

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
DEPARTAMENTO ACADÊMICO DE ELETROTÉCNICA
CURSO DE ENGENHARIA INDUSTRIAL ELÉTRICA – ÊNFASE ELETROTÉCNICA

JOÃO PEDRO JUNGES CARVALHO

MIGUEL KOSMALA NETO

**AGRUPAMENTO DE PADRÕES DE CURVA DE CARGA
UTILIZANDO ALGORITMOS E TÉCNICAS DE AGRUPAMENTO
COMO ALTERNATIVA TARIFÁRIA.**

TRABALHO DE CONCLUSÃO DE CURSO

CURITIBA
2011

JOÃO PEDRO JUNGES CARVALHO

MIGUEL KOSMALA NETO

**AGRUPAMENTO DE PADRÕES DE CURVA DE CARGA
UTILIZANDO ALGORITMOS E TÉCNICAS DE AGRUPAMENTO
COMO ALTERNATIVA TARIFÁRIA.**

Trabalho de Conclusão de Curso de Graduação, apresentado à disciplina de Trabalho de Conclusão de Curso 2, do Curso Superior de Engenharia Industrial Elétrica - Eletrotécnica do Departamento Acadêmico de Eletrotécnica – DAELT – da Universidade Tecnológica Federal do Paraná – UTFPR, como requisito parcial para obtenção de aprovação.

Orientadora: Prof^ª. Andréa Lúcia Costa, Dr. Eng.

Co-orientador: Cláudio Martin, Dr. Eng.

CURITIBA

2011

SUMÁRIO

1 INTRODUÇÃO	5
1.1 TEMA	7
1.1.1 Delimitação do tema	7
1.2 PROBLEMA E PREMISSAS	7
1.3 OBJETIVOS	8
1.3.1 Objetivo geral.....	8
1.3.2 Objetivos específicos.....	8
1.4 JUSTIFICATIVA	9
1.5 ESTRUTURA DO TRABALHO.....	9
2 CONCEITOS DE ESTRUTURA TARIFÁRIA	10
2.1 INTRODUÇÃO	10
2.2 CONCEITOS	10
2.3 SAZONALIDADE	11
2.4 SEGMENTOS HORÁRIOS	12
2.5 SEGMENTOS SAZONAIS	13
2.7 MODALIDADES TARIFÁRIAS	14
2.8 TARIFAS DO GRUPO A	14
2.8.1 Estrutura tarifária convencional.....	15
2.8.2 Estrutura tarifária horossazonal.....	16
2.8.3 Tarifas horossazonal azul	17
2.8.4 Tarifas horossazonal verde	19
2.9 TARIFAS DO GRUPO B	20
2.10 OPÇÃO PELO MELHOR SISTEMA TARIFÁRIO.....	21
2.11 CONCLUSÃO.....	22
3 TÉCNICAS DE AGRUPAMENTO	23
3.1 INTRODUÇÃO	23
3.2 A MÁQUINA DE VETOR DE SUPORTE	25
3.3 TEORIA ESTATÍSTICA DA APRENDIZAGEM	25
3.4 VETORES DE SUPORTE.....	26
3.4.1 Conceitos básicos	26
3.4.2 Conceito de margem e vetor de suporte	28
3.4.3 Classificação de padrões linearmente separáveis	29
3.4.4 Hiperplano ótimo	30
3.4.5 Classificação de padrões não-linearmente separáveis.....	32
3.4.6 Dimensão VC (Vapnik-Chervonenkis)	33
3.4.7 Teoria de Lagrange	35
3.5 FUNÇÃO NÚCLEO	38
3.5.1 Espaço de características com alta dimensionalidade.....	39
3.5.2 Modularidade.....	42
3.6 CONCLUSÃO SOBRE A MÁQUINA DE VETOR DE SUPORTE.....	43
3.7 <i>K-MEANS</i>	44
3.7.1 Cálculos de distância mais rápidos	44
3.7.2 Discussão sobre <i>k-means</i>	45
3.7.3 Atribuindo pontos ao centróide mais próximo	47
3.7.4 Centróides e funções objetivo	47
3.7.5 Escolhendo os centróides iniciais.....	49
3.7.6 Tempo e complexidade do espaço	51

3.7.7 <i>K-means</i> : Aspectos adicionais.....	51
3.8 CONCLUSÃO DE <i>K-MEANS</i>	54
4. RESULTADOS DAS SIMULAÇÕES.....	56
4.1 INTRODUÇÃO	56
4.2 CURVAS DE CARGA	57
4.3 SIMULAÇÕES COMPUTACIONAIS	58
4.3.1 Algoritmos e ferramentas utilizadas.....	58
4.3.2 Conclusões sobre os <i>softwares</i> e algoritmos.....	59
4.3.3 Dados do arquivo ARQ3	62
4.3.4 Dados do arquivo ARQ5	65
4.4 ANÁLISE DOS AGRUPAMENTOS COM DIFERENTES VALORES DE <i>K</i>	68
4.5 CONCLUSÕES DOS RESULTADOS	74
5. CONSIDERAÇÕES FINAIS	76
6. REFERÊNCIAS	78
7. ANEXOS	82
ANEXO A – Algoritmo <i>k-means</i>	82
ANEXO B – Algoritmo para cálculo da distância euclidiana	84
ANEXO C – Algoritmo para cálculo do erro MAE (Mean Error Absolute).....	85
ANEXO D – Algoritmo para plotagem dos agrupamentos	86
8. CRONOGRAMA DETALHADO	87

1 INTRODUÇÃO

A desregulamentação do mercado de energia, ocorrida nos últimos 20 anos, introduziu novas formas de contratação de energia, diferenciadas daquelas praticadas pelas concessionárias e desenvolvidas por empresas independentes do mercado de energia.

Atualmente, critérios para particionamento dos consumidores são estabelecidos em concessionárias de vários países, sendo que a maioria deles é baseada nos valores nominais da carga e parâmetros relacionados ao tipo de atividade do consumidor. Geralmente este critério não se adapta corretamente com o perfil do consumidor.

Hoje no Brasil, uma faixa expressiva de consumidores tem o seu processo de medição efetuado com o medidor de energia horossazonal, uma vez que estão atrelados a contratos entre consumidor e concessionária. O medidor permite efetuar a medição de demanda de potência de 15 em 15 minutos, e, portanto, levantar a curva horária do perfil da demanda, o que se configura a curva de carga daquele consumidor.

No presente estágio da evolução dos sistemas de energia elétrica e mercados, há um específico e crescente interesse em estabelecer um processo de classificação de padrões de carga, especialmente do ponto de vista das empresas distribuidoras de energia (CHICCO; ILIE, 2009). A introdução de tecnologias avançadas de medição permite aos operadores do suprimento receber inúmeros dados de consumo do mercado consumidor, permitindo a análise utilizando algoritmos dedicados.

O crescente interesse na classificação de consumidores tem conduzido os estudos para múltiplas finalidades, entre as quais pode-se destacar:

- com o objetivo de classificar o tipo de consumidor de modo a estabelecer critérios de tarifação (CHEN; HWANG; HUANG, 1997);
- para analisar a interação do consumidor com o mercado frente à resposta a programas de demanda (VALERO et al., 2007);
- com o objetivo de classificação do padrão dos dados de carga de consumidores individuais referentes a diferentes períodos do ano (TSEKOURAS; HATZIARGYRIOU; DIALYNAS, 2007);
- para a detecção de pontos fora da curva exibindo comportamento anômalo, tais como aqueles afetados de várias formas por perdas não-técnicas (MONEDERO et al., 2006).

O particionamento das curvas de carga de consumidores e o seu agrupamento de acordo com certas características podem ser efetuados de várias formas e a literatura tem apresentado inúmeros trabalhos com algoritmos testados em diferentes tipos de dados, tais como o método *follow-the-leader*, *k-means*, *fuzzy k-means*, agrupamento hierárquico, mapas auto-organizáveis de Kohonen (*SOM - Self-organizing map*) e outros métodos estatísticos e de redes neurais.

O algoritmo *k-means* classifica informações de acordo com os próprios dados. Esta classificação, é baseada na análise e na comparação entre os valores numéricos do conjunto. Assim, o algoritmo fornecerá uma classificação sem a necessidade de supervisão. Por causa desta característica, o *k-means* é considerado como um algoritmo de aprendizagem não supervisionado.

Foi a partir da teoria estatística, que a máquina de vetor de suporte foi desenvolvida, sendo que a mesma foi introduzida pelo matemático russo Vladimir Naumovich Vapnik, a qual é constituída por dois estágios. Primeiramente são determinados os vetores suportes a partir dos dados das curvas de carga com diferentes formatos e o segundo estágio utiliza o resultado obtido na primeira fase para formar os agrupamentos (BURGES, 1998).

O trabalho aqui descrito, portanto, tem por finalidade efetuar o particionamento das curvas de carga e posterior agrupamento daquelas que sejam semelhantes dentro de critérios predefinidos. Para atingir este objetivo foram estudadas e analisadas algumas técnicas de agrupamento, como a máquina vetor de suporte e *k-means*, para subsidiar a formação de perfis de carga com a finalidade de estabelecer tarifas diferenciadas para cada perfil.

Os resultados obtidos com este trabalho foram: o entendimento sobre o sistema de contratação e medição de energia praticada pela concessionária; grandezas associadas ao processo de medição, que podem ser obtidos com o medidor horossazonal; estudo de técnicas de Inteligência Artificial e sua utilização para sistemas de energia; estudo de técnicas de programação e utilização de programas disponíveis livremente na Internet ou desenvolvidos via programação própria; obtenção de um agrupamento de perfis a partir dos dados de curvas de carga e dos algoritmos utilizados; relatório com o processo de simulação efetuado juntamente com a descrição dos programas utilizados ou desenvolvidos, resultados alcançados e comparações entre os mesmos.

1.1 TEMA

1.1.1 Delimitação do tema

Este trabalho busca agrupar curvas de demanda diária visando a demonstração de uma alternativa tarifária para consumidores industriais (geralmente com entrada em média-tensão – 13,8 kV) de diversos setores da região metropolitana de Curitiba – PR, independente da potência demandada.

Para isso utilizou-se algoritmos conjuntamente com rotinas e *softwares* livres já desenvolvidos e técnicas de agrupamento. As técnicas conhecidas como Máquina de Vetor de Suporte (*Support Vector Machine*) e *k-means* foram programadas e simuladas no *software* *MATLAB*[®].

1.2 PROBLEMA E PREMISSAS

Segundo Chicco et al. (2004), a abertura do mercado energético tem contribuído para que as empresas distribuidoras de energia elétrica sejam colocadas frente a frente a novas mudanças em fornecer serviços mais satisfatórios aos seus consumidores. A recente abertura do mercado no setor de energia elétrica, aliada a regulamentação no setor de distribuição da mesma, está possibilitando aos fornecedores (concessionárias e empresas de venda de energia) estudar as diferentes classes de consumidores de modo a definir tarifas dedicadas e de acordo com um conjunto de cláusulas de restrição impostas pelo marco regulatório.

Esta mudança requer que os consumidores sejam agrupados adequadamente em classes de consumo, refletindo o comportamento da evolução da curva de carga diária.

1.3 OBJETIVOS

1.3.1 Objetivo geral

Estabelecer um agrupamento de curvas de demanda diária industriais da região metropolitana de Curitiba utilizando algoritmos e técnicas de agrupamento (*clustering*) como alternativa tarifária.

1.3.2 Objetivos específicos

Os objetivos específicos a serem desenvolvidos foram:

- Fazer uma revisão bibliográfica dos sistemas de distribuição de energia, dando ênfase para os assuntos:
 - Contratação de fornecimento de energia;
 - Medição de energia em sistemas de distribuição;
 - Perfis de curva de carga;
 - Tarifação no mercado de energia elétrica;
- Estudar redes neurais, aprendizagem de máquina, *perceptron* e mapas auto-organizáveis de Kohonen (*SOM - Self-organizing map*).
- Estudar a separação de padrões com a utilização da Máquina de Vetor de Suporte e *k-means*;
- Estudar os processos de agrupamento (*clustering*);
- Estudar os algoritmos para determinação dos vetores de suporte, verificando as variáveis de entrada e saída e as adaptações necessárias para a solução do problema específico;
- Realizar simulações para diferentes perfis de carga e parâmetros;
- Efetuar os agrupamentos e compará-los, analisando as diferenças para as simulações efetuadas.

1.4 JUSTIFICATIVA

Já que a introdução de uma tarifação para cada consumidor não é factível, uma boa alternativa é uma estrutura composta por agrupamentos de consumidores similares.

“Uma justa política tarifária deve proporcionar uma tarifa que possa, ao mesmo tempo, induzir o consumidor ao uso racional e econômico da energia elétrica e promover a eficiência da empresa prestadora do serviço, com o máximo de qualidade e produtividade.” (BITU; BORN, 1993).

Formar grupos tarifários com base no perfil de carga individual, em função da semelhança do comportamento da mesma, possibilita agregar os consumidores que impõem custos similares ao sistema (ANDRADE, 2003), para que estes sejam tarifados conforme seus perfis.

“Enfim, a formação de agrupamentos de consumidores com base no perfil de carga individual, além de aumentar a oferta de serviços diferenciados com o atendimento personalizado ao consumidor, pode ainda, ser o alicerce de uma nova tarifa de energia elétrica customizada e capaz de promover a melhoria da racionalidade do uso dos recursos energéticos do país.” (ANDRADE, 2003).

1.5 ESTRUTURA DO TRABALHO

O Trabalho de Conclusão de Curso apresentará 5 capítulos como descrito a seguir.

No Capítulo 1 será apresentado a proposta do trabalho, descrevendo os objetivos Geral e Específicos, o problema a ser estudado, bem como a justificativa para este estudo. Também neste capítulo é descrita a metodologia a ser empregada no trabalho.

No Capítulo 2 serão apresentados os conceitos sobre medição de energia, contratos de fornecimento, perfis de curva de carga e tarifação.

No Capítulo 3 será apresentado os principais conceitos aplicados em aprendizagem de máquina, além de explicar sobre a Máquina de Vetor de Suporte, *k-means* e os processos de agrupamento (*clustering*).

No Capítulo 4 serão apresentados os dados coletados em empresas e os resultados das simulações realizadas.

Por fim, o Capítulo 5 apresentará as principais conclusões obtidas no trabalho.

2 CONCEITOS DE ESTRUTURA TARIFÁRIA

2.1 INTRODUÇÃO

A fatura de energia reflete o modo como a energia elétrica é utilizada e sua análise por um período de tempo adequado, permite estabelecer relações importantes entre hábitos e consumo.

Dadas as alternativas de enquadramento tarifário disponíveis para alguns consumidores, o conhecimento da formação da fatura e dos hábitos de consumo permite escolher a forma de tarifação mais adequada e que resulta em menor despesa com a energia elétrica (PROCEL, 2001).

Este capítulo apresenta os principais conceitos para o entendimento das estruturas tarifárias utilizadas no Brasil.

2.2 CONCEITOS

Define-se estrutura tarifária como sendo o conjunto de tarifas aplicáveis aos Componentes de Consumo de Energia elétrica e/ou Demanda de Potência, de acordo com a modalidade de fornecimento (ANEEL, 2005).

Dessa forma, é importante definir:

- **Demanda:** média das potências elétricas ativas ou reativas, solicitadas ao sistema elétrico pela parcela da carga instalada em operação na unidade consumidora, durante um intervalo de tempo especificado, expressa em quilowatts (kW) e quilovolt-ampère-reactivo (kvar), respectivamente.
- **Demanda Contratada:** demanda de potência ativa a ser obrigatória e continuamente disponibilizada pela distribuidora, no ponto de entrega, conforme valor e período de vigência fixados em contrato, e que deve ser integralmente paga, seja ou não utilizada durante o período de faturamento, expressa em quilowatts (kW).

- **Demanda Faturável:** valor da demanda de potência ativa, considerada para fins de faturamento, com aplicação da respectiva tarifa, expressa em quilowatts (kW).
- **Demanda Medida:** maior demanda de potência ativa, verificada por medição, integralizada no intervalo de 15 (quinze) minutos durante o período de faturamento, expressa em quilowatts (kW).
- **Consumo de Energia:** a energia elétrica consumida por uma instalação/equipamento compõem-se de uma parcela ativa, e outra reativa.
- **Energia Ativa:** é a energia que será convertida em outra forma de energia, e que é cobrada de todos os consumidores, ou seja, é o que se paga na fatura de energia elétrica (SORIA; FILIPINI, 2010).
- **Energia Reativa:** é a energia que circula continuamente entre os diversos campos elétricos e magnéticos, de um sistema de corrente alternada, sem produzir trabalho (SORIA; FILIPINI, 2010).

A estrutura de tarifária depende de alguns critérios relacionados ao cliente consumidor. São eles:

- Demanda contratada;
- Nível de tensão;
- Classes e subclasses de consumo;
- Sazonalidade diária e anual.

2.3 SAZONALIDADE

As estruturas tarifárias são, excetuando o caso da tarifa Convencional (que será visto no item 2.8.1), segmentadas segundo o período em que a energia é consumida. Existem duas diferenciações: diária (Ponta - P, Fora de ponta - FP), obedecendo às variações decorridas durante as vinte e quatro horas do dia, e anual (seco - S, úmido - U), obedecendo às diferenças climáticas entre os meses do ano.

Segundo a ANEEL, em sua Resolução nº 414/2010, a segmentação dos chamados postos tarifários é feita da forma como mostrada a seguir.

2.4 SEGMENTOS HORÁRIOS

Os segmentos horários, dividem-se em horário de ponta (P) e horário fora de ponta (FP):

- a. Horário de ponta (P): período composto por 3 (três) horas diárias consecutivas definidas pela distribuidora considerando a curva de carga de seu sistema elétrico, aprovado pela ANEEL para toda a área de concessão, com exceção feita aos sábados, domingos, terça-feira de carnaval, sexta-feira da Paixão, Corpus Christi, e os seguintes feriados descritos na Tabela 2.1:

Tabela 2.1– Feriados nacionais

Dia e mês	Feriados nacionais	Leis federais
01 de janeiro	Confraternização Universal	10.607 de 19/12/2002
21 de abril	Tiradentes	10.607 de 19/12/2002
01 de maio	Dia do Trabalho	10.607 de 19/12/2002
07 de setembro	Independência	10.607 de 19/12/2002
12 de outubro	Nossa Senhora Aparecida	6.802 de 30/06/1980
02 de novembro	Finados	10.607 de 19/12/2002
15 de novembro	Proclamação da República	10.607 de 19/12/2002
25 de dezembro	Natal	10.607 de 19/12/2002

Fonte: ANEEL, 2010.

- b. Horário fora de ponta (FP): Período composto pelo conjunto das horas diárias consecutivas e complementares àquelas definidas no horário de ponta.

É justamente no horário de ponta que as redes de distribuição assumem o maior valor de carga, atingindo o seu valor máximo aproximadamente às 19h, com pequenas variações de região para região. Conclui-se portanto, que devido ao maior carregamento das redes de distribuição, o atendimento de um novo consumidor neste horário de maior solicitação, custará mais à concessionária, do que em qualquer outro horário do dia, tendo em vista a necessidade de ampliação de seu sistema.

2.5 SEGMENTOS SAZONAIS

A geração de energia elétrica no Brasil é essencialmente hidrelétrica dependendo portanto, do regime de chuvas. Os períodos do ano são definidos como seco e úmido (ANEEL, 2010), determinados da seguinte forma:

- a. Período seco (S): 7 (sete) meses consecutivos, de maio a novembro, inclusive;
- b. Período úmido (U): 5 (cinco) meses consecutivos, dezembro de um ano a abril do ano seguinte.

A Figura 2.1 mostra os períodos seco e úmido durante os meses do ano, sendo que a curva “A” representa a disponibilidade média da quantidade de água nos reservatórios das usinas hidrelétricas, constituindo portanto, o potencial para geração de energia elétrica, já a curva “B” representa o comportamento médio do mercado de energia elétrica, a nível nacional, assumindo um valor máximo justamente no período em que a disponibilidade de água é mínima. Este fato, permite identificar, em função da disponibilidade hídrica do ano, o denominado período seco, aquele compreendido entre maio e novembro do mesmo ano, e período úmido, de dezembro até abril do ano seguinte (RAMOS, 2003).

O atendimento ao mercado no período seco só é possível em virtude da capacidade de acumulação dos reservatórios das usinas que estocam água afluyente durante o ano. Percebe-se assim, que o fornecimento de energia elétrica no período seco, é também mais oneroso, pois leva à necessidade de se construir grandes reservatórios e, eventualmente, colocar em operação usinas térmicas alimentadas, na maioria das vezes, por energéticos importados.

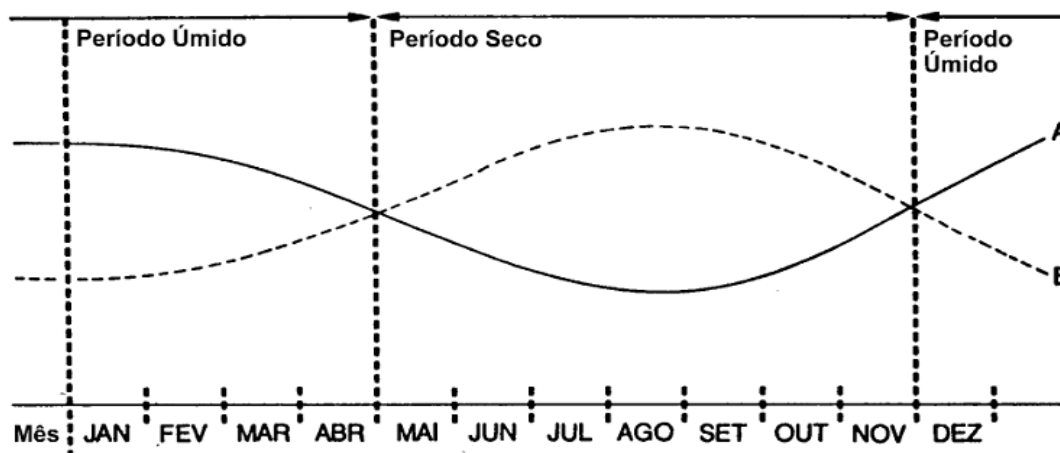


Figura 2.1 – Comportamento do consumo/demanda durante o ano.

Fonte: RAMOS, 2003.

2.7 MODALIDADES TARIFÁRIAS

São duas as modalidades tarifárias (ANEEL, 2010).

Os consumidores do Grupo B (baixa tensão) têm tarifa monômnia, isto é, são cobrados apenas pela energia que consomem.

Os consumidores do Grupo A tem tarifa binômnia, isto é, são cobrados tanto pela demanda quanto pela energia que consomem. Estes consumidores podem enquadrar-se em uma de três alternativas tarifárias:

- Tarifação convencional,
- Tarifação horossazonal verde, ou
- Tarifação horossazonal.

2.8 TARIFAS DO GRUPO A

As tarifas do “Grupo A” são para consumidores com fornecimento em tensão igual ou superior a 2,3 kV, ou, ainda, atendidos a partir de sistema subterrâneo de distribuição em tensão secundária, caracterizado pela estruturação tarifária binômnia e subdividido nos seguintes subgrupos:

- Subgrupo A1 – tensão de fornecimento igual ou superior a 230 kV;
- Subgrupo A2 – tensão de fornecimento de 88 kV a 138 kV;

- Subgrupo A3 – tensão de fornecimento de 69 kV;
- Subgrupo A3a – tensão de fornecimento de 30 kV a 44 kV;
- Subgrupo A4 – tensão de fornecimento de 2,3 kV a 25 kV;
- Subgrupo AS – tensão de fornecimento inferior a 2,3 kV, atendidos a partir de sistema subterrâneo de distribuição.

A Figura 2.2 exemplifica os tipos de consumidores com diferentes modalidades tarifárias.

