testar a consistência de nossos métodos quando a dimensionalidade aumenta, e o conjunto de dados Housing quando o tamanho do conjunto de dados cresce muito. Encerramos com um conjunto de dados bastante complicado, o Abalone, contendo mais de 4 mil dados em um espaço de dimensão 8.

A comparação de resultados é um tema bastante polêmico quando se lida com projeção multidimensional, no tópico a seguir definimos as técnicas de comparação que utilizamos para analisar as projeções obtidas.

3.1 Técnicas de comparação de projeções

Nós utilizamos como técnicas de comparação a análise dos layouts produzidos pelas projeções, ou seja, a qualidade visual produzida pelo método e uma medida conhecida como *stress normalizado* [10]. O stress é um valor real associado a um conjunto de dados e a sua projeção que relaciona as distâncias m-dimensionais com as distâncias bidimensionais. O stress normalizado é dado por

$$Stress = \frac{\sum_{ij}(\delta_{ij} - d_{ij})^2}{\sum_{ij}\delta_{ij}^2}$$

onde δ_{ij} e d_{ij} são as distâncias entre as instâncias i e j nos espaços original e visual, respectivamente. O stress nos dá uma ideia numérica do quanto a distância entre os pontos foi preservada durante o processo de projeção. Valores altos de stress indicam projeções de baixa qualidade, enquanto que valores menores indicam boas projeções.

Outra técnica utilizada para qualificar uma projeção é o gráfico de dispersão. O gráfico de dispersão é um conjunto de pontos no plano, onde cada ponto representa um par de instâncias, a 1ª coordenada do ponto é igual a distância original e a 2ª é igual distância projetada do par de instâncias que representa. Todos os pares de instâncias presentes no conjunto de dados são levados em conta e, antes de serem utilizadas as distâncias originais e visuais são normalizadas nos seus respectivos espaços. O gráfico de dispersão nos dá uma ideia visual do quanto a distância entre os pontos foi preservada durante o processo de projeção. Gráficos com pontos próximos a diagonal representam projeções com uma qualidade boa e gráficos com pontos mais distantes da diagonal projeções ruins.

3.2 Conjunto de dados Íris

Começamos nossas análises de desempenho dos métodos através do conjunto de dados Íris. Esse conjunto de dados é simples, contando com apenas 150 plantas do genero íris cada uma classificada a partir de quatro atributos relacionados as medidas de suas sépalas e pétalas.

As projeções do conjunto de dados Íris produzidas pelos métodos NNP, Force, LSP, LAMP e PLP são exibidas na Figura 3.1. Dois comentários merecem destaque: primeiro o método Force nada mais é do que a projeção do conjunto de dados pelo método NNP seguido do esquema de melhoria Force, como já mencionado, o que vale ressaltar é que essa pré-projeção é a mesma que a do método NNP exibida na Figura 3.1(a); e o segundo é que o método LSP tem três métodos de soluções possíveis para o sistema formado, nessa parte da comparação usamos o denominado anteriormente por Novas Linhas.

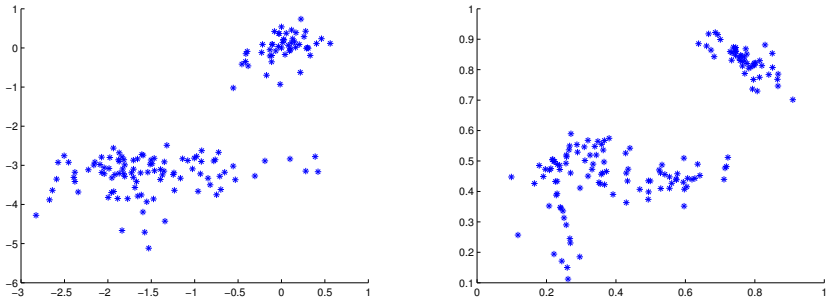
Ao observarmos as projeções constatamos que todos os métodos comportam-se de maneira semelhante. Ao posicionar pontos no plano cartesiano que representam as plantas podemos notar a presença de um grupo menos denso separado de um outro grupo mais denso que esse primeiro.

A Tabela 3.2 mostra o stress produzido nas projeções apresentadas na Figura 3.1. A tabela também mostra o tempo gasto para efetuar as projeções exibidas na figura. A projeção produzida pelo método Force é a que apresenta o segundo menor valor de stress, no entanto, o tempo gasto na projeção através desse método é o maior. Veremos que essa diferença no tempo se torna mais acentuada ainda para conjunto de dados maiores, devido ao grande esforço computacional desse método. Os métodos que fazem uso de pontos de controle (LSP, LAMP e PLP) não apresentam vantagens em termos de tempo para o conjunto de dados Íris como podemos observar na tabela, mas esse comportamento já era esperado uma vez que esse conjunto de dados é razoavelmente pequeno, em conjuntos de dados maiores veremos a capacidade que esse métodos têm de realizar a projeção com um esfoço computacional reduzido, comparado com aqueles que não fazem uso de pontos de controle (NNP e Force). O método PLP realiza a projeção do conjunto de dados Íris com maior rapidez se comparado ao método LSP. O método LAMP realiza a projeção com um tempo comparável aos tempos dos outros métodos, no entanto isso ocorre devido a utilização de 100% dos pontos de controle na projeção de cada ponto, podendo ser reduzido até 25%, nesse caso o esforço computacional é consideravelmente menor.

Método	Stress	Tempo
NNP	0.0558	0.0507 s
Force	0.0141	0.7190 s
LSP	0.0567	0.1050 s
PLP	0.0637	0.0464 s
LAMP	0.0069	0.0982 s

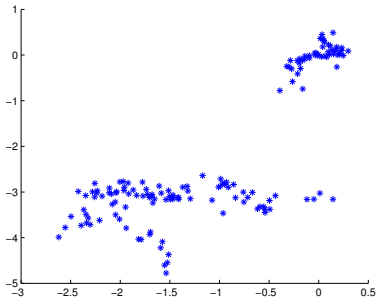
Tabela 3.2: Resultados obtidos nas projeções do conjunto de dados Iris ilustradas na Figura 3.1.

É interessante observar esses resultados com as separações de classes fornecidas pelo próprio criador do conjunto de dados, R.A.Fisher. A Figura 3.2 mostra as mesmas projeções exibidas na Figura 3.1 adicionando cores diferentes a cada

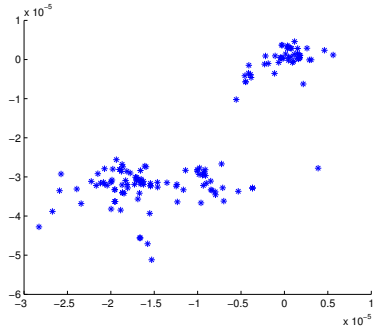


(a) Método NNP

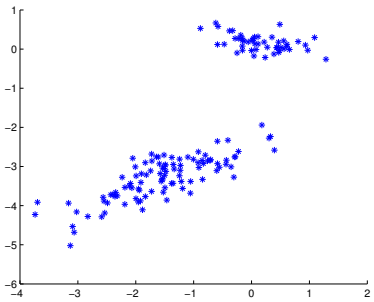
(b) Método Force



(c) Método LSP

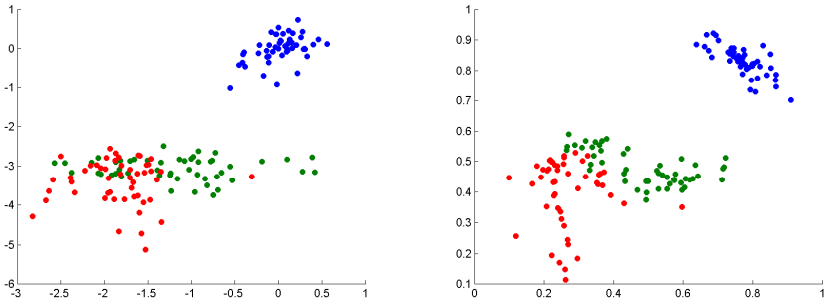


(d) Método PLP



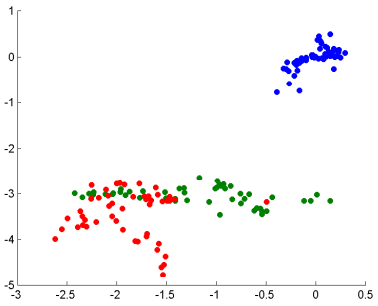
(e) Método LAMP

Figura 3.1: Projeções do conjunto de dados Íris através dos métodos NNP, Force, LSP, LAMP e PLP.

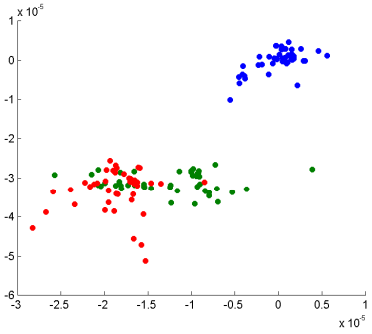


(a) Método NNP

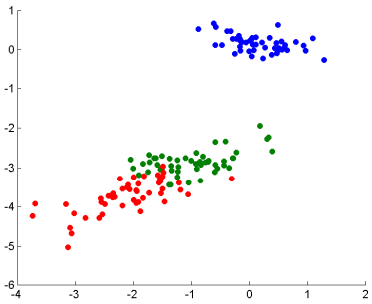
(b) Método Force



(c) Método LSP



(d) Método PLP



(e) Método LAMP

Figura 3.2: Projeções do conjunto de dados Íris através dos métodos NNP, Force, LSP, LAMP e PLP, com destaque para as diferentes classes presentes no conjunto de dados.

uma das classes existentes. A primeira classe com as 50 flores do tipo íris-setosa foi colorida com a cor azul, a segunda com as 50 flores do tipo íris-versicolor com a cor verde e a terceira com as 50 flores do tipo íris-virginica com a cor vermelha.

Pode-se observar na Figura 3.2 um resultado muito próximo a descrição dada por Fisher sobre o conjunto de dados Íris, ele afirma que uma espécie é linearmente separada das outras 2, e estas últimas não são separadas linearmente. Note que a classe azul foi projetada separadamente das classes verde e vermelha, e esta últimas possuem uma certa proximidade não podendo ser separadas. Nesse sentido todos os métodos atuaram como o esperado, fornecendo alguma confiabilidade aos códigos implementados.

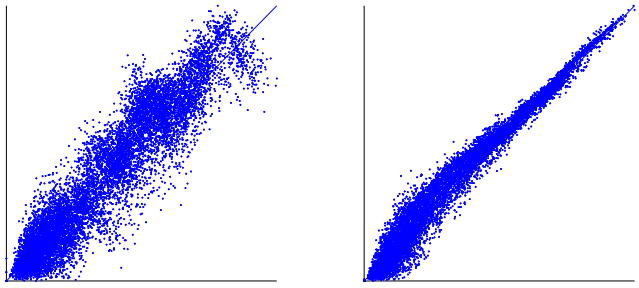
No método LSP os pontos de controle são projetados, em seguida resolve-se dois sistemas lineares cujas soluções são as coordenadas em $\mathbb{R}^2$ dos pontos restante. Os pontos obtidos ao se resolver os sistemas lineares deverão estar no fecho convexo dos pontos de controle, devido a própria formulação do método. Assim caso o número de pontos de controle seja reduzido o layout produzido é "linear" (aproximado a um curva). Caso poucos pontos de controle sejam utilizados pelo método PLP aparecem alguns aglomerados de pontos na projeção, isso ocorre pois o método PLP nada mais é do que a aplicação do método LSP a subconjuntos do conjunto de dados.

A Figura 3.3 mostra o gráfico de dispersão obtido ao se projetar o conjunto de dados Íris através dos nossos 5 métodos de estudo NNP, Force, LSP, LAMP e PLP. Todos os layouts produzidos são muito próximos a diagonal.

No gráfico de dispersão produzido pelo método NNP (Figura 3.3(a)) nota-se muitos pontos dispersos, isto é, que se afastam consideravelmente da diagonal da diagonal. O esquema de melhoria Force ao ser aplicado sobre os resultados do método NNP aproxima da diagonal não só esses pontos dispersos como também muitos outros, assim o layout produzido por esse segundo método se torna mais próximo da diagonal, contudo ocorre a formação de um "buraco" entre os pontos e a diagonal (Figura 3.3(b)), veremos que a medida que o número de iterações utilizada pelo esquema de melhoria Force aumenta esse "buraco" tende a diminuir.

