UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE MATEMÁTICA E ESTATÍSTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM MATEMÁTICA APLICADA

# Classificação da doença de Alzheimer usando algoritmos baseados em distância

por

Yuri Elias Rodrigues

Dissertação submetida como requisito parcial
para a obtenção do grau de

Mestre em Matemática Aplicada

Profª. Drª. Carolina Cardoso Manica
Orientadora

Prof. Dr. Evandro Manica
Coorientador

Porto Alegre, Dezembro 2016.

CIP - Catalogação na Publicação

Elias Rodrigues, Yuri
Alzheimer's disease classification using distance-based algorithms / Yuri Elias Rodrigues. -- 2016.
100 f.

Orientadora: Carolina Cardoso Manica.
Coorientador: Evandro Manica.

Dissertação (Mestrado) -- Universidade Federal do Rio Grande do Sul, Instituto de Matemática, Programa de Pós-Graduação em Matemática Aplicada, Porto Alegre, BR-RS, 2016.

1. pattern recognition. 2. statistical learning. 3. alzheimer's disease. 4. metric learning. 5. k-nearest neighbors. I. Cardoso Manica, Carolina, orient. II. Manica, Evandro, coorient. III. Título.

Elaborada pelo Sistema de Geração Automática de Ficha Catalográfica da UFRGS com os dados fornecidos pelo(a) autor(a).

# Classificação da doença de Alzheimer usando algoritmos baseados em distância

por

Yuri Elias Rodrigues

Trabalho submetido ao Programa de Pós-Graduação em Matemática Aplicada do Instituto de Matemática da Universidade Federal do Rio Grande do Sul, como requisito parcial para a obtenção do grau de

## Mestre em Matemática Aplicada

Linha de Pesquisa: Reconhecimento de Padrões

Orientadora: Prof[a]. Dr[a]. Carolina Cardoso Manica

Coorientador: Prof. Dr. Evandro Manica

Banca examinadora:

Prof. Dr. Paulo Martins Engel
Instituto de Informática, UFRGS

Prof. Dr. Jacob Scharcanski
Instituto de Informática, UFRGS

Prof. Dr. Alexandre Rosa Franco
InsCer, PUCRS

Dissertação submetida e aprovada em
20 de Dezembro de 2016.

Prof. Dr. Carlos Hoppen
Coordenador

# AGRADECIMENTOS

Faz mais de um mês que a dissertação foi defendida e me sinto bem em dedicar alguns sentimentos merecidos ao fim deste ciclo. Ao passar de todas as dificuldades em que nenhuma análise cabia aos meus dados e que não havia nada ali além de um desafio pendente tive momentos agradáveis com amigos. Retrospectivamente, queria agradecer a Cristina Zaniol pelas várias opiniões e pela parceria acadêmica que não teria em nenhum lugar do mundo. Ao pessoal do machine learning, Guilherme Garcia e Eliezer Bernart, pelas pizzinhas e pelas trocas de idéias. Ao meu orientador que confiou em meu trabalho, Evandro Manica. Ao Eduardo Zimmer, por me fazer acreditar no meu trabalho. A Helena Gertz por me ensinar sobre confiança e amor. A minha família eu devo um pedido de desculpas pela ausência. Ao meu amigo Rogerio Valim que me ensinou sobre a saudade. Ao pessoal de Coimbra que vai ser sempre minha segunda família.

Agradeço a política implementada durante a era pré-golpe que permitiu que muitos estudantes tivessem acesso a um conhecimento em universidades no exterior. Espero que nosso pais seja um estado de bem-estar social fortemente voltado para o descobrindo e o desenvolvimento de nossas capacidades como cidadãos. Não entendo esta pesquisa como final, apenas como um intermédio para tópicos resultantes da união entre a neurociências e a matemática.

---

# SUMÁRIO

# LISTA DE FIGURAS

# LISTA DE TABELAS

# LISTA DE SÍMBOLOS

| | |
|---|---|
| $\Omega$ | class space |
| $\omega$ | class label |
| $\hat{\omega}$ | true class |
| $\tilde{\omega}$ | estimated class |
| $\mathcal{F}$ | feature space |
| $x$ | pattern |
| $x_*$ | unclassified pattern |
| $\omega_*$ | true class of unclassified pattern |
| $\mathcal{C}(.)$ | classifier function |
| $\mathcal{C}^*(.)$ | optimal classifier function |
| $\mathcal{N}(.,.)$ | Gaussian distribution |
| $\mathcal{U}(.,.)$ | uniform distribution |
| $\mu$ | mean |
| $\Sigma$ | covariance matrix |
| $\sigma$ | standard deviation |
| $\lambda$ | singular values average |
| $m$ | class space dimension |
| $n$ | feature space dimension |
| $f$ | discriminant function |
| $R$ | decision region |
| $T$ | training set |
| $X^d$ | set of patterns in $\mathbb{R}^d$ |
| $D$ | vectors dataset |
| $L$ | conditional probability of error |

| | |
|---|---|
| $L_n$ | conditional probability of error for $n$ samples |
| $E(.)$ | expectation function |
| $\mathcal{P}$ | confusion matrix |
| $L_N N$ | nearest neighbor conditional probability of error |
| $k$ | number of nearest neighbors or number of folds in cross validation |
| $N(x_*, k)$ | k-neighborhood |
| $d(.,.)$ | distance function |
| $p(.\|.)$ | conditional probability function |
| $p$ | $L_p$ norm value |

# RESUMO

A doença de Alzheimer é a mais comum das patologias neurodegenerativas afetando principalmente pessoas idosas e causando considerável impacto econômico em todos os países. Atualmente não há cura para cessar seus efeitos na memória fazendo do diagnóstico precoce fundamental para eventuais intervenções terapêuticas. Para tal, técnicas de reconhecimento de padrões podem ser utilizadas para diferenciar indivíduos com o mal Alzheimer de indivíduos saudáveis. No entanto, aumentando o número de diagnósticos (classes) que definem estágios intermediários da doença a capacidade de diferenciar classes de tais técnicas é comprometida. Isto ocorre devido ao fato de que as medidas biológicas, ou os biomarcadores, não são discriminantes o suficiente para lidar com classes além do caso extremo, de forma binária, Alzheimer versus controle normal. Outro fator que dificulta o diagnóstico é a distribuição de probabilidade das classes intermediárias serem altamente sobrepostas e com probabilidade a priori representando 60% dos dados.

Neste trabalho vamos demonstrar a capacidade de melhorar a precisão do diagnóstico utilizando classificadores e técnicas de amostragem baseados em funções de distância. Abordagens binárias ou abordagens binárias adaptadas a problemas multiclasse são a regra na literatura de diferenciação das classes da doença de Alzheimer. No presente momento existe apenas um único artigo demonstrando a possibilidade de melhora nas medidas de classificação com um tratamento de reamostragem das probabilidades a priori das classes de Alzheimer. Diferentemente do artigo citado que trabalha com classificação binária, aqui vamos usar uma abordagem de classificação *todos-contra-todos* na avaliação de um problema multiclasse da doença de Alzheimer. Demonstramos que a taxa de classificação do nosso classificador utilizando classes balanceadas e uma função de distância apropriada é superior a classificadores populares. Este trabalho apresenta também duas novas estratégias de ajuste do desequilíbrio tendo como medida de similaridade

a distância entre os padrões. A de subamostragem, denominada Nearmiss 4, que obteve maior redução de sobreposição entre as classes nas comparações entre algoritmos de amostragem, e outra de sobreamostragem, denominada SMOTE-borderline 3, uma versão multiclasse para a família de algoritmos SMOTE que se destaca por ser naturalmente multiclasse ao contrário das abordagens binárias adaptadas para multiclasse tratadas neste texto.

Finalmente, são comparadas técnicas de seleção de características para avaliar o poder discriminativo entre biomarcadores do mal de Alzheimer visando encontrar o subconjunto de biomarcadores que fornece a melhor taxa de classificação. Usando tal abordagem foi possível encontrar o biomarcador com maior capacidade discriminativa em um dado conjunto de biomarcadores.

# ABSTRACT

Alzheimer's disease is the most common neurodegenerative disorder affecting mainly elderly people and causing considerable economic impact in all countries. Currently, there is no cure to cease its effects in memory making early diagnosis fundamental for any therapeutic interventions. To this end, pattern recognition techniques can be used to differentiate individuals with Alzheimer's from healthy individuals. However, increasing the number of diagnoses (classes) which define disease's intermediate stages (e.g. mild cognitive impairment) the classes differentiation for such techniques is compromised. This is due to the fact that biological measures, or biomarkers, are not discriminant enough to deal with classes beyond the binary extreme case, Alzheimer's versus normal control. Another factor that makes diagnosis difficult is the probability distribution of the intermediate classes being highly overlapping and with a priori probability representing 60 % of the data.

In this work, we will demonstrate the ability to improve diagnostic accuracy using classifiers and sampling techniques based on distance functions. Binary approaches or binary approaches adapted to multiclass problems are ubiquitous in the literature of differentiating the classes of Alzheimer's disease. At present, there is only one article demonstrating the possibility of improvement in the classification measures using a sampling treatment to change *a priori* probabilities of the Alzheimer's classes. Differently from the cited article working with binary classification, here we will use an *all-versus-all* classification approach to evaluating an Alzheimer's disease multiclass problem. Also, is shown that classification rate of our classifier using balanced classes and an appropriate distance function is able to overcome popular classifiers choices. This work also presents two new strategies of imbalance adjustment, taking as a measure of similarity the distance between the patterns. The undersampling, called Nearmiss 4, obtained the greatest reduction in class overlapping comparing the sampling algorithms discussed, and another contri-

bution is an oversampling, called SMOTE-borderline 3, a multiclass version for the family of SMOTE algorithms that stands out for being naturally multiclass unlike the binary approaches adapted for multiclass treated in this text.

Finally, feature selection techniques are compared to evaluate the discriminative power between Alzheimer's disease biomarkers in order to find the subset of biomarkers which provides the best classification rate. Using this approach it was possible to find the biomarker with the highest discriminative capacity in a given set of biomarkers.

# 1 INTRODUCTION

Alzheimer's disease (AD) has a profound social and economic impact on the world's population. Mainly elderly people is affected by AD, what poses a financial burden on health and social security for countries worldwide. Actually, there are no AD treatments to cease damage of neurons that causes the symptoms and ultimately leads to death [5]. AD is pathophysiologically characterized by the gradual brain deposition of amyloid plaques, neurofibrillary tangles, and eventual neuronal depletion [40]. The AD spectrum can be explained by preclinical (CN), mild cognitive impairment (MCI) and AD dementia phases [40]. Preclinical AD individuals are those cognitively normal with amyloid plaques and tangles, individuals with MCI have cognitive symptoms without meeting clinical criteria for dementia, and AD dementia individuals present severely compromised cognitive faculties [75]. In recent years, a plethora of disease trackers, called biomarkers, has been developed in order to track AD progression. For instance, biomarkers for beta peptide 1-42 ($A\beta_{1-42}$) and tau proteins that indicate the presence of the hallmark pathological features of AD, namely, amyloid plaques [40] and neurofibrillary tangles [25].

This thesis proposes to analyze the precision of such biomarkers when identifying the region of AD spectrum which a given patient belongs. To this, pattern recognition techniques will be applied to predict a pattern class or label, i.e. a structured category in which patterns are organized. In pattern recognition, the *classifiers* are central objects of study which use pre-labeled observations to identify non-categorized observations. Distance-based classifiers use distance functions as similarity measures observations. Such classifiers are widely known since were essential to early theoretical development in pattern recognition and are used as a benchmark comparison for any new classification technique [19].

Usually, AD datasets contain issues such as missing data due to patients quitting the study or dying, and imbalanced classes, due to different probability

of occurrences in AD spectra [20]. Such artifacts in data quality are detrimental to diagnosis identification [47]. Distance-based pre-processing strategies are one possible solution to deal with such artifacts. For instance, sampling methods in which the main goal is to modify the class priors (resizing dataset classes), and, data imputation algorithms are able to substitute missing data. In this thesis, it will be proposed a study of how such modifications in AD datasets by means of distance-bases algorithms (classifiers and sampling techniques) can benefit AD identification.

In another front, instead of answering which are the classifier modifications that highly improve AD classification, a biomarker-wise comparison is studied in order to answer which biomarker combination provides best chances to identify AD. This is done in the context of feature selection techniques wrapped with distance-based classifiers [46]. Feature selection techniques basically find a subset of features that maximize a chosen criterion of usefulness [30], however there is no optimality ensured [60]. Not restrict to this, feature extraction from raw data [53] and feature construction from other features are found in literature related with AD classification [43]. Using such techniques it is possible to compare biomarkers modalities and the probability of a given biomarker combination to provide a higher precision than other.

This thesis is organized as follows: the second chapter presents basic concepts of pattern recognition as well as the validation scheme for the remaining thesis; the third chapter, how distance-based classifier is defined and can be modified by means of distance functions and normalization schemes; the fourth chapter, distance-based inspired sampling strategies; the fifth chapter explores wrappers feature selection using a distance-based classifier as usefulness measure; the sixth chapter which describes the datasets (should be read first); the last chapter points out future work suggestions and thesis conclusion. Related work will be introduced at the beginning of each section if it is necessary.

# 2 BASIC CONCEPTS IN PATTERN RECOGNITION

This chapter presents some fundamental concepts to understand the main contents of this thesis. It also provides examples of classification problems with analytical treatment for comparison to the non-analytical approach in next chapters. The chapter is organized as follows: first, it is defined and discussed basic objects related to classification, the pattern, and class, as well as the space in which patterns are contained called the feature space. Next, the framework of discriminant functions is introduced in order to define classifiers formally. Finally, is presented the classifier's validation which allows the multiple classifiers comparison. All codes and plots were written in R language and are available at https://github.com/yurier/distance-based-alzheimer.

## 2.1 Basic definitions

A classifier basically assigns a pattern given type into a class. There is no restrained definition for the term *pattern* [32]. It can be, for instance, a real number, a categorical value, an ordinal value, or an image which can be represented as real vectors and even more structured elements are possible, as graphs or temporal series. Here, the term pattern is limited to be a real vector due to the range of objects which can be understood as patterns. Thus, classifiers receive patterns as inputs and categorical values as outputs, the classes. These constraints on pattern and class definitions are not detrimental and will be enough to our purposes. Next are presented definitions for class, pattern, and classifier as given in Devroye *et al.* [19], in order to introduce the functional relationship that will be called classifier. Assuming that any pattern can be assigned in one class, we define the class space as follows.

**Definition 1. (Class space)**

*Any pattern can be assigned to only one of m classes,* $\{\omega_j\}_{j=1}^m$*, which belongs to the finite set* $\Omega = \{\omega_1, ..., \omega_m\}$ *called class space.*

As defined in [19], an observation or pattern is a collection of numerical measurements. Formally, it is an $n$-dimensional real vector in which each dimension is called *feature*. As discussed in the introduction, in medical related classification problems, features would be called for instance biomarkers. Patterns in medical datasets often represent patients with $n$-dimensional biomarkers measures, for instance: weight, height, gender, etc. Alternatively, in single individual studies, patterns can be images' voxels/pixels accounting for different biomarkers, for instance: voxel perfusion, voxel grey-white matter ratio, etc.

Features used in the main datasets on this work are proteomic measures from cerebral spinal fluid (CSF), cognitive measures from neuropsychological tests, positronic emission tomography (PET) measures. Furthermore, constructed and extracted features have been a potential source to understand diseases behavior since new biomarkers are obtained from such processes. For instance, Lopez-de Ipiña K. *et al.* [53] were able to differentiate individuals with Alzheimer's disease from healthy by applying feature extraction techniques on spontaneous speech to obtain discriminant features. Another example can be seen in Khazaee A. *et al.* [43] in which individuals with Alzheimer's disease were differentiated perfectly from healthy group using features obtained by graph theoretical tools through resting state fMRI (functional magnetic resonance imaging). Thus, since features are used to construct the classifier, they play an important role In the performance and classifier's designing. The feature space is defined as follows,

**Definition 2. (Feature space)**

*The feature space* $\mathcal{F}$ *is a n-dimensional space in which patterns are characterized.*

**Property 2.1. (Pattern and class association)**
*Any pattern* $x = (x_1, ..., x_n) \in \mathcal{F}$ *can be associated to a class* $\omega_j$*, such that* $\omega_j \in \Omega$*. This is* $\forall x \in \mathcal{F}$*,* $\exists\ \omega_j \in \Omega$*.*

In supervised learning is not possible to infer if $\Omega$ is suitable to represent the classes of given classification problem through the pairs $\{x \in \mathcal{F}, \omega \in \Omega\}$. Conversely, in unsupervised learning, which deals with unlabeled patterns, the main goal is to seek for a class space for the patterns by clustering them. Recently, unsupervised learning has been used on interesting neuroscience related applications. For instance, similar to the blind source separation problem, spike sorting techniques [50] which have been invaluable for brain-machine interfaces [49] since they are able to cluster spikes based on its shape and give a class structure to the extracellularly recorded neurons [50]. Turning back to supervised learning, let's define the classifier as a functional relationship between the feature space and the class space as follows,

**Definition 3. (Classifier)**
*A classifier* $\mathcal{C} : \mathcal{F}, \rightarrow \Omega$ *is a surjective function such that,*

$$\forall x \in \mathcal{F} \quad \exists\ \omega_j \in \Omega \quad \textit{with} \quad \mathcal{C}(x) = \omega_j.$$

The classifier $\mathcal{C}$ depends on the feature and the class space. In multi-category classification [65], a classifier $\mathcal{C}$ can be represented as a vector of $m$ binary classifiers $\{c_i\}_{i=1}^{m}$, where $c_i(x) = 1$ if $x$ belongs to class $\omega_i \in \Omega$ and $c_i(x) = 0$ otherwise, that is $\mathcal{C}(x) = \{c_1(x), ..., c_m(x)\}$ [19]. An union of such binary classifiers form a well-defined classifier $\mathcal{C}$ if,

$$\forall x \in \mathcal{F},\ c_i(x) = 1 \text{ and } c_j(x) = 0 \text{ for } j \neq i,$$

that is, for $1 \leq i \neq j \leq m$ follows that,

$$\bigcup_{i=1}^{m} \{x : c_i(x) = 1\} = \mathcal{F} \text{ and } \{x : c_i(x) = 1\} \bigcap \{x : c_j(x) = 1\} = \emptyset$$

This simple association between patterns and classes is not a satisfactory definition because any well-defined association between patterns and classes is a valid classifier. Thus, it is reasonable to think how a classifier can be designed for an optimal error. That is to say, a classifier with minimal probability of error. This problem is also known as the *Bayes problem* [19]. Since data generation underlying processes are unknown in real-world classification problems, next examples illustrate how one can use class distributions to solve artificially the Bayes problem.

Prior to discussing the example, it is relevant to remark all AD features and its importance will only be discussed in five. Given two classes representing the diagnosis labels AD and CN as given in chapter six by name of *proteom* dataset, let's suppose to have complete knowledge of these class distributions with respect to the biomarker $A\beta_{1-42}$ (ABETA). The problem here is to classify an unclassified pattern using the previously labeled data, whereas the class space is given by $\Omega = \{\omega_1 = AD, \omega_2 = CN\}$ and the feature space, $\mathcal{F}$, is one-dimensional measure represented by ABETA. Furthermore, let's assume that distribution of each class is completely explained by Gaussians, $\mathcal{N}(\mu_{CN} = 196.67, \sigma_{CN} = 49.96)$ and $\mathcal{N}(\mu_{AD} = 133.20, \sigma_{AD} = 35.84)$ as depicted in figure 2.1 with priors $p(AD) = 39.68\%$ and $p(CN) = 60.32\%$. The line in figure 2.1 represents the value in which both Gaussians distributions are equally likely.

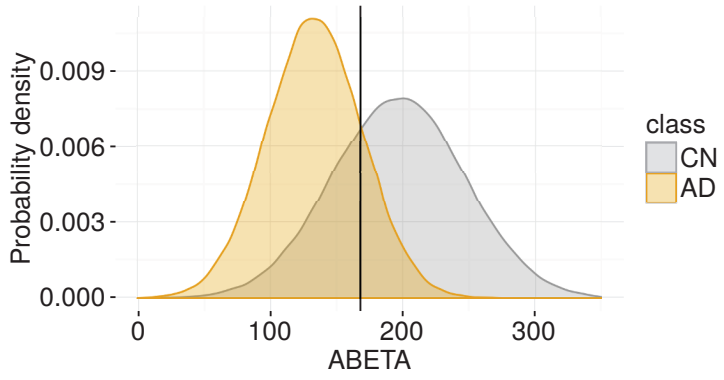



Figure 2.1 Gaussians of both classes and line dividing the domain in two decision regions.