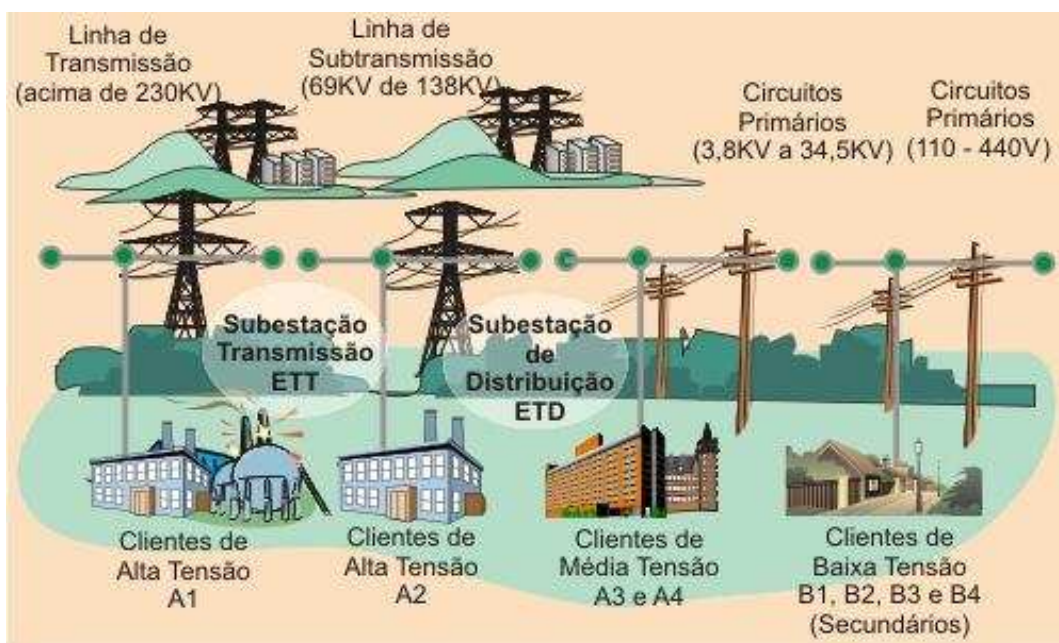


Figura 2.2 – Modalidades tarifárias.

Fonte: COPEL, 2008.

As tarifas do “Grupo A” são construídas em três modalidades de fornecimento:

- Convencional;
- Horossazonal azul;
- Horossazonal verde.

2.8.1 Estrutura tarifária convencional

Esta modalidade é caracterizada pela aplicação de tarifas de consumo de energia elétrica e demanda de potência independentemente das horas de utilização do dia e dos períodos do ano.

O enquadramento na tarifa Convencional exige um contrato específico com a concessionária no qual se pactua um único valor da demanda pretendida pelo consumidor (Demanda Contratada), independentemente da hora do dia (P ou FP) ou período do ano (S ou U).

O consumidor atendido em alta tensão pode também optar pela estrutura tarifária convencional, se atendido em tensão de fornecimento abaixo de 69 kV, sempre que tiver contratado uma demanda inferior a 300 kW.

A fatura de energia elétrica desses consumidores é composta da soma de parcelas referentes ao consumo, demanda e ultrapassagem. A parcela de consumo é calculada multiplicando-se o consumo medido pela tarifa de consumo.

A parcela de demanda é calculada multiplicando-se a tarifa de demanda pela demanda contratada ou pela demanda medida (a maior delas), caso esta não ultrapasse em 5% a demanda contratada.

Nos casos em que ocorre ultrapassagem acima de 5% da demanda contratada é cobrada uma parcela calculada multiplicando-se a tarifa de ultrapassagem pelo valor da demanda medida que supera a demanda contratada.

2.8.2 Estrutura tarifária horossazonal

A estrutura tarifária horossazonal é caracterizada pela aplicação de tarifas diferenciadas de consumo de energia elétrica e de demanda de potência, de acordo com os postos horários, as horas de utilização do dia e dos períodos do ano. O objetivo dessa estrutura tarifária é racionalizar o consumo de energia elétrica ao longo do dia e do ano, motivando o consumidor, pelo valor diferenciado das tarifas, a consumir mais energia elétrica nos horários do dia e nos períodos do ano em que ela for mais barata (ANEEL, 2005). Tem por finalidade deslocar cargas para períodos do dia em que o sistema tenha maior disponibilidade, assim como, orientar o consumo de energia para o período do ano em que há maior disponibilidade hídrica.

Para as horas do dia são estabelecidos dois períodos, denominados postos tarifários: o posto tarifário “Ponta (P)” e o posto tarifário “Fora da Ponta (FP)”.

Já para o ano, são estabelecidos dois períodos: “período seco (S)”, quando a incidência de chuvas é menor, e “período úmido (U)” quando é maior o volume de chuvas. As

tarifas no período seco são mais altas, refletindo o maior custo de produção de energia elétrica devido à menor quantidade de água nos reservatórios das usinas hidrelétricas, provocando a eventual necessidade de complementação da carga por geração térmica, que é mais cara.

A tarifa horossazonal é dividida em duas formas de precificação distintas: Azul e Verde.

2.8.3 Tarifas horossazonal azul

A tarifa horossazonal azul é a modalidade de fornecimento estruturada para a aplicação de tarifas diferenciadas de consumo de energia elétrica, de acordo com as horas de utilização do dia e dos períodos do ano, bem como de tarifas diferenciadas de demanda de potência, de acordo com as horas de utilização do dia (ANEEL, 2010).

É aplicável obrigatoriamente na modalidade tarifária horossazonal azul as unidades consumidoras com tensão de fornecimento igual ou superior a 69 kV, opcional para demandas maior ou igual a 30 kW.

A tarifa horossazonal azul tem a seguinte estrutura:

Demanda de potência (kW):

- ✓ Uma tarifa para o horário de ponta (P);
- ✓ Uma tarifa para o horário fora de ponta (FP).

Consumo de energia (kWh):

- ✓ Uma tarifa para o horário de ponta em período úmido (PU);
- ✓ Uma tarifa para o horário fora de ponta em período úmido (FPU);
- ✓ Uma tarifa para o horário de ponta em período seco (PS);
- ✓ Uma tarifa para o horário fora de ponta em período seco (FPS).

A Figura 2.3 explica, por meio de um diagrama, a estrutura da tarifa horossazonal azul.

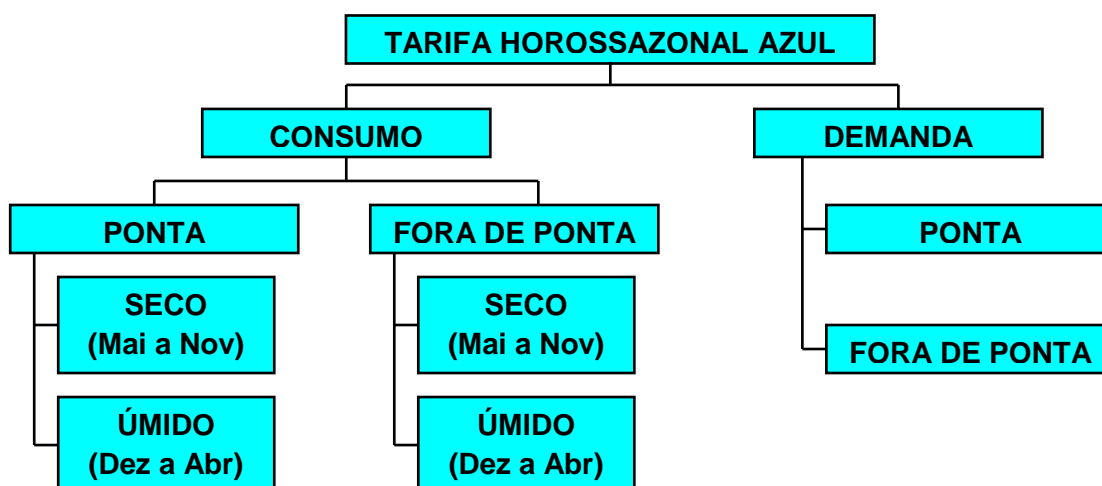


Figura 2.3 – Tarifa horossazonal azul.
Fonte: Aatoria própria.

Essa modalidade tarifária exige um contrato específico com a concessionária no qual se pactua tanto o valor da demanda pretendida pelo consumidor no horário de ponta (demanda contratada na ponta) quanto o valor pretendido nas horas fora de ponta (demanda contratada fora de ponta). As tarifas de consumo na ponta e fora de ponta são diferenciadas por período do ano, sendo mais caras no período seco (maio à novembro).

Para fins de tarifação, a parcela de demanda é calculada somando-se o produto da tarifa de demanda na ponta pela demanda contratada na ponta (ou pela demanda medida na ponta, de acordo com as tolerâncias de ultrapassagem) ao produto da tarifa de demanda fora da ponta pela demanda contratada fora de ponta (ou pela demanda medida fora de ponta, de acordo com as tolerâncias de ultrapassagem). As tarifas de demanda não são diferenciadas por período do ano.

A parcela de ultrapassagem é cobrada apenas quando a demanda medida ultrapassa a demanda contratada acima dos limites de tolerância que é de 5%.

A parcela de ultrapassagem é calculada multiplicando-se a tarifa de ultrapassagem pelo valor da demanda medida que supera a demanda contratada. As tarifas de ultrapassagem são diferenciadas por horário, sendo mais caras nas horas de ponta.

2.8.4 Tarifas horossazonal verde

A tarifa horossazonal verde é a modalidade de fornecimento estruturada para a aplicação de tarifas diferenciadas de consumo de energia elétrica, de acordo com as horas de utilização do dia e os períodos do ano, assim como de uma única tarifa de demanda de potência (ANEEL, 2010).

A tarifa horossazonal verde é opcional às unidades consumidoras atendidas pelo sistema elétrico interligado com tensão de fornecimento inferior a 69 kV e demanda contratada maior ou igual a 30 kW, com opção do consumidor pela modalidade azul ou verde se igual ou maior que 300 kW.

A tarifa horossazonal verde tem a seguinte estrutura:

Demanda de potência (kW):

- ✓ Uma tarifa única.

Consumo de energia (kWh):

- ✓ Uma tarifa para o horário de ponta em período úmido (PU);
- ✓ Uma tarifa para o horário fora de ponta em período úmido (FPU);
- ✓ Uma tarifa para o horário de ponta em período seco (PS);
- ✓ Uma tarifa para o horário fora de ponta em período seco (FPS).

A Figura 2.4 mostra o diagrama de tarifação de tarifa horossazonal verde.

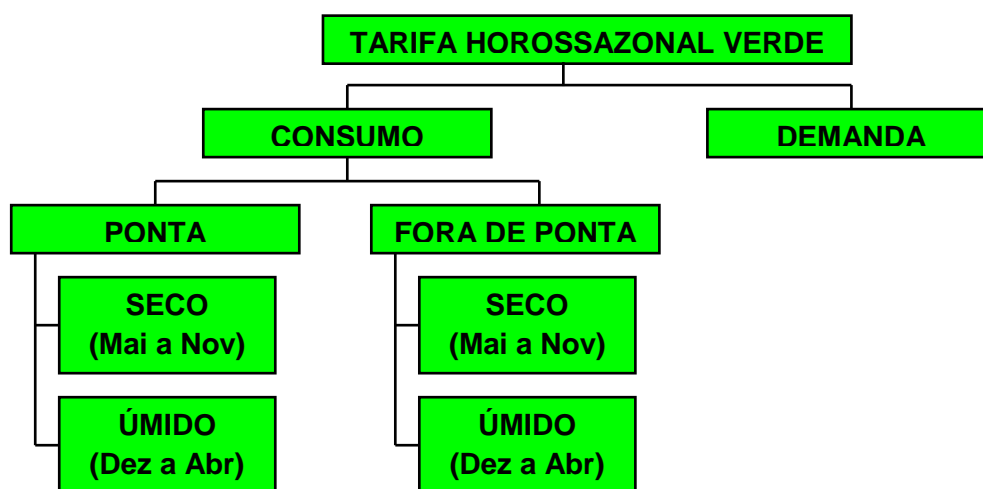


Figura 2.4 – Tarifa horossazonal verde.
Fonte: Autoria própria.

Essa modalidade tarifária exige um contrato específico com a concessionária no qual se pactua a demanda pretendida pelo consumidor (demanda contratada), independentemente da hora do dia (ponta ou fora de ponta).

Para fins de tarifação, a fatura de energia elétrica desses consumidores é composta da soma de parcelas referentes ao consumo (na ponta e fora dela), demanda e ultrapassagem, sendo que no período seco (maio à novembro) as tarifas de consumo na ponta e fora de ponta são mais caras que no período úmido.

A parcela de demanda é calculada multiplicando-se a tarifa de demanda pela demanda contratada ou pela demanda medida (a maior delas), caso esta não ultrapasse em mais de 5% a demanda contratada. A tarifa de demanda é única, independente da hora do dia ou período do ano.

A parcela de ultrapassagem é cobrada apenas quando a demanda medida ultrapassa em mais de 5% a demanda contratada. É calculada multiplicando-se a tarifa de ultrapassagem pelo valor da demanda medida que supera a demanda contratada.

2.9 TARIFAS DO GRUPO B

As tarifas do “Grupo B” se destinam às unidades consumidoras com fornecimento em tensão inferior a 2,3 kV, caracterizado pela estruturação tarifária monômnia e subdividido nos seguintes subgrupos:

- Subgrupo B1 – residencial;
- Subgrupo B2 – rural;
- Subgrupo B3 – demais classes;
- Subgrupo B4 – iluminação pública.

As tarifas do “grupo B” são estabelecidas somente para o componente de consumo de energia, em reais por megawatt-hora, considerando que o custo da demanda de potência está incorporado ao custo do fornecimento de energia em megawatt-hora (ESCELSA, 2004).

Unidades consumidoras ligadas em tensão primária podem optar por faturamento com aplicação da tarifa do grupo B, correspondente à respectiva classe, se atendido pelo menos um dos seguintes critérios:

- a potência nominal total dos transformadores for igual ou inferior a 112,5 kVA;

- a potência nominal total dos transformadores for igual ou inferior a 750 kVA, se classificada na subclasse cooperativa de eletrificação rural;
- a unidade consumidora se localizar em área de veraneio ou turismo cuja atividade seja a exploração de serviços de hotelaria ou pousada, independentemente da potência nominal total dos transformadores;
- quando, em instalações permanentes para a prática de atividades esportivas ou parques de exposições agropecuárias, a carga instalada dos refletores utilizados na iluminação dos locais for igual ou superior a 2/3 (dois terços) da carga instalada total (ANEEL, 2010).

Ainda há a possibilidade, quando a unidade consumidora tiver carga instalada superior a 75 kW e for atendida por sistema subterrâneo de distribuição em tensão secundária, optar pela mudança para o grupo A, com aplicação da tarifa do subgrupo AS.

2.10 OPÇÃO PELO MELHOR SISTEMA TARIFÁRIO

Respeitados os critérios de classificação obrigatória o cliente responsável por uma unidade consumidora poderá optar pela tarifa que melhor lhe convier. Com base nos dados de sua demanda e consumo, poderá fazer uma análise econômica criteriosa, simulando o faturamento de sua conta, para efeito de comparação, com a aplicação das tarifas verde, azul ou convencional.

Em linhas gerais, a estrutura convencional é indicada para unidades consumidoras que utilizam a energia elétrica de uma forma mais irregular, tais como condomínios residenciais.

A estrutura horossazonal verde é mais indicada para a unidade consumidora que possam reduzir a carga no "horário de ponta" e que tenha um Fator de Carga na Ponta baixo.

Já estrutura horossazonal azul é mais recomendada às unidades consumidoras industriais que tenham dificuldade em diminuir ou paralisar suas atividades no "horário de ponta" e que tenham um Fator de Carga na Ponta elevado, e, portanto, apresentam um consumo significativo de energia elétrica nesse período, além de apresentarem um elevado grau de eficiência na utilização da Demanda Contratada, ou seja, apresentam um consumo maior por quilowatt de demanda (ESCELSA, 2004).

2.11 CONCLUSÃO

Este capítulo apresentou os principais conceitos sobre tarifação de energia elétrica, explicando os diferentes tipos de estruturas de tarifação, bem como a classificação dos consumidores.

Este trabalho engloba consumidores de diversos setores da região metropolitana de Curitiba – PR que utilizam uma das três estruturas tarifárias supracitadas: convencional, horossazonal verde e horossazonal azul.

Para o bom entendimento dos resultados obtidos através das técnicas de agrupamento é importante conhecer os termos técnicos fundamentais para a análise dos perfis de carga, que possibilitem uma base para avaliar as tarifas existentes, bem como futuros estudos de alternativa tarifária.

3 TÉCNICAS DE AGRUPAMENTO

3.1 INTRODUÇÃO

Este capítulo apresenta os principais conceitos estudados para o desenvolvimento deste trabalho. Como o objetivo do trabalho é estabelecer os melhores agrupamentos para curvas de carga de indústrias de Curitiba, foram pesquisadas algumas técnicas de agrupamento e ferramentas computacionais que poderiam separar as curvas em grupos de acordo com suas características. Neste trabalho destacam-se o estudo e utilização de algoritmos baseados na Máquina de Vetor de Suporte e também o algoritmo *k-means*.

As técnicas de agrupamento são aplicadas quando não existem classes a serem previstas. Geralmente faz-se uso delas quando as amostras devem ser divididas em grupos com características naturais entre si (HUTCHINSON, 1994).

É chamado *cluster* todo conjunto de amostras que estão próximas entre si. A palavra “próximas” só faz sentido quando as amostras estudadas estão em um espaço métrico (conjunto munido de uma métrica/distância).

Espaço métrico (X, d) é um conjunto X dotado de uma função $d: X^2 \rightarrow R$ chamada métrica ou distância que associa a cada par de elementos de X uma distância entre eles. Esta distância deve satisfazer os seguintes axiomas:

$d(x, y)$ é um número real, não negativo e finito

$d(x, y) = 0 \Leftrightarrow x = y$

$d(x, y) = d(y, x)$ (simetria)

$d(x, z) \leq d(x, y) + d(y, z)$ (desigualdade triangular)

O protótipo natural para um espaço métrico é o mundo real, onde a métrica é a distância euclidiana que se pode medir com uma régua. Cada posição no espaço euclidiano pode ser descrito por três coordenadas. Isto significa que é possível mensurar uma média de várias posições. Portanto, por exemplo, ao se avistar uma revoada de gaiivotas, pode-se falar sobre o centro do bando. Este é um ponto onde pode não haver nenhuma ave. De fato, se as gaiivotas estão todas em um círculo, então o centro do bando está a uma certa distância de qualquer um dos pássaros (HUTCHINSON, 1994).

Sendo a distância euclidiana d_E :

$$\text{Seja } x = (x_1, x_2, \dots, x_n) \quad \text{e} \quad y = (y_1, y_2, \dots, y_n)$$

$$d_E = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \dots + (x_n - y_n)^2}$$

Para realizar o agrupamento de qualquer tipo, é necessário uma pseudo-métrica. A pseudo-métrica é uma função que satisfaz a todos os axiomas da métrica e que excetua a distância entre dois pontos distintos que pode ser zero.

Estes *clusters* refletem mecanismos de funcionamento no domínio destas amostras. Tal mecanismo faz com que em alguns casos os *clusters* tenham uma forte semelhança entre si para com as amostras restantes. Agrupamentos, naturalmente, requerem técnicas diferentes das de classificação, pois são métodos de aprendizagem que buscam realizar associações.

Existem diferentes formas em que o resultado do agrupamento pode ser expresso. Os grupos podem ser identificados como: exclusivos, sobrepostos, probabilísticos, ou hierárquicos (WITTEN, 2005).

Quando exclusivo, qualquer amostra deve pertencer a um único grupo. Caso seja um grupo classificado como sobreposto às amostras podem pertencer a mais de um grupo. Já os grupos com característica probabilística revelam que cada amostra pertence a cada grupo com uma certa probabilidade. E no agrupamento considerado como hierárquico, há uma divisão grosseira dos dados de nível superior, que são refinados posteriormente, normalmente recaindo em casos individuais.

Realmente, a escolha entre essas possibilidades deve ser ditada pela natureza dos mecanismos por trás dos fenômenos particulares de agrupamento. No entanto, tais mecanismos raramente são conhecidos - a própria existência de *clusters* é, afinal de contas, algo que se tenta descobrir - e também por razões pragmáticas, a escolha é geralmente ditada pelas ferramentas de agrupamento que estão disponíveis (WITTEN, 2005).

As diferenças entre os métodos de agrupamento e as redes neurais são qualitativas e quantitativas. Todas as entradas para uma rede neural são simples e similares, e normalmente há muitas delas.

3.2 A MÁQUINA DE VETOR DE SUPORTE

Fundamentada na Teoria da Aprendizagem Estatística, a Máquina de Vetor Suporte, do inglês Support Vector Machine - SVM, foi desenvolvida com o intuito de resolver problemas de classificação de padrões (VAPNIK, 1995).

Segundo Vapnik (2000) a máquina de vetor de suporte é um sistema de aprendizagem que faz uso de um algoritmo baseado na teoria estatística de aprendizagem. Em linhas gerais, os vetores do espaço de entrada são mapeados não-linearmente em um espaço com características de alta dimensionalidade. Neste espaço então é construída uma superfície de decisão linear, que se constitui de um hiperplano de separação ótima e que apresenta propriedades que garantem alta habilidade de generalização da máquina de aprendizagem.

Exemplos de aplicações de sucesso podem ser encontrados em diversos domínios, como na categorização de textos (JOACHIMS, 2002), na análise de imagens (JUNG; KIM; PARK, 2002) (PONTIL; VERRI, 1998), na Bolsa de Valores e no estudo de tarifas, por exemplo.

Devido a sua eficiência em trabalhar com dados de alta dimensionalidade, a Máquina de Vetor de Suporte é reportada na literatura como uma técnica altamente robusta, muitas vezes comparada as Redes Neurais (MUKKAMALA; SUNG, 2003).

3.3 TEORIA ESTATÍSTICA DA APRENDIZAGEM

A Teoria do Aprendizado Estatístico visa estabelecer condições matemáticas que permitem escolher um classificador, com bom desempenho, para o conjunto de dados disponíveis para treinamento e teste (CARVALHO; LORENA, 2003). Em outras palavras esta teoria busca encontrar um bom classificador levando em consideração todo o conjunto de dados, porém se abstendo de casos particulares.

As técnicas de Aprendizado de Máquina (AM) utilizam um princípio de dedução denominado indução, através do qual chegam-se a conclusões genéricas a partir de um conjunto de exemplos (CARVALHO; LORENA, 2007). O aprendizado indutivo pode ser supervisionado ou não-supervisionado.

No aprendizado supervisionado tem-se a figura de um professor externo, o qual apresenta o conhecimento do ambiente por conjuntos de exemplos na forma: entrada, saída desejada (HAYKIN, 1999).

No aprendizado não-supervisionado não há um professor. O algoritmo de AM aprende a agrupar as entradas submetidas segundo uma medida de qualidade. Essas técnicas são utilizadas principalmente quando o objetivo for encontrar padrões ou tendências que auxiliem no entendimento dos dados (SOUTO et al., 2003).

O tipo de aprendizado utilizado neste trabalho será o não-supervisionado dado que o algoritmo terá que, a partir dos dados a ele fornecido, agrupar aqueles que sejam semelhantes num único *cluster*.

Sabe-se que as técnicas de AM devem ser robustas a ponto de serem capazes de lidar com dados que contenham algum tipo de ruído, porém os dados utilizados neste trabalho são determinísticos, por conseguinte não contém ruídos.

3.4 VETORES DE SUPORTE

3.4.1 Conceitos básicos

O desempenho desejado de um classificador f é que o mesmo obtenha o menor erro durante o treinamento, sendo o erro mensurado pelo número de predições incorretas de f . Sendo assim, é definido risco empírico $R_{emp}(f)$ como sendo a medida de perda entre a resposta desejada e a resposta real. A equação (3.1) mostra a definição do risco empírico.

$$R_{emp}(f) = \frac{1}{n} \sum_{i=1}^n c(f(x_i), y_i) \quad (3.1)$$

sendo $c(\cdot)$ a função de custo relacionada a previsão $f(x_i)$ com a saída desejada y_i (CARVALHO; LORENA, 2003), na qual um tipo de função de custo é a “perda 0/1” definida pela equação (3.2).

$$c(f(x_i), y_i) = \begin{cases} 1 & , \text{se } y_i f(x_i) < 0 \\ 0 & , \text{caso contrário} \end{cases} \quad (3.2)$$

O processo de busca por uma função f' que represente um menor valor de $R_{emp}(f)$ é denominado de *Minimização do Risco Empírico*.

Sobre a hipótese de que os padrões de treinamento (x_i, y_i) são gerados por uma distribuição de probabilidade $P(x, y)$ em $\mathfrak{R}^N \times \{-1, +1\}$ sendo P desconhecida. A probabilidade de classificação incorreta do classificador f é denominada de Risco Funcional, que quantifica a capacidade de generalização, conforme é mostrado pela equação (3.3) (SMOLA et al., 1999a) (SMOLA et al., 1999b).

$$R(f) = \int c(f(x_i), y_i) dP(x_i, y_i) \quad (3.3)$$

Durante processo de treinamento, $R_{emp}(f)$, pode ser facilmente obtido, ao contrário de $R(f)$, pois em geral a distribuição de probabilidades P é desconhecida (CARVALHO; LORENA, 2003).