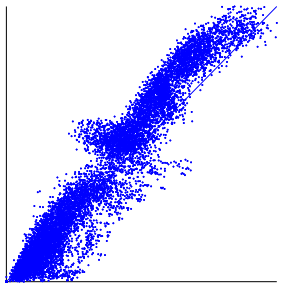
O método LSP cria um gráfico de dispersão onde constatamos a presença de pequenos conjuntos de pontos, uma distribuição dos pontos semelhante a essa também pode ser observada no gráfico de dispersão do método PLP, no entanto a separação dos pontos nesse segundo gráfico de dispersão praticamente não existe, além disso o layout nesse caso é mais próximo à diagonal do que no primeiro (Figuras 3.3(c) e 3.3(d)).

Na Figura 3.3(e) vemos que o método LAMP produz um gráfico de dispersão de ótima qualidade, pois podem ser observados poucos pontos dispersos e, além disso o layout desse método parece ser o mais próximo da diagonal.

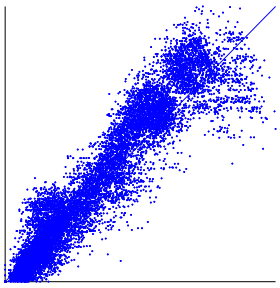


(a) Método NNP

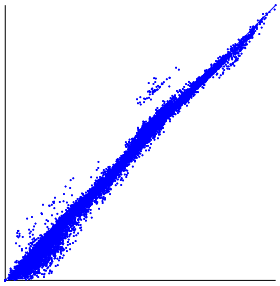
(b) Método Force



(c) Método LSP



(d) Método PLP



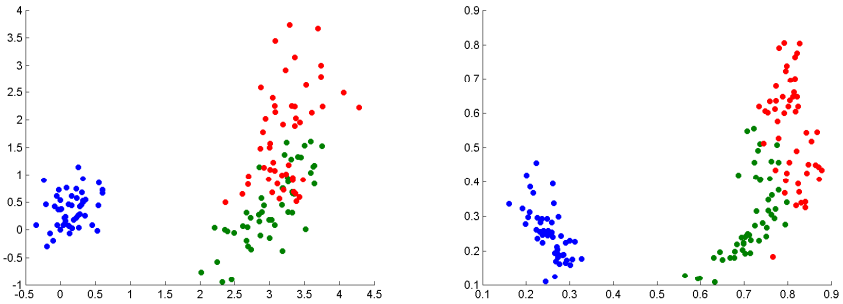
(e) Método LAMP

Figura 3.3: Gráfico de dispersão das projeções do conjunto de dados Íris.

Antes de passarmos ao conjunto de dados Wine, é importante analisarmos o esquema de melhoria Force independentemente, isto é, verificar a melhoria visual e em termos de stress que esse método proporciona a uma projeção.

Constatando a eficiência do esquema de melhoria Force

O método Force tem duas etapas bem distintas, a projeção do conjunto de dados através do método NNP e o esquema de melhoria Force que atua sobre o resultado dessa projeção. A ideia é analisar separadamente essa segunda etapa do método Force, ou seja, deixar clara as vantagens e a desvantagens apresentadas pelo esquema de melhoria Force.



(a) Projeção inicial através do NNP. (b) Após aplicar o esquema de melhoria Force.

Figura 3.4: Método Force = projeção NNP + esquema de melhoria Force.

A Figura 3.4 apresenta uma projeção via método Force, a primeira etapa do método consistente na projeção do conjunto de dados pelo método NNP que pode ser vista na Figura 3.4(a), enquanto que o resultado de se aplicar o esquema de melhoria sobre essa projeção pode ser visto na Figura 3.4(b), o tempo e o stress de ambas as etapas são exibidos na Tabela 3.3.

Nome	Stress	Tempo
Projeção NNP	0.0462	0.0353 s
Esquema Force	0.0286	0.6662 s

Tabela 3.3: Análise de cada uma das etapas que constituem o método Force. Os resultados de tempo e stress apresentados nessa tabela foram obtidos através das projeções presentes na Figura 3.4.

Podemos observar uma redução considerável no valor do stress, contudo o tempo gasto para obter tal melhora foi significativo. Caso o usuário esteja interessado apenas no resultado da projeção esse método se torna uma poderosa ferramenta. Esse aumento no esforço computacional não chega a ser tão grande quando lidamos com conjunto de dados pequenos, no entanto veremos que quando conjuntos de dados maiores são levados em consideração o tempo gasto na projeção cresce muito.

Uma mudança no layout também pode ser observada, na Figura 3.4(b) as classes verde e vermelha não chegam a ser separadas uma da outra (algo espe-

rado devido as características do conjunto de dados Íris), mas os dados projetados dessas duas classes se misturam bem menos se comparados com o resultado da projeção inicial, apresentado na Figura 3.4(a).

Um outro fator interessante de observarmos no método Force diz respeito ao número de iterações utilizadas no esquema de melhoria Force. Apresentamos na Figura 3.5 um gráfico do stress em função do número de iterações para uma projeção em particular, com isso podemos analisar a influência desse fator na qualidade da projeção.

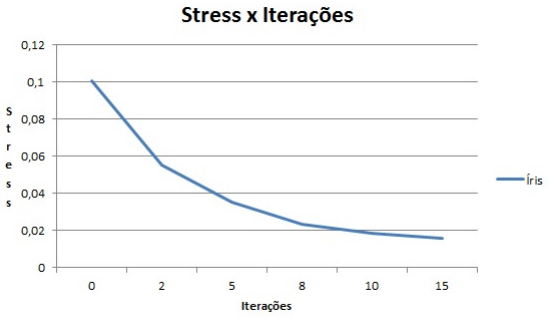
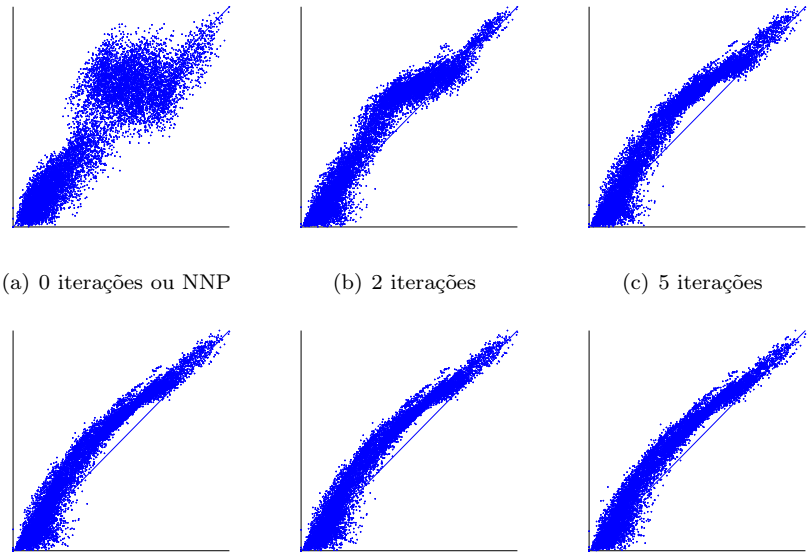


Figura 3.5: Gráfico do Stress x Iterações. Stress produzido pelo método Force em função do número de iterações utilizada pelo mesmo.



(a) 0 iterações ou NNP (b) 2 iterações (c) 5 iterações

(d) 8 iterações (e) 10 iterações (f) 15 iterações

Figura 3.6: Gráficos de dispersão de uma projeção particular do conjunto de dados Íris obtida através do método Force.

Também podemos analisar o comportamento do gráfico de dispersão em função do número de iterações (Figura 3.6). Quando aumentamos o número de

iterações do esquema de melhoria Force os pontos tendem a se aproximar da diagonal, pode-se observar um espaçamento entre os pontos e a diagonal, contudo com o aumento das iterações esse espaçamento tende a diminuir. Portanto os dados projetados estão, em geral, sendo movidos para posições mais adequadas a medida que aplicamos mais iterações do esquema de melhoria Force.

Com base nos gráficos de stress por iterações e de dispersão (Figuras 3.5 e 3.6), adotamos para o conjunto de dados Íris, 15 iterações do esquema de melhoria Force, pois este valor nos pareceu uma ótima escolha se levamos em conta: esforço computacional e qualidade da projeção. Um estudo semelhante ao realizado para o conjunto de dados Íris é feito para os demais conjuntos de dados, isto é, uma análise da qualidade da projeção em função do número de iterações do esquema Force, com base no stress e nos gráficos de dispersão, apartir dos resultado obtidos achamos interessante que o método Force faça uso de 15 iterações do esquema de melhoria Force para todos os conjuntos de dados.

Outro fator que merece destaque é a fração de Δ_{ij} que esse método utiliza. Seu valor tem uma forte influência sobre a projeção e em todos os conjuntos de dados fizemos alguns testes até chegar a valores considerados bons, que são aqueles que resultam em projeções de baixo stress e com bons layouts. Assim a fração de Δ_{ij} para o conjunto Íris foi tomada como sendo 0.01, enquanto que para os conjuntos Wine e Housing usamos 0.001 e para o Abalone utilizamos 10^{-6}.

O método LSP durante a projeção do conjunto de dados gera e resolve um sistema linear. A forma como esse tal sistema é resolvido merece destaque uma vez que foram propostos três formas distintas de resolvê-lo, denominadas por Novas Linha, Penalidade e Evolução. A seguir comparamos os resultados produzidos pela escolha de cada um desses métodos.

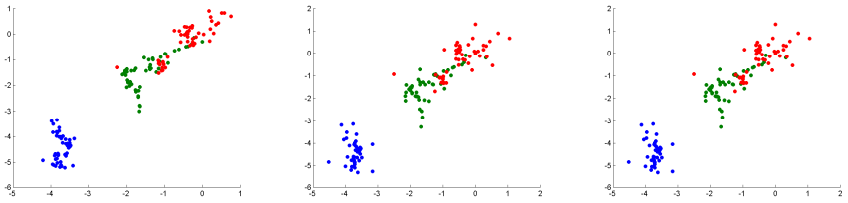
Métodos Novas Linhas, Penalidade e Evolução.

Os métodos Novas Linhas, Penalidade e Evolução são apenas uma componente do algoritmo LSP, então para facilitar na comparação desses métodos designaremos por LSP, LSPpenalidade e LSPevolucao o algoritmo LSP em que se utiliza o método Novas Linhas, Penalidade e Evolução, respectivamente, na solução do sistema formado pelo LSP.

Uma característica do método LSP é que os pontos de controle são fixados pelas novas linhas, no entanto a solução do sistema ainda é aproximada, pois os sistemas são resolvidos através do método de mínimos quadrados. O método LSPpenalidade possibilita pequenos reajustes aos pontos de controle, uma vez que esse método aproxima o sistema original. O método LSPevolucao por sua vez utiliza um sistema iterativo que converge para a solução exata.

A Figura 3.7 apresenta o resultado da projeção do conjunto de dados Íris através de cada um desses métodos. Podemos observar que as projeções diferem muito pouco quando nos preocupamos com o layout produzido (isto é, com os resultados da projeção em $\mathbb{R}^2$) o que é de se esperar para os métodos LSPpenalidade e LSPevolucao uma vez que esses métodos apenas resolvem de modo diferente o mesmo sistema, além disso todos os três métodos separam as espécies de plantas do tipo íris como esperado.

Os resultados de stress e o tempo gasto produzidos pelas projeções apresentadas na Figura 3.7 podem ser observados na Tabela 3.4. Os métodos LSPpenalidade e LSPevolucao apresentam bons valores de stress superando o método



(a) LSP (b) LSPpenalidade (c) LSPevolucao

Figura 3.7: Projeções do conjunto de dados Íris através dos métodos LSP, LSPpenalidade e LSPevolucao com destaque das classes.

LSP, e todos os três métodos realizam as projeções com tempos muito próximos.