Using the line as reference it is possible to assign an unclassified pattern into one class only and a well-defined classifier is achieved. Given that $\forall x \in \mathcal{F}$, $\exists$ $\omega \in \Omega$ the classifier can be written as follows,

$$\mathcal{C}(x) = \begin{cases} \omega_1 & \text{if} \quad x \leq 168.65, \\ \omega_2 & \text{otherwise.} \end{cases}$$

Another solution for the intersection between two Gaussians is $x = -36.84$ however since $x$ is positive it will be ignored. The calculations to obtain the line in figure 2.1 will be explained in next section for multivariate Gaussians. In figure 2.1 notice misclassification errors are unavoidable even for the Bayes classifier which addresses the optimal error solution (with assumptions on distribution shapes). Further sections present examples of a multiclass bivariate problem solved by Bayes classifier. A framework to describe all classifiers in this thesis is presented next. It is known as discriminant functions or the set of rules which compose a classifier [19].

## 2.2 Discriminant functions

An example of a simple classifier is the *maximum a posteriori* (MAP) classifier. The MAP uses the posterior probability for classification, in such way that selected class maximizes the class conditional posterior probability. Let $\tilde{\omega}_j$ denotes the class estimated by the classifier, the MAP classifier is defined as follows,

**Definition 4. (MAP classifier)**
*Let $p(w_j|x)$ be the posterior probability of the pattern $x \in \mathcal{F}$ belong to the class $w_j \in \Omega := \{\omega_1, ..., \omega_m\}$. The MAP classifier is defined as a function that maximizes the posterior probability of classification given the class,*

$$\tilde{\omega}_k = \underset{\omega \in \Omega}{\arg\max} \left\{ p(\omega_1|x), p(\omega_2|x), ..., p(\omega_m|x) \right\},$$

*this is,*

$$\tilde{\omega}_k = \underset{\omega \in \Omega}{\operatorname{argmax}}\ p(\omega|x).$$

*The highest probability defines the classification assignment of* $x$ *to* $\tilde{\omega}_k$ *since* $p(\omega_k|x) \geq p(\omega_j|x)$ *for* $j = 1, .., k-1, k+1, ..., m$.

Any classifier, e.g. MAP classifier, allows feature space be separated in different regions. These regions in the feature space are called decision regions. Basically, all patterns containing in a given region space belongs to the same class. Also, decision regions are limited by its borders which are called decision boundaries. Decision boundaries for the MAP classifier are precisely where class posterior probabilities are equally likely, this is,

$$p(\omega_j|x) = p(\omega_i|x) \quad \forall x \ \in \mathcal{F} \quad \forall\ w_j \neq w_i \in \Omega.$$

It is uncommon in real-world problems patterns laying exactly over the boundary between classes. Decision boundary importance is mainly related to the interpretation of how the feature space will be divided in the class space. This division allows using a classifier as an interpretative tool for observing patterns position in the feature spaces. Next section introduces discriminant functions which generalize MAP classifier.

Discriminant functions are able to describe non-probabilistic and probabilistic classifiers approaches [12] and are defined as follow.

**Definition 5. (Discriminant functions)**
*A discriminant function is a set of functions* $f_j(x) : \mathcal{F} \to \mathbb{R}$ *for* $j = 1, ..., m$ *in which if* $x$ *belongs to class* $\omega_j \in \Omega$ *then*

$$f_j(x) \geq f_i(x) \quad \forall x \in R_j\ \forall i \neq j$$

For instance, if a pattern $x$ belongs to a class $\omega_j$, the discriminant function, $f_j$, has the highest value among discriminant functions. The set of patterns

which assume the highest value for a given discriminant function $f_j$ defines a region in feature space called decision region, denoted by $R_j \subset \mathcal{F}$. That is to say all patterns in $R_j$ are associated with the class $\omega_j$.

Classification by means of the discriminant function is done by maximizing its argument related to the class. Different from the MAP classifier it does not necessarily have a probability function to be maximized, that is,

$$\tilde{\omega}_j = \underset{j=1,\ldots,m}{\operatorname{argmax}} \quad f_j(x) \tag{2.1}$$

A discriminant functions set must create the same decision regions under a monotonic function composition. This property allows classification problems which have the classes described by parametric expressions (e.g. distributions Gamma or Gaussian) to be solved analytically.

**Property 2.2. (Monotonic function composition)**
*Let $h(x)$ be a monotonic function and $\{f_j\}_{j=1}^m$ the set of discriminant functions, then*

$$h(f_j(x)) \geq h(f_i(x)) \quad \forall x \in R_j \quad \forall i \neq j.$$

Suppose one wants the class conditional probability as a discriminant function as used in MAP classifier, in order to this, consider Bayes theorem [57],

$$p(\omega_k|x) = \frac{p(\omega_k, x)}{p(x)} = \frac{p(x|\omega_k)p(\omega_k)}{p(x)}, \tag{2.2}$$

notice that equation (2.1) states the function from discriminant function set which achieves the highest value assigns the associated class to a given pattern $x$. Thus, using equation (2.2) one can write an inequality between class conditional probabilities,

$$\begin{aligned} p(\omega_i|x) &\geq p(\omega_j|x) \\ \frac{p(x|\omega_i)p(\omega_i)}{p(x)} &\geq \frac{p(x|\omega_j)p(\omega_j)}{p(x)}. \end{aligned} \tag{2.3}$$

Assuming $p(x) > 0$, the value $p(x)$ in equation (2.3) does not play any role in the class assignment and can be simplified. Notice that the class prior

probability $p(\omega)$ affects classification, by that in next examples and for the sake of simplicity, let's assume the same prior for all classes. Since classes are proportional (equal priors) the parametric description of all class posterior probability, $\{p(x|\omega_i)\}_{i=1}^{m}$, are enough to describe the classifier. Applying property 2.2 in equation (2.3) and supposing equally likely classes we have,

$$\begin{aligned} log(p(x|\omega_i)p(\omega_i)) &\geq log(p(x|\omega_j)p(\omega_j)) \\ log(p(x|\omega_i)) + log(p(\omega_i)) &\geq log(p(x|\omega_j)) + log(p(\omega_j)) \\ log(p(x|\omega_i)) &\geq log(p(x|\omega_j)). \end{aligned} \tag{2.4}$$

Equation (2.4) shows under the former assumptions of equally likely classes the classification rule depends only on the class conditional probability. Next, let's give an example of a parametric method for a binary classification problem which can be solved analytically.

Suppose the problem of classifying patients with AD. For simplicity, let's classify them as AD or healthy CN as shown figure 2.1. Also, suppose that these classes are bivariate Gaussians. The classification problems between these classes will be performed using biomarkers ABETA and p-tau$_{181}$ (PTAU) as described in *proteom* dataset. Let's assume complete knowledge of the class conditional probability for classes AD and CN which are given by $f_{AD}(x) \sim \mathcal{N}(\mu_{AD}, \Sigma_{AD})$ and $f_{CN}(x) \sim \mathcal{N}(\mu_{CN}, \Sigma_{CN})$, respectively. The terms $\mu$ and $\Sigma$ are the mean and covariance matrices, respectively. Assuming equal priors the class conditional probabilities for AD and CN are given by the Gaussians distributions as follow,

$$\begin{aligned} p(x|AD) = f_{AD}(x) =& \frac{1}{2\pi\sqrt{\det(\Sigma_{AD})}} \exp(-\frac{1}{2}(x-\mu_{AD})^T\Sigma_{AD}^{-1}(x-\mu_{AD})), \\ p(x|CN) = f_{CN}(x) =& \frac{1}{2\pi\sqrt{\det(\Sigma_{CN})}} \exp(-\frac{1}{2}(x-\mu_{CN})^T\Sigma_{CN}^{-1}(x-\mu_{CN})). \end{aligned}$$

Solving $f_{AD}(x) = f_{CN}(x)$ analytically one can determine the decision boundary and decision regions. Applying natural logarithm to the last equation (by the property 2.2 ), this is, $ln(f_{AD}(x)) = ln(f_{CN}(x))$, is possible to write,

$$-(x-\mu_{AD})^T\Sigma_{AD}^{-1}(x-\mu_{AD}) = ln\Big(\frac{|\Sigma_{AD}|}{|\Sigma_{CN}|}\Big) - (x-\mu_{CN})^T\Sigma_{CN}^{-1}(x-\mu_{CN}). \quad (2.5)$$

Equation (2.5) is quadratic and has solution if the covariance matrices $\Sigma_{AD}$ and $\Sigma_{CN}$ are positive-definite which imply to have inverse [28]. That is, if all eigenvalues are strictly positive the solution exists [45], this is required since all covariances matrices are positive semi-definite. The parameters estimated from dataset AD1 are,

$$\Sigma_{AD} = \begin{bmatrix} 1517.7610 & -373.7946 \\ -373.7946 & 4258.3641 \end{bmatrix} \quad \Sigma_{CN} = \begin{bmatrix} 2575.088 & -354.219 \\ -354.219 & 1102.495 \end{bmatrix}$$

$$\mu_{AD} = \begin{bmatrix} 135.7225 \\ 133.8363 \end{bmatrix} \quad \mu_{CN} = \begin{bmatrix} 194.96282 \\ 67.28462 \end{bmatrix}$$

The decision boundary is obtained from equation (2.5) as the distribution functions intersections. Figure 2.2 depicts the decision boundary that divides the feature space into decision regions.

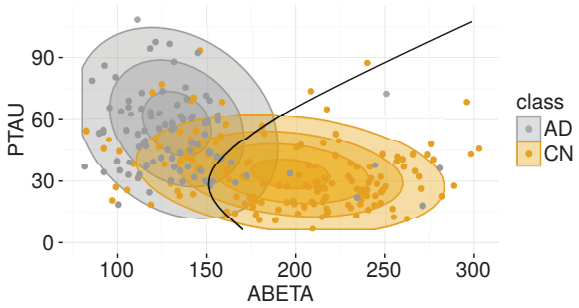



Figure 2.2 Gaussian distribution for AD e CN. Black solid line represents the decision boundary between the two classes.

Figure 2.2 decision boundary's shape is a parabola which depends on the estimated covariance matrix and the mean. A full description of how covariance matrices shapes decision regions can be seen in [57]. A parametric description of data has its advantages and allows the analytical treatment of the decision boundaries, however it makes assumptions on the probability distribution model. Alternatively,

chapter three is focused on model-free approaches to the classification problem in which distribution models are not assumed.

Figure 2.2 one can notice there is the possibility for some training patterns being in the opposite decision region accounting as errors. This is due to the high overlapping between classes AD and CN distributions. Using the Bhattacharyya coefficient [11] obtained through the Bhattacharyya distance one can measure the overlapping between AD and CN which is 0.61 for the *proteom* dataset (assuming Gaussian distributions), where 0 means classes totally separated and 1 means complete overlapped. Therefore, more than half of the patterns are on the wrong side of the decision region. Furthermore, an intermediary class of Alzheimer spectra will be included increasing the overlapping between class distributions. This problem is the main reason why the majority of papers on Alzheimer's disease classification problems deal only with extreme classes (e.g AD and CN) or binary [25].

## 2.3 Validation

A classifier $\mathcal{C}$ is obtained using a *training set* and creates a mapping between the feature space and the class space. One can ask how good this mapping is when compared with another classifier? This question will be addressed using misclassification error measures estimated from an unseen set of patterns called *test set*. This ensures that we are simulating classifier's interaction with real-world data variability. Thus, the validation design plays an important role in the classification performance and classification rate due to data size issues as claimed by [48].

As argued by Devroye *et al.* [19] a good classifier has a property called consistency, which means the more data is available the nearer conditional probability error expectation will be to its optimal value, in the convergence sense. The conditional probability of error can be defined as follows,

**Definition 6. (Conditional probability of error)**

*Consider a training set with n samples,* $T := \{x_i, \omega\}_{i=1}^n$ *and an unclassified pattern* $x_*$ *belonging to the class* $\omega_*$*, then the conditional error probability for the classifier* $\mathcal{C}$ *trained with* $T$*, denoted by* $L_n(\mathcal{C})$*, is expressed by,*

$$L_n(\mathcal{C}) = p(\mathcal{C}(x_*) \neq \omega_* | T).$$

*where* $x_*$ *belongs to the class* $\omega_*$*.*

Therefore, using definition 6, the consistency can be written as follows,

$$\lim_{n\to\infty} E(L_n(\mathcal{C})) = L(\mathcal{C}^*),$$

where $\mathcal{C}^*$ is the Bayes optimal classifier. A consistent classifier is a motivation to get more data since it means the best possible discriminant rule or set of discriminant function [19]. Furthermore, a distance-based classifier to be defined next chapter was essential for early development of consistency concept. For convergence conditions of a wide type of multiclass classifiers see [79]. Since misclassification error is the main concern to understand how good a classifier is, let's define the validation procedure to assess, the multiclass problems treated in this thesis.

### 2.3.1 Confusion matrix

The validation allows one to compare classifiers using a scalar value. There are many scalar measures of misclassification available [72], for instance, accuracy, F-score, G-score, sensitivity, sensibility and so on. Such measures of errors are designed to the binary classification problems, e.g. positive vs. negative classes, and not account for multiclass interactions [26] which will be explored further chapters. Multiclass validation naturally must account for all class assignments errors and would be estimated using a *confusion matrix*, or contingency table, defined as follows.

**Definition 7. (Confusion Matrix)**

*Let* $m$ *be the number of classes in* $\Omega$*,* $\hat{\omega}_j$ *be the true pattern's for a given pattern and*

*$\mathcal{C}$ a hypothetical classifier. Each element of confusion matrix $\mathcal{P} \in \mathbb{R}^m \times \mathbb{R}^m$ is given by,*

$$\mathcal{P}_{j,i} = p(\mathcal{C}(x) = \tilde{\omega}_j | \hat{\omega}_i) = p(\tilde{\omega}_j | \hat{\omega}_i),$$

*that is, the probability of a pattern from class $\omega_i$ be classified into the class $\omega_j$.*

Another way to represent the confusion matrix is showing only the number of wrongly classified elements in each class without any probabilistic measure. However, this simplification will be done in order to use macro-averaged measures which are useful to assess imbalanced classes [72]. Conversely, micro-averaged measures do not treat all classes equally, in turn, it depends on the major classes to achieve high results [72]. By definition 7 the confusion matrix can be seen as a stochastic matrix by lines since the sum of rows are equal to 1,

$$\sum_{j=1}^{m} p(\omega_j | \hat{\omega}_i) = 1 \text{ for } i = 1, ..., m.$$

Using the confusion matrix one can define measures to multiclass classification problem. The ideal situation is the confusion matrix being equal to the identity matrix. In this work empirical evaluation will be used to assess a classifier's confusion matrix, alternatively, one can use analytical estimation error techniques as listed in [76]. Let's use an invariant comparable scalar measure called *precision* as the evaluation criterion. However, a comparable measure can be any function transforming the confusion matrix $\mathcal{P} \in \mathbb{R}^{m^2}$ in a scalar.

**Definition 8. (Macro-averaged precision)**
*The precision measure named as "macro-average precision" [72] is equal to the confusion matrix trace, or the average of correct classified class probabilities,*

$$precision = \frac{\sum_{i=1}^{m} p(\tilde{\omega}_i | \hat{\omega}_i)}{m}$$

A review of relevant measures including invariant analysis is given in [72]. In order to illustrate the confusion matrices' role, consider the following multiclass problem solved by a parametric method (which assumes distribution models

controlled by parameters) [57]. Suppose a trichotomous classification problem between the AD classes (CN, MCI, AD) as defined in *proteom* dataset. The classes distributions in figure 2.3 (left) depicts the overlapping between AD classes, in which the Bhattacharyya coefficient (assuming gaussianity) between classes is 0.61, 0.90 and 0.85, respectively for CN and AD, AD and MCI, AD and MCI.

For this example, it is assumed that each class is well represented by a Gaussian distribution. Seemingly to the figure 2.2 the Gaussian distributions are $f_{AD}(x) \sim \mathcal{N}(\mu_{AD}, \Sigma_{AD})$, $f_{MCI}(x) \sim \mathcal{N}(\mu_{MCI}, \Sigma_{MCI})$ and $f_{CN}(x) \sim \mathcal{N}(\mu_{CN}, \Sigma_{CN})$. In order to show more cases of how the covariance affects the decision boundaries, the covariances matrices are changed to be a scaled identity matrices, as shown by distributions depicted in figure 2.3. The changes are $\Sigma_{CN} = \lambda_{CN} * I$, $\Sigma_{MCI} = \lambda_{MCI} * I$ and $\Sigma_{AD} = \lambda_{AD} * I$ used to generate the training set, where $I$ is the identity matrix and $\lambda$ is the average of singular values in singular value decomposition (SVD) technique [28]. This strategy allows to finding the decision boundary by pair of classes, by solving the following equations $f_{AD} = f_{MCI}$, $f_{CN} = f_{MCI}$ and $f_{AD} = f_{CN}$. Solving $f_{AD} = f_{MCI}$ we have,

$$
\begin{aligned}
ln(f_{AD}) &= ln(f_{MCI}) \\
ln\Big(\frac{1}{2\pi\sqrt{|\Sigma_{AD}|}}\Big) - \frac{1}{2}(x-\mu_{AD})^T\Sigma_{AD}^{-1}(x-\mu_{AD}) &= ln\Big(\frac{1}{2\pi\sqrt{|\Sigma_{MCI}|}}\Big) - \frac{1}{2}(x-\mu_{MCI})^T\Sigma_{MCI}^{-1}(x-\mu_{MCI}) \\
-\sigma_{AD}^{-1}(x^Tx - 2\mu_{AD}^Tx + \mu_{AD}^T\mu_{AD}) &= ln\Big(\frac{\sigma_{AD}}{\sigma_{MCI}}\Big) - \sigma_{MCI}^{-1}(x^Tx - 2\mu_{MCI}^Tx + \mu_{MCI}^T\mu_{MCI})
\end{aligned}
$$

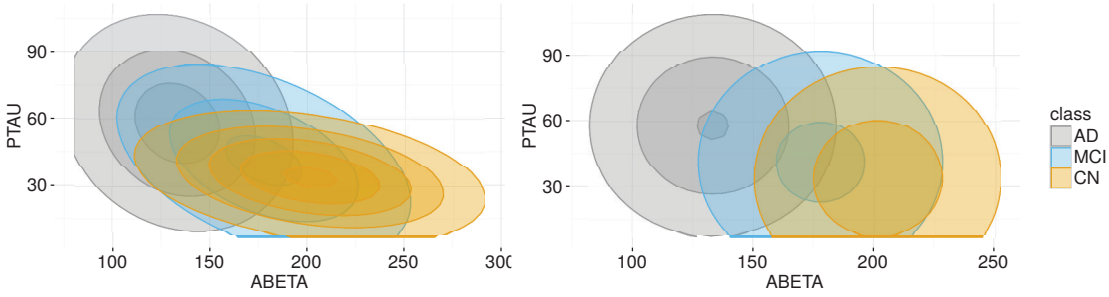



Figure 2.3 At left, original Gaussian distributions for all the three classes. At right, modified covariances matrices that will describe the three classes. Points are the Gaussian mean.

Following this same reasoning is possible to solve the remaining two equations,

$$-\sigma_{AD}^{-1}(x^T x - 2\mu_{AD}^T x + \mu_{AD}^T \mu_{AD}) = ln\Big(\frac{\sigma_{AD}}{\sigma_{CN}}\Big) - \sigma_{CN}^{-1}(x^T x - 2\mu_{CN}^T x + \mu_{CN}^T \mu_{CN}),$$
$$-\sigma_{CN}^{-1}(x^T x - 2\mu_{CN}^T x + \mu_{CN}^T \mu_{CN}) = ln\Big(\frac{\sigma_{CN}}{\sigma_{MCI}}\Big) - \sigma_{MCI}^{-1}(x^T x - 2\mu_{MCI}^T x + \mu_{MCI}^T \mu_{MCI}).$$

For evaluation purposes, original data from *proteom* dataset was divided into 75% for the training set and 25% for the test set. Figure 2.4 depicts how the decision boundaries divide up the feature space, how the training set is classified and where is the true classification of the training set. This plot goal is to show how many training patterns are misclassified in training step. As argued [33] due to over-fitting the training set classification rate is not a reliable measure to evaluate classifier error. Figure 2.5 shows how the test set was classified.

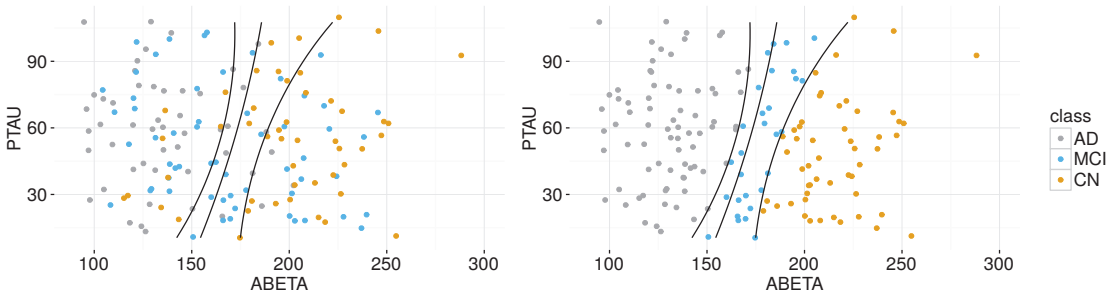



Figure 2.4 At left the decision boundary and the training set and at right the decision boundary and the classification outcome for the training set.