A partir disto, dado um conjunto de dados de treinamento (x_i, y_i) com $x_i \in \mathfrak{R}^N$ e $y_i \in \{-1, +1\}$, $i = \{1, 2, \dots, n\}$, sendo x_i o vetor de entrada e y_i o rótulo da classe.

O objetivo então é estimar uma função $f: \mathfrak{R}^N \rightarrow \{-1, +1\}$. Caso nenhuma restrição seja imposta na classe de funções em que se escolhe a estimativa f , pode ocorrer que a função obtenha um bom desempenho no conjunto de treinamento, porém não tendo o mesmo desempenho em padrões desconhecidos, sendo este fenômeno denominado de “overfitting”.

Em outras palavras, a minimização apenas do risco empírico $R_{emp}(f)$ não garante uma boa capacidade de generalização, sendo desejado um classificador f^* tal que $R(f^*) = \min_{f \in F} R(f)$, onde F é o conjunto de funções f possíveis.

A Figura 3.1 mostra um exemplo onde uma classe de funções pode ser utilizada para separar padrões linearmente separáveis. É necessário determinar uma função que minimize o R_{emp} , representado na figura pela reta do Hiperplano Ótimo.

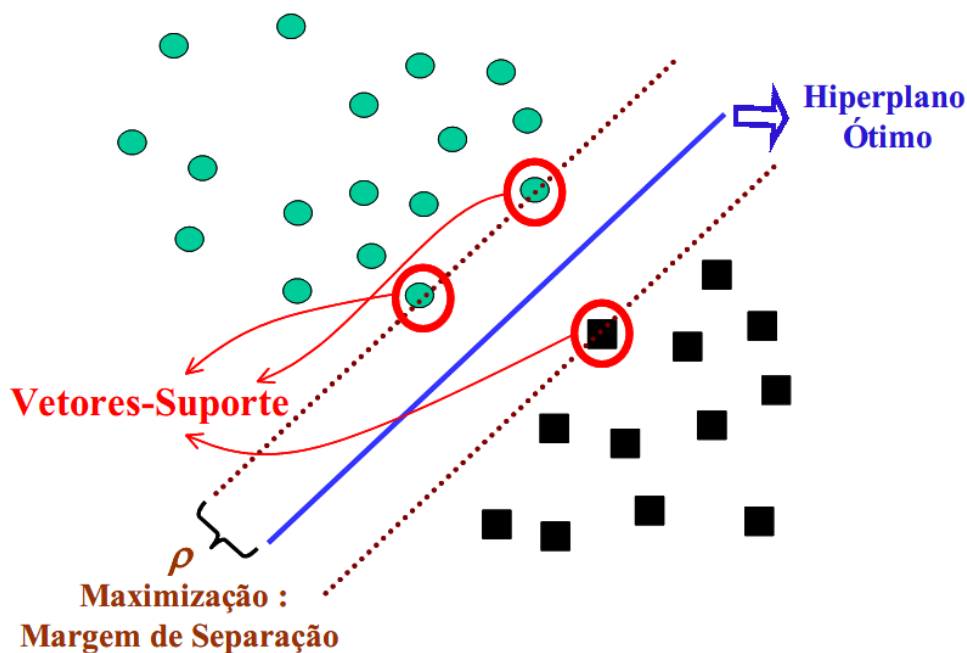


Figura 3.1 – Classe de hiperplanos com um hiperplano ótimo.
Fonte: SEMOLINI, 2002.

A TAE (Teoria de Aprendizado Estatístico) provê formas de limitar a classe de funções (hiperplanos), com o intuito de prevenir modelos ruins, ou seja, que levem ao “overfitting”, implementando uma função com a capacidade adequada para o conjunto de dados de treinamento (HEARST et al., 1998). Estas limitações são impostas ao risco funcional da função. Os limites utilizam o conceito de dimensão Vapnik-Chervonenkis (VC) que será explanado mais adiante.

3.4.2 Conceito de margem e vetor de suporte

Sendo $f(x) = (w \cdot x) + b$ um hiperplano, pode-se definir como margem a menor distância entre os exemplos do conjunto de treinamento e o hiperplano utilizado para separação destas classes (LORENA; CARVALHO, 2003b). A margem determina quão bem duas classes podem ser separadas (SMOLA et al., 1999b). A margem ρ de um classificador f é definida por:

$$\rho = \min_i y_i f(x_i) \quad (3.4)$$

A margem é obtida pela distância entre o hiperplano e os vetores que estão mais próximos a ele, sendo estes vetores denominados de vetores de suporte. De acordo com (SMOLA et al., 1999b) os vetores de suporte são padrões críticos, que sozinhos determinam o hiperplano ótimo, sendo os outros padrões (não-críticos) irrelevantes, podendo ser removidos do conjunto de treinamento sem afetar os resultados. Na Figura 3.2 os vetores de suporte são destacados por círculos externos nos padrões.

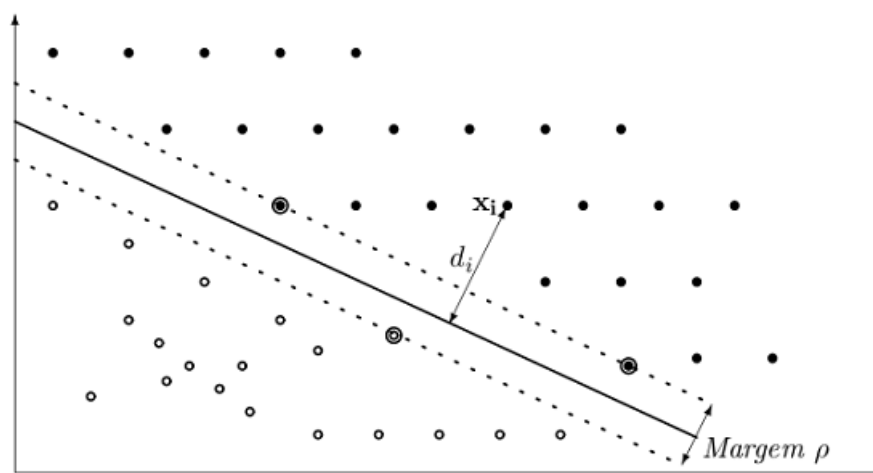


Figura 3.2 – Identificação da margem ρ e dos vetores de suporte sobre a linha pontilhada.
Fonte: LIMA, 2002.

3.4.3 Classificação de padrões linearmente separáveis

Uma classificação linear consiste em determinar uma função $f : X \subseteq R^N \rightarrow R^N$, que atribui um rótulo (+1) se $f(x) \geq 0$ e (-1) caso contrário. Considerando uma função linear, pode-se representá-la pela equação (3.6).

$$f(x) = \langle w \cdot x \rangle + b \quad (3.5)$$

$$f(x) = \sum_{i=1}^n w_i x_i + b \quad (3.6)$$

em que w e $b \in \mathbb{R}^N \times \mathbb{R}^N$, são conhecidos como vetor peso e bias, sendo estes parâmetros responsáveis por controlar a função e a regra de decisão (LIMA, 2002). Os valores de w e b são obtidos pelo processo de aprendizagem a partir dos dados de entrada.

O vetor peso (w) e o bias (b) podem ser interpretados geometricamente sobre um hiperplano. Um hiperplano é um subespaço afim, que divide um espaço em duas partes, correspondendo a dados de duas classes distintas (LIMA, 2002). O vetor peso (w) define uma direção perpendicular ao hiperplano, como mostra a Figura 3.3, e com a variação do bias o hiperplano é movido paralelamente a ele mesmo.

Sendo assim uma SVM linear busca encontrar um hiperplano que separe perfeitamente os dados de cada classe e cuja margem de separação seja máxima, sendo este hiperplano denominado de hiperplano ótimo.

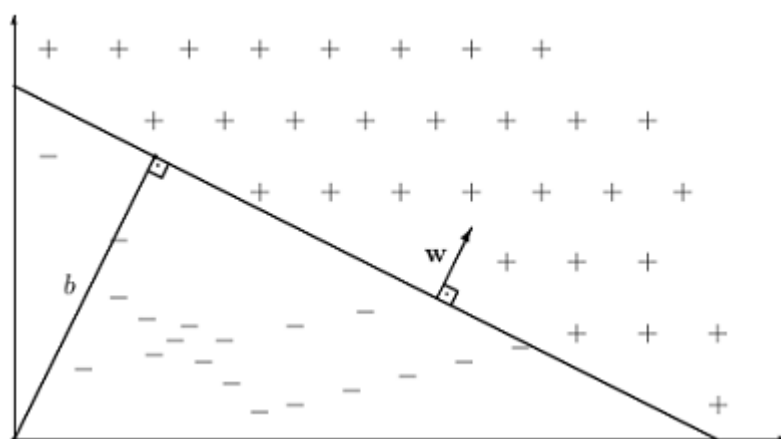


Figura 3.3 – Interpretação geométrica de w e b sobre um hiperplano.
Fonte: LIMA, 2002.

3.4.4 Hiperplano ótimo

Assumindo-se que o conjunto de treinamento é linearmente separável, o hiperplano ótimo é o hiperplano de separação com maior margem. O hiperplano ótimo é definido como:

$$\langle w \cdot x \rangle + b = 0 \quad (3.7)$$

sendo w e b , o vetor peso e o *bias* respectivamente.

Considerando a restrição imposta pela equação (3.8), os classificadores lineares que separam um conjunto de treinamento possuem margem positiva. Ou seja, esta restrição afirma que não há nenhum dado entre $\langle w \cdot x \rangle + b = 0$ e $\langle w \cdot x \rangle + b = \pm 1$, sendo a margem sempre maior que a distância entre os hiperplanos $\langle w \cdot x \rangle + b = 0$ e $|\langle w \cdot x \rangle + b = 1|$. Devido a estas suposições, as SVMs obtidas são normalmente chamadas de SVMs com margens rígidas (ou largas).

$$\begin{aligned} \langle w \cdot x \rangle + b &\geq +1, \text{ para } y_i = +1 \\ \langle w \cdot x \rangle + b &\leq -1, \text{ para } y_i = -1 \end{aligned} \quad (3.8)$$

Estas equações podem ser combinadas em

$$y_i (\langle w \cdot x_i \rangle + b) \geq 1, \quad i = \{1, 2, \dots, n\} \quad (3.9)$$

Seja d_+ (d_-) a distância euclidiana entre os vetores de suporte positivos (negativos) e o hiperplano, define-se como margem ρ de um hiperplano de separação como sendo a maior margem geométrica entre todos os hiperplanos, podemos representar por $\rho = (d_+ + d_-)$.

Denotar-se-á por $d_i(w, b, x_i)$, como a distância de um dado x_i ao hiperplano (w, b) , sendo calculado pela equação a seguir (LIMA, 2002):

$$d_i(w, b, x_i) = \frac{|\langle w \cdot x_i \rangle + b|}{\|w\|} = \frac{y_i (\langle w \cdot x_i \rangle + b)}{\|w\|} \quad (3.10)$$

levando em consideração a restrição imposta pela equação (3.9), pode-se escrever

$$d_i(w, b, x_i) \geq \frac{1}{\|w\|} \quad (3.11)$$

Com isso pode-se identificar $\frac{1}{\|w\|}$ como o limite inferior da distância entre os vetores de suporte x_i e o hiperplano de separação (w, b) . As distâncias d_+ e d_- ficam

$$d_+ = d_- = \frac{1}{\|w\|} \quad (3.12)$$

Como suposto anteriormente que a margem é sempre maior que a última instância, a minimização de $\|w\|$ leva a maximização da margem. A partir disto pode-se definir a margem ρ através da equação (3.13)

$$\rho = (d_+ + d_-) = \frac{2}{\|w\|} \quad (3.13)$$

A Figura 3.4 mostra a distância entre hiperplanos e os vetores de suporte.

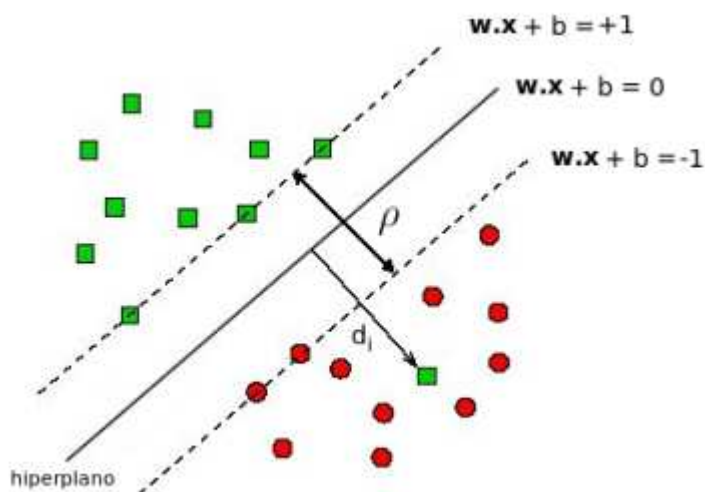


Figura 3.4 – Distância entre hiperplanos e vetores de suporte.
Fonte: LIMA, 2002.

O hiperplano ótimo é dado pela minimização da norma $\|w\|$, considerando a restrição da equação (3.9).

3.4.5 Classificação de padrões não-linearmente separáveis

As *SVMs* apresentadas até agora, trabalham apenas quando os padrões são linearmente separáveis. Em problemas reais esta característica é dificilmente encontrada, sendo a maioria deles complexos e não-lineares. Para estender a *SVM* linear a resolução de

problemas não-lineares foram introduzidas funções reais, que mapeiam o conjunto de treinamento em um espaço linearmente separável, o *espaço de características*.

Um conjunto de dados é dito ser não-linearmente separável, caso não seja possível separar os dados com um hiperplano. A Figura 3.5 mostra um conjunto linearmente e outro não-linearmente separável.

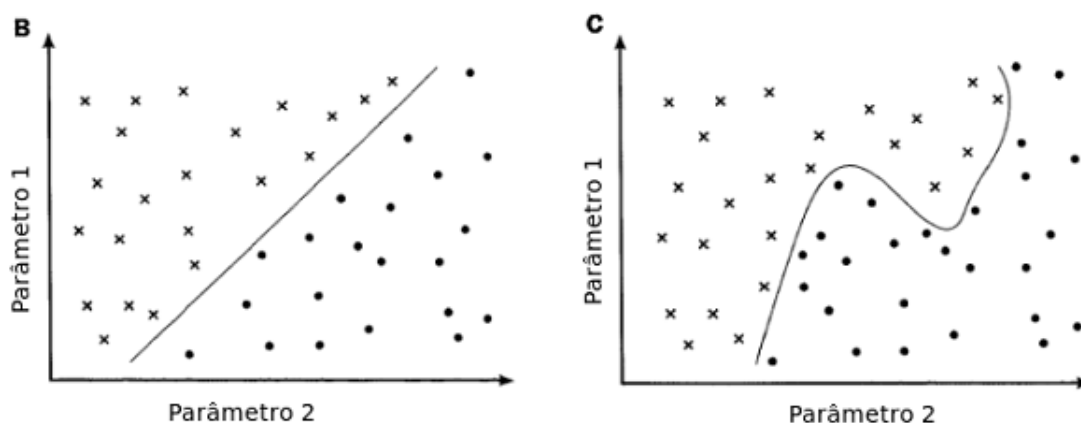


Figura 3.5 – Exemplos de padrões linearmente e não-linearmente separável respectivamente.

Fonte: LORENA; CARVALHO, 2003.

O teorema de Cover afirma que um problema não-linear tem maior probabilidade de ser linearmente separável, em um espaço de mais alta dimensionalidade (SMOLA et al., 1999b). A partir disso, a SVM não-linear realiza uma mudança de dimensionalidade, por meio das funções Kernel, caindo então em um problema de classificação linear, podendo fazer uso do hiperplano ótimo.

3.4.6 Dimensão VC (Vapnik-Chervonenkis)

De acordo com (SMOLA et al., 1999b), dado um conjunto de funções sinal G , sua dimensão VC (Vapnik-Chervonenkis) é definida como o tamanho do maior conjunto de pontos que pode ser particionado arbitrariamente pelas funções contidas em G .

Dessa forma, a teoria estatística da aprendizagem ou teoria VC, estabelece como imperativo restringir a classe de funções de onde $f(x, w)$ é obtido, para aquelas que possuam a capacidade de representar o conjunto de dados de treinamento.

Em outras palavras a dimensão VC do conjunto de funções de classificação G é o número máximo de exemplos de treinamento que pode ser aprendido pela máquina sem erro, para todas as rotulações possíveis das funções de classificação (HAYKIN, 1999).

Matematicamente, a dimensão VC de um conjunto de funções $f(x, w)$ é o máximo número h de vetores x_1, x_2, \dots, x_h que podem ser separados em duas classes, de todas as possíveis formas 2^h , utilizando funções do conjunto, isto é, o máximo número de vetores que pode ser fragmentado pelo conjunto de funções. Se para um dado número l de pontos existir um conjunto l de vetores que podem ser fragmentados pelo conjunto de funções, então a dimensão VC é igual a infinito (VAPNIK, 2000).

Para conceituar melhor fragmentação (separação de dados) e dimVC, considere-se que os dados de entrada estão situados em R^2 , e o conjunto de funções $\{f(x, w)\}$ consiste de linhas. Assim para uma dada linha, os pontos situados em um lado assumem a classe +1, e os situados no outro lado assumem a classe -1.

Conforme mostrado na Figura 3.6, onde a orientação da linha indica o lado em que os pontos assumem a classe +1, podemos observar que é possível obter para três pontos não alinhados, a separação de no máximo três pontos, pelo conjunto de funções, não sendo possível obter o mesmo resultado para quatro pontos em R^2 , conforme mostrado na Figura 3.7. Assim a dimVC de um conjunto de linhas orientadas em R^2 é três.

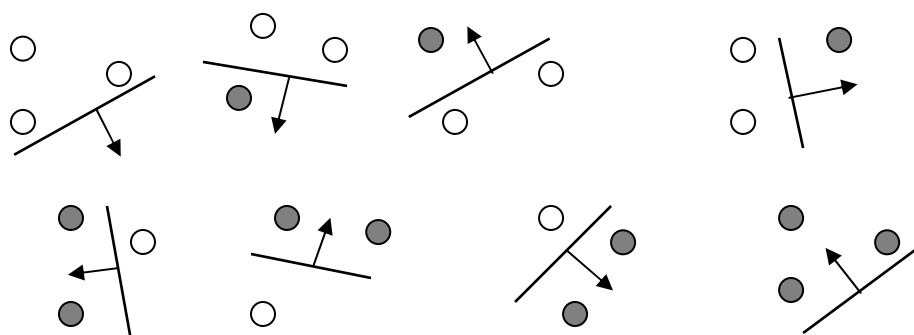


Figura 3.6 – Ilustração de três pontos não alinhados, pertencentes a duas classes diferentes, em um espaço de dimensão R^2 , separados por linhas orientadas.

Fonte: BURGES, 1998; MARKOWETZ, 2003.



Figura 3.7 – Ilustração de quatro pontos em um espaço de dimensão R^2 , onde não é possível efetuar a separação com linhas orientadas de pontos pertencentes a classes diferentes.

Fonte: MARKOWETZ, 2003.

Considerando hiperplanos no espaço R^n , pode-se provar o seguinte teorema (BURGES, 1998):

Considere-se um conjunto de m pontos em R^n . Escolha-se qualquer um dos pontos como origem. Então, os m pontos podem ser separados por planos orientados, se e somente se, a posição vetorial dos demais pontos for linearmente independente.

A dimVC de um conjunto de hiperplanos orientados em R^n é $n+1$, desde que sempre podemos escolher $n+1$ pontos (caso da Figura 3.6), escolhendo um dos pontos como origem, de forma que a posição vetorial dos n pontos remanescentes seja linearmente independente, mas nunca será possível encontrar $n+2$ de tais pontos (caso da Figura 3.7), uma vez que os demais $n+1$ vetores de R^n não são linearmente independentes.

3.4.7 Teoria de Lagrange

Na presença de problemas de otimização que não apresentam restrições, a solução da função objetiva pode ser caracterizada na forma proposta por Fermat, em 1629, e generalizada por Lagrange, em 1797 (PLATT; CRISTIANINI; TAYLOR, 2002).

Teorema 3.1 (Fermat) A condição necessária para que w^* seja um mínimo de $f(w)$, sendo f uma função quadrática convexa, é $\partial f(w^*)/\partial w = 0$, sendo, juntamente com a condição de convexidade de f , uma condição suficiente.

Para problemas com restrições é necessário definir uma função, conhecida por função Lagrangeana, definida pela soma da função objetivo e uma combinação linear da função de restrição, onde os coeficientes α e β são denominados de multiplicadores de Lagrange.

Dado um problema de otimização com função objetiva $f(w)$, e restrição de igualdade $h_i(w) = 0$, $i = 1, 2, \dots, m$, define-se a função Lagrangeana, como:

$$L(w, \beta) = f(w) + \sum_{i=1}^m \beta_i h_i(w) \quad (3.14)$$

Teorema 3.2 (Lagrange) A condição necessária para que um ponto normal w^* seja um mínimo de $f(w)$ sujeito à restrição $h_i(w)$, com $f, h_i \in C^1$, é:

$$\begin{aligned} \frac{\partial L(w^*)}{\partial w} &= 0 \\ \frac{\partial L(w^*, \beta^*)}{\partial \beta} &= 0 \end{aligned} \quad (3.15)$$

A condição descrita pelo Teorema 3.2 é suficiente desde que a função $L(w^*, \beta^*)$ seja função convexa de w . As duas condições geram dois sistemas de equações, que resolvidos juntos apresentam a solução do problema de otimização.

Considerando-se um problema de otimização com restrições de igualdade e desigualdade, pode-se obter uma função Lagrangeana generalizada. Desta forma, dado

$$\begin{aligned} &\text{minimizar} && f(w) && w \in \Omega \\ &\text{sujeito às restrições} && \begin{cases} g_i(w) \leq 0 & i = 1, 2, \dots, k \\ h_i(w) = 0 & i = 1, 2, \dots, m \end{cases} \end{aligned} \quad (3.16)$$

define-se a função Lagrangeana generalizada como:

$$L(w, \alpha, \beta) = f(w) + \sum_{i=1}^m \beta_i h_i(w) = f(w) + \alpha^t g(w) + \beta^t h(w) \quad (3.17)$$

Anteriormente definiu-se o problema de otimização primordial que apresenta uma função objetivo convexa e restrições lineares. Para um problema de otimização restrita como este é possível construir um outro problema chamado de *problema dual*, com a utilização dos multiplicadores de Lagrange para obter a solução ótima.

Dado o problema primordial da definido pela equação (3.4), o problema Lagrangeano dual se apresenta como:

$$\begin{aligned} & \text{maximize} && \theta(\alpha, \beta) = \inf_{w \in \Omega} L(w, \alpha, \beta) \\ & \text{sujeito a} && \alpha \geq 0 \end{aligned} \tag{3.18}$$

A relação fundamental, entre os problemas primordial e dual, é dada pelo teorema da dualidade.

Teorema 3.3 Para uma solução viável w do problema de otimização primordial (equação (3.4)), e, para uma solução viável (α, β) do problema dual descrito em (3.18), existe a relação $f(w) \geq \theta(\alpha, \beta)$.

As condições para uma solução ótima de um problema de otimização geral são apresentadas pelo teorema de Kuhn-Tuchker.

Teorema 3.4 (Kuhn-Tucker) Dado um problema de otimização com domínio convexo $\Omega \subseteq R^N$

$$\begin{aligned} & \text{minimizar} && f(w) && w \in \Omega \\ & \text{sujeito às restrições} && \begin{cases} g_i(w) \leq 0 & i = 1, 2, \dots, k \\ h_i(w) = 0 & i = 1, 2, \dots, m \end{cases} \end{aligned} \tag{3.19}$$

com $f \in C^1$ convexo e, g_i, h_i funções de restrição lineares, as condições necessária suficientes para que um ponto normal w^* , seja um ponto ótimo, é que devem existir valores α^* e β^* , tais que:

$$\begin{aligned} & \frac{\partial L(w^*, \alpha^*, \beta^*)}{\partial w} = 0 \\ & \frac{\partial L(w^*, \alpha^*, \beta^*)}{\partial \beta} = 0 \\ & \alpha_i^* g_i(w^*) = 0, && i = 1, \dots, k, \\ & g_i(w^*) \leq 0, && i = 1, \dots, k, \\ & \alpha_i^* = 0, && i = 1, \dots, k, \end{aligned} \tag{3.20}$$

A terceira relação é conhecida como condição complementar de Karush-Kuhn-Tucker (KKT), a qual implica que para restrições ativas $\alpha_i^* \geq 0$, enquanto que para restrições inativas $\alpha_i^* = 0$.