Método	Stress	Tempo
LSP	0.0251	0.1194 s
LSPpenalidade	0.0222	0.1199 s
LSPevolucao	0.0222	0.1330 s

Tabela 3.4: Resultados obtidos ao se utilizarem os métodos Novas Linhas, Penalidade e Evolução no LSP durante as projeções do conjunto de dados Íris apresentadas na Figura 3.7.

O resultado obtido ao se aplicar o método LSPevolucao ao conjunto de dados Íris (Figura 3.7) foi obtido com 60 iterações. Achamos interessante avaliar o comportamento do stress a medida que o número de iterações cresce. A Figura 3.8 exibe um gráfico do stress em função do número de iterações. A projeção atinge um valor mínimo de stress quando o método LSPevolucao realiza cerca de 20 iterações, acima de 20 iterações o stress parece ficar constante com relação ao número de iterações.

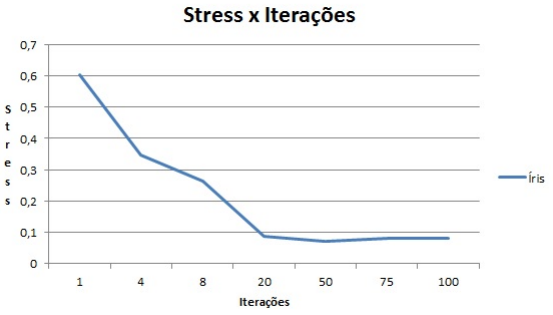
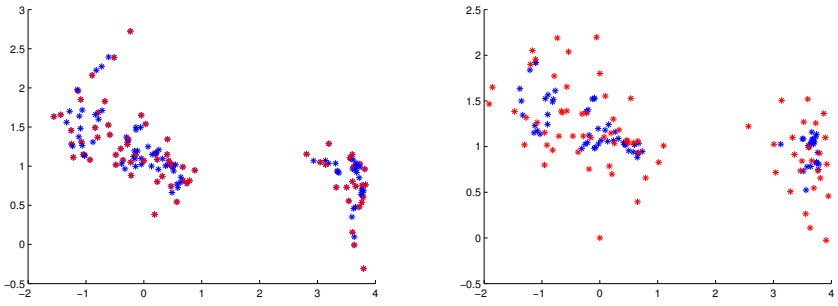


Figura 3.8: Gráfico do Stress x Iterações. Stress produzido pelo método LSPevolução em função do número de iterações utilizada pelo mesmo.

Iteratividade do novo método LSPevolucao.

O novo método LSPevolucao nos permite interagir durante o processo de projeção, enquanto os dados evoluem, daí o nome evolução, para a solução do sistema é permitido ao usuário mover os pontos de controle a fim de melhorar a projeção visualmente tanto quando lhe for possível, resultando também numa melhora do stress.

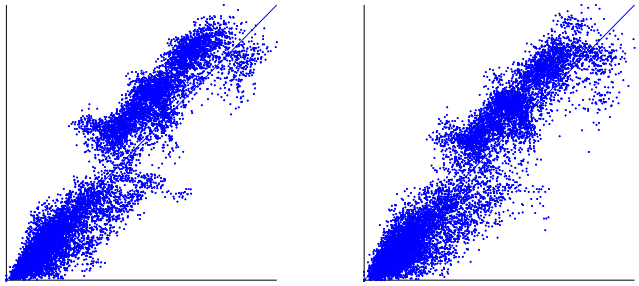
A Figura 3.9 nos mostra uma projeção efetuada por ambos os métodos LSP e LSPevolucao, no método LSPevolucao movemos os pontos de controle (pontos em vermelho) como nos convinha com o objetivo de tentarmos melhorar a projeção. Note que obtemos, após a realocação iterativa dos pontos de controle, um layout mais agradável.



(a) Método LSP (b) Método LSPevolucao

Figura 3.9: Projeções do conjunto de dados Íris através dos métodos LSP e LSPevolucao. No método LSPevolucao os pontos de controle (pontos em vermelho) foram reposicionados de modo a reduzir a linearidade introduzida pelo próprio método.

Um aspecto interessante de se analisar através do gráfico de dispersão é qual dos métodos LSP ou LSPevolucao preserva mais as relações de vizinhança presentes no espaço m-dimensional. A Figura 3.10 ilustra os gráficos de dispersão das projeções do conjunto de dados Íris apresentadas na Figura 3.9. Note que no gráfico de dispersão produzido pelo método LSP os pontos foram distribuidos de forma semelhante aos pontos no gráfico de dispersão do método LSPevolucao, no entanto no segundo gráfico de dispersão vários pontos se aproximaram da diagonal.



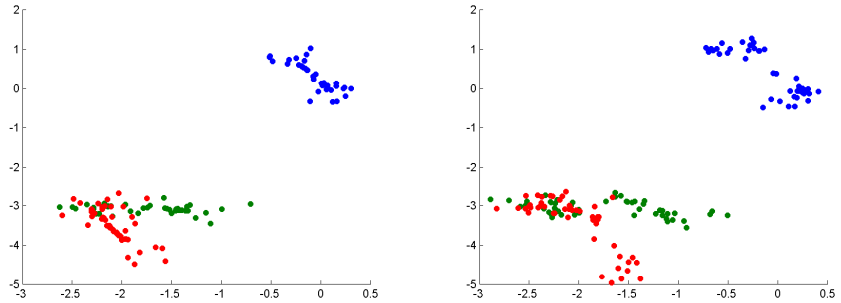
(a) Método LSP (b) Método LSPevolucao

Figura 3.10: Gráfico de dispersão das projeções do conjunto de dados Íris ilustradas na Figura 3.9, realizadas utilizando-se os métodos LSP e LSPevolucao.

Definição de vizinhos no método LSP

Durante a definição do método LSP demos duas possibilidades para a definição de vizinhos de um certo ponto, são elas: pelo número de aglomerados e raio de influência, e pelo número de vizinhos.

Primeiro realizamos projeções do conjunto de dados Íris através do método LSP onde a definição de vizinhança é feita pelo número de aglomerados e raio de influência. Constatamos que na maioria das projeções aparecem muitas soluções repetidas, em alguns casos obtivemos até 16 soluções identicas, o que compromete a qualidade da projeção, a Figura 3.11(a) nos mostra um exemplo de uma projeção LSP usando esse esquema de definição de vizinhança.



(a) (b)

Figura 3.11: Projeção do conjunto de dados Íris através do método LSP, onde utiliza-se a definição de vizinhança como a) o número de aglomerados e raio de influência e b) o número de vizinhos.

Nos experimentos o número de alglomerados é um parâmetro do método, devendo ser fornecido pelo usuário, o raio de influência por sua vez não é considerado como parâmetro e, portanto, é definido durante a execução do método. Para isto, calcula-se uma distância média entre os pontos do conjunto de dados e o valor encontrado é utilizado como raio de influência.

A segunda abordagem onde a definição de vizinhança é feita a partir do número de vizinhos produz resultados com nenhuma solução repetida, salvo em algumas exeções onde aparecem uma ou duas soluções repetidas. Veja na Figura 3.11(b) uma projeção do conjunto de dados Íris utilizando o número de vizinhos para a definição de vizinhaça, nesse caso foi utilizado o número de vizinhos igual a 5. A projeção nesse caso nos dá uma intuição melhor do conjunto de dados do que aquela utilizando o número de aglomerados e raio de influência. Precisamos então definir qual seria o melhor valor para o número de vizinhos a ser utilizado.

O número de vizinhos utilizado na definição de vizinhaça tem impacto na projeção do conjunto de dados, achamos interessante então analisar a influência desse fator nas projeções. Começamos com um número razoavelmente baixo de vizinhos e fomos aumentando e percebemos que há uma alteração significativa no stress e no tempo utilizado para se realizar a projeção, como pode ser observado na Tabela 3.5. Visualmente, percebe-se que aumentando o número de vizinhos exigidos os pontos, que não são pontos de controle, tendem a se aglomerar de forma linear, na Figura 3.12 exibimos projeções do conjunto de dados Íris onde o número de vizinhos utilizado foi 3, 8, 10, 15 e 25.

Número de vizinhos	Stress	Tempo
3	0.0731	0.1006 s
8	0.0735	0.1028 s
10	0.0754	0.1079 s
15	0.0811	0.1053 s
25	0.0945	0.1100 s

Tabela 3.5: Resultados obtidos em uma projeção do conjunto de dados Íris utilizando como técnica de definição de vizinhos o número de vizinhos, com o número de vizinhos igual a 3, 8, 10, 15 e 25.

Com base nessa análise do conjunto Íris e em análises similares dos conjuntos Wine, Housing e Abalone descidimos utilizar o número de vizinhos igual a 3 para os conjuntos de dados Íris e Wine, 20 para o Housing e 40 para o Abalone.

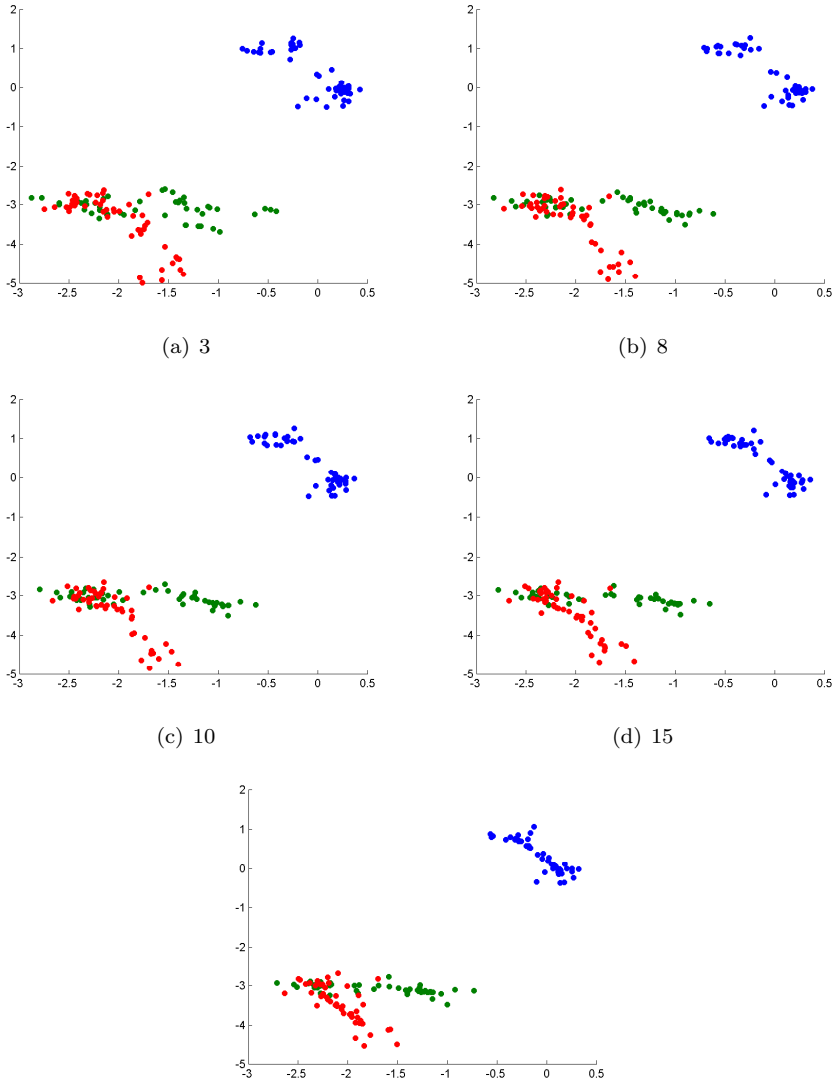
Técnica de definição de vizinhos	Stress	Tempo
Número de aglomerados e raio de influência	0.1013	0.8931 s
Número de vizinhos	0.0734	0.1015 s

Tabela 3.6: Resultados obtidos em uma projeção do conjunto de dados Íris utilizando como técnica de definição de vizinhos o número de aglomerados e raio de influência, e o número de vizinhos.