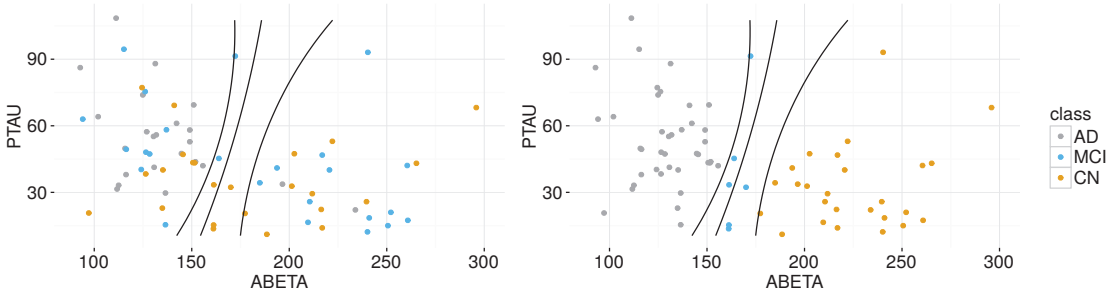



Figure 2.5 At left, the decision boundary and the true class of test set. At right, the classifier outcome for the test set.

Figure 2.6 shows the confusion matrices for different training set sizes 75%, 50% and 25% of the dataset. Each row entry is the probability of classification error for given a class correct. Notice errors in classification of MCI class are higher due to the strong overlap. Using 74% of data as training set misclassification error for intermediary class MCI is $P(AD|\widehat{MCI}) = 88.46\%$ and $P(CN|\widehat{MCI}) = 50\%$ and precision is $P(MCI|\widehat{MCI}) = 7.69\%$. Errors in opposite classes (AD and CN) are less frequent since the intermediary class was generated with equal prior probability relative to the minor class in *proteom* dataset. However, using non-generated data, as it will be presented in next chapter, majority class (MCI) promotes errors in minor classes [35]. The confusion matrices presented in figure 2.6 do not provide the dispersion measures of presented method. To obtain a reliable measure of dispersion one can use $k$-fold cross-validation (CV) [12] and assess the precision from averaged confusion matrices as will be done in next chapter.

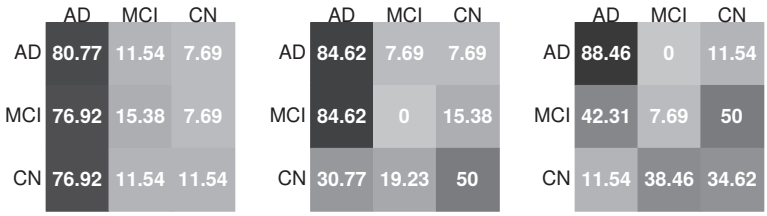



Figure 2.6 From left to right, confusion matrices for training set sizes of 75%, 50% and 25% of dataset, respectively.

The precision of confusion matrices (the average of correctly classified probabilities by class) in figure 2.6 is 35.90%, 44.87% and 43.59%, respectively. The confusion matrices of this same problem will be used as an example for the distance-based classifier in next chapter.

In summary, an important aspect of this chapter are the simulations which give us a first glance of AD classification problem difficulty. Also, basic definitions to develop this thesis are settled. Briefly, the pattern is considered a real vector and the classifier a function composed of discriminant functions. Since the classifier

of interest in next chapter uses all-versus-all strategy for classification the validatory mechanism uses confusion matrix precision instead of binarized strategies, e.g. receiver operator characteristic (ROC) analysis [23]. The parametric classification approach presented in this chapter uses parameters estimated from data to make a decision, see [12] for more parametric methods. Next chapter shows a model-free distance-based classifier which is equivalent to the parametric approach in the sense of class posterior estimation.

# 3 DISTANCE-BASED CLASSIFIER

## 3.1 Introduction

Distance-based classifiers usually incorporate distance functions induced by metrics (e.g. *p-norm metric* or *pseudo-metric*) to determine how similar two patterns are [71]. Alternatively, distance measures participate in the classifier's formulation by penalizing hyperparameters, error estimation, and optimization [32]. In supervised and unsupervised learning classifiers dependence of distance functions are critical for its performance [84]. Thus, the concept of distance-based classifiers extends to a large class of methods, for instance, *support vector machines* (SVM) [15], *artificial neural networks* (ANN) [34], *k-means* [32], *nearest-neighbor* (NN) based algorithms [6].

Exemplifying, SVM is a binary classifier used in many domains which the main goal is to maximize patterns distances from a hyperplane while minimizing misclassification by penalizing linear models weights [32]. Distance-based modifications in SVM include, for instance, *elastic-net* [32] and *lasso* [41] weight penalization in order to obtain faster algorithms or to achieve a specific property, e.g. a given number of non-zero weights [32]. Distance functions participate in ANN supervised and unsupervised learning, for instance, in error adaptation mechanisms (e.g. error back-propagation algorithm) and clustering task orientated ANN (e.g. self-organized maps - SOM), respectively [34]. For instance, a distance modification in ANN is presented by Duch and Adamczak [21] in which non-Euclidean distances are used to provide flexible decision borders for *multi-layer perception* (MLP).

Since the distance-based classifier concept is broad, this thesis focuses on classifiers using distance functions as similarity measures. For instance, the NN algorithm which can be defined straightforwardly: an unsigned query pattern is

classified as belonging to the same class of the nearest labeled pattern [17]. The NN classifier uses only one nearest neighbor to evaluate a query pattern's class. Alternatively, a natural extension which uses more than one neighbor is the k-nearest neighbors (kNN), as well called the majority voting rule [17]. Here, let's study how distance function and normalization would improve kNN precision. Pragmatically, the kNN is a benchmark comparison for machine learning articles and its data pre-processing extensions are competing for state-of-art algorithms when dealing with data issues [47]. Challenging data issues [85], as imbalanced classes [16] and missing data [33], finds state of art solutions in algorithms based on the kNN strategy to prevent decreased classification performance [35]. The next chapter will discuss such algorithms to treat data imbalance.

As claimed in [32], the kNN classifier would be an insecure choice when dealing with high-dimensional feature spaces producing an inferior performance. Such effects are mainly due to the *curse of dimensionality* which causes feature space's sparsity [7]. Since kNN is dangerously simple, an efficient usage goes through the knowledge of what causes its degradation. For instance, as shown by Boiman O. [14], for image related classification task, known by its high-dimensionality, kNN's deteriorated performance can be avoided to excel top leading learning-based image classifiers when used with image descriptors. Hence, to avoid the curse of dimensionality effect in kNN one may find a low-dimensional representation of data or to reduce data sparsity effect by modifying kNN. This chapter will focus on kNN modifications, particularly using Minkowski distance as the similarity measure and different normalization schemes as suggested by [2].

In the following section, the kNN is formally defined by means of the discriminant functions framework. Results are validated using confusion matrices as defined earlier and compared to multiple datasets. AD classification problems in this chapter are used only to illustrate concepts and will be discussed in chapter four.

## 3.2 k-Nearest Neighbor

In the last chapter instances of parametric methods were presented in order to illustrate the Bayes classifier, $L(\mathcal{C}^*)$. However, Bayes classifier requires data complete knowledge on the underlying class distribution. Parametric classifiers, which approximate the distribution model by parameters estimation, are also known as eager learning algorithms. Alternatively, the kNN to be discussed here do not rely on any distribution model. Classifiers using the data structure to describe class distributions are called non-parametric methods. In the non-parametric classification context, analytic manipulation is not possible when obtaining explicitly the decision regions for real-world problems.

Seemingly to the NN classifier henceforward 1NN, the kNN is data dependent [19]. That is, when slightly (even one pattern) different training sets are used to generate the classifier, different classification mappings should arise. This happens since kNN is an instance-based classifier, which requires all dataset distances comparison to make an assignment [39]. Conversely, the parametric methods need only a finite set of parameters to make an assignment. According to [3] classifiers which store data to create the mapping are called: lazy-learning, case-based, instance-based, memory-based [3].

The information to assign an unclassified pattern is contained in its vicinity. Precisely, at least half of the classification information is contained in the nearest neighbor for the infinity-sample case as proved by Cover and Hart [17]. This fact is used to demonstrate the Cover-Hart inequality, a relationship between NN and Bayes optimal error. Considering the conditional probability of error as in definition 6 from chapter 2, in a separable metric space, for any distribution and $n$ samples, we have,

$$\limsup_{n\to\infty} E(L_{NN}) \leq L(\mathcal{C}^*),$$

that is,

$$L(\mathcal{C}^*) \geq L_{NN} \geq 2L(\mathcal{C}^*).$$

Using such estimation with 1NN is possible to obtain information on the optimal solution generally unavailable. Back to kNN formulation, let's define the vicinity of kNN ($k$-neighborhood) as follows.

**Definition 9. ($k$-neighborhood)**
*A $k$-neighborhood of a pattern $x_*$, denoted as $N(x_*, k)$, is a limited sequence of the first $k$ nearest patterns, $N(x_*, k) := \{x_{(1)}, x_{(2)}, ..., x_{(k)}\}$, for a given metric measure $d(.,.)$ over the feature space $\mathcal{F}$.*

More explicitly, given the set of all labeled patterns denoted as $T$, namely, the training set, the $k$-neighborhood of an unclassified pattern $x_*$ is given by $\{x_{(1)}, x_{(2)}, ..., x_{(k)}\} = N(x_*, k) \subset T$. In which $x_{(1)}$ is such that $x_{(1)} = \underset{x \in T}{\operatorname{argmin}} \quad d(x, x_*)$, the closest pattern in $T$ given by the distance function $d(.,.)$. The remaining neighbors $\{x_{(i)}\}_{i=2}^{k}$ are given by,

$$x_{(i)} = \underset{x \in T \setminus \{x_{(j)}\}_{j=1}^{i-1}}{\operatorname{argmin}} d(x, x_*).$$

Each pattern $x$ in $T$ has associated a class $\omega \in \Omega$. Since for each $x_{(i)} \in N(x_*, k)$ we have a class $\omega_{(i)} \in \Omega = \{\omega_1, ..., \omega_m\}$ the classification of an unclassified pattern $x_*$ is obtained by the number of majority classes among the $k$-neighborhood set, $N(x_*, k)$. A critical issue for classification rate is the distance function defined to create the $k$-neighborhood, further examples explore how one can take advantages of metric modification. The kNN classifier can be defined using the discriminant functions as follows [6].

**Definition 10. ($k$-Nearest neighbors)**
*Let $N(x_*, k) := \{x_{(i)}\}_{i=1}^{k}$ be the set of $k$ nearest patterns from $x_*$, in which each $x_{(i)}$ is associated to a class $\omega_{(i)} \in \Omega := \{\omega_1, ..., \omega_m\}$. The estimated class, $\tilde{\omega}$, for the unclassified pattern $x_*$ is given by,*

$$\tilde{\omega} = \underset{\omega \in \Omega}{\operatorname{argmax}} \sum_{x_{(i)} \in N(k, x_*)} \delta(\omega, \omega_{(i)}).$$

The kNN formulation allows one to deal with multiclass classification problems naturally using the all-versus-all strategy in which all classes are mutually affected at once. Conversely, other classifiers dealing with non-binary classification problems require adaptations by means of one-against-all or one-against-one strategies [58]. Examples are depicted in figure 2.3 in the previous chapter. Thus, to solve an $n$-class problem using binary classifiers (e.g. SVM), $\frac{n(n-1)}{2}$ rules are required to build a multiclass classifier. For further discussion on binary adaptations to multiclass problems see [58].

### 3.2.1 Approximating class posterior probabilities

Bishop [12] lists three frameworks to construct classifiers organized by how the discriminant functions are applied. The *generative models* use Bayes theorem to estimate the posterior class probabilities, $p(\omega_i|x)$. In order to this, a parametric distribution for the class conditional probabilities, $p(x|\omega_i)$, is searched and the class prior probabilities, $p(\omega_i)$ are estimated as usual [12]. This approach does not require to carry out all dataset to estimate classifier, only a set model parameters that would generate the classes. Examples of algorithms commonly used in generative modeling are *maximum likelihood* and *expectation maximization*.

Alternatively, the *discriminative models* do not need to estimate the class conditional probability neither the class prior. Instead, the posterior class probabilities for each class are estimated directly as an inference problem. Instances of discriminative modeling are NN-based classifiers treated in this thesis. When compared to generative, the discriminative approaches are more precise in discrimination task [12].

The third framework is the case in which probabilities play no role, being neither generative nor discriminative. However, discriminative functions framework are employed to define the decision regions in this case. A widely known

example is the SVM classifier. For a detailed comparison of the listed approaches see [42].

Considering $\delta(.,.)$ as Kronecker delta let's show that kNN classifier locally estimates the posterior class probability. kNN classifies a given query pattern, $x_*$, as follows,

$$\tilde{\omega} = \underset{\omega \in \Omega}{\operatorname{argmax}} \sum_{x_{(i)} \in N(k, x_*)} \delta(\omega, \omega_{(i)}) = C(x). \tag{3.1}$$

Since $\{x_{(i)}\}_{i=1}^{k} \in N(k, x_*)$ are the nearest neighbors to the unclassified pattern $x_*$ and each nearest neighbor is associated to a class $\omega_{(i)} \in \Omega := \{\omega_1, ..., \omega_c\}$, one can rewrite equation (3.1) as,

$$\tilde{\omega} = \underset{\omega \in \Omega}{\operatorname{argmax}} \left\{ \sum_{x_{(i)} \in N(x_*, k)} \delta(\omega, \omega_1), ..., \sum_{x_{(i)} \in N(x_*, k)} \delta(\omega, \omega_c) \right\}. \tag{3.2}$$

Using the monotonic property of discriminant functions to equation (3.2) and dividing it by $k$ we have,

$$\tilde{\omega} = \underset{\omega \in \Omega}{\operatorname{argmax}} \left\{ \sum_{x_{(i)} \in N(x_*, k)} \frac{\delta(\omega, \omega_1)}{k}, ..., \sum_{x_{(i)} \in N(x_*, k)} \frac{\delta(\omega, \omega_c)}{k} \right\}. \tag{3.3}$$

Thus, each term in equation (3.3) is the probability of class assignment given a training set $T$ and the size of $k$-neighborhood,

$$p(\omega_j | x_*, T, k) = \sum_{x_{(i)} \in N(x_*, k)} \frac{\delta(\omega_j, \omega_i)}{k}, \qquad \text{for} \quad j = 1, ..., c. \tag{3.4}$$

By means of equation (3.4) the kNN's discrimination function can be written as,

$$\tilde{\omega} = \underset{\omega \in \Omega}{\operatorname{argmax}}\ p(\omega | x_*, T, k). \tag{3.5}$$

The equation (3.5) have the same structure of MAP classifier which use the posterior class probability depending on the training set $T$ and the hyperparameter $k$. That is, dataset modifications, even in the absence of one pattern, it would imply in different outcomes from equation (3.5) succeeding a different classification

mapping. Since it is not possible to obtain the decision regions of kNN analytically, further is shown the decision regions from the equation (3.4).

The next example, depicted in figure 3.1, uses the same dataset of binary classification problem depicted in figure 2.2. Instead, it is used the original class priors, $p(AD) = 39.68\%$ and $p(CN) = 60.32\%$ solved with 5NN. Here, the decision region for kNN is drawn using a lattice with specific resolution limited to the maximum and minimum pattern features. Once this instance is a binary problem the decision boundary is when $p(AD|x) = p(CN|x) = 0.5$, or equivalently, $p(AD|x) = p(AD|x) - 1$, thus, one class outcome is enough to estimate the probability of assignment for both classes. However, in kNN we have a discrete values and the probability value 50% is not always available. For instance, in a binary problem 5NN allows the values, 1/5, 2/5, 3/5, 4/5 and 5/5, with 2.5/5=50% reached only for even $k$. Thus, the decision border is drawn between class transitions. In figure 3.1 the probability scale is proportional to the size of probability values allowed by $k$ parameter.

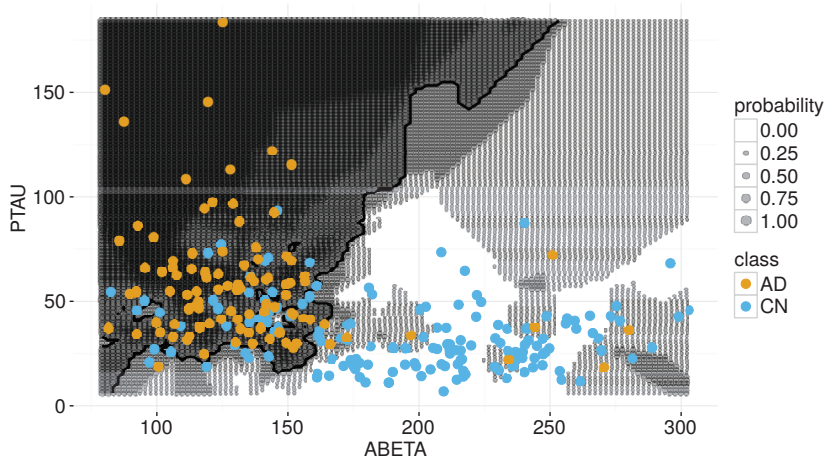



Figure 3.1 The decision border between the classes is represented by the black line.

The normalization is required in any distance-based algorithms since distances compared to different scales are meaningless. In order to compare with the examples from last chapter the problem in figure 3.1 was not normalized (re-scaled). When data is normalized classification mapping interpretative component may be compromised, however, to disregard re-scaling could potentially prejudice the classification performance [12]. Regarding the parameter $k$ in kNN as higher it is more stratified can be the probability outcomes for equation (3.5). In the next section, normalization schemes effect on kNN will be analyzed [54].

### 3.2.2 Tie breaking strategies

In binary classification problems, one can avoid ties using kNN if the $k$ parameter is an odd number. This fact does not hold for multiclass problems and there are no guidelines to the optimal $k$ hyperparameter. Despite this, one can limit the range of exhaustive search for $k$ parameter by the square root of sample number in the training set as suggested in [10]. Also, since kNN is a voting scheme any choice of $k$ parameter different from 1 (1NN) will imply in ties for the multiclass case. To deal with that, breaking ties strategies in kNN are incorporated to avoid ties. Unlikely binary cases solved with kNN, in which ties can be avoided, the decision boundaries in a multiclass problem depend on how we break ties. Tie breaking strategies discussed by [6] are:

- Replace $N(k, x_*)$ by $N(k-1, x_*)$ iterativelly until solve the tie (*minus*);
- Replace $N(k, x_*)$ by $N(k+1, x_*)$ iterativelly until solve the tie (*plus*);
- Random tie-breaking (*random*).

Figure 3.2 depicts the $k$-neighborhood for 5NN and the tie-breaking effect. As mentioned, using parameter $k$ as an odd number prevents ties in binary classification, however for multiclass problems this is not true as depicts figure 3.2.

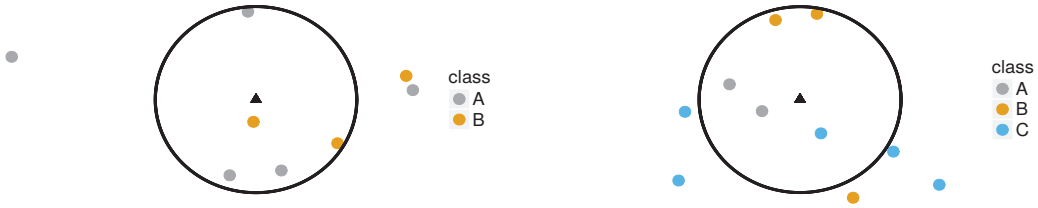



Figure 3.2 At left a binary classification problem and the query point (triangle) and its 5-neighborhood. At right a multiclass problem and the query point (triangle) and its 5-neighborhood.

Next example shows decision regions for a multiclass problem accounting for tie-breaking strategies, two validation procedures, and dataset-wise comparison. In the multiclass case, decision boundary definition are similarly defined as binary case, drawn between class transitions. Considering the AD classification problem as depicted in figure 2.3 and original priors and normalized with min-max [54]. Figure 3.3 depicts the decision region obtained with 5NN for each class with random break tie.

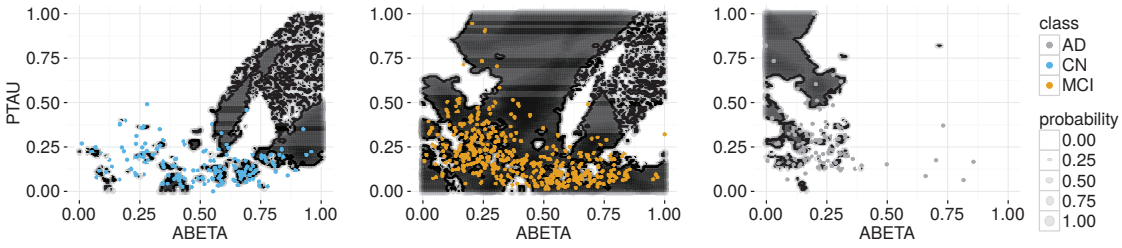



Figure 3.3 Respectively each plot shows the decision region for CN, MCI and AD for 5NN classifier with random tie-breaking.