3.5 FUNÇÃO NÚCLEO

Para muitos problemas complexos do mundo real, requerem-se hipóteses de espaço mais expressivas do que simples representação linear. Muitos problemas não podem ser resolvidos simplesmente através da combinação linear dos atributos (vetor de entrada), mas, a partir da aplicação de características mais abstratas aos dados a serem explorados (PLATT; CRISTIANINI; TAYLOR, 2000).

Uma boa performance na generalização de classificação de padrões pode ser determinada quando a capacidade da função de classificação é maior do que o tamanho do conjunto de treinamento. Classificar com um número elevado de parâmetros a ajustar e também com uma grande capacidade, provavelmente conduzirá a um processo de aprendizagem sem erro, mas, que apresentará um baixo desempenho de generalização. Por outro lado, classificar com capacidade insuficiente, o treinamento não consegue apreender todas as características do problema. No entanto, entre esses dois extremos existe um processo de classificação ótimo capaz de minimizar a generalização de erro esperada para um dado conjunto de dados de treinamento (BOSER; GUYON; VAPNIK, 1992). A representação através de funções núcleo oferece uma alternativa de solução a partir da projeção dos dados em um espaço de características com alta dimensionalidade, incrementando a capacidade da máquina de aprendizagem linear naquele espaço.

Definição: Função núcleo é definida como sendo uma função $k(x, z)$, tal que para todo x e z pertencente a X , tem-se:

$$k(x, z) = \langle \phi(x), \phi(z) \rangle \quad (3.21)$$

em que ϕ é o mapeamento de X para um (produto interno) espaço de características \mathcal{V} (PLATT; CRISTIANINI; TAYLOR, 2000).

A idéia da máquina de vetor de suporte é: mapear os dados de treinamento para um espaço de características com alta dimensionalidade, através da função ϕ , e construir o hiperplano de separação, com a máxima separação, nesse espaço. Este procedimento resulta em uma superfície de decisão não linear no espaço de entrada. Utilizando a função núcleo k (equação (3.21)), é possível computar o hiperplano de separação, sem a necessidade de projetar o mapeamento no espaço de características (SMOLA; BARTLETT; SCHÖLKOPF; SCHUURMANS, 2000).

Este é um ponto importante da função núcleo, pois o produto escalar é implicitamente efetuado no espaço de características, sem a necessidade de mapear ϕ explicitamente. Como consequência direta deste procedimento, tem-se (SCHÖLKOPF; SMOLA; MULLER, 1998): *Todo algoritmo(linear) que somente utiliza produto escalar, pode implicitamente executá-lo no espaço de características pela utilização de funções núcleo, isto é, pode-se construir, de uma forma elegante, uma versão não-linear de um algoritmo linear.*

3.5.1 Espaço de características com alta dimensionalidade.

A complexidade da função alvo a ser apreendida depende da forma como é representada, e a dificuldade apresentada pelo processo de aprendizagem varia de acordo com essa representação. Assim, uma estratégia de pré-processamento nas máquinas de aprendizagem envolve introduzir modificações na representação dos dados, transformando os dados para um espaço de alta dimensionalidade, de uma forma não-linear, satisfazendo o teorema de Cover de separabilidade de padrões, que em termos qualitativos pode ser formulado como (COVER, 1965 apud HAYKIN, 2001):

Um problema complexo de classificação de padrões, disposto não linearmente, em um espaço de alta dimensionalidade tem maior probabilidade de ser linearmente separável do que em um espaço de baixa dimensionalidade.

Mapeando o espaço de vetores de entrada X dentro de um novo espaço $\vartheta = \{\phi(x)/x \in X\}$ onde $\phi: X \rightarrow \vartheta$ transforma o espaço de entrada de dimensão n em um

novo espaço de alta dimensionalidade de dimensão N , conforme mostrado na Figura 3.8, que pode ser escrito:

$$x = (x_1, x_2, \dots, x_n) \Rightarrow \phi(x) = (\phi_1(x), \phi_2(x), \dots, \phi_N(x)) \quad (3.22)$$

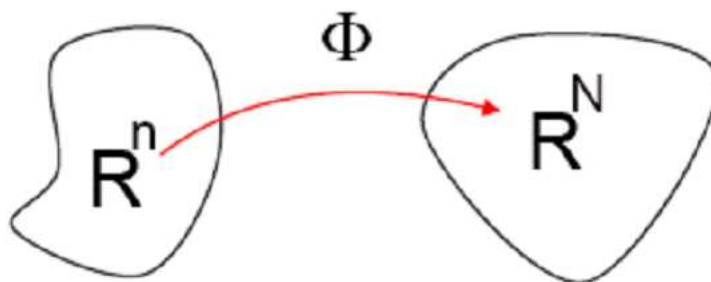


Figura 3.8 – Transformação dos dados de entrada do espaço de dimensão R^n , para um espaço da alta dimensionalidade R^N , através da aplicação de um mapeamento não linear Φ .

Fonte: SMOLA, 1996.

Exemplificando de como através de uma determinada representação em outro espaço, um problema se transforma de não-linear em um problema de solução linear, considere-se a lei de Coulomb, dada por:

$$f(q_1, q_2, r) = K \frac{q_1 q_2}{r^2} \quad (3.23)$$

onde a lei é expressa pelas quantidades observáveis, cargas elétricas q_1, q_2 e distância entre as mesmas r . Efetuando uma transformação de coordenadas, $(q_1, q_2, r) \Rightarrow (x, y, z) = (\ln q_1, \ln q_2, \ln r)$ tem-se a representação nesse novo espaço:

$$g(x, y, z) = \ln f(q_1, q_2, r) = \ln K + \ln q_1 + \ln q_2 - 2 \ln r = k + x + y - 2z \quad (3.24)$$

que pode ser apreendida por uma máquina linear.

A Figura 3.9 mostra uma ilustração de um mapeamento de características de um espaço de entrada de dimensão dois para um espaço de características da mesma dimensão, onde os dados no espaço de entrada não podem ser separados por função linear, no entanto permitem a separação linear no espaço de características.

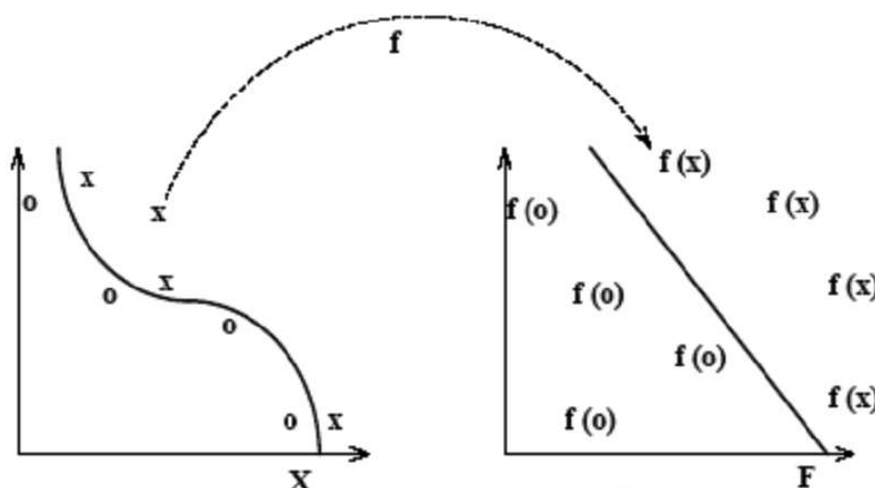


Figura 3.9 – Um mapeamento em um espaço de características pode simplificar o processo de separação de padrões.

Fonte: CRISTIANINI; TAYLOR, 2001.

O espaço de características resultante da transformação Φ apresenta uma alta dimensionalidade, o que pode redundar em dificuldades na otimização do problema considerando o que se denomina de maldição da dimensionalidade, uma vez que para funções complexas do espaço de entrada necessitar-se-á de pontos de amostra densos, para bem representá-las. A complexidade aumenta exponencialmente com a dimensionalidade, levando à deterioração das propriedades de preenchimento do espaço para pontos distribuídos aleatoriamente em espaços de dimensões mais elevadas (HAYKIN, 2001).

Apesar da maldição da dimensionalidade, apreender em um espaço de alta dimensionalidade, segundo a teoria estatística da aprendizagem, pode ser mais simples utilizando baixa complexidade, isto é, classes simples de regras de decisão, por exemplo classificação linear (MULLER; MIKA; RÄTSCH; TSUDA; SCHÖLKOPF, 2001). Desta forma toda a variabilidade necessária para melhor representar os pontos de entrada e obter assim uma classe de funções capaz de representá-los, é introduzido através do mapeamento para o espaço de alta dimensionalidade Φ , isto é, o ponto de interesse não é a dimensionalidade, mas sim, a classe de funções (VAPNIK, 2000). Intuitivamente pode-se ver na Figura 3.10, onde a superfície de decisão do espaço de dimensão dois é extremamente complicada e não-linear, enquanto no espaço de características de monômios de 2º grau a separação é um hiperplano linear.

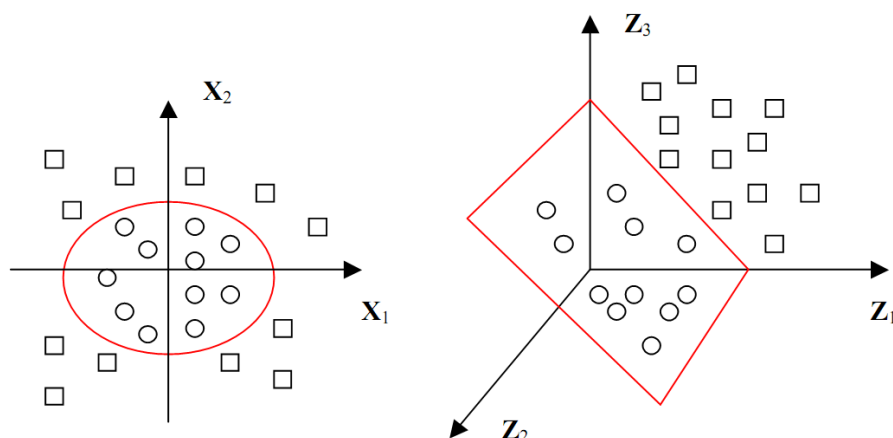


Figura 3.10 – Exemplo de classificação bidimensional. No espaço de entrada (esquerda) a superfície de decisão é elipsoidal e não-linear, enquanto, no espaço de características de monômios de 2º grau(direita), a superfície de decisão é um hiperplano linear.

Fonte: MULLER; MIKA; RÄTSCH; TSUDA; SCHÖLKOPF, 2001.

3.5.2 Modularidade

Uma das características importantes das funções núcleos é a sua modularidade. O mesmo algoritmo pode trabalhar com qualquer núcleo e em qualquer domínio. A matriz núcleo, K , que é totalmente desvinculada do processo de solução do problema, pode ser reutilizada para diferentes algoritmos, uma vez que os procedimentos de solução são adaptados para utilizar somente produtos escalares entre valores de entrada, e, a matriz núcleo é formada com os produtos escalares das imagens de dois valores no espaço de características, possibilitando a implementação do algoritmo em um espaço de alta dimensionalidade. A Figura 3.11 mostra os estágios envolvidos na implementação da análise de padrões através de núcleos. Os dados são processados utilizando núcleos que constroem a matriz núcleo, K , que por sua vez é utilizada pelo algoritmo de análise de padrões para gerar a função do padrão a ser analisado.

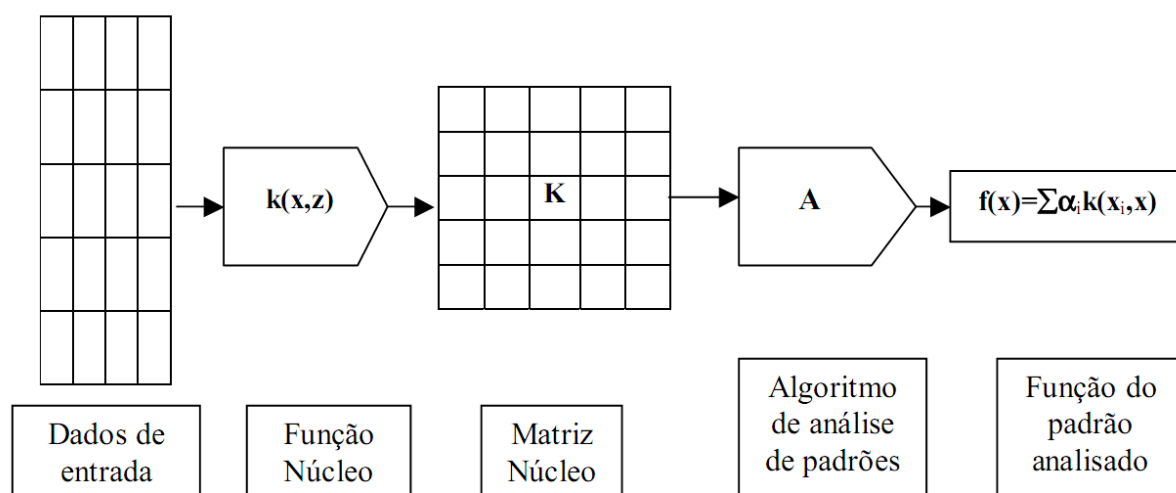


Figura 3.11 – Os estágios envolvidos na aplicação dos métodos com função núcleo.
Fonte: CRISTIANINI ; TAYLOR, 2004.

3.6 CONCLUSÃO SOBRE A MÁQUINA DE VETOR DE SUPORTE

A Máquina de Vetor de Suporte (SVM) destacam-se pela forte fundamentação teórica existente, possuindo como base a teoria da aprendizagem estatística, sendo esta característica um diferencial sobre outras técnicas como redes neurais, que não possui um modelo teórico.

A capacidade em trabalhar com padrões de alta dimensionalidade é outra característica interessante desta técnica, sendo ideal para aplicação em problemas de visão computacional, como reconhecimento de padrões e filtragem de dados.

Mesmo com características atrativas, algumas ressalvas devem ser feitas, como descreve (LORENA; CARVALHO, 2003):

Velocidade de classificação pode se menor do que outras técnicas como Redes Neurais;

Alta complexidade computacional na busca de soluções, agravando ainda mais quando um grande número de dados estão disponíveis para treinamento;

Conhecimento adquirido não é facilmente interpretável.

Diversos estudos foram realizados como intuito de minimizar estas deficiências, o que juntamente com a robustez desta técnica, faz da SVM uma das técnicas mais exploradas atualmente.

3.7 K-MEANS

A técnica clássica de agrupamento chamada de *k-means* permite estabelecer o número de *clusters* desejado. O primeiro passo a ser dado quando se deseja trabalhar com o *k-means* é especificar o parâmetro k , ou seja, quantos *clusters* estão sendo procurados. Então k pontos são escolhidos aleatoriamente como centros dos *clusters*. Todas as ocorrências são atribuídas ao centro do *cluster* mais próximo de acordo com a métrica ordinária da distância euclidiana. Próximo do centróide, ou média, as amostras de cada *cluster* são calculadas - esta é a parte "*means*". Estes centróides são tidos como novos valores de centro para seus respectivos *clusters*. Finalmente, todo o processo é repetido. A iteração continua até que os mesmos pontos sejam atribuídos a cada grupo em rodadas consecutivas, ou até o estágio no qual os centros dos *clusters* se estabilizem e continuem a serem os mesmos (WITTEN, 2005).

Este método de agrupamento é simples e eficaz. É fácil provar que a escolha de um centróide minimiza a distância quadrática total de cada um dos pontos do *cluster* ao seu centro. Uma vez que a iteração se estabilizou, cada ponto é atribuído ao seu centro de *cluster* mais próximo, de modo que o efeito geral é minimizar a distância quadrática total. Mas o mínimo é um mínimo local, não há garantia de que ele seja mínimo global. Os *clusters* finais são bastante sensíveis ao centro do *cluster* inicial. Arranjos completamente diferentes podem surgir a partir de pequenas mudanças na escolha aleatória inicial. Na verdade, isso é um fato para todas as técnicas de ordem prática: é quase sempre impossível encontrar grupos globalmente ótimos. Para aumentar a chance de encontrar um mínimo global, muitas vezes executa-se o algoritmo várias vezes com diferentes escolhas iniciais e toma-se o melhor resultado final - aquele com a menor distância quadrática total (WITTEN, 2005).

3.7.1 Cálculos de distância mais rápidos

O algoritmo de agrupamento *k-means* geralmente requer várias iterações, cada uma envolvendo a descoberta da distância dos centros de agrupamento k de cada amostra para determinar um *cluster*. Há aproximações simples que aceleram isso consideravelmente. Por exemplo, pode-se projetar o conjunto de dados e fazer cortes ao longo de eixos selecionados, em vez de usar as divisões arbitrárias por hiperplano que estão envolvidos com a escolha do

centro mais próximo do *cluster*. Mas isso inevitavelmente compromete a qualidade dos *clusters* resultantes (WITTEN, 2005).

3.7.2 Discussão sobre *k-means*

Muitas variantes do procedimento *k-means* básico tem sido desenvolvidas. Algumas produzem um agrupamento hierárquico, aplicando o algoritmo com $k = 2$ para o conjunto de dados em geral e, em seguida, repetindo, de forma recursiva, dentro de cada *cluster*.

Como se escolhe k ? Muitas vezes nada se sabe sobre o número provável de *clusters*, e todo o ponto de agrupamento deve ser descoberto. Uma maneira é tentar diferentes valores e escolher o melhor.

A técnica *k-means* é descrita formalmente pelo algoritmo a seguir (WITTEN, 2005).

A operação de *k-means* é ilustrada na Figura 3.12, que mostra como, a partir de três centróides, os *clusters* finais são encontrados em quatro passos de atribuição-atualização.

Algoritmo básico de *K-means*:

- 1: Selecionar o número de pontos k como centróides iniciais;
- 2: **Repetir**
 - 3: Formar k *clusters* atribuindo a cada ponto à centróide mais próxima.
 - 4: Recalcular a centróide de cada *cluster*.
- 5: **Até** Centróides não mudam.

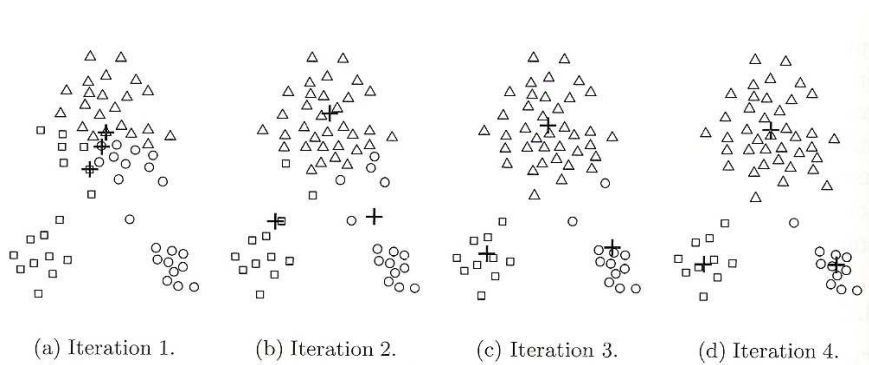


Figura 3.12 – Algoritmo *k-means* utilizando três centróides.
Fonte: TAN; STEINBACH; KUMAR, 2006.

Na primeira etapa, mostrada na Figura 3.12 (a), os pontos são atribuídos a centróides iniciais, que estão todos no grupo maior de pontos. Para este exemplo, usa-se a média como o centróide. Depois, pontos são atribuídos a um centróide, e o centróide é atualizado. Mais uma vez, a figura de cada etapa mostra a evolução e a atribuição de pontos aos centróides. Na segunda etapa, os pontos são atribuídos aos centróides atualizados, e estes são atualizados novamente. Nos passos 2, 3 e 4, que são mostrados nas figuras 3.12 (b), (c) e (d), respectivamente, dois dos centróides movem-se para os dois pequenos grupos de pontos na parte inferior das figuras. Quando o algoritmo *k-means* termina na Figura 3.12 (d), porque nenhuma mudança ocorreu, os centróides identificaram os agrupamentos naturais de pontos.

Para algumas combinações de funções de proximidade e os tipos de centróides, *k-means* sempre converge para uma solução, ou seja, *k-means* atinge um estado em que nenhum ponto muda de um *cluster* para outro. Devido à maioria das convergências ocorrerem nas etapas iniciais, a condição na linha 5 do algoritmo é muitas vezes substituída por uma condição mais branda, por exemplo, repetir até que apenas 1% dos pontos dos *clusters* mudem (TAN; STEINBACH; KUMAR, 2006).

Na Figura 3.13 é apresentado o algoritmo *k-means* no formato de um fluxograma, abrangendo todas as etapas a serem executados pelo computador para realizar o agrupamento.

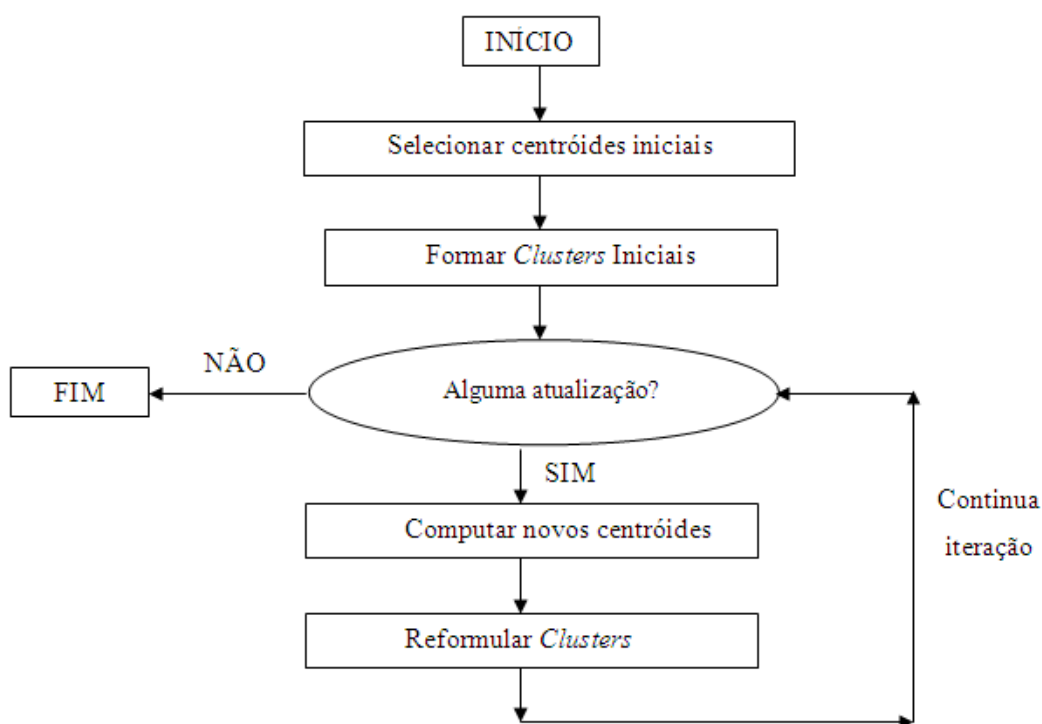


Figura 3.13 – Fluxograma do algoritmo *k-means*.
Fonte: Autoria própria.

3.7.3 Atribuindo pontos ao centróide mais próximo

Para atribuir um ponto ao centróide mais próximo, precisa-se de uma medida de proximidade que quantifica a noção de "mais próximo" para os dados específicos em consideração. A distância euclidiana (d_E) é freqüentemente utilizada para pontos de dados no espaço euclidiano, enquanto a similaridade do cosseno é mais adequada para documentos. No entanto, pode haver vários tipos de medidas de proximidade que são apropriadas para um determinado tipo de dado.

Normalmente, a semelhança de medidas utilizadas para *k-means* é relativamente simples, já que o algoritmo repetidamente calcula a similaridade de cada ponto a cada centróide. Em alguns casos, no entanto, como quando os dados estão em um espaço euclidiano de baixa-dimensão, é possível evitar muitos cálculos de semelhanças, assim, acelerando significativamente o algoritmo. Dividir *k-means* é outra abordagem que acelera o processo, reduzindo o número de semelhanças computadas (TAN; STEINBACH; KUMAR, 2006).