A Tabela 3.6 nos fornece o stress e o tempo gasto nas projeções do conjunto de dados Íris ilustradas nas Figuras 3.11(a) e 3.11(b), o que podemos reparar é que o método que utiliza o número de vizinhos realiza suas rotinas com tempo inferior ao tempo utilizado pelo outro método. No conjunto de dados Íris escolhemos definir os vizinhos de um ponto a partir do número de vizinhos pela melhor qualidade dos layouts produzidos e também pelo reduzido tempo gasto para realizar a projeção.

Também realizamos essa análise nos outros conjuntos de dados chegando a resultados que mostram essas mesmas tendências, isto é, o método que utiliza

o número de vizinho possui melhor layout e realiza a projeção com um menor esforço computacional que o método que utiliza o número de vizinhos e raio de influência.



(a) 3

(b) 8

(c) 10

(d) 15

(e) 25

Figura 3.12: Projeção do conjunto de dados Íris atrvés do método LSP, onde utiliza-se a definição de vizinhança como o número de vizinhos. Com o número de vizinhos igual a 3, 8, 10, 15 e 25.

O método LAMP como método local.

Na sua formulação, o método LAMP trabalha com um somatório (equação 2.2) para projetar cada dado p, esse somatório incorpora informações de todos os

pontos de controle à projeção do dado em questão, percorrendo assim todos os pontos de controle. Contudo, esse método pode ser modificado para levar em conta somente os pontos de controle mais próximos do dado a ser projetado. Com isso o método LAMP torna-se um método verdadeiramente local, essa restrição é considerada levando em conta uma porcentagem dos pontos de controle mais próximos a p, como por exemplo 75%, 50% ou 25%.

A redução da porcentagem de pontos de controle utilizada tem forte influência na projeção, a Figura 3.13 mostra gráficos do stress e do tempo gasto pelo método LAMP, na projeção do conjunto de dados Íris, em função da porcentagem de pontos de controle utilizados no somatório 2.2. Em termos de esforço computacional, pode ser observada uma redução considerável no tempo gasto a medida que a porcentagem de pontos de controle utilizados diminui. Além disso o stress produzido pelo método LAMP não sofre grande alteração quando a porcentagem de pontos de controle mais próximos é reduzida de 100% para 25%, contudo uma pertubação maior já pode ser constatada quando a porcentagem é de 25%, isso indica uma instabilidade nos valores de stress para porcentagens menores que 25%.

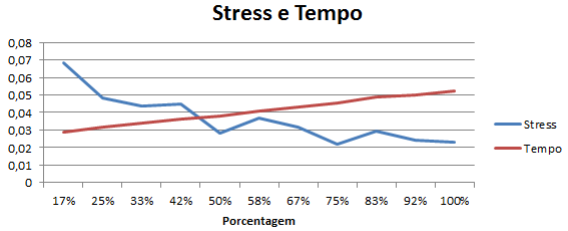
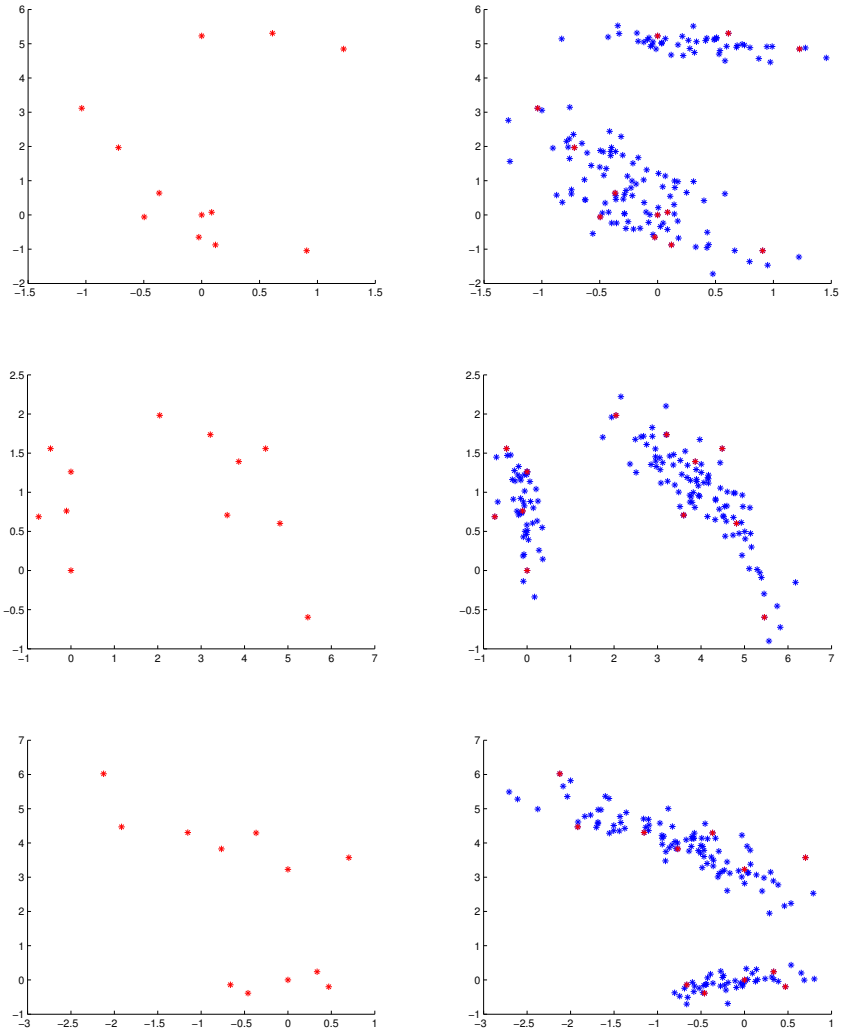


Figura 3.13: Tempo gasto pelo método LAMP para projetar o conjuto de dados Íris, onde a porcentagem de pontos de controle utilizada na equação 2.2 é 100%, 75%, 50%, 25%.

Análise da influência dos pontos de controle na projeção.

Alguns dos principais métodos de projeção multidimensional utilizam pontos de controle para realizar a projeção do conjunto de dados, esses métodos utilizam os pontos de controle para orientar a projeção, com isso o posicionamento dos pontos de controle tem influência no resultado final da projeção, esta seção tem como objetivo estudar justamente essa influência.

A Figura 3.14 ilustra três projeções do conjunto de dados Íris obtidas através do método LAMP, onde os pontos de controle, que orientam a projeção, tem disposições diferentes no espaço. Os pontos de controle são escolhidos a partir do conjunto de dados utilizando o método k-means e um método de seleção aleatória, em seguida esses pontos são projetados utilizando o método NNP e finalmente orientam a projeção. A Tabela 3.7 fornece os valores de stress obtido nas três projeções apresentadas na Figura 3.14



(a) Pontos de controle (b) Projeções LAMP

Figura 3.14: Influência dos pontos de controle na projeção. As imagens na primeira coluna mostram a disposição dos pontos de controle e as imagens na segunda coluna a projção LAMP do conjunto de dados Íris, com os pontos de controle em destaque na cor vermelho.

Método	Stress
k-means$_1$	0.0080
k-means$_2$	0.0052
Aleatório	0.0071

Tabela 3.7: Stress produzido nas projeções do conjunto de dados Íris através do método LAMP.

3.3 Conjunto de dados Wine

Passamos agora ao conjunto de dados Wine, os dados presentes nesse conjunto de dados são o resultado de uma análise de vinhos, cujas videiras cresceram na mesma região da Itália, mas são provenientes de três cultivares, foram analisados 59 vinhos do primeiro cultivare, 71 do segundo e 48 do terceiro, totalizando 178 vinhos. Uma característica que se sobressai nesse conjunto de dados é a maior dimensão do espaço ao qual seus dados estão incorporados. Cada um de seus vinhos é representado por 13 atributos, dentre eles destacamos álcool, ácido málico, magnésio, fenóis totais, a intensidade da cor e matiz, atributos esses que podem ser observados em destaque e nessa ordem no seguinte dado que representa o quinto vinho analisado

$$p_5 = \mathbf{13.24}\ \mathbf{2.59}\ 2.87\ 21\ \mathbf{118}\ \mathbf{2.8}\ 2.69\ 0.39\ 1.82\ \mathbf{4.32}\ \mathbf{1.04}\ 2.93\ 735$$

Aplicamos então os métodos de projeção multidimensional ao conjunto de dados Wine e os resultados dessas projeções podem ser observados na Figura 3.15, bem como a Tabela 3.8 informa os parâmetros de saida (tempo e stress) para essas projeções.

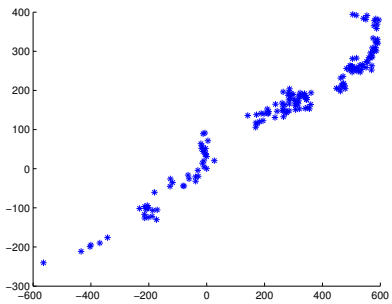
Método	Stress	Tempo
NNP	0.0203	0.0473 s
Force	0.0303	0.9817 s
LSP	0.0350	0.1359 s
PLP	0.0217	0.0540 s
LAMP	0.0050	0.1141 s

Tabela 3.8: Resultados obtidos nas projeções do conjunto de dados Wine ilustradas na Figura 3.15.

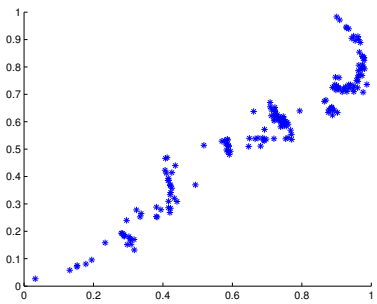
Ao aplicar o método Force ao conjunto de dados Wine observamos algumas peculiaredades, em alguns casos, principalmente quando a projeção NNP inicial é muito boa, o método Force não desempenha muito bem seu papel piorando as projeções em vez de melhorá-las. Fica justificado com isso um valor de stress mais alto no método Force do que no método NNP (Tabela 3.8). O método LAMP apresenta o melhor valor de stress ao projeta o conjunto de dados Wine, o stress produzido por sua projeção é uma ordem de precisão menor que o stress produzido pelos demais métodos.

Como mencionado esse conjunto de dados também está dividido em três classes, segundo o próprio doador do conjunto de dados Riccardo Leardi. Na Figura 3.16 incluimos as informações de classes ao resultado das projeções apresentada na Figura 3.15. As três classes ficam bem definidas nas projeções resaltando a diferença de cultivare ao qual pertencem, no entanto, pode-se observar pequenas regiões nas projeções nas quais todas as três classes possuem pontos projetados, sugerindo uma certa proximidade das mesmas, o que é plausível, pois os dados pertencentes a esse conjunto de dados representam vinhos que foram cultivados numa mesma região da Itália.

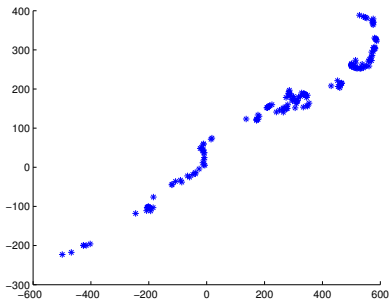
Observe que diferente do conjunto de dados Íris que apresenta a classe das flores íris-setosa bem separada das demais, o conjunto de dados Wine não apresenta um classe que se distância consideravelmente das demais.