One can notice, due to the random tie-breaking, overlapped regions between three plots act like an uncertainty region, creating a diffused decision region. Such randomness can add variability even to the LOOCV procedure as it is shown in table 3.2. Figures 3.4 and 3.5 depict the decision regions obtained by tie-breaking decreased $k$-neighborhood (minus) and increased $k$-neighborhood (plus), respectively.

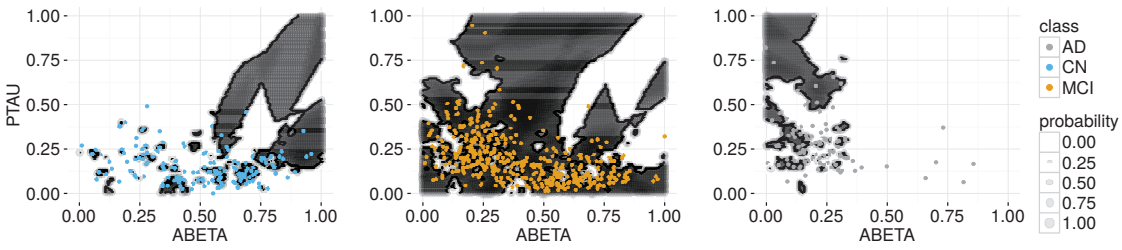



Figure 3.4 Respectively each plot shows the decision region for CN, MCI and AD for 5NN classifier with decreasing neighborhood tie-breaking.

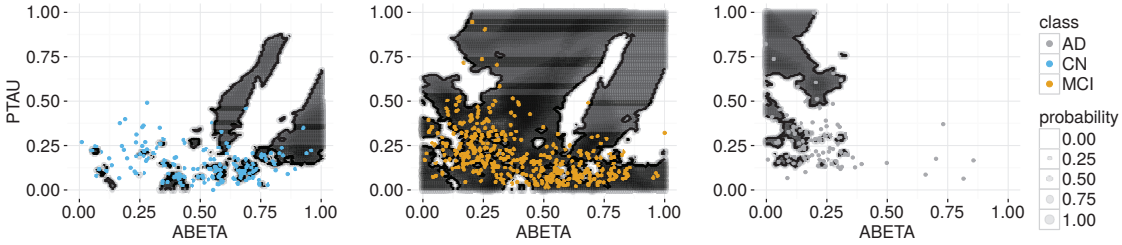



Figure 3.5 Respectively each plot shows the decision region for CN, MCI and AD for 5NN classifier with increasing neighborhood tie-breaking.

The confusion matrices' trace (precision) for 10-fold CV are depicted in 3.6 for all listed tie-breaking strategies. Confusion matrices averaged from 10-fold CV are depicted in figure 3.7. Furthermore, figure 3.7 shows that intermediary class, which is also the major class MCI, tends to concentrate most of the class assignments and is misclassified than minor classes (AD and CN). This is expected due to class overlapping as observed previously (chapter two).

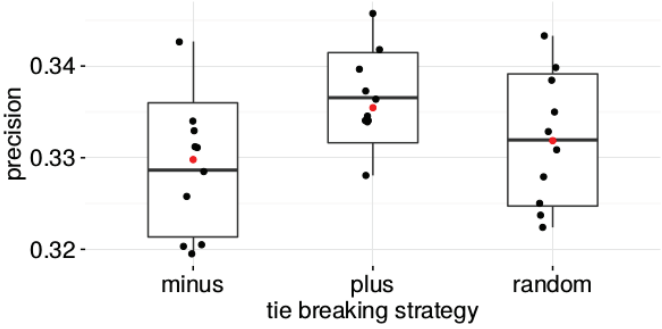



Figure 3.6 10-fold CV boxplot for each tie-breaking strategy and its precision using 5NN. The red dot stands for median.

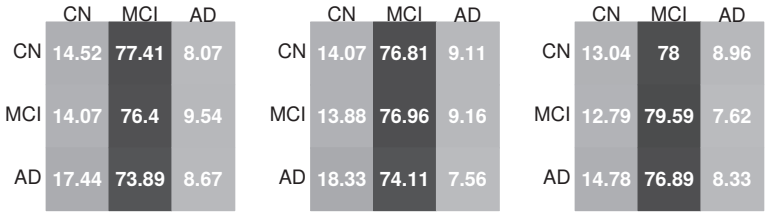



Figure 3.7 From left to right, confusion matrices for tie-breaking strategies random, minus and plus, respectively. Its precision are, 33.19%, 32.86% and 33.65%, respectively .

Confusion matrices of kNN (e.g. 3.7) in comparison with figure's 2.3 parametric classification approach was inferior, however, further modifications in kNN will change this. Table 3.2 presents a dataset-wise comparison between the three tie-breaking strategies. The validation was done with leave-one-out CV (LOOCV) and 2-fold CV. The $k$-fold number is small due to the converter datasets (*plasma*, *psyconvert*, *bloodabeta*) have classes with only two patterns as described in chapter six. The number of features for each dataset is described in chapter six.

Table 3.1 LOOCV comparison for each tie-breaking strategies.

| dataset | minus (%) | $k$ | plus (%) | $k$ | random (%) | $k$ |
|---|---|---|---|---|---|---|
| plasma | 37.91 | 3 | 36.58 | 32 | 38 ± 3.01 | 3 |
| psyconvert | 37.71 | 13 | 37.77 | 4 | 37.72 ± 0.18 | 14 |
| bloodabeta | 15.95 | 19 | 16.02 | 21 | 16.33 ± 0.30 | 7 |
| adni | 79.43 | 1 | 79.43 | 1 | 79.43 | 1 |
| proteom | 41.60 | 1 | 41.60 | 1 | 42.29 ± 1.32 | 2 |
| neuropsy | 88.71 | 1 | 88.71 | 1 | 88.71 | 1 |
| neuroimag | 47.91 | 11 | 47.91 | 10 | 47.83 ± 0.36 | 10 |
| parkinson | 96.22 | 1 | 96.22 | 1 | 96.22 | 1 |
| average | 55.67 | - | 55.52 | - | 55.81 | - |

Table 3.2 2-fold CV comparison for each tie-breaking strategies.

| dataset | minus (%) | k | plus (%) | k | random (%) | k |
|---|---|---|---|---|---|---|
| plasma | 35.77 ± 0.57 | 16 | 37.15 ± 1.13 | 10 | 36.31 ± 1.45 | 14 |
| psyconvert | 37.11 ± 0.13 | 7 | 37.55 ± 2.03 | 17 | 37.32 ± 1.14 | 6 |
| bloodabeta | 16.30 ± 0.42 | 32 | 16.22 ± 0.21 | 32 | 16.38 ± 0.24 | 8 |
| adni | 79.13 ± 5.44 | 4 | 74.78 ± 1.67 | 1 | 77.36 ± 4.72 | 1 |
| proteom | 38.61 ± 0.96 | 1 | 39.07 ± 1.99 | 1 | 43.40 ± 4.61 | 2 |
| neuropsy | 86.50 ± 0.49 | 1 | 88.60 ± 0.78 | 1 | 88.30 ± 1.99 | 1 |
| neuroimag | 46.92 ± 1.64 | 9 | 47.00 ± 3.64 | 2 | 47.41 ± 1.21 | 6 |
| parkinson | 93.09 ± 4.88 | 1 | 91.71 ± 4.90 | 1 | 91.70 ± 0.97 | 1 |
| average | 54.17 | - | 54.00 | - | 54.77 | - |

All tie-breaking strategies differ from less than 1% for both validation modalities. Regarding a tie-breaking strategies comparison, plus and minus are less computational advantageous than random tie-breaking [6]. This is, As claimed by [6] strategies that require $k$-neighborhood reassessment are slower than random tie-breaking. Henceforward, the random tie-breaking will be adopted in this work.

Regarding table 3.2 datasets *plasma*, *psyconvert* and *bloodabeta* which have seven classes are highly overlapped and imbalanced (classes with unequal prior probabilities). Furthermore, datasets plasma, psyconvert and bloodabeta are a cohort study, that is, a combination of follow-up and cross-sectional data. Here, it is assumed that follow-up patterns are non-temporal in order to observe how conversion classes affect overall performance, that is, a cross-sectional usage of cohort data [56]. Data-wise comparisons in this section use min-max normalization, however, re-scaling schemes can affect kNN classifier precision given the magnitude between features. In the sequel is analyzed how the classifier can be affected by different normalization schemes.

## 3.3 Normalization schemes

Since kNN is a distance-based classifier data normalization is required in order to avoid meaningless distance measures [8]. With respect to interpretability of decision regions, normalization can be a drawback since features must be re-scaled in order to be dimensionless. Indeed, to disregard normalization would affect the validation phase, for instance when using an unnormalized dataset, the ANNs converges slower [73] and *k-means* clustering provides poorer results [59]. Such drawbacks are increased the more features magnitude differ, causing patterns to cluster in short scales, the more compacted they are.

Settled the normalization's importance, what is the best normalization scheme for the kNN given a classification problem? An instance of normalization effect was given by Ma *et al.* [54] when solving the binary classification problem for the *parkinson*'s dataset (described chapter six). They were able to show that kNN accuracy increases by assessing four normalization strategies and validation strategies ($k$-fold CV and LOOCV) when compared to the random choice of kNN hyperparameters with unnormalized data. Apart from the normalization requirement, there are classifiers as Naive Bayes and Fischer's Linear Discriminant in which re-scaling is unnecessary and have no impact on classification measures [12]. In the last section, only min-max normalization was applied to illustrate tie-breaking simulations. Here, the effect of different normalization schemes is compared when $k$ parameter is optimized. In order to do this, the probability of one normalization scheme to provide higher precision than other is evaluated. The compared normalization schemes are: maximum (max), min-max (min), z-score (zsc) and decimal (dec) [54]. Considering $v$ the feature and $v'$ the normalized feature, we have.

$$\text{min: } v' = \frac{(v - min(v))}{max(v) - min(v)} \qquad \text{max: } v' = \frac{v}{max(v)}$$

$$\text{zsc: } v' = \frac{v - \mu(v)}{\sigma(v)} \qquad \text{dec: } v' = \frac{v}{10^j} \text{ for } j \in \mathbb{Z},\ \|v'\| \leq 1$$

First let's define a prior reduction technique discussed in next chapter, the *random undersampling*, in which majority classes go under pattern removal until it achieves a desired proportion. This changes the classification problem since it modifies the prior probability. Let's compare different degrees of pruning with different normalization schemes. Figure 3.8 shows the precision of AD three-class classification problem depicted in figure 3.1 for the mentioned normalization techniques and using 0%, 50%, 75% and 100% of random undersampling degree.

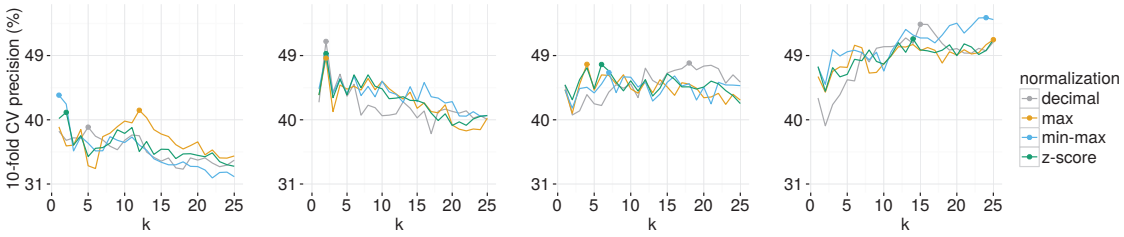



Figure 3.8 From left to right, 10-fold CV precision for four normalization strategies, respectively for 0%, 50%, 75% and 100% of random undersampling degree. The dots in each line depict the maximum precision for each normalization scheme.

Figure 3.8 shows that increasing undersampling degree the optimal $k$ parameters for each normalization scheme changes as well. Considering each undersampling degree in figure 3.8 there is not significant improvement in the highest 10-fold CV precision by each normalization scheme. That is, highest precision for each undersampling degree (0%, 50%, 75% and 100%) in terms of mean and standard deviation is 43.45 $\pm$ 4.55% for min-max, 50.98 $\pm$ 5.10% for decimal, 47.95 $\pm$ 7.64 % for max and 54.30 $\pm$ 6.06 % for decimal, respectively to each listed undersampling degree. The last kNN score surpasses the parametric method in last chapter.

Despite the lack of significance, there is a small difference of highest precision by normalization scheme for each undersampling degree of figure 3.8 being the standard deviation between them, 1.83%, 1.04%, 0.60% and 1.52%. It is interesting to verify whether the precision changes in other validation schemes in which standard deviation is less than 10-fold CV, e.g. LOOCV.

Changing the validation procedure parameter $k$ optimum value can be not the same. Figure 3.9 depicts if a dominance exists among normalization schemes through the search range for the parameter $k$. In order to do this, each undersampling degree is evaluated for the probability of a normalization scheme to provide higher precision than other.

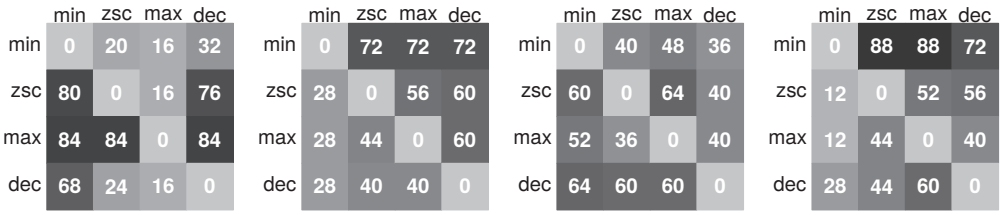



Figure 3.9 From left to right, probability matrices of a normalization scheme to provide strictly higher 10-fold CV precision than other for $k$ parameter ranging from 1 to $\sqrt{\text{number of samples}}$ for each undersampling degree, 0%, 50% 75% and 100%, respectively.

Probability matrices in figure 3.9 comparisons would be not symmetrical. For instance, the fourth matrix in figure 3.9 depicts that $P(\text{zsc} < \text{min}) = 44\%$, whereas $P(\text{zsc} < \text{max}) = 52\%$. This happens since ties would occur for some $k$ values. Since for each undersampling degree the 10-fold CV precision was not statistically significant. Let's repeat the same experiment for LOOCV procedure. Figure 3.10 depicts the LOOCV precision for different undersampling degrees and normalization schemes. As it can be seen in figure 3.8, the same effect observed for 10-fold CV in which undersampling degree of 100% exhibits highest precision and undersampling degree 0% (original dataset) inferior are evident for LOOCV as well.

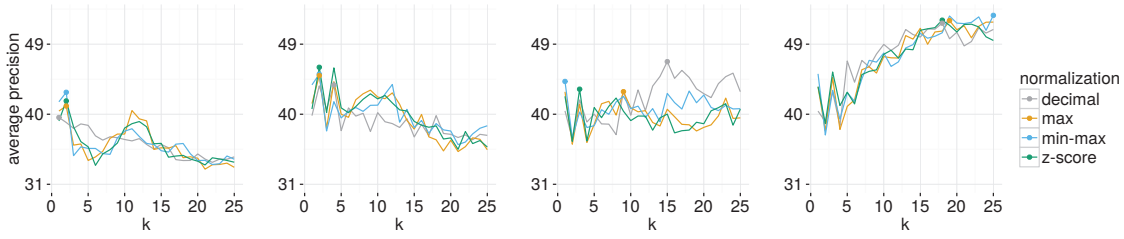



Figure 3.10 From left to right, LOOCV precision for four normalization strategies, respectively for 0%, 50%, 75% and 100% of random undersampling degree. The dots in each line depict the maximum precision for each normalization scheme.

Using LOOCV each undersampling degree (0%, 50%, 75% and 100%) achieves the following highest precision respectively: $42.82 \pm 0.09\%$ with min-max, 46.03% with z-score, 46.76% with decimal and 52.70% with min-max. Except for undersampling degree of 75%, the difference between highest value are nearer compared to 10-fold CV highest scores. The standard deviation between the maximum for each normalization scheme by undersampling degree (0%, 50%, 75% and 100%) is 1.35%, 0.93%, 1.74% and 0.41%, respectively. Figure 3.11 is depicted the probability matrices of a given normalization scheme to provide higher precision than remaining through parameter $k$ search path. One can notice re-scaling dominance along parameter $k$ values does not hold for different validation schemes.

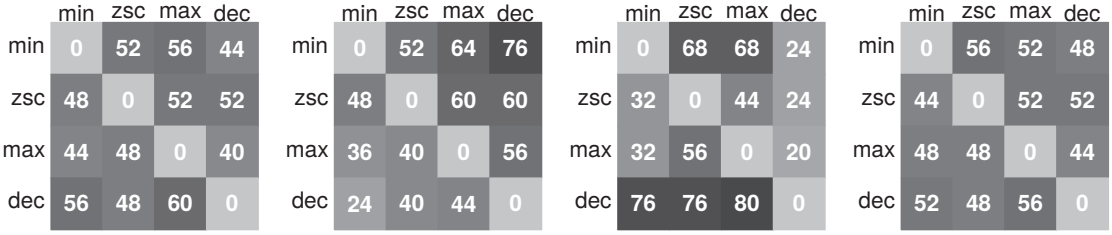



Figure 3.11 From left to right, probability matrices of a normalization scheme to provide strictly higher LOOCV precision than other for $k$ parameter ranging from 1 to $\sqrt{\text{number of samples}}$ for each undersampling degree, 0%, 50% 75% and 100%, respectively.

In order to show how the normalization behaves for other $k$-fold validation values, the optimization path (through $k$ parameter) for $k$-fold CV varying from 2 to 30 folds and different normalization schemes for imbalanced setup are shown in the interactive plot, link https://plot.ly/∼yurier/190/ . Despite the optimization path's for $k$-fold CV being the same, standard deviation curves become less disperse as the k-fold size decreases as seen in https://plot.ly/∼yurier/192/ . This happens due to the statistical dissimilarities in sample size for different validation folds. Let's shown that it is possible to increase precision by experimenting different normalization schemes. Below, simulation comparing normalization schemes and different datasets.

### 3.3.1 Data-wise comparison of normalization schemes

The last section presented comparisons between 10-fold CV and LOOCV for the *proteom* dataset with optimized $k$. Furthermore, despite LOOCV being more computationally intensive than 10-fold CV it has less standard deviation allowing a significant improvement comparison between normalization schemes. Figure 3.12 depicts different datasets and the average precision difference for listed normalization schemes with kNN optimized. None of the datasets are treated with balance aid strategies as done in figures 3.8 and 3.10.

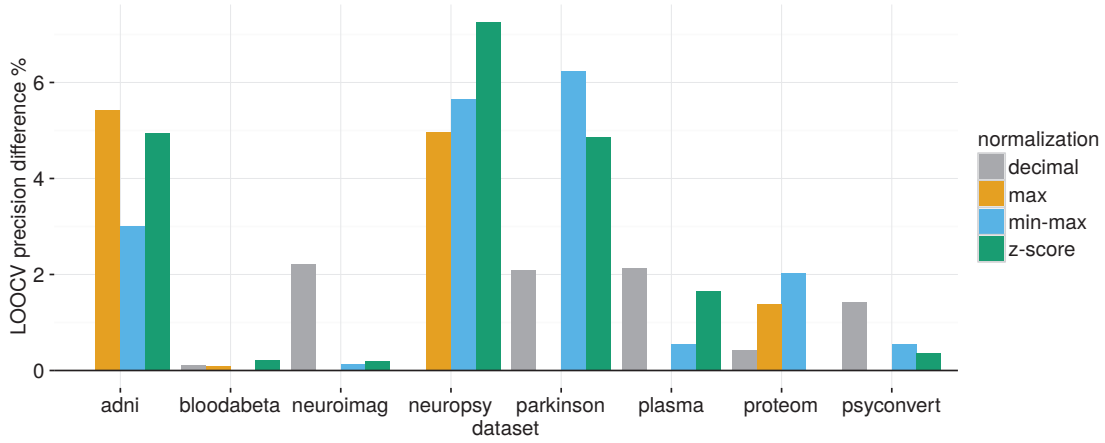



Figure 3.12 Relative difference between the lesser average precision normalization scheme and the remaining ones. The missing color label means the lesser value.

For *bloodabeta* dataset the normalization increases LOOCV precision from $16.27 \pm 0.22\%$ using min-max to $16.49 \pm 0.38\%$ using z-score. Conversely, there was a slight improvement for other datasets, an instance, *neuropsy* dataset in which best use of normalization schemes can raise LOOCV precision from 83.04% using decimal to $90.30 \pm 0.71\%$ using z-score. Thus, kNN and normalization optimization can be done in order to raise the average precision. Later on, a greedy approach will be adopted since other optimization procedures will increase the number of hyperparameters. Next section proposes to find a suitable $p$-norm metric for kNN in order to increase its precision.

## 3.4 Minkowski distance modification

The *curse of dimensionality* [7] refers to poor classification measures [72] in problems related to high-dimensional feature spaces [12], it is generally avoided with dimensionality reduction [30]. Dimensionality reduction techniques' goal is to find a representative feature subspace to describe the same classification problem with equal or higher classification properties, e.g. principal component analysis (PCA) and Fischer's discriminant analysis. An interesting curse of dimensionality effect explanation is given by [32], in which it is shown the mean distance between the closest pattern and the origin for a given uniform distribution becomes *meaningless*, in the sense defined by [8]. Due to the meaningless metric behavior in high-dimensional feature spaces the usage of kNN becomes limited. Let's study here the possibility of increasing kNN's precision by searching a suitable Minkowski distance function.