3.7.4 Centróides e funções objetivo

O passo 4 do algoritmo *k-means* é geralmente estabelecido como um "recálculo do centróide de cada *cluster*", uma vez que o centróide pode variar, dependendo da medida de proximidade para os dados e o objetivo do agrupamento. O objetivo do agrupamento é tipicamente expresso por uma função objetivo que depende das proximidades dos pontos de um para o outro ou para os centróides do *cluster*; por exemplo, minimizar a distância quadrática de cada ponto para o seu centróide mais próximo. No entanto, o ponto-chave é este: uma vez que tenha sido especificada uma medida de proximidade e uma função objetivo, o centróide que deve ser escolhido muitas vezes pode ser verificado matematicamente.

Dados no espaço euclidiano: Considerar os dados cuja proximidade medida é a distância euclidiana. Para a função objetivo, que mede a qualidade de um agrupamento, usa-se a Soma dos Erros Quadráticos (SSE na sigla em inglês), que também é conhecido como dispersão. Em outras palavras, calcula-se o erro de cada ponto, ou seja, a sua distância euclidiana para o centróide mais próximo, e depois se calcula a soma total dos erros

quadráticos. Dado dois conjuntos diferentes de *clusters* que são produzidos por duas execuções diferentes de *k-means*, prefere-se aquela com o menor erro quadrático uma vez que este significa que os protótipos (centróides) deste agrupamento são a melhor representação dos pontos em seu *cluster*. Usando a notação na Tabela 3.1, o SSE é formalmente definido como segue:

$$SSE = \sum_{i=1}^K \sum_{x \in C_i} dist(c_i, x)^2 \quad (3.25)$$

Sendo *dist* a norma euclidiana de distância (d_E) entre dois objetos no espaço euclidiano.

Tabela 3.1– Descrição dos símbolos usados na SSE

Símbolo	Descrição
x	Um objeto.
C_i	<i>Cluster</i> iésimo.
c_i	Centróide do <i>cluster</i> C_i .
C	Centróide de todos os pontos.
m_i	Número do objeto no iésimo <i>cluster</i> .
m	Número de objetos no conjunto de dados.
K	Número de <i>clusters</i> .

Fonte: TAN; STEINBACH; KUMAR, 2006.

Dadas essas premissas, pode-se demonstrar que o centróide que minimiza o SSE do *cluster* é a média. Usando a notação na tabela 3.1, o centróide (média) do *cluster* iésimo é definido pela equação 3.26.

$$c_i = \frac{1}{m_i} \sum_{x \in C_i} x \quad (3.26)$$

Os passos 3 e 4 do algoritmo *k-means* diretamente tentam minimizar o SSE (ou, mais geralmente, a função objetivo). O passo 3 forma *clusters* por atribuição de pontos aos centróides mais próximos, o que minimiza o SSE para o conjunto de centróides. O passo 4 recalcula os centróides, de modo a minimizar ainda mais a SSE. No entanto, as ações do *k-*

means nos passos 3 e 4 são apenas garantidas para encontrar um mínimo local com respeito à SSE, uma vez que são baseados na otimização do SSE para escolhas específicas dos centróides e *clusters*, ao invés de todas as escolhas possíveis (TAN; STEINBACH; KUMAR, 2006).

3.7.5 Escolhendo os centróides iniciais

Quando a inicialização aleatória de centróides é usada, diferentes execuções de *k-means* normalmente produzem diferentes SSEs.

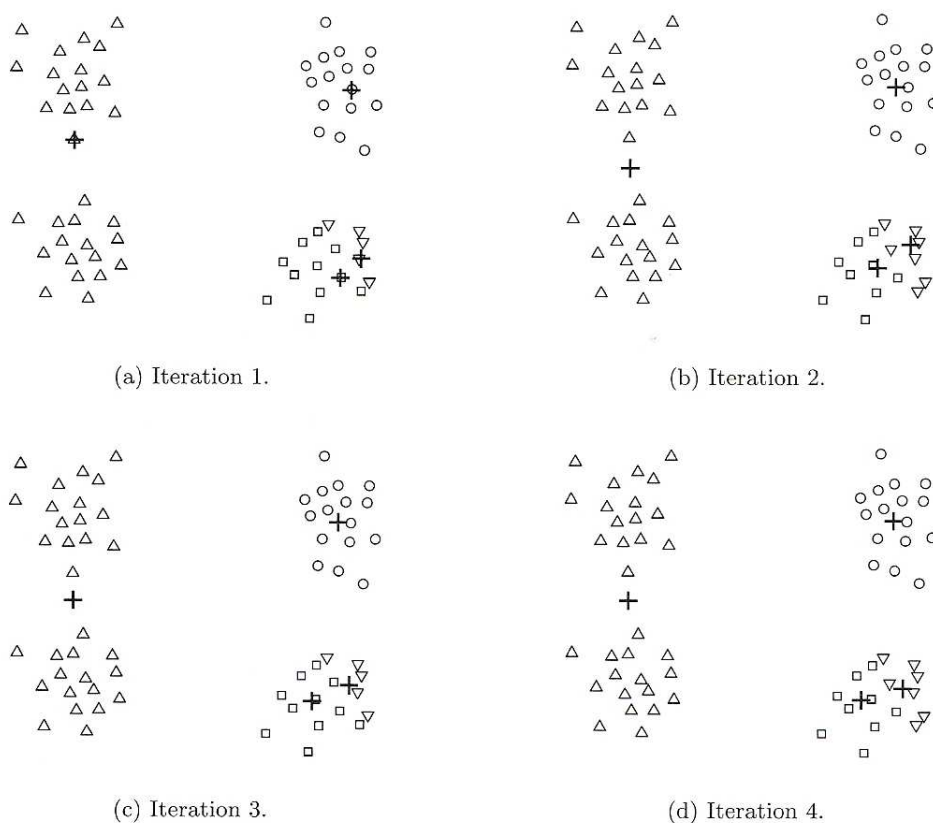
O passo fundamental do processo de *k-means* é escolher centróides adequados. Uma abordagem comum é escolher centróides iniciais de forma aleatória, mas o resultado, muitas vezes, são *clusters* pobres.

Uma técnica que é comumente usada para tratar o problema de escolher centróides é a realização de várias execuções, cada uma com um conjunto diferente escolhido aleatoriamente de centróides iniciais, e depois selecionar o conjunto de *clusters* com o SSE mínimo. Embora simples esta estratégia pode não funcionar muito bem, dependendo do conjunto de dados e o número de *clusters* procurado. (TAN; STEINBACH; KUMAR, 2006)

Devido ao problema com o uso de centróides iniciais selecionados aleatoriamente, que mesmo depois de repetidas execuções podem não ser superadas, outras técnicas são frequentemente utilizadas para a inicialização. Uma abordagem eficaz é tomar uma amostra de pontos e agrupá-los usando uma técnica de agrupamento hierárquico. *K clusters* são extraídos do agrupamento hierárquico, e os centróides dos *clusters* são utilizados como centróides iniciais. Esta abordagem, muitas vezes funciona bem, mas é prática somente se (1) a amostra é relativamente pequena, por exemplo, de algumas centenas a alguns milhares, e (2) *k* é relativamente pequeno comparado ao tamanho da amostra.

O procedimento a seguir é outra abordagem para seleção de centróides iniciais. Seleciona-se o primeiro ponto ao acaso ou toma-se o centróide de todos os pontos. Então, para cada centróide sucessivo inicial, seleciona-se o ponto que está mais distante de qualquer um dos centróides iniciais já selecionados. Desta forma, obtêm-se um conjunto de centróides iniciais com o qual garante-se não só serem selecionados aleatoriamente, mas também bem separados. Infelizmente, tal abordagem pode selecionar pontos remotos, ao invés de pontos em regiões densas (*clusters*). Além disso, é dispendioso calcular o ponto mais distante do

atual conjunto de centróides iniciais. Para transpassar estes problemas, esta abordagem é muitas vezes aplicada a uma amostra dos pontos. Sabendo-se que pontos remotos são raros, eles tendem a não aparecerem em uma amostra aleatória. Em contraste, os pontos de todas as regiões densas são suscetíveis de serem incluídos, a menos que o tamanho da amostra seja muito pequena. Além disso, o cálculo envolvido para encontrar os centróides iniciais é muito reduzido, porque o tamanho da amostra é geralmente muito menor do que o número de pontos. Na Figura 3.14 é apresentando um exemplo de convergência indesejável (TAN; STEINBACH; KUMAR, 2006).



**Figura 3.14 – Algoritmo *k-means* não convergindo de forma eficaz.
Fonte: TAN; STEINBACH; KUMAR, 2006.**

3.7.6 Tempo e complexidade do espaço

O espaço necessário para *k-means* é modesto porque só os pontos de dados e centróides são armazenados. Especificamente, o armazenamento necessário é $O((m + K)n)$, sendo m o número de pontos e n o número de atributos. O tempo necessário para *k-means* também são modestos - basicamente linear. Em particular, o tempo necessário é $O(I * K * m * n)$, sendo I o número de iterações necessárias para a convergência. Como mencionado, I é muitas vezes pequeno e geralmente pode ser delimitado de forma segura, já que a maioria das mudanças ocorre normalmente nas iterações iniciais. Portanto, *k-means* é linear em m , o número de pontos, e é eficiente, bem como *simplex*, desde que k , o número de *clusters*, é significativamente menor do que m (TAN; STEINBACH; KUMAR, 2006).

3.7.7 *K-means*: Aspectos adicionais

➤ Manipulação de *Clusters* Vazios

Um dos problemas com o algoritmo básico de *k-means* dado anteriormente é que os *clusters* vazios podem ser obtidos se nenhum ponto for a eles atribuído durante a etapa de atribuição. Se isso acontecer, então é necessária uma estratégia para escolher um centróide de substituição, pois caso contrário, o erro quadrático será maior do que o necessário. Uma abordagem é escolher o ponto que está mais distante de qualquer centróide atual. Isso elimina o ponto que atualmente mais contribui para o erro total quadrático. Outra abordagem é escolher o centróide substituto do *cluster* que tem o maior SSE. Isto tipicamente dividirá o *cluster* e reduzirá o SSE geral do processo. Se houver vários *clusters* vazios, então esse processo pode ser repetido várias vezes (TAN; STEINBACH; KUMAR, 2006).

➤ Casos Anômalos

Quando o critério de erro quadrático é utilizado, valores atípicos podem influenciar indevidamente os *clusters* que são encontrados. Em particular, quando estes valores estão presentes, os centróides aglomerados resultantes (protótipos) podem não ser tão representativos quanto deveriam e assim, o SSE será maior também. Por causa disto, muitas vezes é útil descobrir valores discrepantes e eliminá-los de antemão. É importante, no entanto, perceber que existem casos nos quais tais valores não devem ser eliminados. Quando o agrupamento é utilizado para compressão de dados, cada ponto deve ser agrupado e, em alguns casos, tais como na análise financeira, aparentes discrepâncias, por exemplo, os clientes pouco rentáveis, podem ser os pontos mais relevantes.

Uma questão óbvia é como identificar pontos anômalos. Se se utiliza de abordagens que os removem antes do agrupamento, evita-se pontos que não serão agrupados como esperado. Alternativamente, pontos discrepantes também podem ser identificados em uma etapa de pós-processamento. Por exemplo, pode-se acompanhar a contribuição ao SSE de cada ponto, e eliminar os pontos com contribuições elevadas, especialmente ao longo de vários ensaios. Além disso, pode-se querer eliminar pequenos grupos, uma vez que freqüentemente eles representam grupos de pontos discrepantes (TAN; STEINBACH; KUMAR, 2006).

➤ Reduzindo o SSE com Pós-processamento

Uma maneira óbvia para reduzir o SSE é encontrar mais *clusters*, ou seja, usar um maior número de k . No entanto, em muitos casos, gostar-se-ia de melhorar o SSE, mas não aumentar o número de *clusters*. Isto é possível porque muitas vezes *k-means* normalmente converge para um mínimo local. Várias técnicas são usadas para "arrumar" os *clusters* resultantes, a fim de produzir um agrupamento que tem SSE mais baixos. A estratégia é concentrar-se em *clusters* individuais já que o SSE total é simplesmente a soma da SSE de cada *cluster*. (Usa-se a terminologia SSE total e SSE do *clusters*, respectivamente, para evitar qualquer confusão). Pode-se mudar o SSE total executando-se várias operações nos *clusters*, tais como separação (divisão de *clusters*), na qual grupos serão separados, ou fusão de *clusters* (fusão de fases), quando são combinados. Desta forma, muitas vezes é possível escapar de SSE mínimos locais e ainda produzir uma solução de agrupamento com o número de *clusters* desejado. A seguir algumas técnicas utilizadas na separação e fusão de fase.

Duas estratégias que diminuem o SSE total aumentando o número de *clusters* são:

- ✓ Dividir um *cluster*: o *cluster* com o maior SSE é normalmente escolhido, mas também poder-se-ia dividir aquele com maior desvio padrão para um atributo particular.
- ✓ Introduzir um novo centróide: muitas vezes o ponto que está mais distante de qualquer centro de *cluster* é escolhido. Pode-se facilmente determinar isso quando se acompanha a contribuição de cada ponto ao SSE. Outra abordagem é escolher aleatoriamente a partir de todos os pontos ou a partir dos pontos com maior SSE.

Duas estratégias que diminuem o número de *clusters*, ao tentar minimizar o aumento da SSE total, são os seguintes:

- ✓ Dispersão: realizado através da remoção do centróide que corresponde a um *cluster* e reatribuir os pontos a outros. Idealmente, o *cluster* que é disperso deve ser aquele que menos acrescenta ao SSE total.
- ✓ Fusão: os *clusters* com os centróides mais próximos são escolhidos, embora, a melhor abordagem é a junção dos dois grupos que resultem no menor aumento do SSE total.

Estas duas estratégias de fusão são as mesmas que são utilizadas nas técnicas de agrupamento hierárquico conhecido como o método centróide e método de Ward, respectivamente (TAN; STEINBACH; KUMAR, 2006).

➤ Atualizando Centróides de Forma Incremental

Em vez de atualizar os centróides dos *clusters* após todos os pontos serem atribuídos, os centróides podem ser atualizados de forma incremental, após cada atribuição de um ponto a um *cluster*. Observa-se que isso requer zero ou duas atualizações para centróides de *clusters* em cada etapa, já que um ponto ou se move para um novo (duas atualizações) ou permanece no atual (zero atualizações). Usando uma estratégia de atualização incremental garante-se que os *clusters* vazios não serão produzidos, já que todos eles começam com pelo menos um

ponto, e se algum sempre tem apenas um ponto, este será sempre transferido para o mesmo *cluster*.

Além disso, se a atualização incremental é usada, o peso relativo do ponto que está sendo adicionado pode ser ajustado, por exemplo, o peso dos pontos é frequentemente diminuída com o processo de agrupamento. Embora isso possa resultar em uma melhor precisão e convergência mais rápida, pode ser difícil fazer uma boa escolha para o peso relativo, especialmente em uma ampla variedade de situações. Estes problemas de atualização são semelhantes aos envolvidos na atualização de pesos para redes neurais artificiais.

Mais um benefício das atualizações incrementais tem a ver com o uso de outros objetivos que "minimizam o SSE". Suponha-se que é dada uma função objetivo arbitrária para medir a eficácia de um conjunto de *clusters*. Quando se processa um ponto individual, pode-se calcular o valor da função objetivo para cada atribuição de *cluster* possível e então escolher o que otimiza o objetivo.

Em relação aos agrupamentos vê-se que a aderência das curvas em cada *cluster* é satisfatória, cabendo ao programador, ao utilizar *k-means*, identificar para qual valor de *k* obtêm-se o agrupamento mais coerente para a base de dados que ele está utilizando.

Por outro lado, a atualização incremental de centróides introduz uma dependência de ordem. Em outras palavras, os aglomerados produzidos podem depender da ordem na qual os pontos são processados. Embora isso possa ser tratado de forma randômica a abordagem de *k-means* básica, atualizar os centróides depois de todos os pontos serem atribuídos a grupos, não gera nenhuma dependência de ordem. Além disso, atualizações incrementais são ligeiramente mais demoradas. No entanto, *k-means* converge muito rapidamente e, portanto, o número de pontos de comutação dos grupos rapidamente se torna relativamente pequeno (TAN; STEINBACH; KUMAR, 2006).

3.8 CONCLUSÃO DE *K-MEANS*

K-means destaca-se pela simplicidade tanto na programação quanto na execução de comandos. É prática, não requerendo muita memória ou tempo de processamento e sua formulação matemática é elegante e concisa.

Outra de suas vantagens é a escolha, a priori, de quantos *clusters* se deseja trabalhar, o que possibilita a visualização de como os agrupamentos se desenvolvem com o aumento de k .

Para algumas aplicações, porém, isso causa perdas de tempo até a determinação de quantos *clusters* são necessários para a representação de um agrupamento ideal.

Seu bom entendimento é um passo fundamental para a iniciação em técnicas de agrupamentos mais sofisticadas.

4. RESULTADOS DAS SIMULAÇÕES

4.1 INTRODUÇÃO

A pesquisa científica pressupõe o estabelecimento de um método científico que corresponde a quatro aspectos, a observação que pode ser visual ou com a utilização de instrumentos, a descrição para que o experimento possa ser replicável em qualquer tempo, a previsão para que as hipóteses sejam válidas em observações efetuadas em qualquer época e a explicação das causas que devem se correlacionar com as observações e preceder os efeitos observados.

Este trabalho baseia-se em uma pesquisa científica, desta forma alguns passos foram seguidos até a finalização do mesmo:

- Obtenção de curvas de carga representativas do universo de consumidores com contrato de fornecimento;
- Agrupamento das curvas de carga segundo critérios estabelecidos;
- Utilização de técnicas e algoritmos para efetuar o agrupamento das curvas de carga;
- Simulações computacionais;
- Análises e comparações de resultados das simulações;
- Interpretação dos resultados observando os agrupamentos definidos.

As ferramentas computacionais utilizadas foram obtidas através de programação no MATLAB® com utilização de algumas rotinas já desenvolvidas. Também foram pesquisados programas livres (freeware) dedicados tais como o SVM^{Light} de Thorsten Joachims, LibSVM de Chang e Ling e outros disponíveis livremente na Internet.

Este capítulo apresenta os resultados das simulações computacionais utilizando a Máquina de Vetor de Suporte e o algoritmo *k-means* com o objetivo de separar as curvas de carga em grupos distintos e apresentar uma nova abordagem no estudo do sistema tarifário.

4.2 CURVAS DE CARGA

As curvas de carga utilizadas foram obtidas junto à concessionária de energia local a qual repassou valores tabelados de curvas diárias de diversas empresas que utilizam medidores com memória de massa. As mesmas não trouxeram informações pessoais ou jurídicas, sendo totalmente anônimas.

Um total de 47 arquivos foi obtido contendo cada qual, no mínimo, a medição de duas semanas consecutivas entre os meses de outubro a novembro de 2010. Cada curva é discretizada em 96 (noventa e seis) pontos, que representam os 96 intervalos de 15 minutos que há em 24 horas, os quais, por sua vez, representam a integralização da demanda requerida num intervalo de 15 (quinze) minutos daquele consumidor. Portanto, os arquivos podem ser tratados como matrizes com um número variável de linhas, sendo que cada linha representa um dia do mês, e cada uma das 96 colunas um período de 15 minutos, totalizando as 24 horas diárias.

Cada arquivo foi normalizado para que as curvas de carga apenas representassem o perfil de demanda, e não sua grandeza. Para isso buscou-se em cada linha que representa uma curva de determinado dia, o valor que indica a maior demanda do respectivo dia e posteriormente dividiram-se todos os valores dessa mesma linha pelo valor encontrado, assim a variação máxima dos valores está restrita a 0 e 1. Tal procedimento foi repetido em todas as linhas de todos os arquivos obtidos. A normalização permite a comparação de perfis de curva de carga de empresas de portes diferentes. Se duas empresas, que impõem perfis de consumo parecidos, porém em níveis de demanda diferentes, fossem analisadas sem a normalização, certamente não seriam passíveis de comparação já que haveria uma distância, no eixo y, entre elas. Como o foco deste trabalho é agrupar perfis parecidos e não demandas, a normalização permitiu que isto se realizasse sem maiores contratempos.

4.3 SIMULAÇÕES COMPUTACIONAIS

4.3.1 Algoritmos e ferramentas utilizadas

Durante o estudo selecionou-se alguns algoritmos capazes de realizar várias funções envolvendo a máquina de vetor de suporte. Entretanto, apenas um grupo seletivo permitia a obtenção dos agrupamentos, haja vista, que a maioria buscava a classificação de dados para posterior regressão de dados. A seguir são citados os algoritmos que foram selecionados para os testes.

Bioinformatic Toolbox™

O software MATLAB® conta com inúmeras toolboxes que na realidade são bibliotecas externas, capazes de realizar um rol de aplicações específicas, seja para as soluções de problemas de engenharia elétrica, mecânica, aeroespacial, genética ou bioinformática.

A Bioinformatic Toolbox™ fornece algoritmos e técnicas de visualização para o seqüenciamento, análise e espectrometria de massa, e ontologia de genes, dentre outras finalidades.

A Bioinformatic Toolbox™ estende o ambiente MATLAB® para fornecer um ambiente de software integrado para inúmeras análises na qual cientistas e engenheiros podem resolver problemas, construir aplicações, dentre elas análise do genoma (toda a informação hereditária de um organismo que está codificada em seu DNA) e análise do proteoma (conjunto de proteínas que pode ser encontrado numa célula específica quando esta está sujeita a um certo estímulo) (MATLAB, 2011).

A maioria das funções são implementadas na linguagem de programação MATLAB®. Este ambiente aberto permite explorar e personalizar a toolbox dos algoritmos existentes ou desenvolver o seu próprio.

Pode-se usar as funções básicas de Bioinformatic Toolbox™ para criar algoritmos mais complexos e aplicações diversas.

SVM^{Light}

O SVM^{Light}, desenvolvido por Joachims, é um dos algoritmos de classificação por SVM. Ele tem um algoritmo de otimização rápida, pode ser aplicado a conjuntos de dados muito grande e tem uma implementação muito eficiente de cross-validation. Distribuído em linguagem C++ para Linux, Windows, Cygwin, e Solaris. As funções núcleo disponíveis são: polinomial, função de base radial, e neural (JOACHIMS, 2011).

LibSVM

O LIBSVM (Biblioteca de Support Vector Machines) foi desenvolvido por Chang e Lin e contém C-classification, Classification ν , ϵ -regression, e ν -regression. Desenvolvido em C++ e Java, suporta também multi-classe, classificação SVM ponderada para dados desbalanceados, cross-validation e de seleção do modelo automático. Possui interfaces para Python, R.Splus, MATLAB®, Perl, Ruby, e LabVIEW. As funções núcleo disponíveis são: linear, polinomial, função de base radial, e neural (CHANG; LIN, 2011).

4.3.2 Conclusões sobre os *softwares* e algoritmos

Durante a etapa de testes percebeu-se que era inviável realizar o agrupamento das curvas de carga através do Bioinformatic Toolbox™, devido ao fato de o mesmo realizar apenas a classificação de forma supervisionada, ou seja, para o treinamento é necessário a introdução prévia do grupo a qual pertencem os dados. Desta forma, trata-se de um classificador de dados.

Visando esclarecer a diferença entre classificação e agrupamento é apresentada a Tabela 4.1.

Tabela 4.1– Diferença entre classificação e agrupamento

Classificação	Agrupamento
<p>A tarefa de atribuir amostras para classes pré-definidas.</p> <p>Exemplo: Decidir se um registro de paciente em particular pode ser associado a uma doença específica.</p>	<p>A tarefa de agrupamento de dados relacionados com os pontos distintos, sem rotulá-los.</p> <p>Exemplo: Agrupamento de registros de pacientes com sintomas semelhantes sem saber o que os sintomas indicam.</p>

Fonte: Autoria própria.

O algoritmo SVM^{Light}, diferentemente da Bioinformatic Toolbox™, oferece uma imensa gama de aplicações, entretanto, é esperado que o usuário possua uma boa base teórica em programação para operá-lo de forma correta. A interface foi considerada pouco amigável, todavia, é um aspecto que não é levado muito em consideração para o grupo dos especializados em computação comparado com o público leigo no assunto. Nota-se que o SVM^{Light} permite o ajuste de seu próprio código-fonte, sendo possível otimizar os resultados.

O algoritmo LibSVM mostrou-se o mais simples no quesito programação pois possui interface com MATLAB®, e a linha de programação é muito intuitiva.