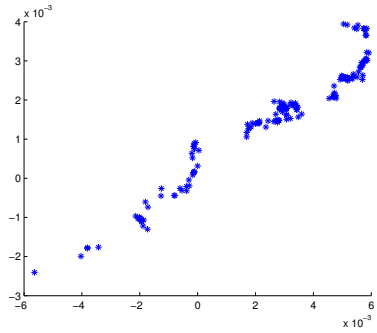
(a) Método NNP



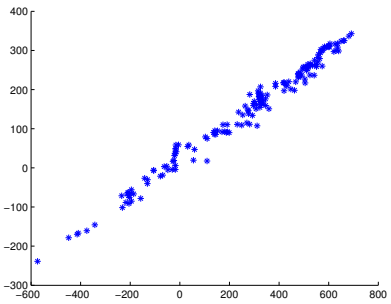
(b) Método Force



(c) Método LSP

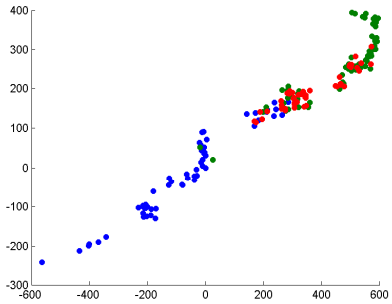


(d) Método PLP

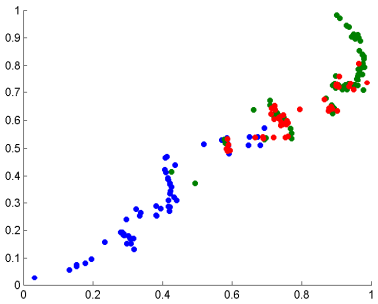


(e) Método LAMP

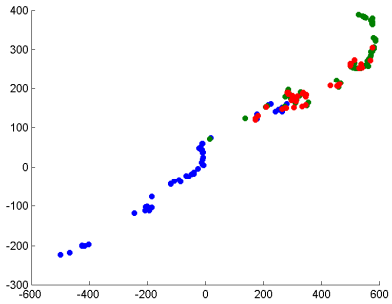
Figura 3.15: Projeções do conjunto de dados Wine através dos métodos NNP, Force, LSP, LAMP e PLP.



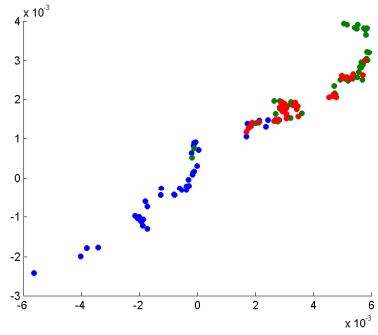
(a) Método NNP



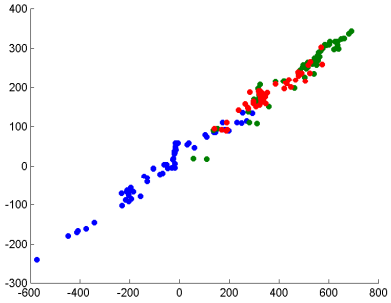
(b) Método Force



(c) Método LSP



(d) Método PLP



(e) Método LAMP

Figura 3.16: Projeções do conjunto de dados Wine através dos métodos NNP, Force, LSP, LAMP e PLP com destaque para as diferentes classes presentes no conjunto de dados.

3.4 Conjunto de dados Housing

Um outro conjunto de dados bastante interessante que utilizamos em nossas comparações é o conjunto de dados Housing, que diz respeito à valores de 506 alojamentos no subúrbio de Boston, cada uma das residências é descrita por 14 atributos, sendo 13 deles continuos e 1 binário. A variável binária aparece como quarto atributo dos dados, o valor 1 é atribuido aquelas que fazem limites com o rio Charles, e 0 aquelas que não fazem. Damos exemplo de duas residências uma que faz limite com o rio e uma que não faz, atente para o quarto atributo

$$\begin{aligned} p_{88} &= 0.07\ 0.0\ 04.49\ \mathbf{0}\ 0.45\ 6.12\ 056.8\ 3.75\ 3\ 247.0\ 18.5\ 395.15\ 08.44\ 22.2 \\ p_{143} &= 3.32\ 0.0\ 19.58\ \mathbf{1}\ 0.87\ 5.40\ 100.0\ 1.32\ 5\ 403.0\ 14.7\ 396.90\ 26.82\ 13.4 \end{aligned}$$

Assim como nos conjunto de dados anteriores realizamos a projeção do conjunto de dados Housing através dos nossos métodos de estudo, os resultados podem ser observados na Figura 3.17.

É também interessante fornercermos resultados de stress e tempo gasto na projeções observadas na Figura 3.17 para que possamos comparar os métodos, valores estes que aparecem na Tabela 3.9. Observe que em termos de stress o método Force supera o método NNP, no entanto para um conjunto de dados como o Housing, que contém 506 dados, o tempo que esse método leva para obter a projeção é muito maior se comparado ao método NNP, que resolve o problema quase que instantaneamente. Note também que os métodos baseados em pontos de controle ainda não apresentam um esforço computacional reduzido, no entanto veremos que para conjuntos de dados maiores a redução no tempo gasto ficara evidente.

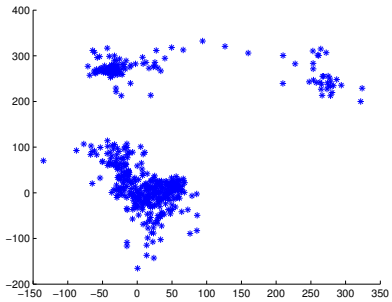
Método	Stress	Tempo
NNP	0.0968	0.2806 s
Force	0.0211	8.0746 s
LSP	0.0796	1.0934 s
PLP	0.0880	0.2466 s
LAMP	0.0450	0.5083 s

Tabela 3.9: Resultados obtidos nas projeções do conjunto de dados Housing ilustradas na Figura 3.17.

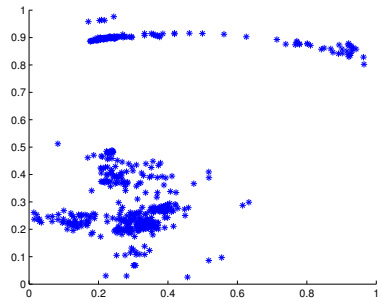
Diferente dos conjunto de dados Íris e Wine o conjunto de dados Housing não possui classes pré-definidas por seus criadores, desejamos justamente inferir se existem tendências de agrupamento em seus dados, para tal finalidade realizamos a redução dimensional através dos nossos métodos de estudo que podem ser observadas na Figura 3.17.

Os resultados das projeções sugerem que estão presentes três grupos bem definidos nos dados, cada um deles com uma concentração diferente de dados. Pode-se notar um primeiro grupo bastante denso de dados, um segundo grupo que apresenta uma densidade um pouco menor que este, e um terceiro com poucos dados.

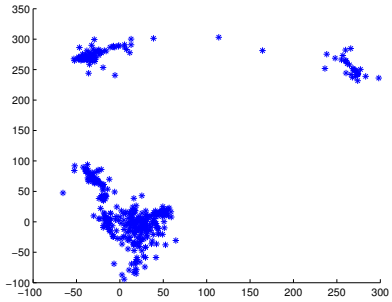
Observamos a projeção e a analisamos ponto a ponto classificando-os em três classes conforme o grupo ao qual pertence, obtemos a seguinte distribuição das residências



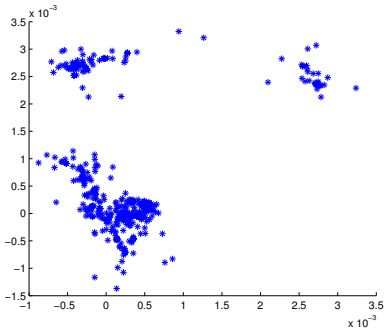
(a) Método NNP



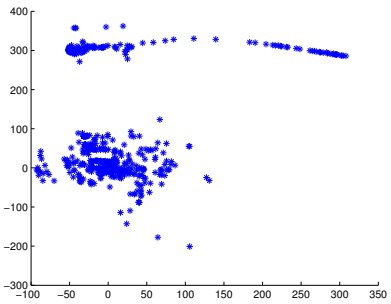
(b) Método Force



(c) Método LSP

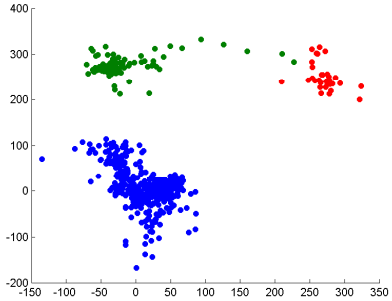


(d) Método PLP

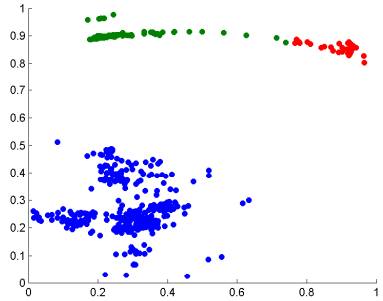


(e) Método LAMP

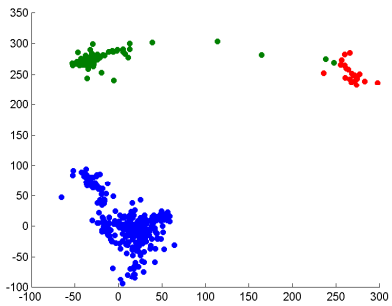
Figura 3.17: Projeções do conjunto de dados Housing através dos métodos NNP, Force, LSP, LAMP e PLP.



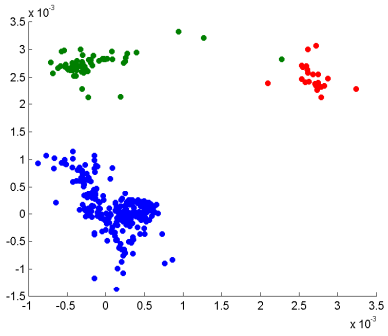
(a) Método NNP



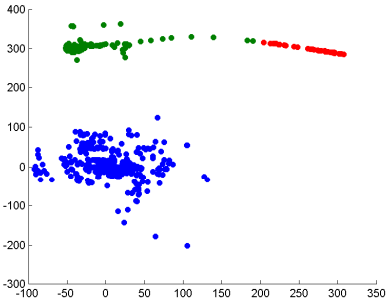
(b) Método Force



(c) Método LSP



(d) Método PLP



(e) Método LAMP

Figura 3.18: Projeção do conjunto de dados Housing através dos métodos NNP, Force, LAMP, LSP e PLP, com destaque para as classes presentes no conjunto de dados.

1. **Classe 1:** da 1ª à 356ª e da 494ª à 506ª

2. **Classe 2:** da 357ª à 493ª menos os elementos que pertencem a classe 3.

3. **Classe 3:** 411ª, 412ª ,413ª, 415ª, 416ª, 417ª, 419ª, 420ª, da 424ª à 439ª, 446ª, 451ª, 455ª, 456ª, 457ª, 458ª , 467ª

O processo para obtenção das classes é realmente trabalhoso, mas em contrapartida muito simples. A fim de testar esse agrupamento que obtivemos, rodamos novamente os algoritmos com essa separação de classe por nós definida. Os resultados dessas projeções podem ser observados na Figura 3.18, os grupos ficam muito bem definidos como podemos observar, constatando uma ótima definição das classes. Assim a partir de um lista de dados conseguimos inferir tendências de agrupamento.

Após obter esse agrupamento através das projeções surge naturalmente uma questão: seria possível obter uma separação de classes que representasse bem os grupos presentes na projeção a partir de algum atributo ou atributos presente no conjunto de dados. A resposta para esssa questão no conjunto de dados Housing é muito próxima de sim.

Observamos o conjunto de dados e a analisamos os atributos que tem maior influência na projeção. A nosso ver o nono atributo parecia ter forte influência no conjunto de dados, criamos então duas faixas para esse atributo, são elas:

1. **Faixa 1:** 0 a 20

2. **Faixa 2:** 21 acima

Com essas faixas conseguimos praticamente separar o grupo mais denso dos demais, todos os pontos pertencentes ao grupo mais denso tem o nono atributo pertencente a Faixa 1, além de alguns pontos pertencentes ao grupo de densidade moderada de dados, veja na Figura 3.19.

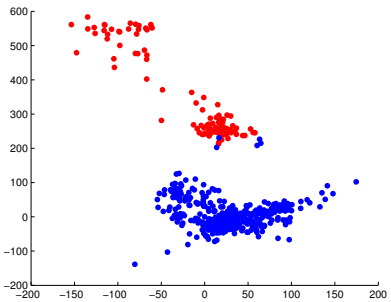


Figura 3.19: Influência do nono atributo na separação dos grupos. Os pontos em azul tem o nono atributo dentro da Faixa 1 e os pontos em vermelho dentro da Faixa 2.

Outro atributo que contribui na separação dos grupos é o décimo segundo atributo, criamos também duas faixas para esse atributo:

1. **Faixa 3:** 0 a 110

2. **Faixa 4:** 110 acima

A partir das Faixas 3 e 4 conseguimos separar quase que totalmente o grupo menos denso dos demais, todos os pontos pertencentes ao grupo menos denso tem o décimo segundo atributo dentro da Faixa 3, além de alguns pontos pertencentes ao grupo mais denso, veja na Figura 3.20.