The term of high-dimensional evolved within popularization of video and image pattern recognition and micro-array gene analysis. Early developments owing to reduce the feature space dimensionality are designed to deal with nearly 40 features [46]. Nowadays, high-dimensional classification problems vary broadly, ranging from 6000 to 60000 features [30]. Examples of datasets which receive high-dimensional feature spaces aids are genetic data, neuroimaging data, image/video data, bank transactions, etc.

A central problem regarding high-dimensionality is the feature space's sparsity. That is, adding features, patterns become isolated in the feature space and distances between farthest and nearest patterns become unrecognizable. Meaning that, the distance between patterns become meaningless [2]. Let's explore and simulate a relative distances measure in an uniform data and the fractional metric solution proposed by [2] to prevent distances functions to become meaningless.

### 3.4.1 Contrast measure for distance meaningfulness

Particularly, the kNN classifier is affected by the curse of dimensionality since it is a distance-based classifier [8]. Bishop [12] exemplifies the dimensionality effect on kNN by means of increasing hypercubic volume and the sparsity between patterns in it. However, as shown by Aggarwal [2], one may use the fractional distance function in order to improve kNN's performance. That is, one can reduce the high-dimensionality effect with metric searching and parameter $k$ optimization. However, no empirical study on the search range for fractional metric was conducted. The behavior of $p$-norm induced metric in high-dimensionality feature space for uniform data is simulated after the *contrast* measure definition.

Let $X^d \subset \mathcal{F} := \mathbb{R}^d$ be a set of patterns in an $d$-dimensional feature space. Assume $\|X^d\|_p$ as the vector of the distances between each pattern in $X^d$ and the origin ($0 \in \mathbb{R}^d$) for a given Minkowski distance, defined by $\|x\|_p = (\sum_{i=1}^{d} |x_i|^p)^{1/p}$, in which $x = x_1, ..., x_d \in X^d$. Also, consider the ratio between the distance of nearest and farthest patterns in $X^d$ to the origin as *contrast* [8], given by, $constrast^{-1} = Dmin_d^p / Dmax_d^p$, where $Dmax_d^p = max\{\|X^d\|_p\}$ and $Dmin_d^p = min\{\|X^d\|\}_p$ for $Dmin_d^p \neq 0$, that is, the origin do not belong to $X^d$. The behavior of $L_p$-norm metric induced in high-dimensionality feature space is obtained observing the contrast convergence when dimensionality ($d$) is increased and fixing the previous set of patterns $X^{d-1}$. That is, new features are introduced. For further discussion and definition on contrast see [8].

To illustrate contrast in high-dimensional feature space, consider 2000 patterns in the set $X^d$, in which each pattern in each dimension is an uniform distribution realization, $\mathcal{U}(0, 1)$. Figure 3.13 depicts fractional and non-fractional $p$-norms contrast for the following $p$ values: 2/3, 2/5, 1/7, 3, 2 (Euclidean distance), 1 (Manhattan distance).

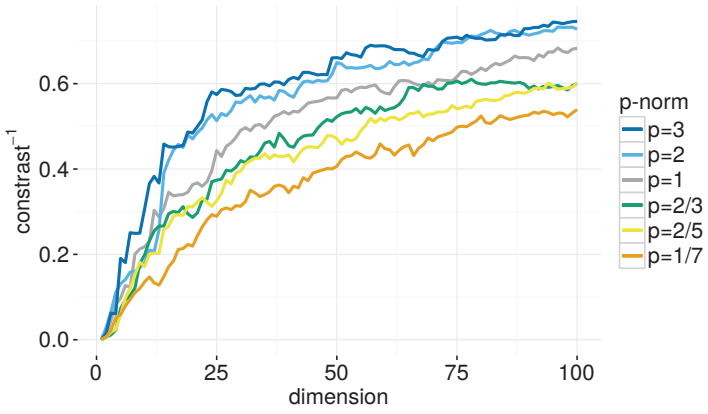



Figure 3.13 Contrast convergence for uniform data.

Figure 3.13 depicts the inverse contrast when dimensionality increases. As it can be seen by contrast measure the furthest and nearest patterns have about the same distance from the origin. This behavior depends on data distribution, however as claimed by [8] the convergence of contrast is valid for a wide variety of data distribution functions. Convergence of fractional measures is theoretically discussed in [2]. Figure 3.14 shows the probability of a given $p$-norm to have higher inverse contrast than other.

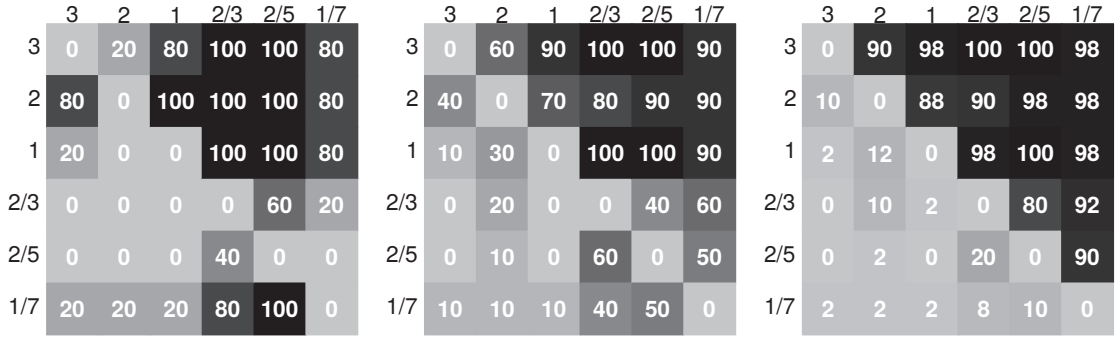



Figure 3.14 From left to right the probability of an inverse contrast of a given p-norm being higher than another, summing up until 5, 10 and 50 dimensions of figure 3.13 respectively.

The higher is the $p$-norm value the faster is its contrast convergence (in probability), as depicted in figure 3.14 in which is shown that increasing the number of features the farthest and closest vector have about the same distance from origin. In the curse of dimensionality context, there are query point instability issues.

Specifically, how perturbations in the query point imply in changes for classification outcome. Pattern classifiers must be robust against noise or perturbations since in real applications one would have pattern artifacts difficulties which modify actual pattern value [12]. However, as demonstrated in [2] the fractional metric is able to overcome the integral $p$-norm in noisy mask simulations. Due to the dataset's low-dimensional character in this work fractional metric effect in contrast measure would be subtle. Next chapter illustrates Minkowski distance modifications using *parkinson* dataset.

### 3.4.2 Parkinson dataset and distance function search

It is possible that metric optimization benefits low-dimensional classification problems. Let's show it through *parkinson* dataset and fractional distances optimization since it have the largest number of features and it is a prion-like disease as AD. At end of section, combined strategies will be applied in the AD datasets and its contrast measures will be compared. Figure 3.15 depicts the kNN's LOOCV precision for $p$-norm induced metrics using $k$ parameter optimized and non-optimized (fixed at $k = 5$). The simulation in figure 3.15 was driven using min-max normalization due to score achieved in figure 3.12.

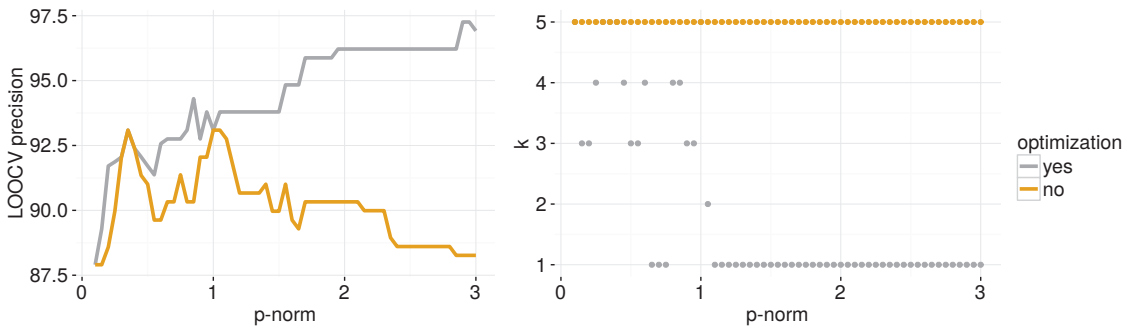



Figure 3.15 At left precision each $p$-norm induced metric ranging from 0.1 to 3 by 0.05 for $k$ parameter optimized (yes) and fixed at $k = 5$ (no). At right, optimum $k$ parameter for each $p$-norm.

LOOCV higher precision in figure 3.15 achieves 97.25% for $p$-norms 2.90 and 2.95 with optimized $k$ and min-max normalization. Euclidean and Manhattan distances achieve 96.21% and 93.09% respectively for optimized $k$ and min-max normalization. Regarding *parkinson* dataset and measures obtained from experiment depicted in figure 3.15, distance function searching had few improvement relative to usual euclidean distance. However, comparing our results by accuracy instead of precision, both defined in [72], the metric optimization is able to surpass the score obtained described by [54] with the margin of 0.51% for the same z-score normalization and LOOCV but using $p$-norm 3 instead of Euclidean distance. Also, by using accuracy measure metric, the optimization achieves 96.92% for min-max against 96.41% described as kNN's highest precision [54]. R. Ramani and G. Sivagami [64] solve *parkinson* classification problem and compare 13 popular algorithms. They were able to shown that kNN accuracy is only inferior to random trees classifier [32] which achieves accuracy of 100%. In a further, chapter imbalance aiding strategies based on kNN will be included and the classification rate would increase. Next experiment in figure 3.16 shows the behavior between $p$-norms 3, 2, 1, 2/3, 2/5 and 1/7 whereas *parkinson* dataset features vary. First, this is done for $k$ parameter fixed at 5. In order to observe $p$-norms behavior, features were also added randomly without substitution to compare with non-randomized added features. Moreover, in order to observe each $p$-norm precision the curve was smoothed and the real value was added with the same color in the background.

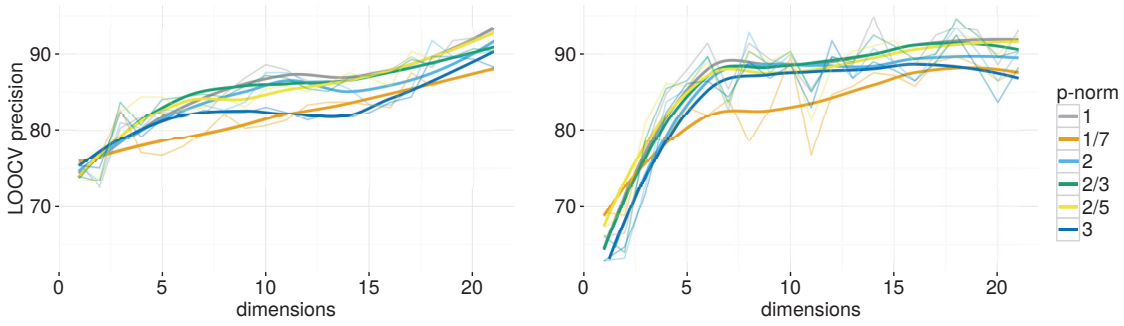



Figure 3.16 At left, precision curves for $p$-norm distances when varying features. At right, precision curves for $p$-norm distances when varying the dimension (features) randomly.

Figure 3.16 shows using 14 random features the LOOCV precision achieves 94.83 % for $p$-norm 1, whereas the highest precision combination for non-random added features achieve 93.09%. That is, different features combinations would provide different classification rates. Figure 3.17 shows the probability matrix of a $p$-norm to provide higher precision than other.

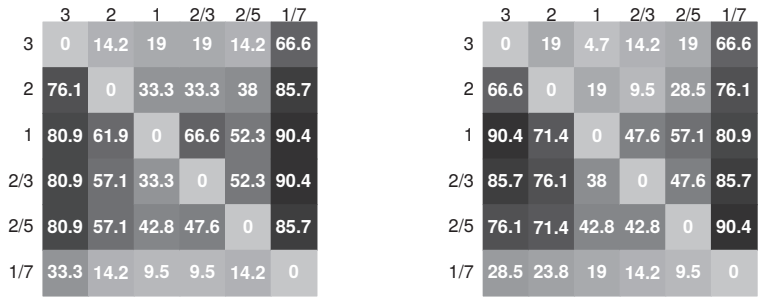



Figure 3.17 At left, the probability matrix for *parkinson* features not randomly added. At right, the probability matrix for *parkinson* features randomly added.

Both probability matrices in figure 3.17 shows extreme $p$-norm, e.g. 3 and 1/7, the matrices lines averaged are lower than remaining, this is comparable to the experiment in figure 3.15. Also, $p$-norm 1 and .35 have the same precision for 5NN as depicted in figure 3.15. Interesting, p-norm 1 had the highest average line in both probability matrices. Suggesting that the $p$-norm behavior regarding lower and higher precision can affect the probability of dominance through other features combinations. Next, the $p$-norm behavior when $k$ parameter is optimized as depicted in figure 3.18 for random and non-random features of *parkinson* dataset.

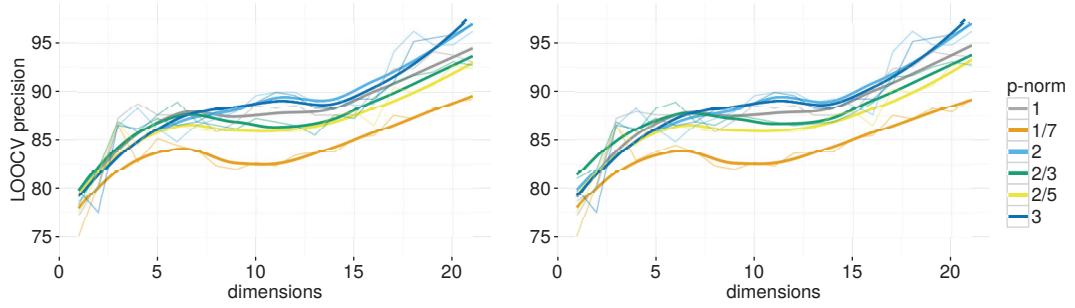



Figure 3.18 At left, the probability matrix for *parkinson* features nor randomly added. At right, the probability matrix for *parkinson* features randomly added.

In figure 3.18 precision for $k$ parameter optimized achieves 96.91% using $p$-norm 3. Despite this, for fractional metric, it was found that precision reaches 97.25% for $p$-norms 2.90 and 2.95. Dominance behavior of $p$-norm metric is depicted in figure 3.19.

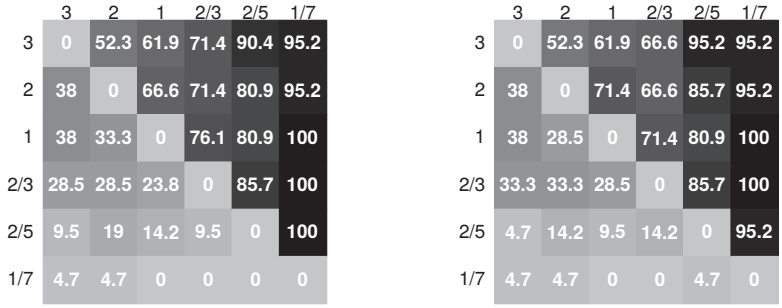



Figure 3.19 At left, the probability matrix for *parkinson* features not randomly added. At right, the probability matrix for *parkinson* features randomly added.

Using the same method to assess the dominance of one metric over another the random and non-random probability matrices have the same dominance structure in the sense of averaged matrix lines. A suggestion of why this happens is because the preference by metric in probability is reflected through the features as well for the non-optimized $k$ case. From measures taken in figure 3.19, $p$-norm 3 and 2 achieves the highest precision along the dimensions for random and non-random features adding. Such measures are coherent with the experiment in figure 3.15.

Both experiments, for optimized and non-optimized $k$, show that random added features do not largely modify the dominance's structure along dataset features. Such structure of metric through the features would be useful for dimensionality reduction mechanisms while searching the most suitable metric in high-dimensional feature spaces. However, the study remains to be done by comparing the optimal $p$-norm distribution through the multiple features combinations. Furthermore, other approaches as metric learning [82] by linear programming being used in a comparative study with for kNN distance function modifications. Advances in this direction can be found in the following works: Neighborhood Components

Analysis by [27] and Stochastic Neighbor Embedding [37]. It is clear that metric modifications increase kNN performance in next section a data-wise comparison is presented.

### 3.4.3 Data-wise comparison for distance optimization

The section 3.4.2 uses only *parkinson* dataset since it has the largest number features here is presented a comparison of AD datasets accounting to the previous normalization scheme obtained in the experiment from figure 3.12. Figure 3.20 (left) depicts the difference between LOOCV precision with Euclidean distance and LOOCV precision using fractional distances, both with optimized $k$. It is shown also the standard deviation for the outcome $k$ optimized outcome for each $p$-norm.

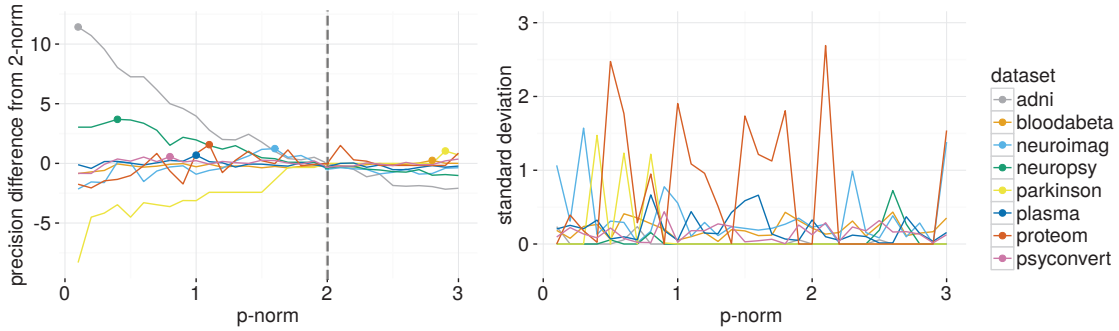



Figure 3.20 At left, the difference between precision in Euclidean distance by dataset and $p$-norm precision. At right, the standard deviation for optimized $k$ in each $p$-norm.

Regarding improvements by metric optimization, there were datasets, for instance, *bloodabeta*, with high standard deviation and oscillating precision around Euclidean distance. There was none dataset in which the highest average precision occur for Euclidean distance. The highest improvement was in the *adni* dataset in which LOOCV precision increases $11.43 \pm 0.23\%$ with $p$-norm 0.01 over the 81.84% achieved by searching only a suitable normalization scheme with Euclidean distance. The metric searching shows to be capable of improving kNN's precision when associated to the normalization scheme. Comparison of such techniques discussed here and other classifiers will be performed next section.

In summary, this chapter shows how distance functions and normalization schemes modifications would improve the kNN's performance. These procedures also can be applied in other distance-based classifiers. NN algorithms are plenty and easy to modify since similarity measures and voting scheme can be arranged in several ways [9]. The number of publications regarding popular classifiers' extension is notoriously increasing, for instance, in science direct database a search for "kNN extension" outcome 37 titles in 2000 and increases to 610 titles in 2016. A kNN extension, the weighted kNN (wkNN) [29], attributes a weight for each voting pattern, whereas the kNN assumes that all patterns have the voting weight regardless of distance. The wkNN uses kernels to distribute the weights through voting patterns and since it is an distance-based classifier it also can use the approaches discussed here. Additionally, wkNN can would benefits from kernel optimization as proposed by [36]. Another extension is the informative kNN (ikNN) [74] which uses *informativeness* criterion as similarity measure and is able to surpass popular algorithms as SVM and boosting methods in classification task.

Instead of exhaustive searching as done here, metric learning for nearest neighbor algorithms are found in kNN extensions, e.g. the Large Margin Nearest Neighbor in which the Mahalanobis distance function is learned by semi-definite programming [82]. The discussed Minkowski distance searching , suitable for kNN, produces a non-convex optimization problem which is incompatible to the usual optimization techniques. An instance of a non-convex surface is depicted in figure 3.20 for *bloodabeta* dataset. However, the optimization procedure for metric learning would be done between two patterns as follows. Considering a dataset $D$ in which $x_i, x_j \in D \subset \mathbb{R}$ the metric learning approach can be done by finding a matrix $A$ which minimizes the distance function $d(x_i, x_j)_A^2 = \|x_i - x_j\|_A^2 = (x_i - x_j)^T A(x_i - x_j)$ while ensuring that $d(.,.)_A$ satisfies the metric (or pseudo-metric) definitions. A model to find the optimal distance function to kNN without an exhaustive search would be useful since there are other parameters to optimize, e.g. optimal k-fold CV partition and parameter $k$. However, more data is required to make a predictor for

such relation between contrast and the optimal $p$-norm in order to have at least an smart start to search the optimal distance function. An exploratory data analysis was driven using the average class inverse contrast, contrast, highest improvement and optimal distance function as depicted in figure 3.21.