Tentou-se o algoritmo LibSVM, por ser o mais simples de lidar-se. No entanto, os resultados obtidos mostraram claramente que o agrupamento não estava adequado. Buscou-se de diversas formas obter-se os agrupamentos, porém houve apenas insucessos.

Testes feitos com o MATLAB® mostraram que o método SVC (Support Vector Clustering) não agrupava os perfis de curva de demanda, apenas, após um treinamento prévio com qualquer curva escolhida aleatoriamente (Figura 4.1), encontrava as curvas semelhantes àquela primeira, como mostra a Figura 4.2. Esse resultado não está de acordo como os objetivos deste trabalho, pois espera-se que ao final do mesmo seja possível estabelecer grupos (*n clusters*) com as curvas de carga, independentemente de curvas previamente escolhidas.

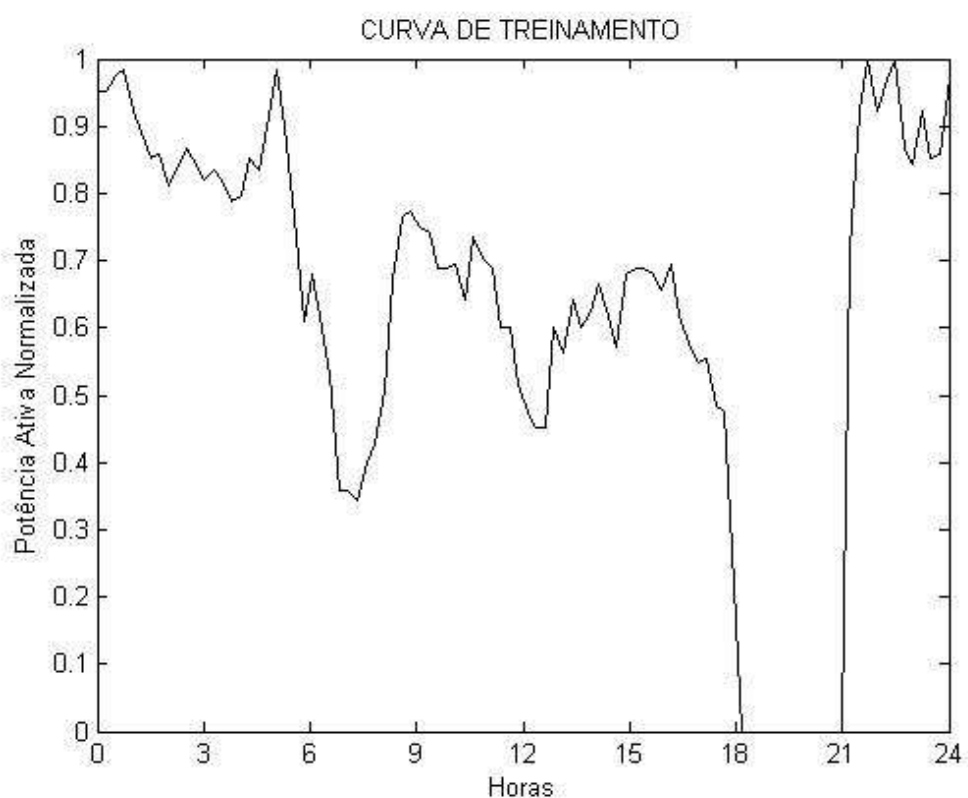


Figura 4.1 – Curva de treinamento do ARQ3 para a Máquina de Vetor de Suporte.
Fonte: Autoria própria.

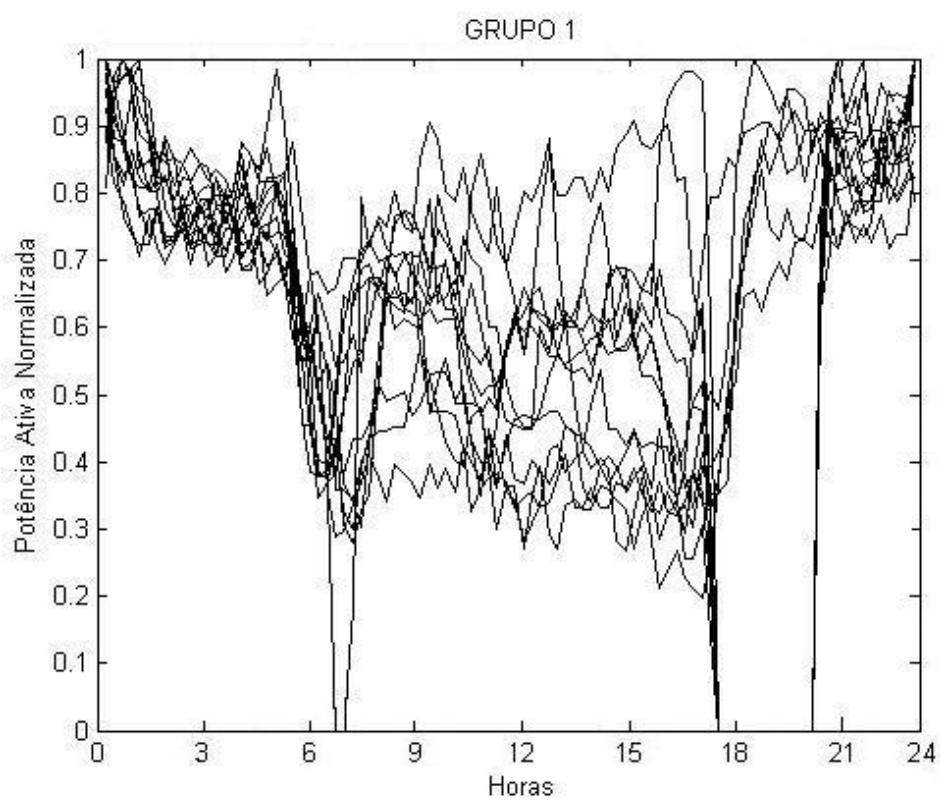


Figura 4.2 – Simulação no arquivo ARQ3, utilizando Máquina de Vetor de Suporte.
Fonte: Autoria própria.

Passou-se então às simulações utilizando *k-means*. Inicialmente, buscou-se analisar e testar se realmente tal método agruparia os dados aleatórios que fossem semelhantes. Para isso, decidiu-se realizar simulações de agrupamento com dados de uma mesma empresa, o que garantiria que várias curvas seriam agrupadas. Uma análise preliminar selecionou duas empresas-teste: os arquivos denominados ARQ3 e ARQ5.

4.3.3 Dados do arquivo ARQ3

O arquivo ARQ3 possui dados que compreendem as curvas de carga do dia 12/10/2010 até o dia 09/11/2010, ou seja, um total de 29 curvas diárias. Cabe ressaltar que neste período ocorrem os seguintes feriados nacionais: 12/10/2010 (Dia de Nossa Senhora Aparecida) e 02/11/2010 (Dia de Finados).

A Tabela 4.2 mostra os agrupamentos obtidos utilizando o algoritmo *k-means* (cujo código está nos Anexos A e B) . O arquivo ARQ3 foi testado utilizando $k = 1$ (apenas 1 *cluster*), $k = 2$ (2 *clusters*) e $k = 3$ (3 *clusters*).

Tabela 4.2 – Arquivo ARQ3

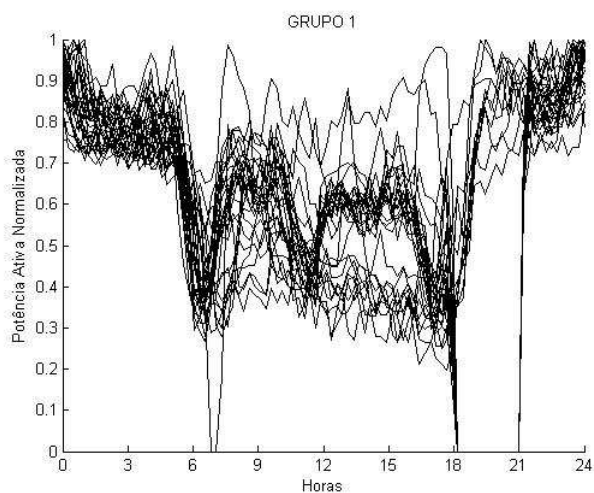
ARQ3				
Data:	Dia da semana	$k=1$	$k=2$	$k=3$
		Grupo	Grupo	Grupo
12/10/2010	Terça-feira	1	2	3
13/10/2010	Quarta-feira	1	2	2
14/10/2010	Quinta-feira	1	2	2
15/10/2010	Sexta-feira	1	2	2
16/10/2010	Sábado	1	1	1
17/10/2010	Domingo	1	1	1
18/10/2010	Segunda-feira	1	2	2
19/10/2010	Terça-feira	1	2	2
20/10/2010	Quarta-feira	1	2	2
21/10/2010	Quinta-feira	1	2	3
22/10/2010	Sexta-feira	1	2	2
23/10/2010	Sábado	1	1	1
24/10/2010	Domingo	1	1	1
25/10/2010	Segunda-feira	1	2	2
26/10/2010	Terça-feira	1	2	2

Tabela 4.2 – Arquivo ARQ3 - continuação

ARQ3				
Data:	Dia da semana	$k=1$	$k=2$	$k=3$
		Grupo	Grupo	Grupo
27/10/2010	Quarta-feira	1	1	1
28/10/2010	Quinta-feira	1	1	1
29/10/2010	Sexta-feira	1	2	2
30/10/2010	Sábado	1	1	1
31/10/2010	Domingo	1	1	1
01/11/2010	Segunda-feira	1	2	3
02/11/2010	Terça-feira	1	2	3
03/11/2010	Quarta-feira	1	2	2
04/11/2010	Quinta-feira	1	2	2
05/11/2010	Sexta-feira	1	2	2
06/11/2010	Sábado	1	1	1
07/11/2010	Domingo	1	1	1
08/11/2010	Segunda-feira	1	2	2
09/11/2010	Terça-feira	1	2	2

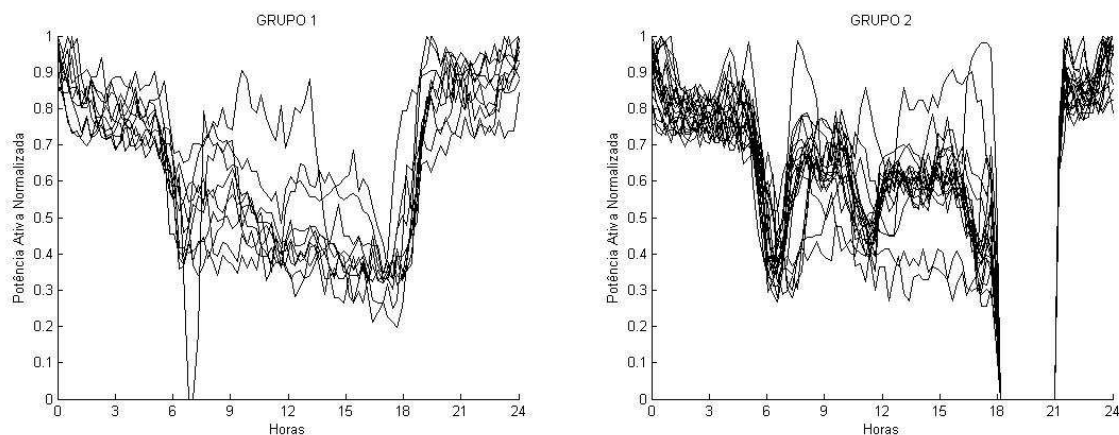
Fonte: Autoria própria

Considerando-se $k=1$ (Figura 4.3), obviamente, obteve-se como resposta todas as curvas do arquivo ARQ3. Já é possível notar que várias curvas se sobrepõem, o que já era esperado por tratar-se de uma mesma empresa e que apresenta uma demanda característica.



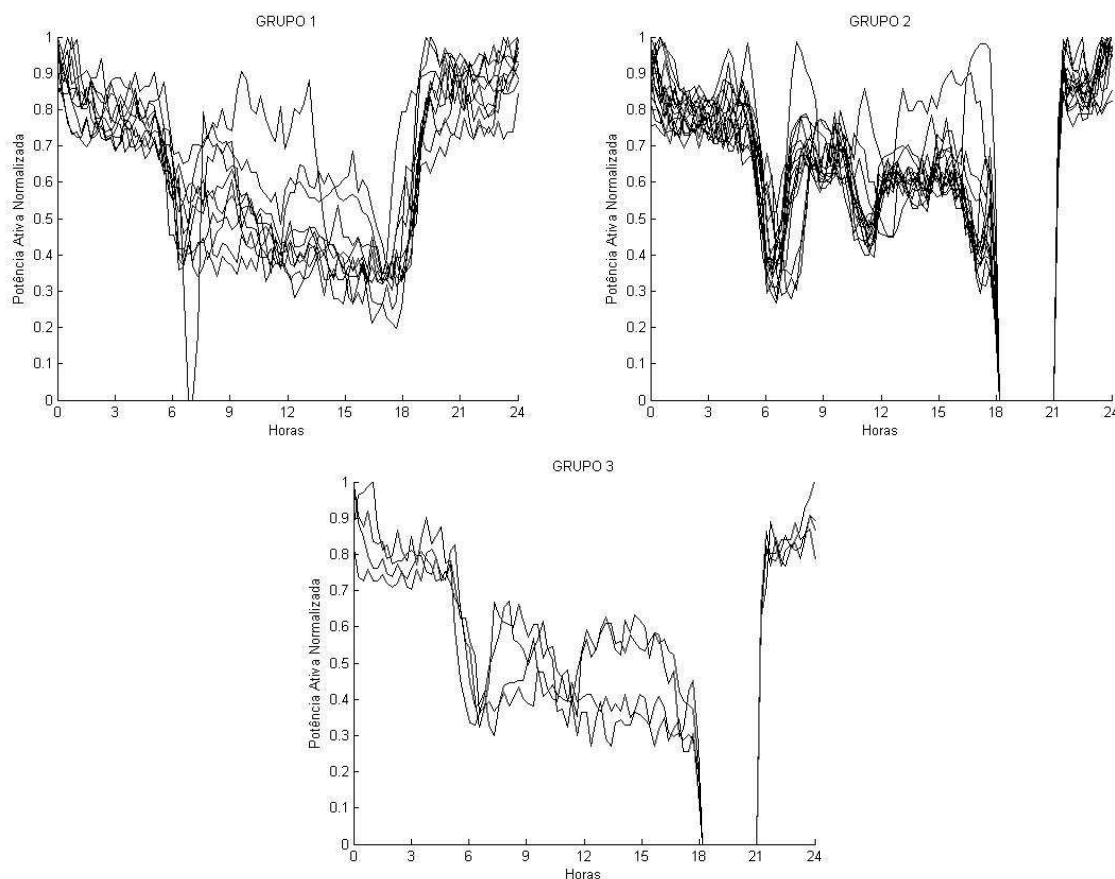
**Figura 4.3 – Simulação no arquivo ARQ3, utilizando $k=1$.
Fonte: Autoria própria.**

Para $k = 2$ (Figura 4.4), como resultado obtêm-se dois agrupamentos distintos. Nota-se que este é o número ideal de *clusters* para este arquivo, visto que para qualquer valor de $k \geq 3$ (Figura 4.5), o agrupamento cria *clusters* redundantes, ou seja, forçadamente o algoritmo separa curvas que deveriam estar juntas.



**Figura 4.4 – Simulação no arquivo ARQ3, utilizando $k=2$.
Fonte: Autoria própria.**

A Figura 4.5, apresenta o arquivo ARQ3 na ótica de três agrupamentos distintos. Visualmente é perceptível que não foi uma boa escolha utilizar $k = 3$.



**Figura 4.5 – Simulação no arquivo ARQ3, utilizando $k=3$.
Fonte: Autoria própria.**

Para se verificar a aderência das curvas de carga foi utilizado o erro MAE (Mean Error Absolute) (Anexo C). O MAE é uma das várias maneiras usadas em estatística para medir o quão perto as previsões estão dos resultados finais.

$$MAE\% = \frac{1}{N} \sum_{i=1}^N \left| 1 - \frac{y_{Pi}}{y_{Ri}} \right| \cdot 100 \quad (3.27)$$

Sendo N o número de elementos no conjunto, y_{Pi} o valor a ser comparado e y_{Ri} o valor de referência.

O valor médio do erro MAE para os valores do primeiro grupo (Figura 4.4) é da ordem de 27,239%. Já o segundo grupo apresentou 34,055% de erro.

No entanto quando se compara as curvas do primeiro para com o segundo grupo obteve-se um erro de 59,26%, mostrando que realmente as curvas não têm tanta aderência. A diferença não é tão grande visto que são da mesma empresa.

Fazendo-se um comparativo entre os dados da Tabela 4.2 com os gráficos da Figura 4.4, vê-se que o gráfico do Grupo 1 apresenta na sua grande maioria as curvas de carga durante os sábados e domingos, e no gráfico do Grupo 2 as curvas de carga dos dias típicos, dado a redução drástica no consumo no horário de ponta (período no qual as empresas horossazonal pagam tarifas mais elevadas).

4.3.4 Dados do arquivo ARQ5

O arquivo ARQ5 possui dados que compreendem as curvas de carga do dia 02/10/2010 até o dia 02/11/2010, ou seja um total de 32 curvas diárias, sendo chamado de ARQ5. Cabe ressaltar que neste período ocorrem os seguintes feriados nacionais: 12/10/2010 (Dia de Nossa Senhora Aparecida) e 02/11/2010 (Dia de Finados).

A Tabela 4.3 mostra os agrupamentos obtidos utilizando o algoritmo *k-means* (Anexo A e B) . O arquivo ARQ5 foi testado utilizando $k=1$ (apenas 1 *cluster*), $k=2$ (2 *clusters*) e $k=3$ (3 *clusters*).

Tabela 4.3 – Arquivo ARQ5

ARQ5				
Data:	Dia da semana	$k=1$	$k=2$	$k=3$
		Grupo	Grupo	Grupo
02/10/2010	Sábado	1	2	1
03/10/2010	Domingo	1	2	2
04/10/2010	Segunda-feira	1	1	3
05/10/2010	Terça-feira	1	1	3
06/10/2010	Quarta-feira	1	1	3
07/10/2010	Quinta-feira	1	1	3
08/10/2010	Sexta-feira	1	1	3
09/10/2010	Sábado	1	2	1
10/10/2010	Domingo	1	2	2
11/10/2010	Segunda-feira	1	1	3
12/10/2010	Terça-feira	1	2	2
13/10/2010	Quarta-feira	1	1	3
14/10/2010	Quinta-feira	1	1	3
15/10/2010	Sexta-feira	1	1	3
16/10/2010	Sábado	1	2	1
17/10/2010	Domingo	1	2	2
18/10/2010	Segunda-feira	1	1	3
19/10/2010	Terça-feira	1	1	3
20/10/2010	Quarta-feira	1	1	3
21/10/2010	Quinta-feira	1	1	3
22/10/2010	Sexta-feira	1	1	3
23/10/2010	Sábado	1	2	1
24/10/2010	Domingo	1	2	2
25/10/2010	Segunda-feira	1	1	3
26/10/2010	Terça-feira	1	1	3
27/10/2010	Quarta-feira	1	1	3
28/10/2010	Quinta-feira	1	1	3
29/10/2010	Sexta-feira	1	1	3
30/10/2010	Sábado	1	2	1
31/10/2010	Domingo	1	2	2
01/11/2010	Segunda-feira	1	1	3
02/11/2010	Terça-feira	1	2	2

Fonte: Autoria própria.

As mesmas considerações já verificadas anteriormente no arquivo ARQ3 podem ser verificadas nas figuras a seguir, ressaltando-se que neste caso o melhor agrupamento ocorreu para $k = 3$.

Inicialmente com $k = 1$ vislumbra-se de forma vaga a sobreposição de 3 conjuntos distintos, como mostrado na Figura 4.6. Quando $k=2$ ainda no Grupo 2 da Figura 4.7 é possível notar a possibilidade de mais uma separação.

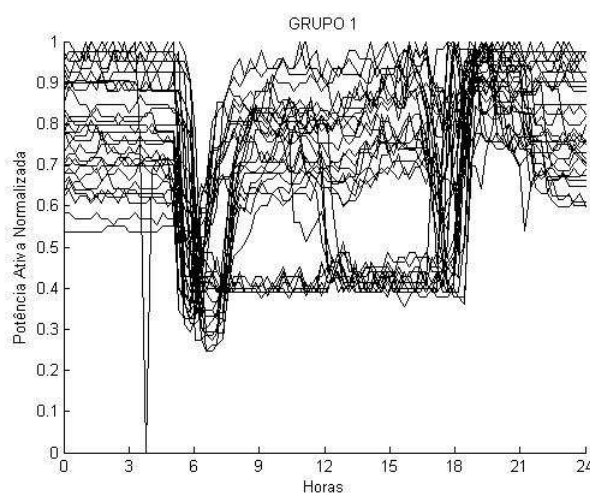


Figura 4.6 – Simulação no arquivo ARQ5, utilizando $k=1$.
Fonte: Autoria própria

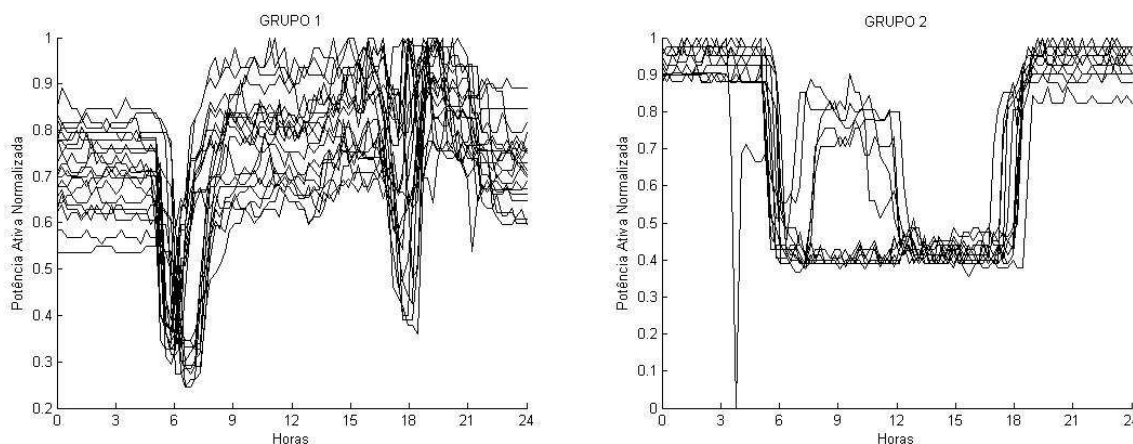
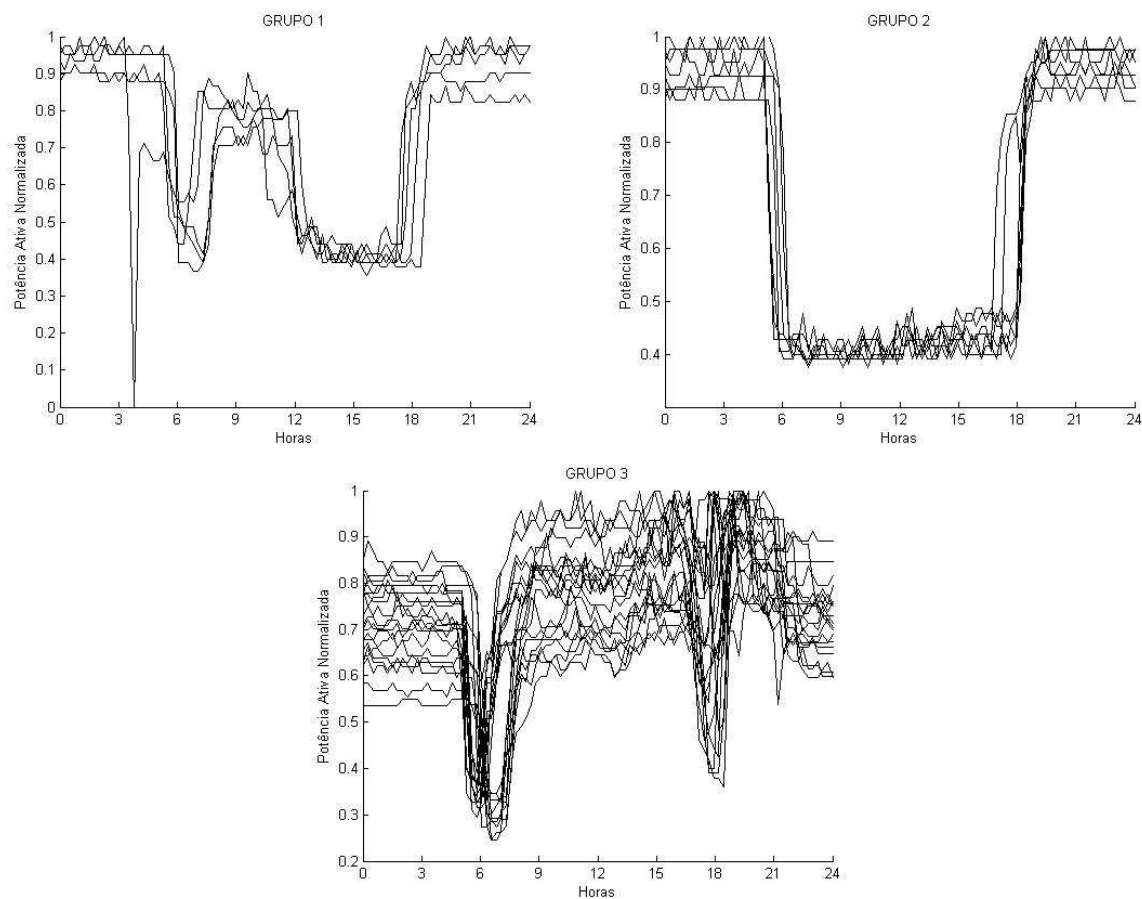


Figura 4.7 – Simulação no arquivo ARQ5, utilizando $k=2$.
Fonte: Autoria própria

É curioso observar que para $k = 3$ (Figura 4.8), foi obtido, conforme a Tabela 4.3: no primeiro grupo estão as curvas de carga de sábado desta empresa, já no segundo grupo estão contidos os comportamentos das curvas de carga de domingos, e no terceiro grupo os dias típicos da empresa. Desta forma, tem-se $k = 3$, como agrupamento ideal para esta empresa.