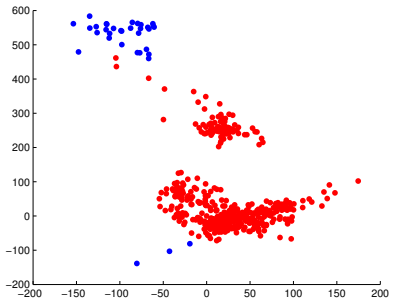


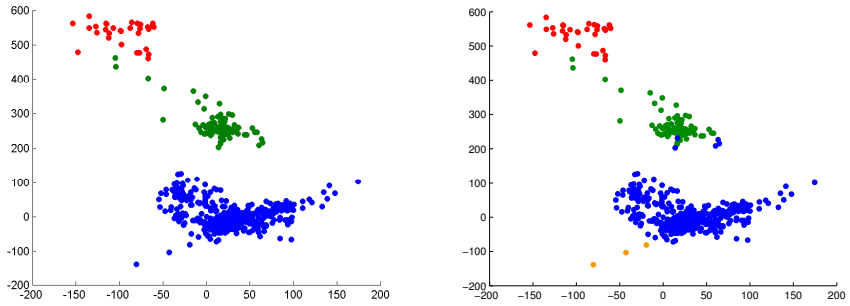
Figura 3.20: Influência do décimo segundo atributo na separação dos grupos. Os pontos em azul tem o décimo segundo atributo dentro da Faixa 3 e os pontos em vermelho dentro da Faixa 4.

Finalmente, utilizamos um produto cartesiano entre as quatro faixas apresentadas acima para os nono e décimo segundo atributos, conseguimos uma separação dos grupos muito próxima a obtida segundo a separação de classe definidas anteriormente, veja na Figura 3.21 a separação dos gurpos utilizando as classes e o produto cartesiano dos atributos 9 e 12. Na Figura 3.21(b) os pontos são coloridos de acordo com as faixas as quais pertencem, sendo

1. **Pontos azuis:** (9º, 12º) ∈ (Faixa 1, Faixa 4);
2. **Pontos vermelhos:** (9º, 12º) ∈ (Faixa 2, Faixa 3);
3. **Pontos verdes:** (9º, 12º) ∈ (Faixa 2, Faixa 4);
4. **Pontos amarelos:** (9º, 12º) ∈ (Faixa 1, Faixa 3).

Os atributos 9 e 12 têm sem duvida bastante influência no conjunto de dados, pois com eles conseguimos uma separação dos grupos quase tão boa quanto aquela obtida analisando a projeção ponto a ponto.

Para a determinação das Faixas 1, 2, 3 e 4 desenvolveu-se uma rotina no MATLAB que funciona como um equalizador. Realiza-se inicialmente uma projeção, os pontos então são coloridos com as cores azul ou vermelha, de acordo com um atributo, pontos com esse atributo abaixo de um determinado valor são coloridos com a cor azul e os pontos com esse atributo acima desse valor são coloridos com a cor vermelha. O usuário pode então alterar esse tal valor através de um medidor, com base nessa mudança o código redefine as cores. Assim pode-se analisar a influência de um determinado atributo em uma projeção



(a) (b)

Figura 3.21: Separação dos grupos utilizando (a) nossa definição de classes e (b) o produto cartesiano definido pelos nono e décimo segundo atributos

3.5 Conjunto de dados Abalone

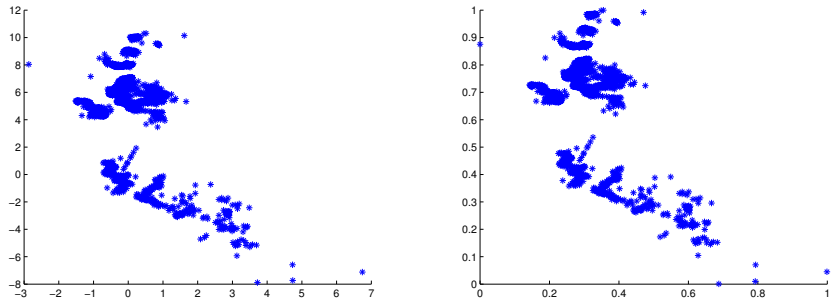
Passamos finalmente ao nosso último conjunto de dados conhecido como conjunto de dados Abalone, trata-se de haliotes (ou abalones) um tipo de molusco encontrado na maioria dos mares temperados, dos nossos conjuntos de dados esse é o que apresenta o maior número de dados, são 4177 moluscos, cada um deles representados por 8 atributos, uma característica marcante desses dados é a presença de 28 classes distintas, a quantidade de haliotes em cada uma dessas classes pode ser observada na Tabela 3.10.

Classe	Nº de dados	Classe	Nº de dados	Classe	Nº de dados
1ª	1	11ª	487	21ª	14
2ª	1	12ª	267	22ª	6
3ª	15	13ª	203	23ª	9
4ª	57	14ª	126	24ª	2
5ª	115	15ª	103	25ª	1
6ª	259	16ª	67	26ª	1
7ª	391	17ª	58	27ª	2
8ª	568	18ª	42	28ª	1
9ª	689	19ª	32		
10ª	634	20ª	26		

Tabela 3.10: Classes do conjunto de dados Abalone.

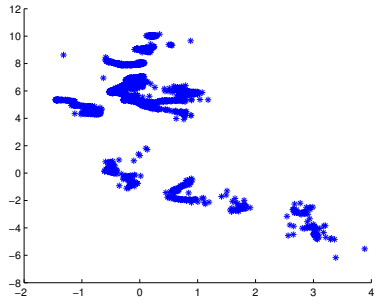
Novamente aplicamos os métodos de projeção NNP, Force, LSP, LAMP e PLP ao conjunto de dados Abalone. Os resultado dessas projeções podem ser observados na Figura 3.22. Podemos constatar visualmente a presença de muitos grupos, além de alguns pontos isolados, esse pontos representam provavelmente as classes constituidas de apenas um molusco.

Assim como fizemos no conjunto de dados Housing é possível analisar o conjunto de dados Abalone ponto a ponto e definir a que classe cada um deles pertence, dificilmente vamos recuparar as 28 classes fornecidas pelo criador do conjunto de dados mais boa parte delas poderão ser identificadas.

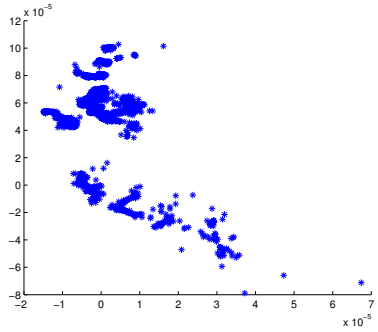


(a) Método NNP

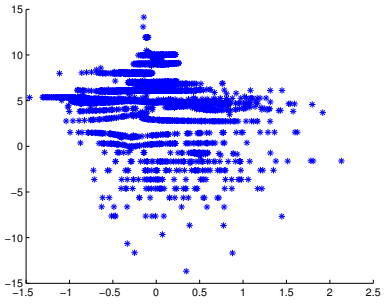
(b) Método Force



(c) Método LSP



(d) Método PLP



(e) Método LAMP

Figura 3.22: Projeções do conjunto de dados Abalone através dos métodos NNP, Force, LSP, LAMP e PLP.

A Tabela 3.11 fornece os valores de stress e tempo gasto nas projeções visualizadas na Figura 3.22. Esses resultados deixam bem claro o benefício obtido ao se utilizar pontos de controle para realizar projeções. Os métodos PLP e LAMP, que fazem uso dessa técnica, projetam os dados com um esforço computacional consideravelmente menor se comparados os métodos NNP e Force, que não fazem uso dessa técnica. No método LSP optamos por projeções mais custosas e de melhor qualidade, apenas por esse motivo este método não superou os métodos NNP e Force.

O método LAMP projeta os dados com um valor de stress que supera os demais métodos em uma ordem de precisão, em particular supera os métodos NNP e Force além de realizar a projeção com um esforço computacional menor.

Método	Stress	Tempo
NNP	0.2671	16.7592 s
Force	0.1227	554.2662 s
LSP	0.4212	99.4822 s
PLP	0.2711	7.6286 s
LAMP	0.0201	10.8164 s

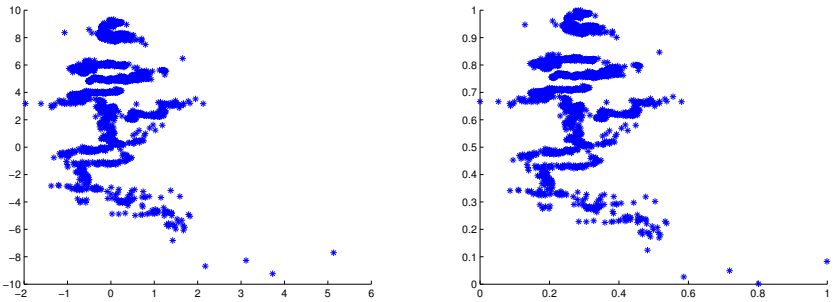
Tabela 3.11: Resultados obtidos nas projeções do conjunto de dados Abalone ilustradas na Figura 3.22.

O layout produzido pelo método LAMP diferiu muito do layout produzido pelos demais métodos. No conjunto de dados Abalone, estão presentes muitos grupos e alguns desses grupos são formados por um número muito grande de dados, com isso o método k-means acaba por escolher poucos ou até mesmo nelhum ponto de controle nos grupos que têm poucos dados, assim os pontos de controle não conseguem orientar de forma adequada a projeção. Por esse motivo o layout do método LAMP parece tão diferente dos demais. Note contudo que ainda assim o valor de stress produzido é pequeno superando os demais métodos.

Até aqui seguimos um padrão para realizar as projeções através do método LAMP: o número de pontos de controle $n_c = \sqrt{n}$ e a porcentagem de vizinhos mais próximos, utilizado na projeção de cada ponto, igual a 100%. Para obter uma layout mais parecido com os layouts produzidos pelos outros métodos aumentamos o número de pontos de controle para $n_c = 500$ e reduzimos para 0.01% a porcentagem de vizinhos mais póximos utilizado. Os resultados podem ser observados na Figura 3.23 e na Tabela 3.12. O tempo gasto para realizar a projeção através do método LAMP com essa alteração é maior, além disso o stress obtido no método LAMP se aproxima um pouco mais do stress obtido pelo método NNP, devido ao uso de um número maior de pontos de controle.

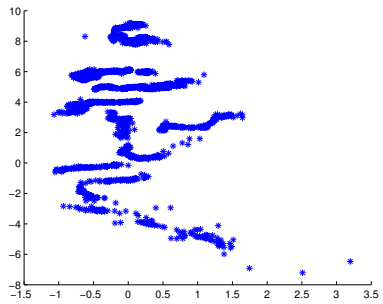
Método	Stress	Tempo
NNP	0.3192	15.4337 s
Force	0.1988	583.4274 s
LSP	0.5485	101.0432 s
PLP	0.3815	7.5239 s
LAMP	0.2749	12.6758 s

Tabela 3.12: Resultados obtidos nas projeções do conjunto de dados Abalone ilustradas na Figura 3.23.

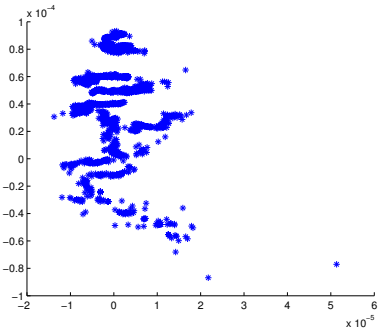


(a) Método NNP

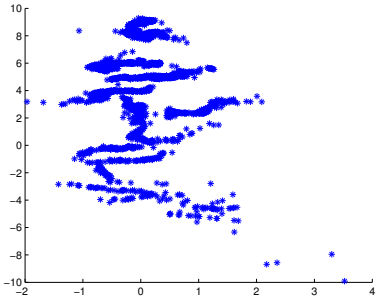
(b) Método Force



(c) Método LSP



(d) Método PLP



(e) Método LAMP

Figura 3.23: Projeções do conjunto de dados Abalone através dos métodos NNP, Force, LSP, LAMP e PLP, com mudança no número de pontos de controle utilizado pelo método LAMP.