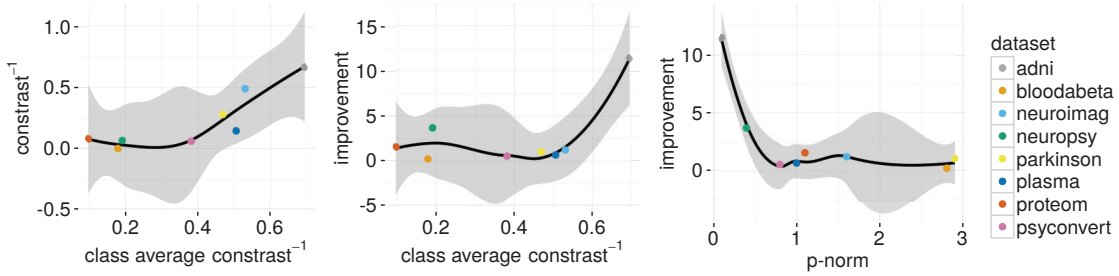



Figure 3.21 Exploratory data analysis on constrast and precision relationship.

The objective of figure 3.21 is to show how contrast plays an important role in the use of kNN modifications developed in this chapter. The theoretical improvement modeling, that is how better the metric optimized would reach remains to be cleared and deserves further attention. Next chapter brings aid balance solutions for imbalanced classification problems, which means datasets with a different number of patterns. Figure 3.8 shows normalization schemes with a different undersampling degree to illustrate how much improvement a classification problem would expect. Also in imbalanced class setup, the major class tends to have higher classification rate than minority ones as depicted in figure 3.7 for tie-breaking strategies. The AD classification problem with imbalance aids would increase the capability of detect patterns in minor classes. As demonstrated in the pioneer work comparing balance aid strategies to improve classification rate using ADNI (Alzheimer's Disease Neuroimaging Initiative) data [20]. The next chapter explores distance-based strategies to deal with imbalance setup in AD classification problems and compares novel distance-based strategies to do it.

# 4 IMBALANCED CLASSIFICATION

## 4.1 Introduction

Data imbalance happens when a number of instances for a given class is higher than the remaining ones [35]. It is considered among the ten most challenging issues in data analysis since classifiers dealing with data imbalance situations tend to misclassify minor class samples. In medical research and industry, consequences of imbalanced data in computer aided diagnosis are overwhelmingly costly [35]. Unfortunately, imbalances in medical databases are ubiquitous due to the manifestation of different disease spectra's regions which have its own probability of occurrence [20]. An instance from medical images is the number of voxels/pixels representing abnormalities which are inferior compared to all image's voxels/pixels, characterizing an imbalanced problem. Breast cancer detection using digital mammography [16] is a classification problem which congregates such difficulties. The number of pixels representing cancerous regions is scarce and hard to detect, additionally, cancerous breasts images in training set are few (10,923 healthy vs 260 cancer [35]). The urgency of this problem gives rise to the *data dream digital mammography challenge* in 2016 which an $1.2M prize was offered for the most accurate classifier. Similar data dream challenges for AD classification called attention in 2014.

In AD, it is relevant to predict if an individual will convert to another AD's class. Despite the importance of early intervention, an accurate solution for AD conversion classification problem remains a challenge. This happens due to the few instances in conversion classes and MCI classes stratification which prevent an accurate non-binary or binary approaches. For instance, Cuingnet R. *et al* [18], using MRI extracted features from 509 subjects of ADNI cohort study, proposed three binary classification designs between AD classes: CN vs AD, CN vs MCI/AD (MCI patients which become AD within 18 months) and MCI/AD vs MCI (MCI

non-converters within 18 months). Cuingnet R. *et al* were able to correctly classify: CN vs AD with 81% sensitivity and 95% specificity; classify CN vs MCI/AD with specificity ranging from 51% to 73% and sensitivity 85%; and, MCI/AD vs MCI achieving 32% sensitivity and 91% specificity. Even using a binary perspective over the classes Cuingnet R. *et al* obtained low scores for highly overlapped classes and conversion classes. Here, instead of non-binary classifiers largely applied in AD classification [25] let's use a natural multiclass classifier, described in the last chapter aided with sampling strategies.

The main goal of sampling techniques is to balance the accuracy between unrepresented and overrepresented classes. Imbalanced setup complexity is a wide topic, covering from small sample-size imbalanced classes, majority and minority class *concepts*, overlapped imbalanced classes to the noise data [35]. Such topics will be highlighted and discussed in this chapter using distance-based sampling algorithms, that is, inspired by kNN (with exception of random undersampling).

Regarding state-of-art algorithms for imbalanced data Haibo He *et al.* [35] divide up the current focus of study in sampling methods, cost-sensitive methods, kernel-based and active learning methods. This chapter compares the oversampling and undersampling distance-based algorithms effects in AD datasets precision. The undersampling methods presented are the random undersampling and four informed undersampling. Whereas oversampling methods presented are the SMOTE (**S**ynthetic **M**inority **O**versampling **TE**chnique) and three adaptive synthetic sampling variations from SMOTE's family. In this context we describe two contributions, one algorithm for informed undersampling, Nearmiss4, and one for oversampling, SMOTE-borderline3. Respectively, the first show to be the best choice to reduce class overlapping while the other is a multiclass orientated SMOTE's version. Next section illustrates the imbalance problem in the kNN context and relates the integration of kNN modifications with sampling methods. At end of chapter a

comparison between kNN with sampling methods and previous chapter modifications against popular classifiers choices.

## 4.2 Imbalanced dataset issues

Minority classes have a less representative number of instances, implying in reduced size of decision regions. Since they are underrepresented, misclassification errors in minor classes tend to be higher than in major ones for all confusion matrices previously presented, for instance, figures 3.7 and 2.6.

To illustrate the decreased size of the minor class decision region let's assume a circle representing the minority class decision region. Figure 4.1 depicts the class A (minority class) where patterns were generated using a circular uniform distribution centered in origin with unitary radius, whereas class B (majority class) was generated by a Gaussian distribution following $\mathcal{N}(\mu = [2, 2], \Sigma = 4 * I)$. The imbalanced proportion between class A and B is 1:13.

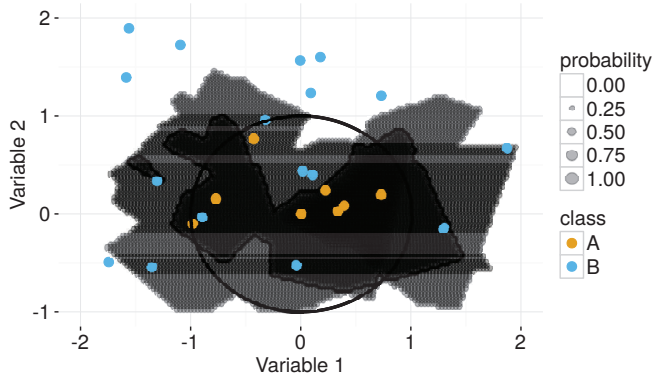



Figure 4.1 Binary class decision region obtained with 3NN.

The circle in figure 4.1, in which class A was generated, is underrepresented and give rise for minor class classification errors. Other sources of misclassification include minor class sub-clusters and noise samples. These topics have a concise presentation in [35], here only usual imbalance will be discussed. Classifi-

cation methods (e.g. SVM, MLP, and kNN) have performance issues when dealing with imbalanced datasets [35]. However, imbalance aid can be noneffective depending on the dataset as will be shown. Another case, are classifiers in which a degree of imbalance allow better results than balance dataset [35]. The kNN's formulation approximates $p(\omega|x)$ directly disregarding the class priors as was shown. Here let's discuss how imbalance setup affects kNN's precision and how class prior modifications can prevent the shrinking effect for AD classification problems.

## 4.3 Random undersampling

The previously mentioned random undersampling (RU) is a sampling strategy which prunes randomly the major classes in order to have equal proportion between minor ones [35]. The major classes will loss instances and it will enlarge the minor class decision regions [55]. Also there is the possibility to recycle pruned data to use together with the validation set. In the end of this section an AD dataset comparison showing how RU increases the kNN classifier precision for LOOCV and 2-fold CV. Assume the AD three class problem for *proteom* dataset normalized with min-max. Figure 4.2 shows the decision regions using 100% of undersampling degree and 5NN. That is, the class priors of majority classes will have the same prior number than minor class.

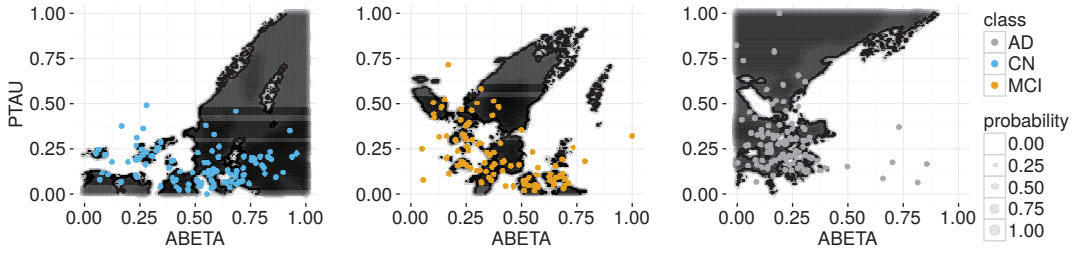



Figure 4.2 Decision regions for CN, MCI and AD respectively for RU with 100% degree and 5NN.

Notice majority class decision region in figure 4.2 was reduced when compared with imbalanced problem decision region in figure 3.3. The undersampling

degree parameter will not be optimized since this would increase the computational cost associated to optimize the $k$ parameter. Thus, let's use $p$-norm value and normalization scheme previously obtained for all data-wise simulations in further tables. Figure 4.3 show how changes the $k$ parameter when the undersampling degree is modified due to this the $k$ value must be optimized.

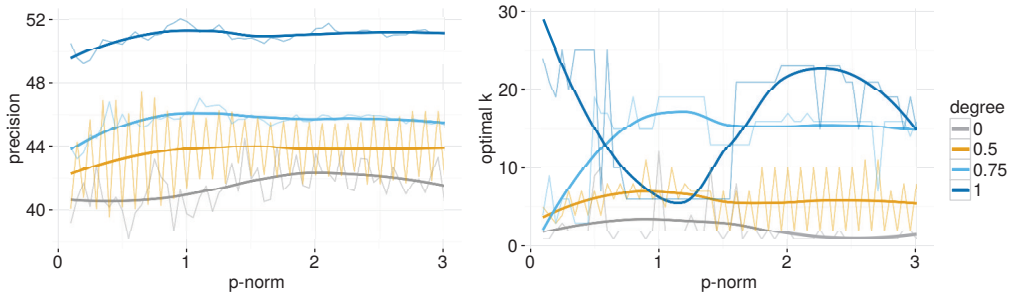



Figure 4.3 At left, precision obtained from optimal $k$ and varying $p$-norm and RU degree. At right, optimal $k$ for each example in left figure.

In figure 4.4 the AD classification problem is solved using 5NN when the undersampling degree is less than 75% and 25NN otherwise. It was also applied random tie-breaking, Euclidean distance, 10-fold CV. The pruned data was included in validation to increase available data for the test set.

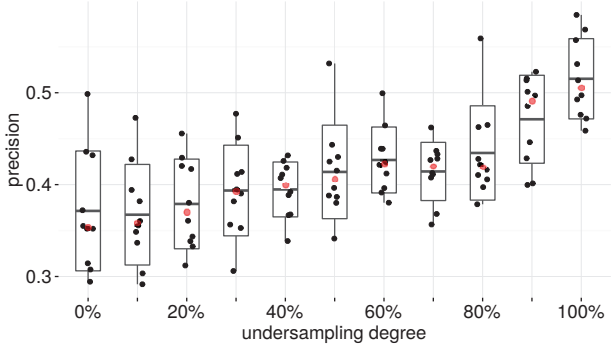



Figure 4.4 From left to right, decision regions for CN, MCI and AD, respectively.

Comparing the imbalance degree from figure 4.4 in which the RU degree vary from 0% to 100%, it is possible to see that there is a significant improvement when compared to parametric approach and random tie experiment, depicted in figures 2.6 and 3.7, respectively. Such improvement can be assessed in confusion

matrices depicted in figure 4.5. However, it is noticeable the trade-off between major an minor classes under imbalance aid.

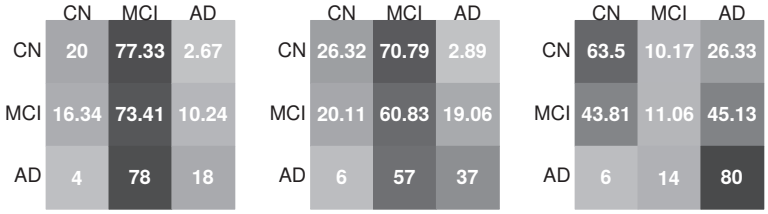



Figure 4.5 Respectively each plot shows the decision region for CN, MCI and AD and undersampled relative to the minor class from the major classes.

It is possible to reduce the class overlap using undersampling strategy by decreasing the major class decision region shown in figures 4.4 and 4.5, but losing precision in majority class. The confusion matrices in figure 4.5 show precision improvement in average from 37.13% (imbalanced) to 51.52% using 100% degree random undersampling. Table 4.1 shows the dataset-wise comparison for undersampled classification problems using 2-fold CV and LOOCV. The number of folds is not bigger because there are classification problems with only two patterns per class. Due to this let's set the undersampling degree in 50%.

Table 4.1 Dataset comparisons for random undersampling and two validation procedures.

| dataset | 2-fold CV (%) | k | LOOCV (%) | k |
|---|---|---|---|---|
| plasma | 38.78 ± 4.94 | 2 | 43.15 | 3 |
| psyconvert | 38.28 ± 0.69 | 4 | 38.72 | 7 |
| bloodabeta | 15.86 ± 0.31 | 14 | 16.44 | 7 |
| adni | 94.13 ± 1.33 | 25 | 94.46 | 22 |
| proteom | 43.87 ± 1.93 | 3 | 44.54 | 2 |
| neuropsy | 94.79 ± 3.46 | 24 | 95.16 | 7 |
| neuroimag | 51.46 ± 0.78 | 9 | 53.62 | 4 |
| parkinson | 89.18 ± 5.86 | 1 | 96.24 | 1 |
| average | 58.29 | - | 60.29 | - |

For every dataset, LOOCV and 2-fold CV precision is higher in average using RU (50% of degree) than imbalanced setup shown in table 3.2. The number of patterns to prune is based on the convex combination between the number of patterns in major and minor classes whereas degree is the new proportion to the major class. For instance, with a minor class being 100 and a major 414, the convex combination is,

$$(1 - degree) * 414 + degree * 100,$$

for undersampling degree of 50% the new major class is 257. Next section, another approach for the undersampling degree which considering structural data aspects is given.

## 4.4 Informed undersampling

The random undersampling technique randomly prunes data until it achieves the desired equilibrium between major and minor classes, however, as a drawback, it disregards data structural aspects. Alternatively, informed undersampling techniques try to overcome information loss introduced by the random undersampling using distance rules to prune. Examples of such techniques include *EasyEnsemble*, *BalanceCascade* [52], *Nearmiss* family of algorithms and *one-sided selection*, all discussed in [35] and from the best of our knowledge none applied in AD classification imbalance problems.

In the pioneer work of R. Dubey *et al* [20] done with imbalanced AD classification the following balance aid techniques are compared: random undersampling, random oversampling [35], $k$-medoids undersampling [63] and SMOTE oversampling [16]. As claimed by Dubey R. *et al* [20] $k$-medoids shows to be the balance aid strategy with higher overall classification measures among designed binary classification studies. Aside from the oversampling techniques discussed in next section, $k$-medoids is a cluster-based undersampling technique based on un-

supervised learning algorithm called $k$-means [12]. Distance-based algorithms, as *k-means* and *k-medoids*, uses properties of local approximation as kNN would use, then it is reasonable to think that metric-learning strategies can be performed in the context of sampling strategies. In object recognition field, an instance of such combination is given by Ebert S. *et al.* [22] by comparing several sampling strategies and learning a representative distance function to improve object recognition. Ebert S. *et al.* strategies was able to improve overall accuracy up to 23% for object recognition challenges.

Here, let's use the kNN which is a natural multiclass classifier to evaluate sampling strategies in ADNI datasets. Also the informative algorithms to be evaluated in this section are the *Nearmiss* family and one Nearmiss algorithm adapted from it.

### 4.4.1 Nearmiss undersampling

Let's explore informed undersampling techniques to increase minor classes precision [86], namely: *NearMiss-1* (N1), *NearMiss-2* (N2), *NearMiss-3* (N3), *Most distant* (MD), and a version of N3 named *Nearmiss-4* (N4).

• N1 [55], selects (keeps) the major class samples which are closest to minority class samples. The selected major class samples must have the smallest average distance to three closest patterns minor class.
• N2 [55], selects the major classes samples in their average distance whose three farthest minority samples are the smallest.
• N3 [55], removes a given number of the closest majority class samples for each minority class sample.
• N4, an experimental version of N3 in which is removed a given number among the farthest majority class samples for each minority class sample.

• MD [55], selects the majority class samples whose average distances to three closest minor class samples are the largest.

Let's describe the pseudo-code for Nearmiss4 which is an original contribution in this work:

**input**: data $D$, classes $\omega \in \Omega = \{\omega_1, ..., \omega_c\}$ , undersampling degree $d$, $L_p$-norm $p$, number of elements to be removed by each minority pattern $n$

$C \leftarrow$ major classes, $\Omega \backslash \omega_{\min}$

$\omega_{\min} \leftarrow$ minor class

**for all** $\omega \in C$ **do**

dist matrix $\leftarrow$ calculate distance matrix between pattern of $\omega$ and $\omega_{\min}$ for a given norm (or semi) $p$

degree number $\leftarrow floor[(1-d) * \sharp D(\omega) + d * \sharp D(\omega_{\min})]$

**while** $\sharp D(\omega) \geq$ degree number **do**

randomly choice a column $c$ in distance matrix

order column $c$ by decreasing order

remove the first $n$ elements relative to the greatest distances from class $\omega$

**end while**

**end for**

**if** $\sharp D(\omega) <$ degree number **then**

retrieve removed patterns from last $D$ in order to $\sharp D(\omega) =$ degree number

**end if**

**output** reduced $D$

The algorithm is written in R programming language and is available on line in github. For the NM3 and NM4 the number of elements to be removed by minority example is 2.

Nearmiss family original implementations [55] does not control the undersampling degree, here, instead, the algorithm continues to prune until achieves

the desired balance and repeated data selection/removal is not allowed as in the original implementation. Decision regions for Nearmiss family using *proteom* dataset are depicted in figures from 4.6 to 4.9. These mappings are solved with 5NN, Euclidean distance, random tie-breaking for undersampling degree 100%.

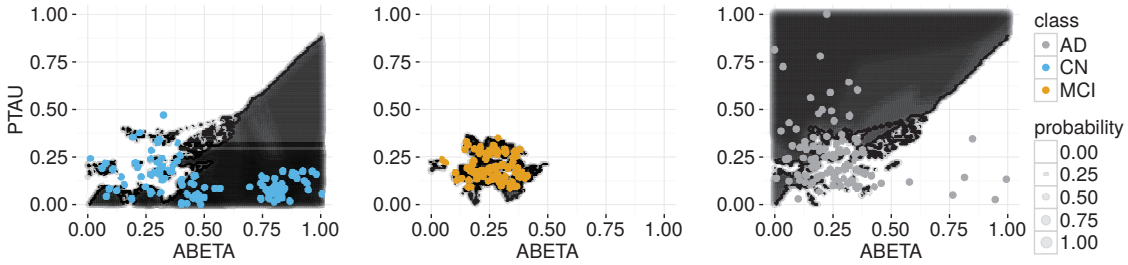



Figure 4.6 Decision regions for CN, MCI and AD respectively with N1.

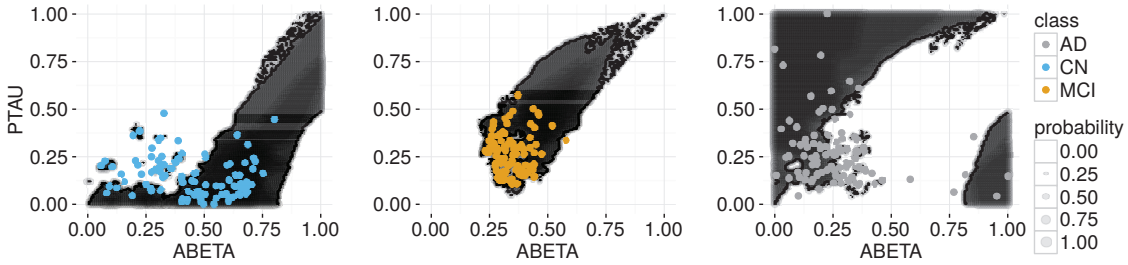



Figure 4.7 Decision regions for CN, MCI and AD respectively with N2.