**Figura 4.8 – Simulação no arquivo ARQ5, utilizando $k=3$.
Fonte: Autoria própria.**

4.4 ANÁLISE DOS AGRUPAMENTOS COM DIFERENTES VALORES DE K

Tendo-se verificado que o *k-means* fornecia resultados conforme esperados, partiu-se então para as simulações de agrupamento de curvas de consumidores diversos.

O próximo passo foi reunir em um mesmo arquivo Excel uma curva de cada um dos 47 arquivos obtidos para que estas fossem então agrupadas em *clusters*. Optou-se pelas curvas de terça-feira posto que estas representassem a curva típica de um dia útil de meio de semana com horário de ponta e chamou-se este arquivo de “ARQ_TERCA”. Tomou-se o cuidado para que não fossem incluídas curvas de feriados nacionais - coincidentemente os feriados de 12 de outubro e 02 de novembro ocorreram em terças-feiras no ano de 2010 – e para que não fossem incluídas duas curvas de um mesmo arquivo, já que estas certamente seriam agrupadas em um mesmo *cluster* e comprometeriam o resultado final. Curvas de sábados e domingos

também não foram incluídas já que, por não haver horário de ponta nos fins de semana, não representam a curva típica de consumo.

A Tabela 4.4 mostra os agrupamentos obtidos utilizando o algoritmo *k-means* (Anexo A e B). O arquivo ARQ_TERCA foi agrupado utilizando $k=1$ (apenas 1 *cluster*), $k=4$ (4 *clusters*), $k=6$ (6 *clusters*), $k=8$ (8 *clusters*), $k=10$ (10 *clusters*), $k=12$ (12 *clusters*) e $k=16$ (16 *clusters*).

Tabela 4.4 – Arquivo ARQ_TERCA

ARQ_TERCA							
ARQUIVO	$k=1$	$k=4$	$k=6$	$k=8$	$k=10$	$k=12$	$k=16$
	Grupo	Grupo	Grupo	Grupo	Grupo	Grupo	Grupo
ARQ3	1	1	1	1	1	1	1
ARQ4	1	2	5	2	2	2	2
ARQ5	1	3	3	3	3	3	3
ARQ6	1	4	4	4	4	4	4
ARQ7	1	2	5	5	5	5	5
ARQ8	1	3	6	6	6	6	6
ARQ9	1	2	2	7	7	7	7
ARQ10	1	2	2	8	8	8	8
ARQ11	1	3	6	6	9	9	9
ARQ12	1	3	6	6	10	10	10
ARQ13	1	3	3	3	3	11	11
ARQ14	1	3	3	3	3	12	12
ARQ15	1	3	6	6	9	9	13
ARQ16	1	2	2	8	9	9	14
ARQ17	1	2	2	8	8	8	15
ARQ18	1	2	2	8	8	8	16
ARQ19	1	2	5	2	2	2	2
ARQ20	1	3	6	6	6	6	13
ARQ21	1	3	6	6	9	6	13
ARQ22	1	2	6	6	6	9	13
ARQ23	1	1	1	1	1	1	1
ARQ24	1	2	2	8	8	8	8
ARQ25	1	2	2	8	9	9	14
ARQ26	1	3	3	3	9	11	11
ARQ27	1	2	5	2	2	2	2

Tabela 4.4 – Arquivo ARQ_TERCA - continuação

ARQ_TERCA							
ARQUIVO	$k=1$	$k=4$	$k=6$	$k=8$	$k=10$	$k=12$	$k=16$
	Grupo	Grupo	Grupo	Grupo	Grupo	Grupo	Grupo
ARQ28	1	1	1	1	1	1	1
ARQ29	1	1	1	1	1	1	9
ARQ30	1	1	1	1	1	1	9
ARQ31	1	3	3	3	3	11	11
ARQ32	1	2	2	2	2	2	2
ARQ33	1	2	2	8	9	9	14
ARQ34	1	3	3	3	3	12	12
ARQ35	1	2	6	8	9	9	9
ARQ36	1	2	2	8	8	8	16
ARQ37	1	1	3	3	3	3	3
ARQ38	1	4	4	4	4	4	1
ARQ39	1	3	3	3	3	11	11
ARQ40	1	1	1	1	1	1	1
ARQ41	1	3	6	6	9	9	13
ARQ42	1	2	2	8	8	8	15
ARQ43	1	3	6	6	9	9	9
ARQ44	1	3	6	6	9	9	13
ARQ45	1	1	1	1	1	1	1
ARQ46	1	2	6	6	9	9	13
ARQ47	1	1	1	1	1	1	9
ARQ48	1	2	4	2	2	11	11
ARQ49	1	3	6	6	6	11	11

Fonte: Autoria própria.

A Figura 4.9 serve para mostrar a grande discrepância existente entre as empresas analisadas, e que provavelmente precisa-se de uma quantidade razoável de grupos para se obter uma solução otimizada. Ou seja, conter em cada agrupamento as empresas com padrões similares de uso de energia.

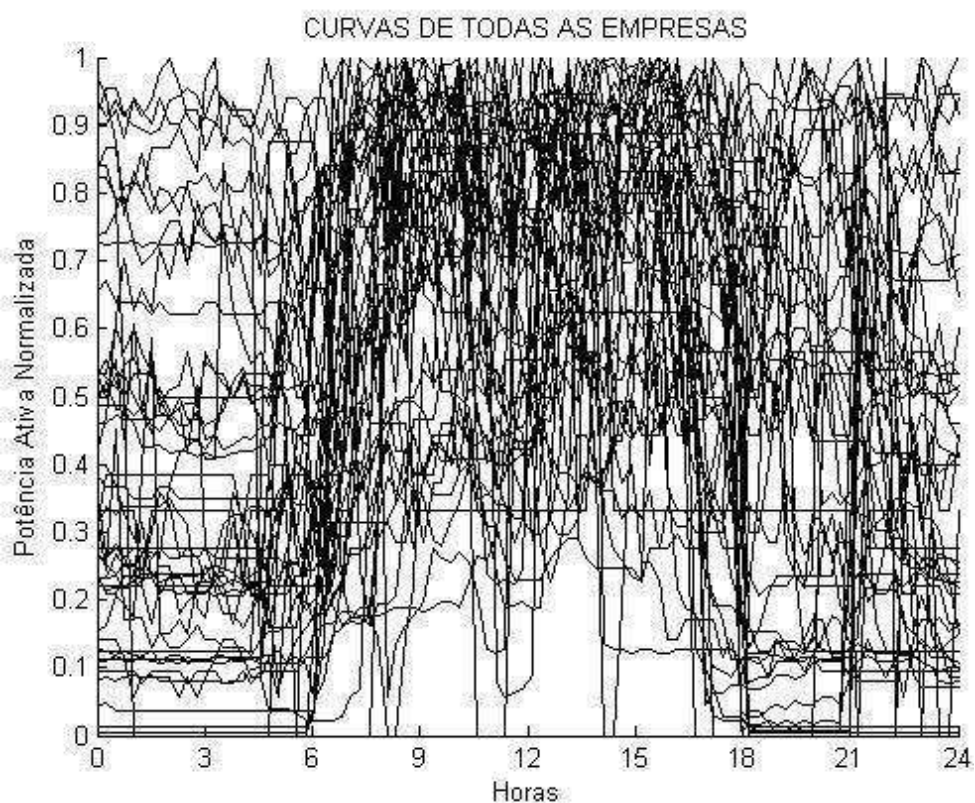


Figura 4.9 – Simulação no arquivo ARQ_TERCA, utilizando $k=1$.
Fonte: Autoria própria.

Utilizando-se $k = 4$ (Figura 4.10) e $k=6$ (Figura 4.11) notou-se que muitas curvas ainda poderiam ser agrupados em novos *clusters*.

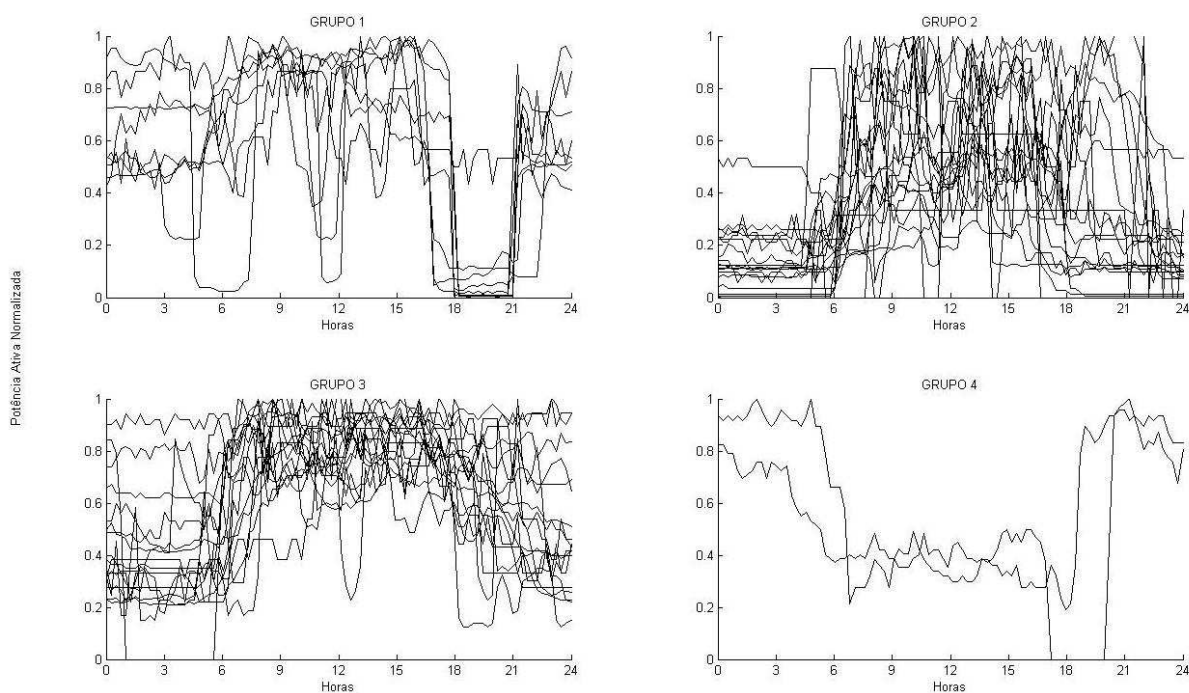
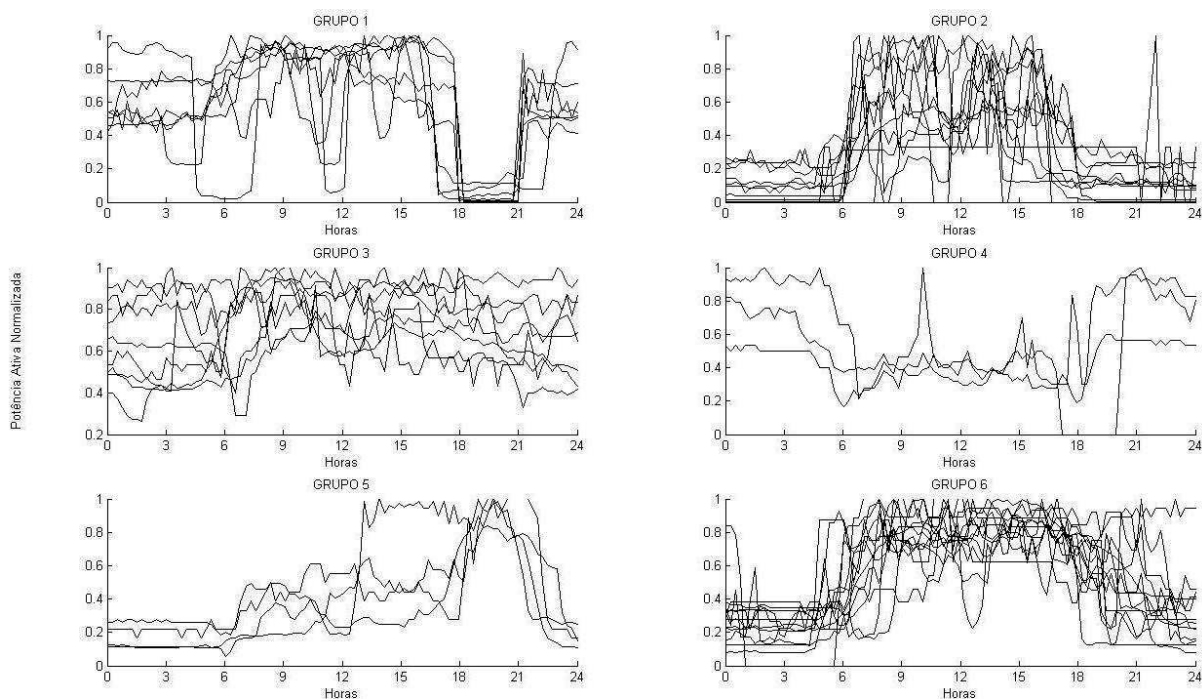
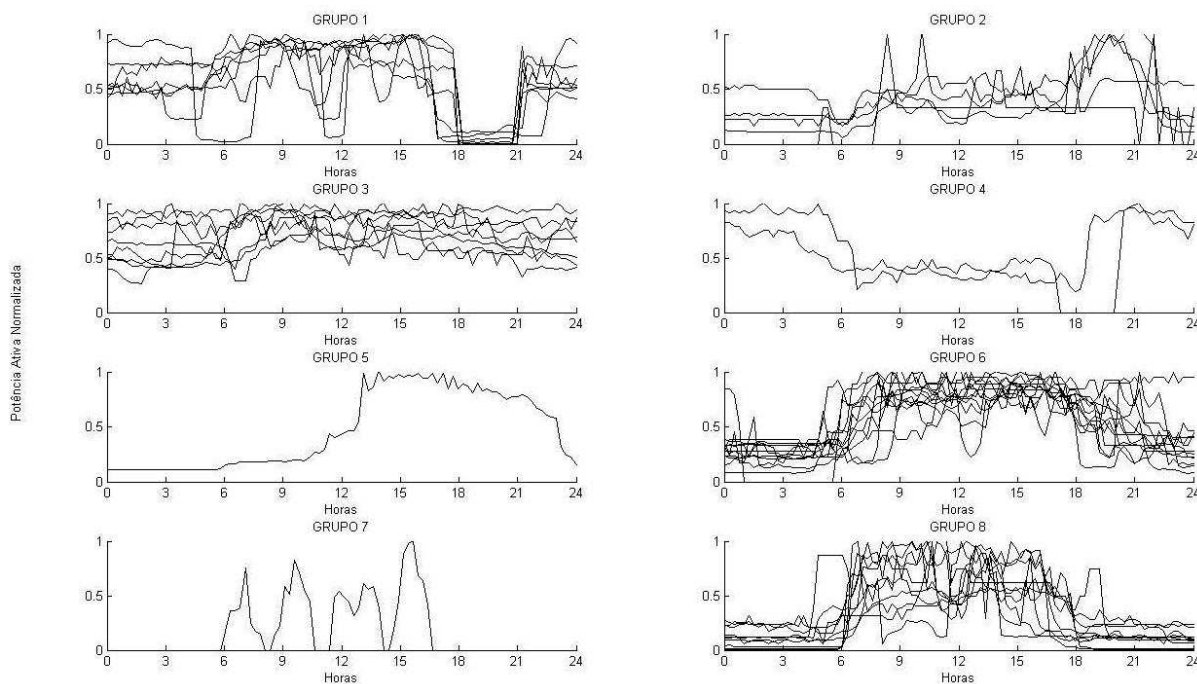


Figura 4.10 – Simulação no arquivo ARQ_TERCA, utilizando $k=4$.
Fonte: Autoria própria.



**Figura 4.11 – Simulação no arquivo ARQ_TERCA, utilizando $k=6$.
Fonte: Autoria própria.**

Quando se têm $k=8$, como mostrado na Figura 4.12, os valores estão no seu formato ideal de agrupamento.



**Figura 4.12 – Simulação no arquivo ARQ_TERCA, utilizando $k=8$.
Fonte: Autoria própria.**

Na situação em que $k=10$ os valores estão em uma representação não otimizada, como mostra a Figura 4.13. A partir de deste valor nota-se que o algoritmo *k-means* está piorando a solução desejada, pois desune grupos que apresentam similaridades entre si, como mostrado nas figuras 4.14 e 4.15.

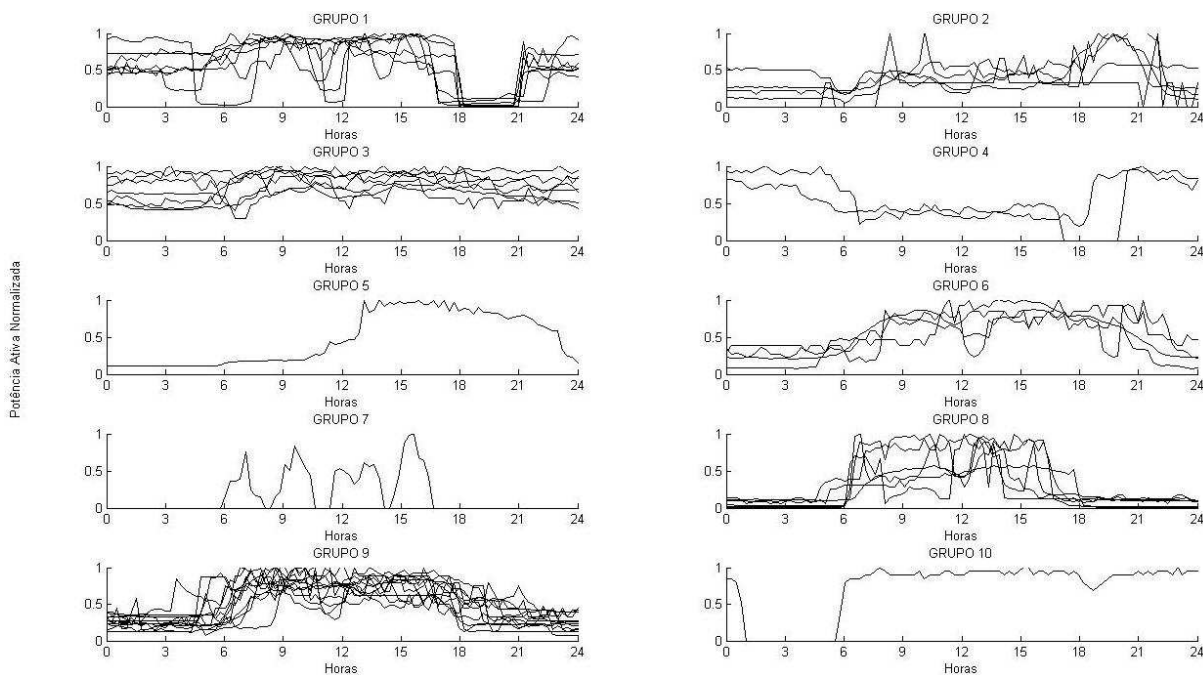


Figura 4.13 – Simulação no arquivo ARQ_TERCA, utilizando $k=10$.
Fonte: Autoria própria.

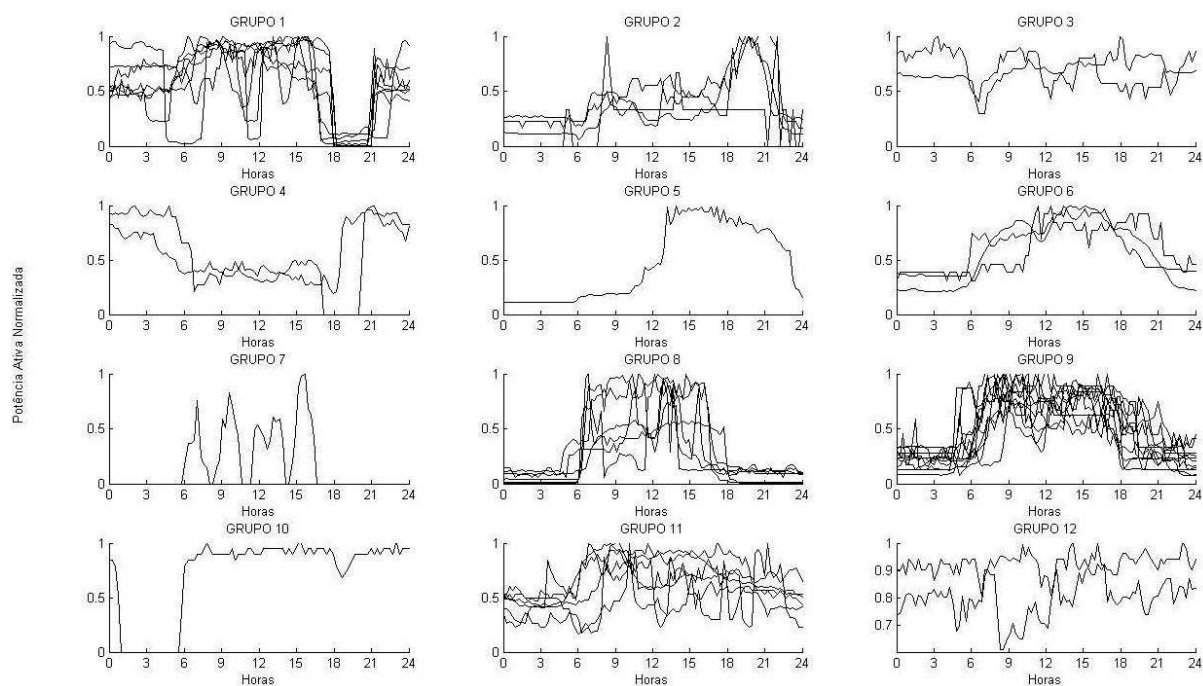
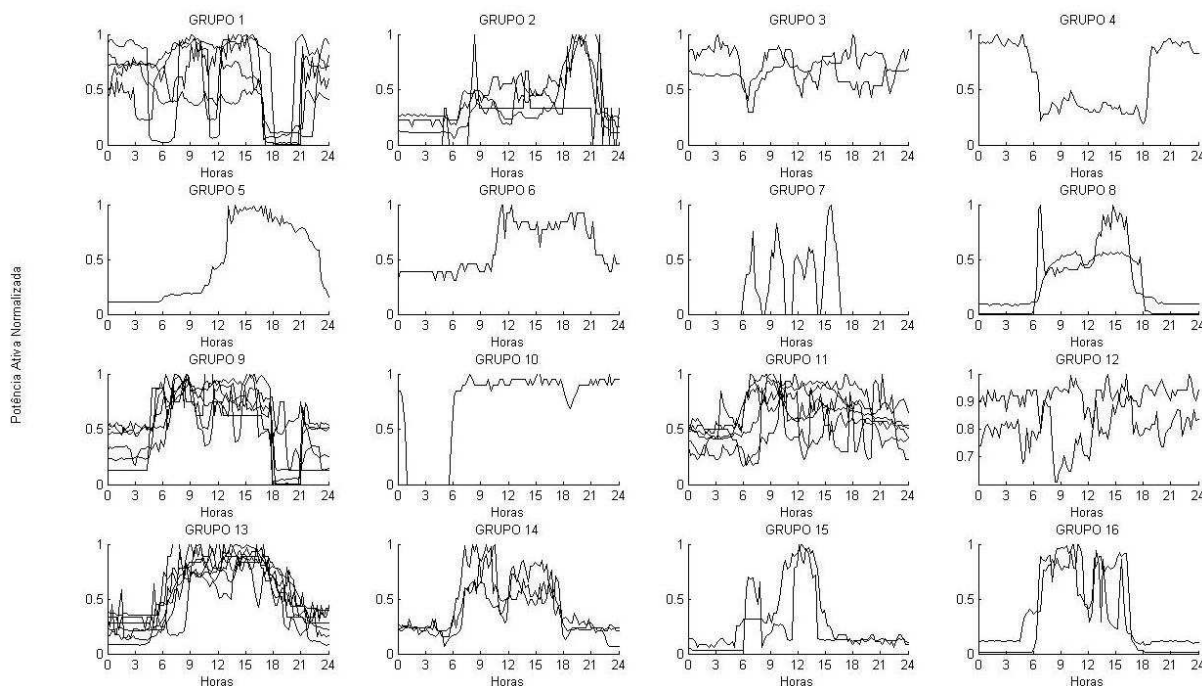


Figura 4.14 – Simulação no arquivo ARQ_TERCA, utilizando $k=12$.
Fonte: Autoria própria.