Surgiu uma questão bastante sutil nos estudo desses métodos de projeção, não teriamos uma perda de informação menor ao projetarmos os pontos no espaço em vez de projetá-los no plano a final de contas teriamos uma dimensão a mais para guarda as informações. Motivado por essa possibilidade adaptamos o método NNP para realizar projeções no espaço, a esse método demos o nome de NNP3.

3.6 Método NNP3

O método NNP3 inicia o algoritmo de modo semelhante ao método NNP, os dois primeiros pontos são projetados em $\mathbb{R}^3$, então para cada novo ponto p a ser projetado realiza-se uma pesquisa no conjunto dos pontos que já foram projetados a procura dos dois pontos q e r mais próximos (em $\mathbb{R}^m$) de p, considera-se esferas E_q e E_r de centro q' e r' e raios d_q e d_r, respectivamente (no lugar dos círculos C_q e C_r). A partir dessas esferas é escolhido um ponto no espaço para representar p.

A Figura 3.25 mostra a projeção dos conjuntos de dados Íris, Wine, Housing e Abalone obtidas através do método NNP3. Na projeção do conjunto de dados Íris, é possível observar uma diferença de altura entre as classes verde e vermelha (íris-versicolor e íris-virginica). Segundo o autor desse conjunto de dados, essas duas classes não se separam linearmente, como obtivemos, suas coordenadas x e y tem uma forte relação, no entanto, com a projeção NNP3 conseguimos incorporar a diferença de classes a projeção e ainda manter essas classes linearmente unidas.

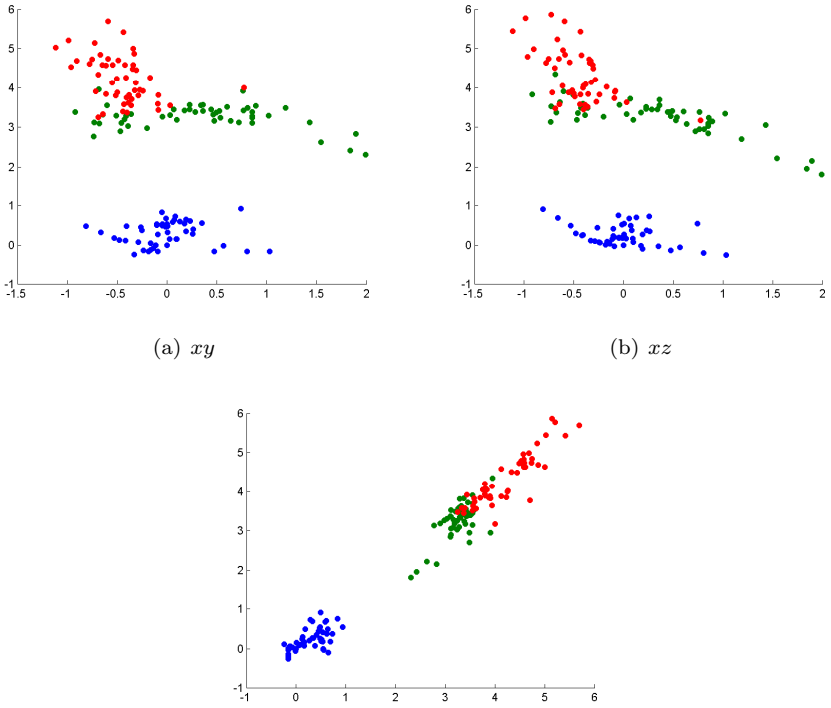
Passamos então a comparação dos métodos NNP e NNP3, vamos analisá-los utilizando o stress e o tempo gasto para realizar as projeções. A Tabela 3.13 fornece o stress obtido por ambos os métodos ao projetar os conjuntos de dados Íris, Wine, Housing e Abalone. Pode-se notar que em alguns casos o método NNP3 obtém valores de stress inferiores aqueles obtidos pelo método NNP, como por exemplo nos conjuntos de dados Íris e Housing. Em termos de esforço computacional os dois métodos apresentam resultados muito próximos (Tabela 3.13).

	Stress		Tempo	
	NNP	NNP3	NNP	NNP3
Íris	0.0415	0.0304	0.0356 s	0.0261 s
Wine	0.0208	0.0743	0.0484 s	0.0416 s
Housing	0.1148	0.0438	0.2847 s	0.2714 s
Abalone	0.1795	0.5249	16.0566 s	15.0808 s

Tabela 3.13: Stress e tempo gasto nas projeções NNP e NNP3 dos conjuntos de dados Íris, Wine, Housing e Abalone.

As projeções no espaço são bastante interessantes no seguinte sentido, podemos obter para cada projeção em $\mathbb{R}^3$ várias projeções em $\mathbb{R}^2$, basta projetar os pontos obtidos em um plano qualquer do espaço, como exemplo apresentamos na Figura 3.24 as projeções nos planos xy, xz e yz da projeção NNP3 do conjunto de dados Íris apresentada na Figura 3.25.

O método NNP3 demonstra ser um ótimo método de projeção multidimensional, isso nos motiva a estudar melhor projeções multidimensionais no espaço. Apartir do método NNP3 também podemos alterar o método Force para realizar projeções no espaço. Além disso, podemos projetar os pontos de controle, necessários aos métodos LSP, LAMP e PLP, no espaço desenvolvendo assim os métodos LSP3, LAMP3 e PLP3. Esse raciocínio pode ser utilizado a muitos outros métodos de projeção existentes.

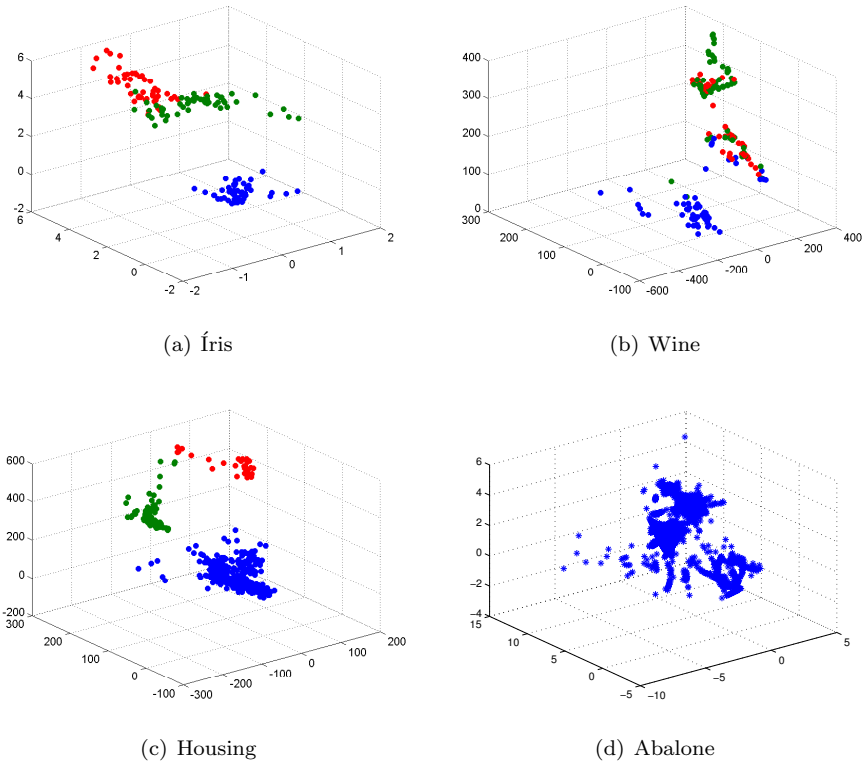


(a) xy

(b) xz

(c) yz

Figura 3.24: Projeção nos planos xy, xz e yz da projeção NNP3 do conjunto de dados Íris.



(a) Íris (b) Wine

(c) Housing (d) Abalone

Figura 3.25: Projeção dos conjunto de dados Íris, Wine, Housing e Abalone através do novo método de projeção multidimensional NNP3.

Capítulo 4

Conclusão e trabalhos futuros

Destinamos essa seção para apontar as principais conclusões tiradas a cerca dos métodos, e também damos a direção que pretendemos seguir nos nossos estudos.

O método NNP tem uma importância muito grande, como método de projeção multidimensional ele apresenta bons resultados em termos de precisão. No entanto, aplicando o esquema de melhoria Force, o resultado mostra-se melhor sob as métricas introduzidas. Além disso os demais métodos ao projetar os pontos de controle utilizam o método NNP. Então é bastante evidente a importância desse método.

O esquema de melhoria Force em geral reduz os valores de stress de uma projeção. Esse método apesar de ser computacionalmente caro produz ótimos valores de stress sendo ideal para aplicações onde o tempo computacional não é muito importante e sim o resultado final da projeção.

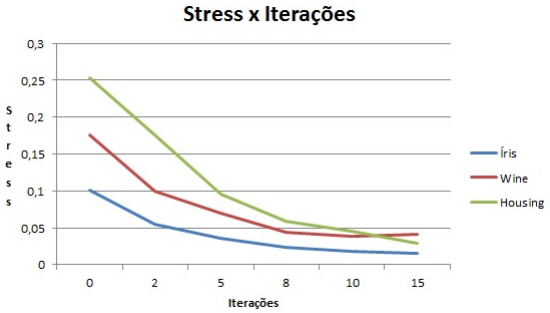


Figura 4.1: Gráfico do Stress x Iterações. Stress produzido pelo método Force em função do número de iterações utilizada pelo mesmo.

Uma análise do stress em função do número de interações do esquema de melhoria Force é feita para os conjuntos de dados Íris, Wine e Housing, os resultados aparecem na Figura 4.1. Pode-se notar que o stress diminui rapidamente a medida que o número de iterações aumenta e, que esse valor tende a ficar constante depois de um certo número de iterações entre 10 e 15 iterações para os

conjuntos de dados Íris e Wine e 15 iterações para o conjunto de dados Housing.

Os métodos abordados neste trabalho que fazem uso dos pontos de controle são os métodos LSP, PLP e LAMP, nota-se que o esforço computacional para realizar a projeção de conjuntos de dados maiores é consideravelmente menor para métodos que utilizam essa opção. A Tabela 4.1 apresenta o tempo gasto pelos métodos NNP, Force, LSP, PLP e LAMP para projetar o conjunto de dados Abalone que contém 4177 dados. Note que os métodos PLP e LAMP superam os métodos que não fazem uso de pontos de controle (NNP e Force). O método LSP apresenta dificuldade para projetar esse conjunto de dados, provavelmente devido ao grande número de classes presentes nesses dados, então optamos por projeções mais custosas e de melhor qualidade, somente por esse motivo este método não superou os métodos NNP e Force.

Método	Tempo
NNP	16.7592 s
Force	554.2662 s
LSP	99.4822 s
PLP	7.6286 s
LAMP	10.8164 s

Tabela 4.1: Tempo gasto nas projeções do conjunto de dados Abalone ilustradas na Figura 3.22.

Pontos de controle carregam informações geométricas que são utilizadas de diferentes formas pelos métodos durante o processo de projeção. O método LSP trabalha com alguns sistemas laplacianos para realizar a projeção, onde as informações fornecidas pelos pontos de controle devem ser incorporadas a esses sistemas. Este trabalho apresenta uma nova técnica para incorporar as informações dos pontos de controle aos sistemas, denominada Evolução. Ao utilizar essa técnica o método LSP ganha o nome LSPevolucao.

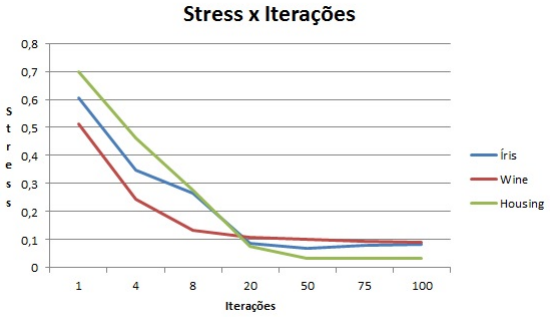


Figura 4.2: Gráfico do Stress x Iterações. Stress produzido pelo método LSPevolução em função do número de iterações utilizada pelo mesmo.