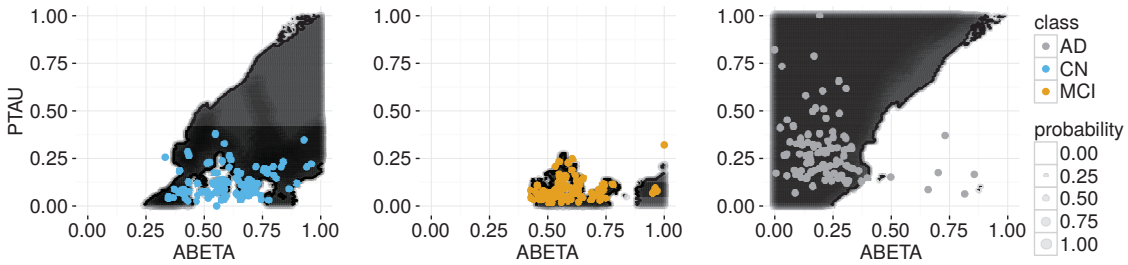



Figure 4.8 Decisions region for CN, MCI and AD respectively with N3.

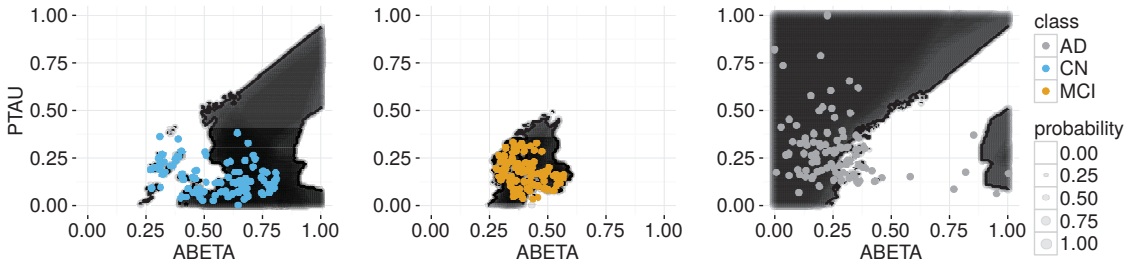



Figure 4.9 Decision regions for CN, MCI and AD respectively with N4.

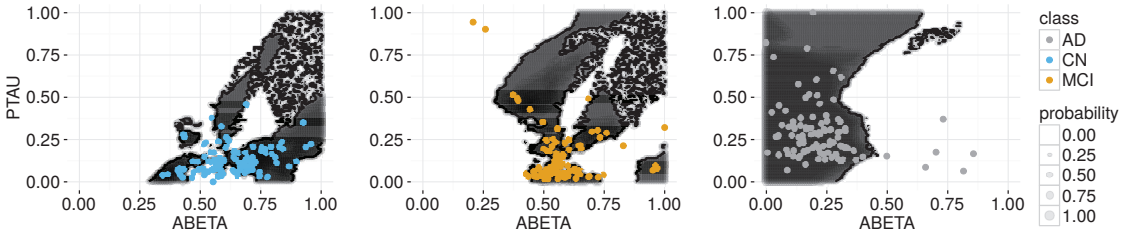



Figure 4.10 Decision region for CN, MCI and AD respectively with MD.

One can observe from figure 4.6 to 4.10 intermediary class becomes compacted reducing the class overlap, in the last section a comparison on sampling methods overlap is presented. Figures from 4.6 Using *proteom* dataset figures 4.11 (N1 and N2) and 4.12 (N3, N4, and MD) shows the effect when varying the undersampling degree. None of the informed undersampling includes the pruned data into the test set since they are structurally different, although the test set has the randomness of cross-validation folds. Here, instead of 5NN, when the undersampling degree is higher than 75% let's use 25NN.

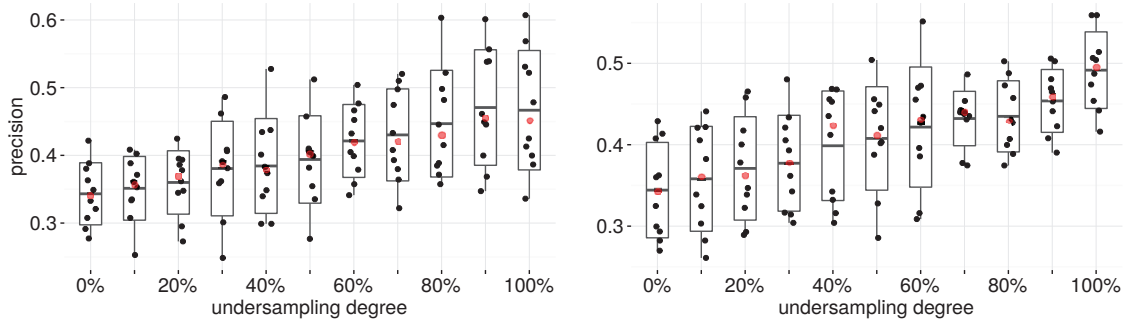



Figure 4.11 At left, 10-fold CV precision for N1. At right, 10-fold CV precision for N2. The red dot stands for the median value.

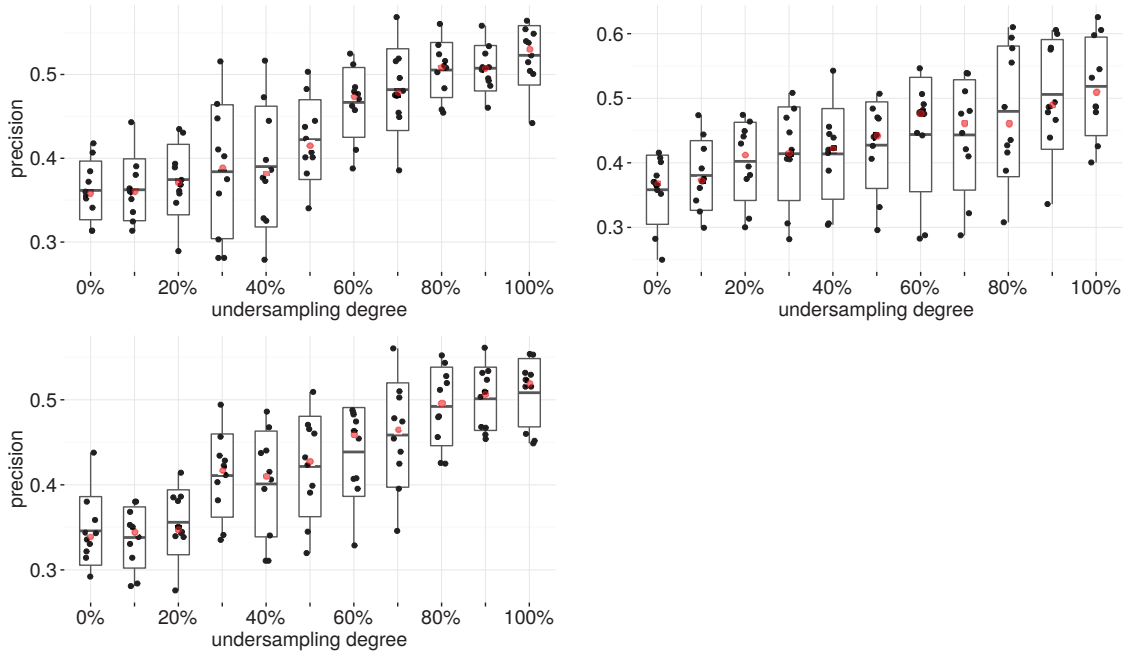



Figure 4.12 At first row left, 10-fold CV precision for N3. At first row right, 10-fold CV precision for N4. At second row, 10-fold CV precision for MD. The red dot stands for the median. value.

Figures from 4.13 to 4.17 depict confusion matrices for N1, N2 ,N3 ,N4 and MD, respectively.

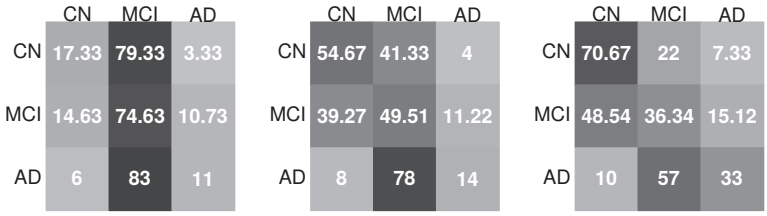



Figure 4.13 NM1 confusion matrices for 0%, 50% and 100% of undersampling degree with average precision of 34.32%, 39.39% and 46.67% respectively.

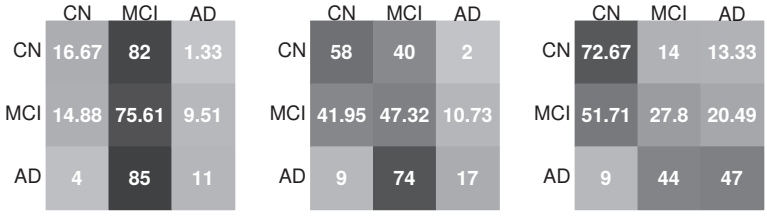



Figure 4.14 N2 confusion matrices for 0%, 50% and 100% of undersampling degree with average precision of 34.42%, 40.77% and 49.15% respectively.

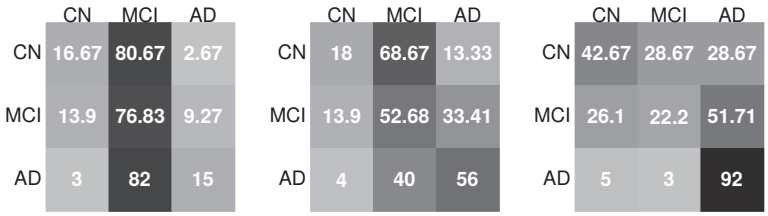



Figure 4.15 N3 confusion matrices for 0%, 50% and 100% of undersampling degree with average precision of 36.16%, 42.22% and 52.29% respectively.

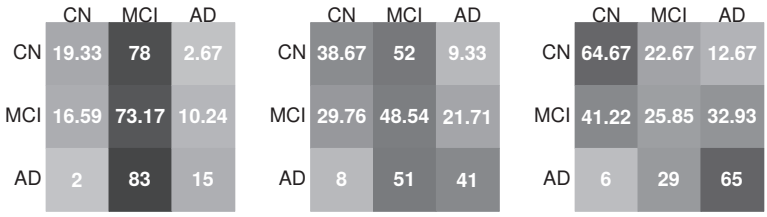



Figure 4.16 N4 confusion matrices for 0%, 50% and 100% of undersampling degree with average precision of 35.83%, 42.73% and 51.84% respectively.

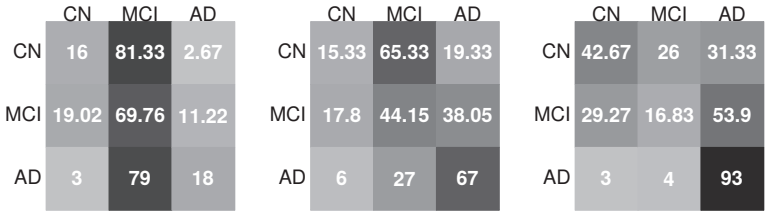



Figure 4.17 MD confusion matrices for 0%, 50% and 100% of undersampling degree with average precision of 34.58%, 42.16% and 50.83%.

Data-wise comparisons for the informed undersampling techniques are available in table 4.2 for two validation modalities (LOOCV and 2-fold CV). The undersampling degree to each class was 50% relative to the smallest class as explained earlier. When the minor class is small in such way the algorithm cannot be performed, for instance, NM1 needs three patterns in minor class to evaluate the average distance from the major class, the algorithm then uses only the number of patterns available to assess the average.

Table 4.2 Dataset comparison for informed undersampling.

| dataset | N1 2-fold CV (%) | $k$ | N1 LOOCV (%) | $k$ | N2 2-fold CV (%) | $k$ | N2 LOOCV (%) | $k$ |
|---|---|---|---|---|---|---|---|---|
| plasma | 35.92 ± 0.59 | 16 | 36.48 ± 0.22 | 7 | 29.59 ± 0.88 | 7 | 36.44 ± 0.28 | 8 |
| psyconvert | **40.28 ± 2.62** | 1 | 40.03 | 1 | 29.94 ± 0.23 | 19 | 40.03 | 1 |
| bloodabeta | 16.98 ± 0.28 | 5 | 16.40 ± 0.07 | 17 | 15.57 ± 0.56 | 7 | 16.52 ± 0.14 | 21 |
| adni | 74.94 ± 2.23 | 3 | 90.78 ± 0.01 | 14 | 72.00 ± 0.12 | 1 | 90.96 ± 0.23 | 14 |
| proteom | 45.23 ± 2.93 | 2 | 44.12 ± 0.47 | 15 | 42.72 ± 5.14 | 10 | 43.20 ± 0.18 | 11 |
| neuropsy | 85.01 ± 7.02 | 3 | 88.76 | 3 | 79.03 ± 1.48 | 2 | 88.04 | 11 |
| neuroimag | 50.94 ± 0.23 | 15 | 54.03 ± 0.21 | 5 | 54.85 ± 2.34 | 3 | 55.10 ± 0.55 | 4 |
| parkinson | 87.89 ± 9.34 | 1 | 97.26 | 1 | 84.13 ± 15.10 | 1 | 97.26 | 1 |
| average | 54.64 | - | 58.48 | - | 50.97 | - | 58.44 | |

| dataset | N3 2-fold CV (%) | $k$ | N3 LOOCV (%) | $k$ | N4 2-fold CV (%) | $k$ | N4 LOOCV (%) | $k$ |
|---|---|---|---|---|---|---|---|---|
| plasma | **43.20 ± 3.96** | 1 | 35.57 ± 0.03 | 3 | 33.02 ± 3.20 | 2 | 36.58 ± 0.54 | 7 |
| psyconvert | 34.07 ± 10.35 | 2 | 36.24 ± 0.02 | 11 | 30.33 ± 0.61 | 20 | 36.35 ± 0.13 | 7 |
| bloodabeta | 20.70 ± 7.72 | 2 | 18.91 ± 7.97 | 3 | 15.50 ± 0.54 | 7 | 16.28 ± 0.10 | 32 |
| adni | 73.82 ± 2.83 | 5 | 94.52 ± 0.06 | 15 | 73.40 ± 0.34 | 8 | 92.57 | 21 |
| proteom | 47.03 ± 0.36 | 21 | 44.83 ± 0.28 | 19 | 47.29 ± 0.59 | 1 | 45.73 ± 0.35 | 13 |
| neuropsy | 82.35 ± 0.58 | 1 | 95.33 ± 0.41 | 20 | 78.25 ± 3.78 | 3 | 90.96 | 3 |
| neuroimag | 55.14 ± 2.98 | 5 | 55.48 ± 0.34 | 4 | 50.14 ± 6.54 | 23 | 55.55 ± 0.16 | 5 |
| parkinson | 87.97 ± 3.19 | 1 | 91.50 | 1 | 90.14 ± 0.72 | 1 | 97.26 | 1 |
| average | 55.53 | - | **59.04** | - | 52.25 | - | 58.91 | |

| dataset | MD 2-fold CV (%) | $k$ | MD LOOCV (%) | $k$ |
|---|---|---|---|---|
| plasma | 41.34 ± 1.08 | 1 | 33.83 ± 0.21 | 11 |
| psyconvert | 34.94 ± 10.29 | 2 | 33.73 ± 0.14 | 10 |
| bloodabeta | **23.24 ± 10.41** | 3 | 16.16 ± 0.08 | 19 |
| adni | 73.63 ± 3.77 | 1 | **94.59 ± 0.53** | 16 |
| proteom | **49.65 ± 1.68** | 6 | 47.10 ± 1.39 | 2 |
| neuropsy | 81.57 ± 2.85 | 9 | **95.81** | 25 |
| neuroimag | **56.47 ± 0.63** | 4 | 56.34 ± 0.27 | 5 |
| parkinson | 87.08 ± 3.69 | 1 | 90.82 | 1 |
| average | 55.99 | - | 58.54 | - |

The sampling techniques' goal is to increase the minority samples, however, without losing precision in other classes. Thus, the macro-average precision is suitable since it not benefits the major class as accuracy does. For instance, the review paper [35] in which sampling techniques for binary classification are presented, the following measures are claimed to be the most used to evaluate imbalanced prob-

lems: accuracy, recall, F-Measure, G-mean. Such measures to compare imbalance solutions are not appropriate as shown by [81] since they are originally created for binary problems and not account for multiclass interactions, e.g. classifier degradation which tends to be caused by multi major classes. In order to illustrate the class precision imbalance let's suggest a measure as being the average difference between class precision. Instead of measuring the degree of exactness, it simply evaluates the average difference between the class probability of correct classified samples (e.g. accuracy, precision). Moreover, for such measure, the classifier would achieve 0% for all classes and be considered "fair" since this regards only to the difference between classes. As shown in table 4.2 the MD algorithm reaches most of the high precision whereas NM3 reaches the highest average. However, comparing the average difference between precision for the *proteom* dataset classification problem one may observe NM3 and MD, indeed, increase precision for minor classes but creating a precision difference higher than the imbalanced version. That is, the imbalanced problem achieves 47.50% of average difference precision using random tie-breaking, with RU it is reduced to 45.96%, with MD undersampling it increases back to 50.78%. For NM3 which achieves second higher average in table 4.2 the average difference precision is 46.31%. Since all informed undersampling algorithms here uses pair-wise relations to prune data minor classes' interactions are not accounted leading to increased distance between class individual scores and minor class overlapping as depict in figures 4.8 and 4.10. A comparison between all balance aid techniques by means of overlap will be presented at end of this chapter.

## 4.5 Synthetic oversampling methods

Instead of pruning data as the under-sampling algorithms one would like to raise the low priors of the minor classes by adding synthetic samples. The oversampling techniques are designed to deal with imbalanced problems by modifying the training balance artificially. The strategies of oversampling techniques discussed

here use distances in order to insert synthetic samples in training set. Let's show in AD classification problems the well-known oversampling algorithms, SMOTE (Synthetic Minority Over-sampling TechniquE) and three of its derivations. Let's show also how the oversampling parameters affect the original distribution and how it differ from the undersampling approach.

Basically, SMOTE evaluates the $k$-nearest neighbors of the minority classes and add a synthetic data between a random pair of the $k$ neighbors. This process is repeated until it achieves the desired proportion between classes. For details regarding the algorithm see [16]. The algorithm uses as entries the number of $k$ neighbors and the proportion to be reached. As higher is the number of neighbors, $k$, more distant the synthetic data would be included. Figure 4.18 shows how the $k$ parameter modifies the original distribution when synthetic data is included in AD class by means of Bhattacharyya coefficient and assuming Gaussianity.

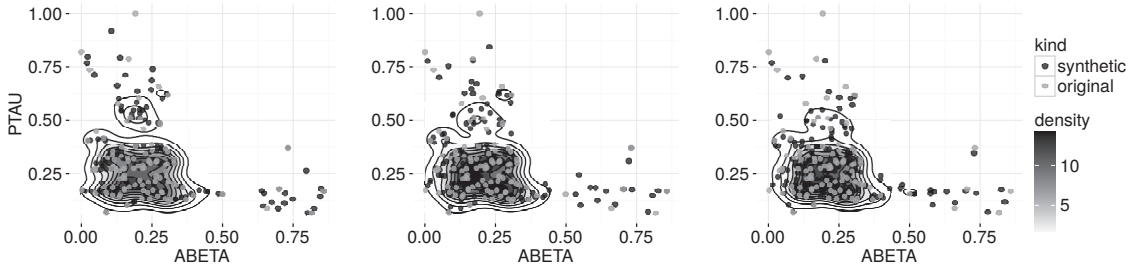



Figure 4.18 From left to right the synthetic data generated with SMOTE using parameter $k = 7$, $k = 15$ and $k = 30$, respectively. The Bhattacharyya coefficients are, 0.9719190, 0.9881616 and 0.9943559, respectively.

Using Bhattacharyya coefficient there is a small ($< 0.03\%$) difference between original and SMOTE pre-processed distributions shown in figure 4.18. However, when the parameter $k$ is increased the synthetic samples tend to be placed distantly. Figure 4.19 shows the effect of proportion control, this is, the inclusion of synthetic data relative to the difference between the number of major and minor classes samples.

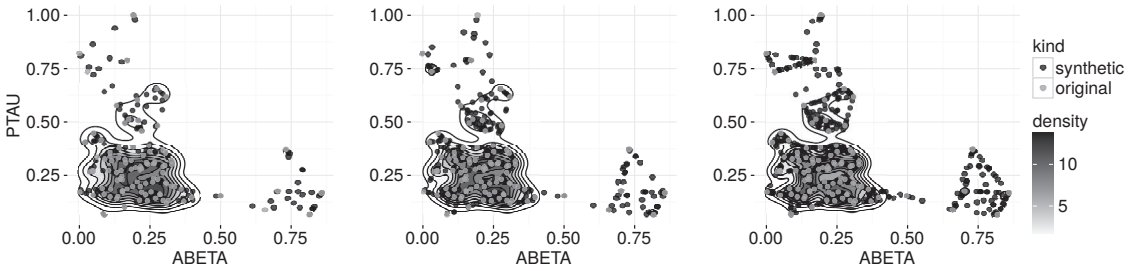



Figure 4.19 From left to right the synthetic data generated with SMOTE using parameter $k = 5$ and proportions, 150%, 300% and 500%, respectively. The Bhattacharyya coefficients are, 0.9970899, 0.9968130 and 0.9961496, respectively.