**Figura 4.15 – Simulação no arquivo ARQ_TERCA, utilizando $k=16$.
Fonte: Autoria própria.**

4.5 CONCLUSÕES DOS RESULTADOS

Com $k=8$ obteve-se um agrupamento o melhor agrupamento na opinião dos autores, pois os *clusters* têm características distintas e bem definidas. Percebe-se que o Grupo 5 (Figura 4.12) se destaca por um consumo elevado a partir das 12h. Já as empresas do Grupo 7 (Figura 4.12) provavelmente não operam durante a noite, pois sua demanda é maior entre 6 e 17h. A maioria das empresas, entretanto, têm atividades noturnas, como mostram os grupos 2, 3, 4, 6 e 8 (Figura 4.12). O Grupo 1 (Figura 4.12) destaca-se pela grande redução das atividades no horário de ponta.

Outra análise que pode ser feita, agora pelo ponto de vista dos sistemas tarifários, é levantar hipóteses em relação a qual sistema está enquadrado determinado *cluster*. Pode-se dizer que aqueles grupos com parada total ou redução de suas atividades no horário de ponta (grupos 1 e 8 (Figura 4.12), por exemplo) provavelmente têm tarifa horossazonal. O horário de ponta se caracteriza por ter uma tarifa mais elevada, tanto no período úmido quanto no seco, o que induz os consumidores englobados na tarifa horossazonal a reduzirem seu consumo e sua demanda gerando economia.

Os grupos 2, 3 e 6 (Figura 4.12) provavelmente estão enquadrados no sistema tarifário convencional ou do grupo B. Estes sistemas têm como característica principal a não diferenciação da tarifa nos segmentos horários nem nos períodos do ano, existindo apenas a tarifa de consumo para o grupo B, e de consumo e demanda no sistema convencional, logo, é indiferente, do ponto de vista financeiro, se há consumo no horário de ponta ou fora dele. Assim, o perfil da curva de carga destas empresas mostra que não há uma preocupação com relação à hora em que a energia está sendo demandada.

Ao incrementar-se o valor de k , para 10, 12 ou 16, forçadamente novos *clusters* foram criados, mesmo sendo estes redundantes e que já não representavam um agrupamento coerente das amostras. Portanto, para a amostra de 47 curvas obtidas anteriormente, o agrupamento em 8 *clusters* é o ideal.

Quando o valor de k é menor que 8, 4 ou 6, excluindo-se 1, por este não ser um agrupamento e sim somente uma amostra de todas as curvas, há poucos *clusters* ainda, fazendo com que curvas com perfis pouco parecidos estejam juntas, dificultando o discernimento do padrão de cada *cluster*.

5. CONSIDERAÇÕES FINAIS

Em relação às técnicas de agrupamento (*clustering*) notou-se que são dois os motivos principais para o seu uso nos diversos campos da ciência:

- Encontrar funções que modelem um determinado comportamento;
- Quantificar conjuntos representativos de um espaço amostral (otimizar função custo).

Dada a definição acima, destaca-se que o segundo motivo foi aquele que norteou o agrupamento dos padrões de curva de carga, já que obter uma função matemática que modelasse determinado comportamento das mesmas ou dos consumidores da região de Curitiba não era o foco principal.

Após o estudo do algoritmo *perceptron* (que pode ser considerado um neurônio artificial) e mapas auto-organizáveis de Kohonen (SOM - *Self-organizing map*) chegou-se a conclusão que estes métodos não eram viáveis para a obtenção dos agrupamentos, visto que exigiam um grande domínio da lógica computacional presente na ciência da computação. Constatou-se que a Máquina de Vetor de Suporte e o *k-means* eram os métodos com maior confiabilidade em termos de programação.

Durante o estudo foram selecionados alguns algoritmos capazes de realizar várias funções envolvendo a Máquina de Vetor de Suporte, entretanto não houve progresso ao se tentar criar agrupamentos, haja vista, que a maioria buscava a classificação de dados para posterior regressão dos mesmos, além do que o resultado alterava-se imprevisivelmente com a variação dos parâmetros envolvidos. Decidiu-se então pela técnica de agrupamento conhecida por *k-means* que alcançou resultados favoráveis para diversos testes e finalmente para o agrupamento final dos dados analisados, com somente um parâmetro, *k*.

Optou-se por se fazer agrupamentos (*clustering*) porque não existe, a priori, um número de classes a serem previstas. Outra característica do método que pesou na sua escolha foi a similaridade das amostras, nesse caso, as diversas curvas de carga, que apresentam características em comum, como por exemplo, a não utilização da energia no horário de ponta, ou somente a utilização durante o expediente diurno (das 6 às 18 horas).

Com uma amostra o mais heterogênea possível de padrões de demanda da região metropolitana de Curitiba (ARQ3 a ARQ49) e tendo-se feita a normalização das curvas em cada arquivo e posterior refinamento dos dados de um dia típico (ARQ_TERCA), partiu-se para as simulações com a técnica escolhida, neste caso, *k-means*. Feitos os primeiros testes com o ARQ3 e o ARQ5, demonstrando que a técnica era coerente, havia fortes indícios de que o objetivo estava mais próximo de ser alcançado.

Efetuada as simulações com o ARQ_TERCA, viu-se que o *k-means* oferecia a liberdade necessária para que os agrupamentos fossem criados. Infelizmente, ainda, a definição de quantos *clusters* deveriam ser criados dependia do julgamento do programador, e não era algo definido pela técnica de agrupamento.

Os agrupamentos futuramente poderão ser, se ainda não o são, incorporados nas análises das concessionárias para que estas avaliem as tarifas e os custos que consumidores com perfis de carga semelhantes impõem ao sistema, com isso melhorando a eficiência. Idealmente, consumidores com perfis de carga semelhantes (Grupo 1 ou Grupo 8 da Figura 4.12), quando estas não são constantes nas 24 horas do dia – como o Grupo 2 da Figura 4.12, não deveriam estar localizados em regiões próximas, já que estes demandariam uma grande energia em determinadas horas do dia e no restante as redes de transmissão e distribuição permaneceriam ociosas.

Dessa forma, vê-se que as técnicas de agrupamento são uma ferramenta poderosa para o planejamento e dimensionamento do Sistema Interligado Nacional (SIN).

6. REFERÊNCIAS

ANDRADE, F. **Formação de agrupamentos de consumidores de energia elétrica com base nos perfis de carga individuais.** UFSC, 2003.

ANEEL. **Resolução ANEEL N°414, Tarifas de Fornecimento de Energia Elétrica.** Brasília. Abr, 2010.

ANEEL. **Resolução ANEEL N°90, Tarifas de Fornecimento de Energia Elétrica.** Brasília. Abr, 2005. 30 p.

BITU, R. S.; BORN, P. H. **Tarifas de Energia Elétrica – Aspectos Conceituais e Metodológicos.** 1a ed. São Paulo: MM Editora, 1993.

BOSER, B. E.; GUYON, I. M.; VAPNIK, V. N. **A training algorithm for optimal margin classifiers.** In D. Haussler, edition, *Proceedings of the Fifth Annual Wokshop on Computational Learning Theory, Pittsburgh*, pg.144-152, 1992.

BURGES, C. **A Tutorial on Support Vector Machines for Pattern Recognition. Data Mining and Knowledge Discovery.** Boston, MA: Kluver, 1998. v.2, p 121-167.

CHAN,C. C.; LIN,C. J. **LibSVM - A Library for Support Vector Machines.**
Disponível em: < <http://www.csie.ntu.edu.tw/~cjlin/libsvm/>>.

CHEN, C. S.; HWANG, J. C.; HUANG, C.W. **Application of Load Survey Systems to Proper Tariff Design.** IEEE Transactions on Power Systems, v. 12, n. 4, p. 1746-1751. Nov, 1997.

CHICCO, G.; NAPOLI, R.; POSTOLACHE, P.; SCUTARIU, M.; TOADER, C. **Load Pattern – Based Classification of Electricity Customers.** IEEE Transactions on Power Systems, v. 19, n. 2, p. 1232-1239. May, 2004.

CHICCO, G.; ILIE, I.S. **Support Vector Clustering of Electrical load Pattern Data.** IEEE Transactions on Power Systems, v. 24, n. 3, p. 1619-1628. Aug, 2009.

COPEL. **Revisão tarifária compreendendo as tarifas,** 2008.

COVER, T. M. **Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition.** IEEE Transactions on Electronic Computers, vol. EC-14, pg.326-334, 1965.

CRISTIANINI N., TAYLOR J. S. **“Kernel Methods for Pattern Analysis”.** Cambridge University Press, 2004.

CRISTIANINI N., TAYLOR J. S. **“An Introduction to Support Vector Machines and other kernel-based learning methods”.** Cambridge University Press, 2001.

ESELSA ENERGIAS DO BRASIL. **Manual do Cliente Horo-Sazonal.** Vitória. Dez, 2004. 87 p.

HAYKIN, S. **Neural Network – A Comprehensive Foundation**. 2º editin, Prentice-Hall, New Jersey, 1999.

HAYKIN, S. **Redes neurais: Princípios e prática**. 2ºedição Bookman Companhia Editora. Porto Alegre - Brasil , 2001.

HEARST, M. A. et al. **Support vector machines**. IEEE Intelligent Systems, IEEE Computer Society, Los Alamitos, CA, USA, v.13, n.4, p.18-28, 1998. ISSN1094-7167.

HUTCHINSON, A. **Algorithmic Learning**. Oxford: Clarendon, New York 1994.

JOACHIMS, T. **Learning to classify texts using support vector machines: methods, theory and algorithms**. Kluwer Academic Publishers, 2002.

_____. **SVM^{Light} Support Vector Machine**

Disponível em: < <http://svmlight.joachims.org/>>.

KARDI, T., **K-Mean Clustering Code in Matlab**.

Disponível em:

<http://people.revoledu.com/kardi/tutorial/kMean/matlab_kMeans.htm>

KIM, K. I.; JUNG, K.; PARK, S. H.; KIM, H. J. **Support vector machines for texture classification**. IEEE Transactions on Pattern Analysis and Machine Intelligence, 24 (11): 1542-1550, 2002.

LIMA, A.R.G. **Máquinas de Vetores Suporte na Classificação de Impressões Digitais**. Dissertação (Mestrado) - Universidade Federal do Ceará, Fortaleza, Ceará, 2002.

LORENA, A. C.; CARVALHO, A. C. P. L. **Introdução às Máquinas de Vetores Suporte**. São Carlos-SP, Abril 2003.

_____. **Introdução aos Classificadores de Margens Largas**. São Carlos-SP, Maio 2003.

_____. **Uma Introdução às Support Vector Machines**. RITA, Volume XIV, Número 2. São Carlos-SP, 2007.

MARKOWETZ, F. **Support Vector Machines in Bioinformatics**. Diplomarbeit Universität Heidelberg. 2003.

MATLAB. Bioinformatics ToolboxTM

Disponível em: < <http://www.mathworks.com/products/bioinfo/>>.

MIRKIN, B. G. **Clustering for Data Mining: a data recovery approach**. Chapman & Hall/CRC, London 2005.

MONEDERO, I.; BISCARRI, F.; LEÓN, C.; BISCARRI, J.; MILLÁN, R. M. **Detection of non-technical Losses in Electrical Consumption Using Neural Networks and Statistical Techniques**. Proc. Computational Science and its Applications 2006. Pág. 725-734, 2006.

MUKKAMALA, S.; SUNG, A. H. **Identifying important features for intrusion detection using support vector machines and neural networks.** *Applications and the Internet. Proceedings.* 2003 Symposium, p.209-216, January 2003.

MÜLLER, K. R.; MIKA, S.; RÄTSCH, G.; TSUDA, K.; SCHÖLKOPF, B. **An Introduction to Kernel-Based Learning Algorithms.** *IEEE Transactions on Neural Networks*, vol.12, N°2,pg 181-202, 2001.

OLIVEIRA, G. F. de; BRUNONI, R. E. **Criação de uma Ferramenta para Gerenciamento do Consumo da Energia Contratada na Universidade do Rio de Janeiro.** Rio de Janeiro. Abr, 2007. 41p.

PEREIRA, J. C. **Tarifa Horó-sazonal.** Rio de Janeiro. 2003. 18 p.

PERFECTUM, **Tarifa Horó-sazonal.** Curso de conservação de energia, 2003.

Disponível em:

<http://www.perfectum.eng.br/energia/tarifa_horo.html>

PLATT, J.; CRISTIANINI, N.; SHAWE-TAYLOR, J. **Large margin dags for multiclass classification.** In: SOLLA, S., LEEN, T.; MUELLER, K.-R. (Ed.). *Advances in Neural Information Processing Systems12.* [S.l.: s.n.], 2000. p. 547-553.

PONTIL, M.; VERRI, A. **Support Vector Machines for 3-D object recognition.** *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 20(6): 637-646, 1998.

PROCEL. **Manual de Tarifação da Energia Elétrica.** 1° Edição. Mai, 2001. 44 p.

RAMOS, M. C. G., **Avaliação dos pontos de desperdício de energia elétrica na indústria.** Curso de conservação de energia, 2003

Disponível em:

<ftp://ftpaluno.umc.br/profs/Mario_Cesar/Curso_Conserva%20E3o_EE/Texto%20pdf/Cap%20EDtulo%202.pdf>

SCHÖLKOPF, B.; SMOLA, A. J.; MÜLLER, K. –R. **Nonlinear component analysis as a kernel eigenvalue problem.** *Neural computation*, vol.10, pg.1299-1319, 1998.

SEMOLINI, R., **Support Vector Machines, Inferência Transdutiva.** Tese de Mestrado. UNICAMP, Campinas – SP, 2002

SMOLA, A. J. **Regression estimation with Support Vector Learning Machines.** Master's thesis. Physik Department, Technische Universität München, 1996.

SMOLA, A. J.; BARTLETT, P.L.; SCHÖLKOPF, B.; SCHUURMANS, D. **Introduction to large margin classifiers.** In A. J. Smola, P.L. Bartlett, B. Schölkopf, D. Schuurmans, editors, *Advances in large margin classifiers.* pg 1-29, Cambridge, 2000

SMOLA, A. J. et al. **Introduction to large margin classifiers.** In: .[S.l.]:Morgan-Kauman, 1999. cap.1 , p.1-28.

_____. **Advances in Large Margin Classifiers.** [S.l.]: Morgan-Kauffman,1999.

SILVA, A. L. R. da; JANNUZZI, G. D. M. **Segmentação de clientes de energia elétrica por nichos especiais**. Universidade de Campinas – UNICAMP. ISSN: 1807-8575 – Número 10. Abr, 2009. 10 p.

SORIA, A. F. S.; FILIPINI, F. A. **Eficiência Energética**. Base Editorial, Curitiba, 2010.

SOUTO, M. C. P.; LORENA, A. C.; DELBEM, A. C. B.; CARVALHO, A. C. P. L. F. **Técnicas de Aprendizado de Máquina para problemas de Biologia Molecular**, pg 103–152. Minicursos de Inteligência Artificial, Jornada de Atualização Científica em Inteligência Artificial, XXIII Congresso da Sociedade Brasileira de Computação, 2003.

SOUZA, J. L. M. **Planejamento na utilização da energia elétrica no meio rural: tarifas horo-sazonais**. Apresentação da Disciplina de Eletrificação Rural ministrada na UFPR. 50 p.

TAN, P.; STEINBACH, M.; KUMAR, V. **Introduction to data mining**. Pearson, Boston 2006.

TSEKOURAS, G. J.; HATZIARGYRIOU, N. D.; DIALYNAS, E. N. **Two-stage Pattern Recognition of Load Curves for Classification of Electricity Customers**. IEEE Transactions on Power Systems, v. 22, n. 3, p. 1120-1128. Aug, 2007.

VALERO, S.; ORTIZ, M.; SENABRE, C.; ALVAREZ, C.; FRANCO, F. J. G.; GABALDÓN, A. **Methods for Customer and Demand Response Policies Selection in New Electricity Markets**. IET Generation, Transmission, Distribution, v.1, p. 104-110. Jan, 2007.

VAPNIK, V. N. **The nature of statistical learning theory**. New York, NY, USA: Springer-Verlag New York, Inc., 1995. ISBN0387945598.

Disponível em: <<http://portal.acm.org/citation.cfm?id=211359>>.

VAPNIK, V. **The Nature of Statistical Learning Theory**. 2ª edição, Springer – Verlag. New York, 2000.

WITTEN, I. H; FRANK, E. **Data mining: Practical Machine Learning Tools and Techniques**. 2ª edição, Morgan Kaufmann, Amsterdam 2005.

7. ANEXOS

ANEXO A – Algoritmo *k-means*

Baseado em (KARDI, 2011):

```
function [y,k,order_ARQ,MAE]=kMeansCluster(ARQ,k,isRand)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% kMeansCluster - Simple k means clustering algorithm
% Author: Kardi Teknomo, Ph.D.
%
% Purpose: classify the objects in data matrix based on the attributes
% Criteria: minimize Euclidean distance between centroids and object points
% For more explanation of the algorithm, see
http://people.revoledu.com/kardi/tutorial/kMean/index.html
% Output: matrix data plus an additional column represent the group of each
object
%
% Example: m = [ 1 1; 2 1; 4 3; 5 4] or in a nice form
%           m = [ 1 1;
%                 2 1;
%                 4 3;
%                 5 4]
%           k = 2
% kMeansCluster(m,k) produces m = [ 1 1 1;
%                                   2 1 1;
%                                   4 3 2;
%                                   5 4 2]
% Input:
% m      - required, matrix data: objects in rows and attributes in
columns
% k      - optional, number of groups (default = 1)
% isRand - optional, if using random initialization isRand=1, otherwise
input any number (default)
%        it will assign the first k data as initial centroids
%
% Local Variables
% f      - row number of data that belong to group i
% c      - centroid coordinate size (1:k, 1:maxCol)
% g      - current iteration group matrix size (1:maxRow)
% i      - scalar iterator
% maxCol - scalar number of rows in the data matrix m = number of
attributes
% maxRow - scalar number of columns in the data matrix m = number of
objects
% temp   - previous iteration group matrix size (1:maxRow)
% z      - minimum value (not needed)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
close all
clear i
clear j
clear y
clear tam
clear tamy
clear ans
```

```

format long
tam=size(ARQ);
ARQ=Norma(ARQ);

if nargin<3,          isRand=0;   end
if nargin<2,          k=1;        end

[maxRow, maxCol]=size(ARQ);
if maxRow<=k,
    y=[ARQ, 1:maxRow];
else
    % initial value of centroid
    if isRand,
        p = randperm(size(ARQ,1));      % random initialization
        for i=1:k
            c(i,:)=ARQ(p(i),:);
        end
    else
        for i=1:k
            c(i,:)=ARQ(i,:);            % sequential initialization
        end
    end
end

temp=zeros(maxRow,1); % initialize as zero vector

while 1,
    d=DistMatrix(ARQ,c); % calculate objects-centroid distances
    [z,g]=min(d,[],2); % find group matrix g
    if g==temp,
        break; % stop the iteration
    else
        temp=g; % copy group matrix to temporary variable
    end
    for i=1:k
        f=find(g==i);
        if f % only compute centroid if f is not empty
            c(i,:)=mean(ARQ(find(g==i),:),1);
        end
    end
end

y=[ARQ,g];

end

% Plot routine
plot_clusters(ARQ,y,k);
% MAE error routine
[order_ARQ,MAE]=errormae(y,k);

```

ANEXO B – Algoritmo para cálculo da distância euclidiana

Baseado em (KARDI, 2011):

```
function d=DistMatrix(A,B)
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    % DISTMATRIX return distance matrix between points in A=[x1 y1
... w1] and in B=[x2 y2 ... w2]
    % Copyright (c) 2005 by Kardi Teknomo,
    http://people.revoledu.com/kardi/
    %
    % Numbers of rows (represent points) in A and B are not
necessarily the same.
    % It can be use for distance-in-a-slice (Spacing) or distance-
between-slice (Headway),
    %
    % A and B must contain the same number of columns (represent
variables of n dimensions),
    % first column is the X coordinates, second column is the Y
coordinates, and so on.
    % The distance matrix is distance between points in A as rows
% and points in B as columns.
    % example: Spacing= dist(A,A)
    % Headway = dist(A,B), with hA ~= hB or hA=hB
    %           A=[1 2 3; 4 5 6; 2 4 6; 1 2 3]; B=[4 5 1; 6 2 0]
    %           dist(A,B)= [ 4.69    5.83;
    %                       5.00    7.00;
    %                       5.48    7.48;
    %                       4.69    5.83]
    %
    %           dist(B,A)= [ 4.69    5.00    5.48    4.69;
    %                       5.83    7.00    7.48    5.83]
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    [hA,wA]=size(A);
    [hB,wB]=size(B);
    if wA ~= wB, error(' second dimension of A and B must be the
same'); end
    for k=1:wA
        C{k}= repmat(A(:,k),1,hB);
        D{k}= repmat(B(:,k),1,hA);
    end
    S=zeros(hA,hB);
    for k=1:wA
        S=S+(C{k}-D{k}').^2;
    end
    d=sqrt(S);
end
```

ANEXO C – Algoritmo para cálculo do erro MAE (Mean Error Absolute)

Autoria própria:

```
function [order_ARQ,MAE]=errormae(y,k)

format long
clear order_ARQ
clear MAE

tam=size(y);
j=1;
for cont=1:1:k
    for i=1:1:tam(1)
        if(cont==y(i,tam(2)))
            order_ARQ(j,:)=y(i,:);
            j=j+1;
        end
    end
end

for cont=1:1:k
    tamp=size(order_ARQ);
    for m=1:1:tamp(1)
        for n=1:1:tamp(1)
            if(m~=n & order_ARQ(m,tamp(2))==cont & order_ARQ(n,tamp(2))==cont)
                V(1,:)=1-order_ARQ(n,:)./order_ARQ(m,:);
                V=abs(V);
                V(isnan(V))=0;
                V(isinf(V))=0;
                MAE1(m,n,cont)=1/96*(sum(V(1,:)));
            else
                MAE1(m,n,cont)=0;
            end
        end
    end
end
MAE{cont}=full(MAE1(:,: ,cont));
clear MAE1
end
```

ANEXO D – Algoritmo para plotagem dos agrupamentos

Autoria própria:

```
function plot_clusters(ARQ,y,k)

tam=size(ARQ);
tamy=size(y);
for i=1:1:k
    figure
    hold on
    for j=1:1:tam(1)
        if(y(j,tamy(2))==i)
            xData = linspace(0,24,96);
            plot(xData,ARQ(j,:), 'k')
            xlim([0 24])
            set(gca, 'XTick', [0:3:24])
        end
    end
end

figure

for i=1:1:k
    hold on
    for j=1:1:tam(1)
        if(y(j,tamy(2))==i)
            xData = linspace(0,24,96);
            plot(xData,ARQ(j,:), 'k')
            xlim([0 24])
            set(gca, 'XTick', [0:3:24])
        end,
        hold on
        subplot(4,3,i)
    end
end
```