Nos conjuntos de dados Íris, Wine e Housing são feitas análises do stress em função do número de interações utilizada pelo método LSPevolucao para resolver o sistema gerado pelo mesmo durante a projeção dos dados, os resultados aparecem na Figura 4.2. O stress decaí rapidamente a medida que o número de iterações cresce e, após 50 iterações tende a se manter constante.

Quando um número reduzido de pontos de controle é utilizado, as projeções obtidas através do método LSP apresentam uma certa "linearidade" (isto é, aproximam-se de uma curva), esse problema é resolvido utilizando um número maior de pontos de controle.

O método PLP figura entre os métodos mais vantajosos se levamos em conta a qualidade da projeção e o esforço computacional, contudo ele nada mais é do que a aplicação do método LSP a subconjuntos do conjunto de dados, chamados de amostras, assim ao se utilizar dessa estratégia o método PLP pode criar projeções nas quais nota-se pequenos aglomerados, cada aglomerado desses sendo o resultado da projeção de uma amostra. Contudo, vale destacar que projeção desse tipo só são obtidas quando o número de pontos de controle utilizado é pequeno.

Por fim o método LAMP tem uma caracterítica bastante interessante, os pontos assim como no método NNP são projetados um a um permitindo que projeções parciais sejam exibidas e com elas podemos ter uma ideia de como o método evolui. Esse método é o que apresenta o melhor negócio em termos de qualidade e esforço computacional. Sua formulação também merece destaque, pois facilita muito a implementação do método.

Na seção anterior fornecemos uma medida visual para qualificar uma projeção, o gráfico de dispersão. Utilizamos essa medida para analisar o conjunto de dados Íris, no entanto também foram produzidos os gráficos de dispersão de todas as projeções dos conjunto de dados Wine, Housing e Abalone como podemos observar na Figura 4.3.

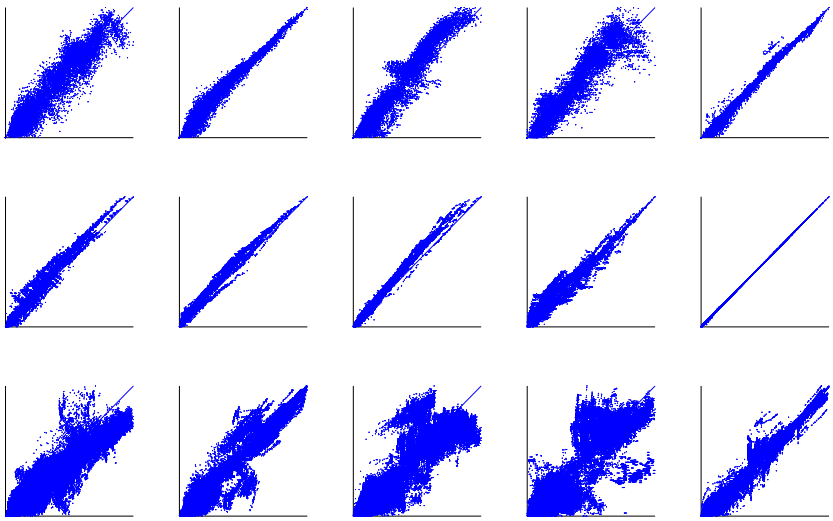


Figura 4.3: Gráficos de dispersão das projeções dos conjuntos de dados Íris, Wine, Housing (de cima para baixo) através dos métodos NNP, Force, LSP, PLP e LAMP (da esquerda para a direita).

Também obtivemos as medidas de stress de todos os conjuntos de dados: Íris, Wine, Housing e Abalone. A Tabela 4.2 mostra todos os resultados de stress, bem como, os de tempo utilizados pelos método na projeção dos conjunto de dados. Essa tabela é na verdade uma forma de resumir todos os resultados que

já mostramos anteriormente de forma separada.

	Íris (150 × 4)		Wine (178 × 13)		Housing (506 × 14)		Abalone (4177 × 8)	
Método	S	T	S	T	S	T	S	T
NNP	0.0558	0.05	0.0203	0.05	0.0968	0.28	0.2671	16.76
Force	0.0141	0.72	0.0303	0.98	0.0211	8.08	0.1227	554.27
LSP	0.0567	0.11	0.0350	0.14	0.0796	1.09	0.4212	99.48
PLP	0.0637	0.05	0.0217	0.05	0.0880	0.25	0.2711	7.63
LAMP	0.0069	0.10	0.0050	0.11	0.0450	0.51	0.0201	10.82

Tabela 4.2: Valores de stress (S) e tempo gasto (T) obitidos nas projeções de todos os conjuntos de dados. Os valores que aparecem entre parenteses são o número de dados e a dimensão do espaço em que os dados estão incorporados.

Os métodos apresentados neste trabalho são alguns dos métodos de projeção multidimensional existentes, futuramente pretendemos estudar outros métodos de projeção multidimensional, como por exemplo: PLMP [17], Hybrid [12], Glimer [9], e, além disso realizar um estudo das aplicações possível para os métodos de projeção multidimensional. Outra meta é analisar melhor projeções tridimensionais e seus benefícios.

O método LAMP faz uso de um mapeamento afim $f_p(x) = xM + t$ para cada ponto p a ser projetado, a projeção desse ponto p é então dada por

$$f_p(p) = pM + t$$

na tentativa de melhorar a qualidade da projeção testamos uma nova abordagem. Inicialmente calculamos todos os mapeamentos afins, isto é, dado um conjunto de dados $S = \{p_1, p_2, ..., p_n\}$ obtemos o conjunto

$$\{f_{p_1}, f_{p_2}, ..., f_{p_n}\},$$

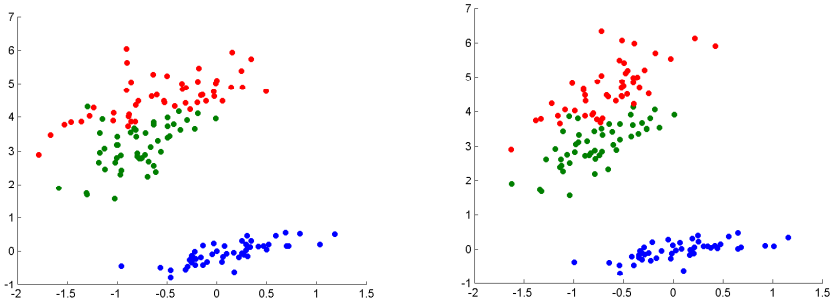
em seguida, utilizamos uma ideia de partição da unidade para definir uma única transformação f que será utilizada para projetar todos os pontos, essa função é dada por

$$f = \sum_k \beta_k f_k,$$

onde $\sum_k \beta_k = 1$. Por fim definimos a projeção de um ponto p_i como sendo

$$p_i' = f(p_i).$$

Foram realizadas projeções do conjunto de dados Íris utilizando o método LAMP e essa modificação do método LAMP, denotada por LAMP$_m$, e os resultados obtidos podem ser observados na Figura 4.4. Resultados de tempo e stress também podem ser observados na Tabela 4.3, os resultados parecem promissores. Estudar melhor essa modificação do método LAMP é outro dos nossos objetivos, utilizando-se para isso conjuntos de dados mais complexos.



(a) LAMP (b) LAMP$_m$

Figura 4.4: Projeções do conjunto de dados Íris que comparam o método LAMP com a sua versão modificada LAMP$_m$.

Método	Stress	Tempo
LAMP	0.0063	0.0644 s
LAMP$_m$	0.0019	0.1481 s

Tabela 4.3: Resultados de stress e tempo gasto nas projeções do conjunto de dados Íris apresentadas na Figura 4.4, nas quais foram utilizadas os métodos LAMP e a sua versão modificada LAMP$_m$.

Pekalska aborda em seu trabalho outros métodos de projeções, além da Triangulação, como Sammon mapping [19] e ANN [5]. Outros trabalhos estão fortemente relacionados à essa dissertação, dentre eles merecem destaque: Isomap [21], Chalmers [3], PCA [11] (o método PLMP pode ser visto como uma generalização do PCA), FDP model [14].

Referências Bibliográficas

[1] Åke Björck. *Numerical methods for least squares problems*. Siam, 1996.

[2] I Borg and P Groenen. Modern multidimensional scaling. 1997. *NY Springer*.

[3] Matthew Chalmers. A linear iteration time layout algorithm for visualising high-dimensional data. In *Visualization'96. Proceedings.*, pages 127–131. IEEE, 1996.

[4] Carlos Eduardo Ribeiro de Mello. *Agrupamento de regiões: Uma abordagem utilizando acessibilidade*. PhD thesis, UNIVERSIDADE FEDERAL DO RIO DE JANEIRO, 2008.

[5] Dick De Ridder and Robert PW Duin. Sammon's mapping using neural networks: a comparison. *Pattern Recognition Letters*, 18(11):1307–1316, 1997.

[6] Christos Faloutsos and King-Ip Lin. *FastMap: A fast algorithm for indexing, data-mining and visualization of traditional and multimedia datasets*, volume 24. ACM, 1995.

[7] Andrew Frank and Arthur Asuncion. Uci machine learning repository. 2010.

[8] Gene H Golub and Charles F Van Loan. matrix computations, 3rd. *Johns Hopkins, Baltimore*, 1996.

[9] Stephen Ingram, Tamara Munzner, and Marc Olano. Glimmer: Multilevel mds on the gpu. *Visualization and Computer Graphics, IEEE Transactions on*, 15(2):249–261, 2009.

[10] Paulo Joia, Fernando V Paulovich, Danilo Coimbra, José Alberto Cuminato, and Luis G Nonato. Local affine multidimensional projection. *Visualization and Computer Graphics, IEEE Transactions on*, 17(12):2563–2571, 2011.

[11] Ian T Jolliffe. *Principal component analysis*, volume 487. Springer-Verlag New York, 1986.

[12] Fabien Jourdan and G Melangon. Multiscale hybrid mds. In *Information Visualisation, 2004. IV 2004. Proceedings. Eighth International Conference on*, pages 388–393. IEEE, 2004.

[13] Richard C. T. Lee, James R. Slagle, and H Blum. A triangulation method for the sequential mapping of points from n-space to two-space. *Computers, IEEE Transactions on*, 100(3):288–292, 1977.

[14] Alistair Morrison, Greg Ross, and Matthew Chalmers. A hybrid layout algorithm for sub-quadratic multidimensional scaling. In *Information Visualization, 2002. INFOVIS 2002. IEEE Symposium on*, pages 152–158. IEEE, 2002.

[15] Fernando V Paulovich, DM Eler, J Poco, Charl P Botha, R Minghim, and LG Nonato. Piece wise laplacian-based projection for interactive data exploration and organization. In *Computer Graphics Forum*, volume 30, pages 1091–1100. Wiley Online Library, 2011.

[16] Fernando V Paulovich, Luis G Nonato, Rosane Minghim, and Haim Levkowitz. Least square projection: A fast high-precision multidimensional projection technique and its application to document mapping. *Visualization and Computer Graphics, IEEE Transactions on*, 14(3):564–575, 2008.

[17] Fernando V Paulovich, Claudio T Silva, and Luis G Nonato. Two-phase mapping for projecting massive data sets. *Visualization and Computer Graphics, IEEE Transactions on*, 16(6):1281–1290, 2010.

[18] Elzbieta Pekalska, Dick de Ridder, Robert PW Duin, and Martin A Kraaijveld. A new method of generalizing sammon mapping with application to algorithm speed-up. In *Fifth Annual Conference of the Advanced School for Computing and Imaging*, pages 221–228, 1999.

[19] John W Sammon Jr. A nonlinear mapping for data structure analysis. *Computers, IEEE Transactions on*, 100(5):401–409, 1969.

[20] Eduardo Tejada, Rosane Minghim, and Luis Gustavo Nonato. On improved projection techniques to support visual exploration of multi-dimensional data sets. *Information Visualization*, 2(4):218–231, 2003.

[21] Joshua B Tenenbaum, Vin De Silva, and John C Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000.