It is noticeable by the Bhattacharyya coefficient and varying the synthetic inclusion degree that SMOTE does not promotes great changes in original distribution (figure 4.19). However, SMOTE is considered a blind over-sample because it is possible to propagate noise samples compromising precision and class' decision regions. In order to avoid propagation of noise (or outliers) samples a variation of SMOTE, called *SMOTE-borderline*, is used. Such technique inserts synthetic samples only near to instances that have increased chances to be misclassified in comparison to remaining, such patterns belonging to the minority classes are called *borderlines*. The SMOTE-borderline variations discussed here are:

• *SMOTE-borderline1* (S1) identifies borderline instances in minor class and add synthetic samples between borderline and nearest patterns of its class;
• *SMOTE-borderline2* (S2) identifies borderline instances in minor class and add synthetic samples between borderline and nearest patterns in majority class;
• *SMOTE-borderline3* (S3), an S1 modification, named here as S3, which identifies borderline instances in minor class and adds synthetic samples between borderline and nearest patterns in all dataset.

There is no requirement to include S3 pseudo-code, it can be understood from pseudo-code SMOTE borderline described in [31]. Next is presented a comparison between SMOTE (S), S1, S2 and S3 for the *proteom* dataset. Figures from 4.20 to 4.23 show the decision regions for the classification problem solved

with 5NN, random tie-breaking and min-max normalization for each oversampling strategy and the oversampling degree 100%.

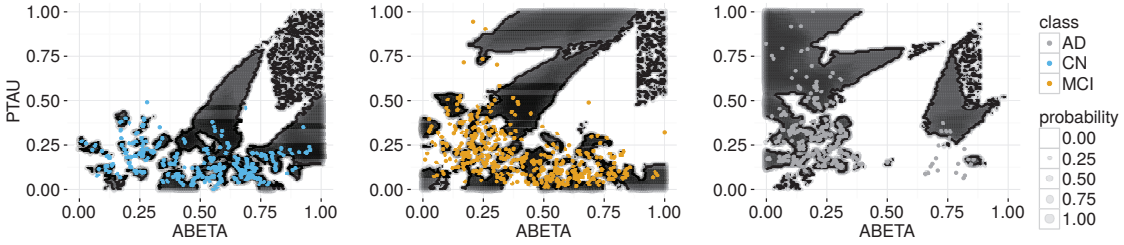



Figure 4.20 Decisions region using S for classes CN, MCI and AD, respectively.

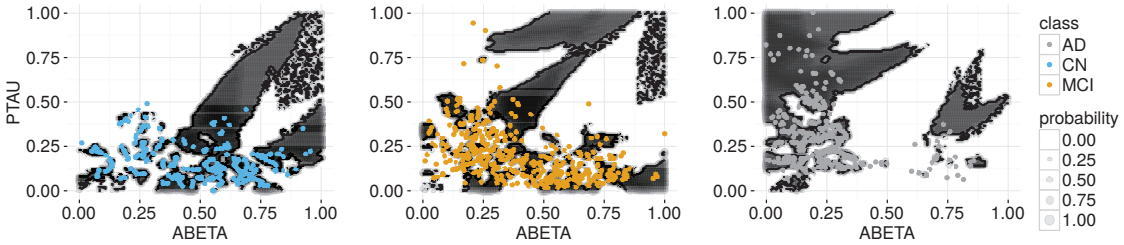



Figure 4.21 Decisions region by S1 for classes CN, MCI and AD, respectively.

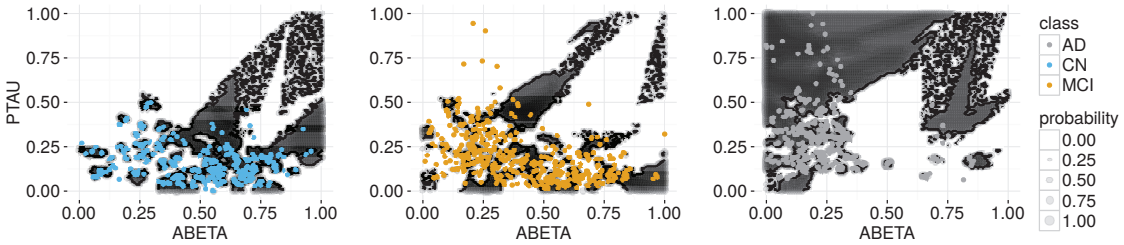



Figure 4.22 Decisions region by S2 for classes CN, MCI and AD, respectively.

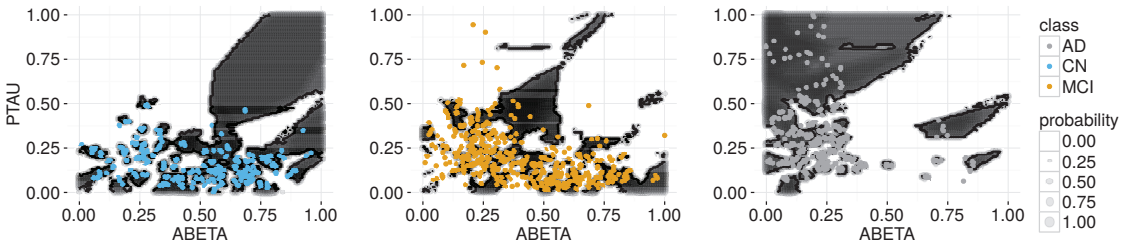



Figure 4.23 Decisions region by S3 for classes CN, MCI and AD, respectively.

Figure 4.24 shown a comparison of 10-fold CV precision for the oversampling techniques varying the degree between 0% to 150%, the $k$ value is changed from $k = 5$ to $k = 20$ when degree is higher than 75%.

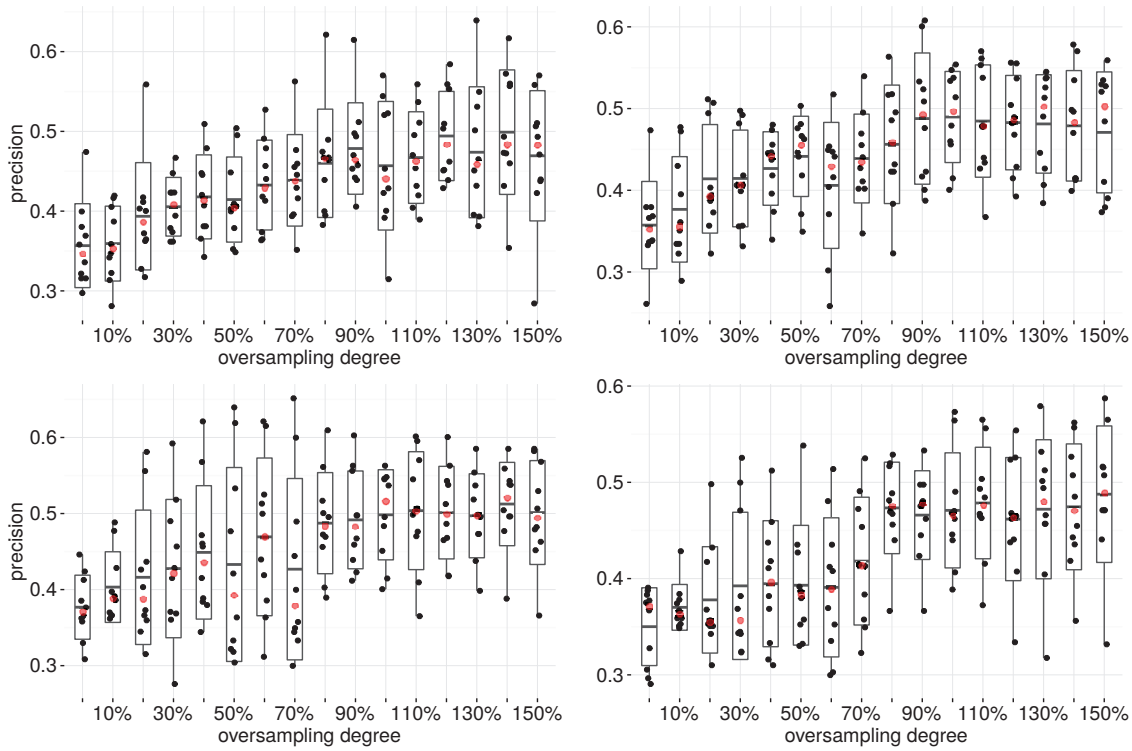



Figure 4.24 From left to right at first row, the 10-fold CV outcomes for SMOTE and SMOTE-borderline1, respectively. From left to right at second row, the 10-fold CV outcomes for SMOTE-borderline2 and SMOTE-borderline3, respectively. Red dot corresponds to for the median value.

One can notice in above figures that standard deviation for SMOTE's family algorithms are higher than previous undersampling and imbalanced instances, synthetic samples would account for such effect [68]. Conversely, LOOCV precision is capable of reducing this effect as it will be shown. Beyond 100% of the oversampling degree, all SMOTE techniques become noneffective for this choice of $k$. Indeed, for such cases above 100%, the majority classes turn into the minority ones causing 10-fold CV precision to be stagnated. Further data-wise comparison gives shows precision and standard deviation for AD classification problems. Figures from 4.25 to 4.28 depicts the 10-fold CV confusion matrices for oversampling algorithms and 0%, 50%, 100% and 150% of oversampling degrees.

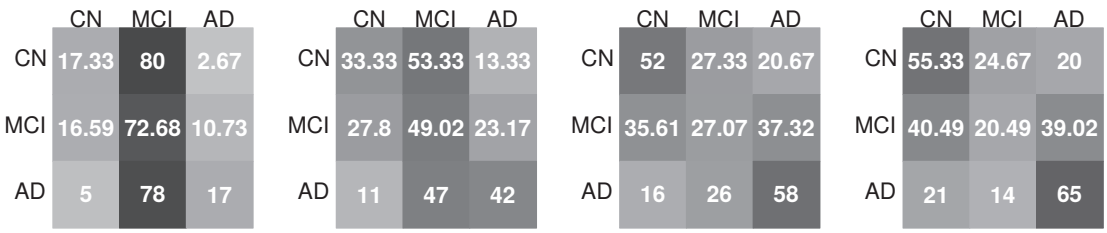



Figure 4.25 From left to right confusion matrices obtained by S with 35.67%, 41.45%, 45.69% and 46.27% of 10-fold CV precision, respectively.

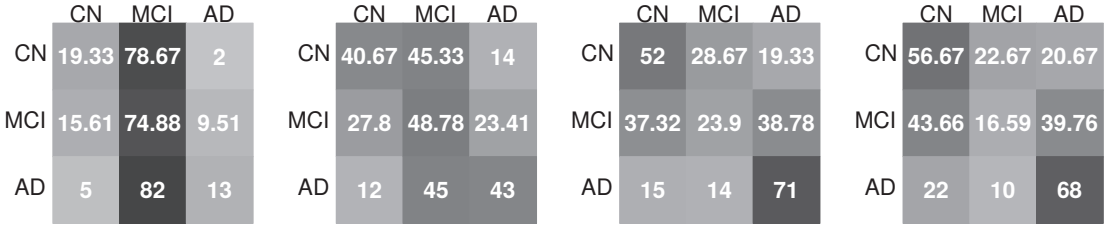



Figure 4.26 From left to right confusion matrices obtained by S1 with 35.73%, 44.15%, 48.96% and 47.08% of 10-fold CV precision respectively.

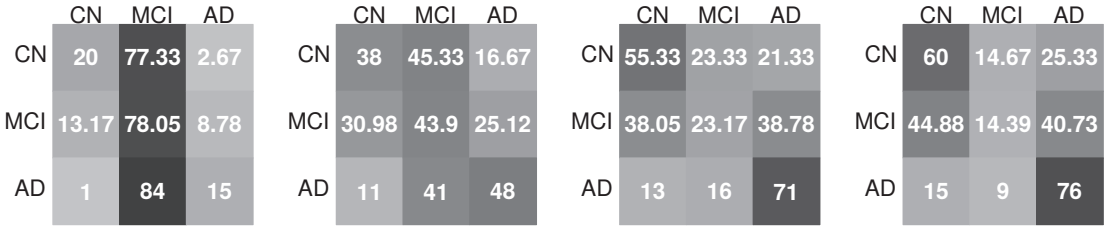



Figure 4.27 From left to right confusion matrices obtained by S2 with 37.68%, 43.03%, 49.83% and 50.13% of 10-fold CV precision, respectively.

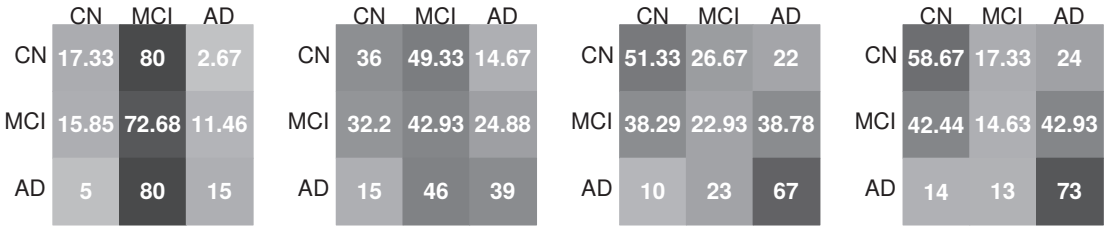



Figure 4.28 From left to right confusion matrices obtained by S3 with oversampling degree of 0%, 50%, 100% and 150%, respectively. The precision achieved is 35.00%, 39.31%, 47.08% and 48.76%, respectively.

Evaluating the average class precision difference, each oversampling technique achieves for 100% of oversampling degree, 20.62%, 31.67%, 31.88%, and 29.68%, respectively for S, S1, S2, and S3.

Table 4.3 shows data-wise comparison for algorithms S, S1, S2 and S3. All datasets use previously estimated parameters for $p$-norm and normalization. The comparisons are done using 2-fold CV and LOOCV. Since the SMOTE's family needs at least two samples in training set generate synthetic ones the classes with only that are not suitable for SMOTE generates perturbed values around such pattern.

Table 4.3 Datasets with oversampling degree of 50% and optimized $k$.

| dataset | S 2-fold CV (%) | k | S LOOCV (%) | k | S1 2-fold CV (%) | k | S1 LOOCV (%) | k |
|---|---|---|---|---|---|---|---|---|
| plasma | **44.20 ± 1.97** | 22 | 39.83 ± 0.03 | 3 | 44.47 ± 1.31 | 3 | 42.52 ± 0.25 | 3 |
| psyconvert | **47.94 ± 8.99** | 31 | 45.84 ± 0.16 | 18 | 46.16 ± 5.52 | 32 | 45.42 ± 0.11 | 16 |
| bloodabeta | 14.47 ± 1.56 | 3 | 13.23 ± 0.31 | 3 | 15.25 ± 0.37 | 1 | 14.83 | 1 |
| adni | **94.49 ± 2.45** | 25 | 81.25 ± 0.06 | 17 | 94.05 ± 2.80 | 24 | 81.00 | 7 |
| proteom | 52.35 ± 0.58 | 22 | 46.53 ± 0.18 | 19 | **54.29 ± 0.02** | 24 | 43.92 ± 0.20 | 25 |
| neuropsy | 95.05 ± 3.58 | 24 | 90.24 | 9 | **96.02 ± 0.78** | 20 | 89.99 | 5 |
| neuroimag | 58.45 ± 0.11 | 21 | 56.56 ± 0.16 | 21 | **59.12 ± 2.64** | 13 | 54.55 ± 0.28 | 21 |
| parkinson | 94.51 ± 0.99 | 1 | 96.92 | 1 | 87.89 ± 3.41 | 1 | 97.26 | 1 |
| average | **62.68** | - | 58.79 | - | 62.15 | - | 58.68 | - |

| Dataset | S2 2-fold CV (%) | k | S2 LOOCV (%) | k | S3 2-fold CV (%) | k | S3 LOOCV (%) | k |
|---|---|---|---|---|---|---|---|---|
| plasma | 38.06 ± 4.09 | 6 | 39.93 ± 0.04 | 20 | 39.00 ± 2.74 | 1 | 40.57 ± 0.08 | 17 |
| psyconvert | 46.49 ± 7.91 | 2 | 39.48 ± 0.19 | 12 | 41.90 ± 6.97 | 5 | 39.05 ± 0.22 | 16 |
| bloodabeta | 15.57 ± 1.52 | 1 | 15.58 ± 0.07 | 2 | **15.86 ± 1.40** | 3 | 14.21 ± 0.21 | 3 |
| adni | 93.40 ± 3.51 | 9 | 79.92 | 1 | 93.63 ± 1.98 | 18 | 80.63 | 1 |
| proteom | 53.09 ± 1.28 | 25 | 50.34 ± 0.80 | 25 | 48.77 ± 1.92 | 25 | 50.74 ± 0.74 | 22 |
| neuropsy | 95.27 ± 2.13 | 17 | 88.06 ± 0.02 | 10 | 95.11 ± 0.47 | 12 | 89.32 | 3 |
| neuroimag | 57.37 ± 1.01 | 21 | 56.74 ± 0.49 | 24 | 57.48 ± 3.22 | 22 | 58.01 ± 0.41 | 25 |
| parkinson | 90.31 ± 7.85 | 1 | 97.26 | 1 | 94.83 ± 5.37 | 1 | 97.26 | 1 |
| average | 61.19 | - | 58.41 | - | 60.82 | - | 58.72 | - |

Despite SMOTE not shown the highest precision among other oversampling techniques for 100% of oversampling degree, as depicted in figure 4.25, it achieves the highest average for 10-fold CV and LOOCV in data-wise comparison for 50% of oversampling degree. SMOTE's family average precision in 2-fold CV achieves higher results than RU, however, for LOOCV the dominance does not hold

since random undersampling achieves higher average precision. Despite the oversampling techniques have a higher average precision for 2-fold CV validation, the standard deviation is discouraging.

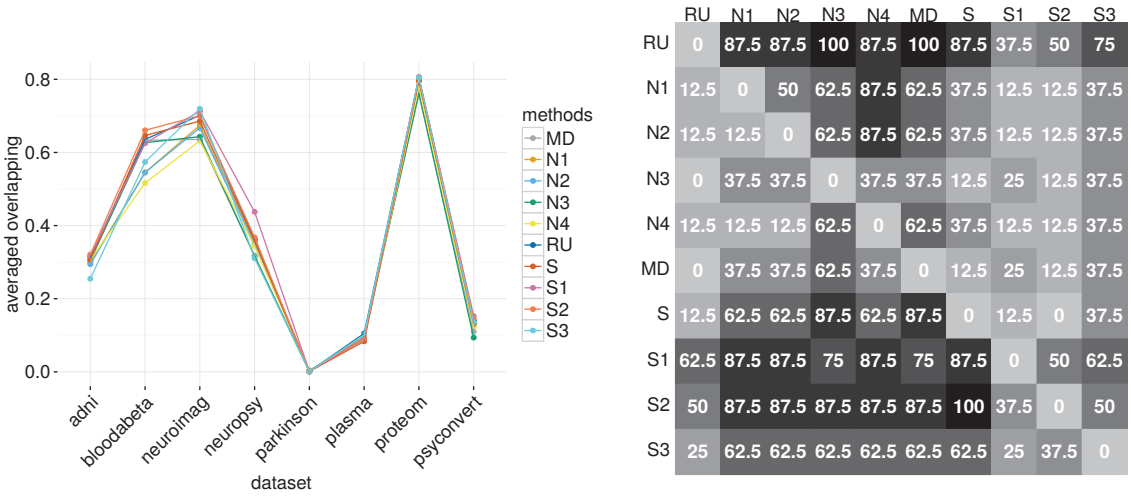
Comparing all previous tables one may observe that validation scheme is one of the main factors to decide the most precise sampling technique (in average) [68]. As claimed in [68], such factors rather than imbalance ratio and validation scheme is the amount of overlapping between classes and other distribution characteristics [26]. In order to assess the overlapping amount driven by such methods, a comparison is made in next section.

## 4.6 Imbalance strategies comparison

The oversampling and undersampling are supposed to be equivalent since they both modifies the original dataset by providing a balance solution. However, each method adds its own biases and problematic consequences, for instance, noise increasing by synthetic samples in oversampling, reduced training set for RU, discarded pruned data since informed undersampling changes training set structure. Not only imbalance degree affects learning, but the class overlap is an important factor to prevent classifier deterioration. It is possible to measure the degree of overlap using caused by sampling methods by using Bhattacharyya coefficient (assuming Gaussianity). However, for extreme minor classes accounting with only two samples, e.g. converters datasets, the covariance matrices no meaning. Also, overlapping is not the only source of error in imbalanced setup as claimed in [68] and [26], with class distribution parameters being decisive as well.

In figure 4.29 a dataset comparison is presented for averaged class overlap caused by the sampling techniques and its probability matrix. All sampling methods had its degree settled to 50% and the datasets were normalized. Covari-

ance matrices which become singular evaluating Bhattacharyya distance were not included in the average ranging from 0 to 1.




| | RU | N1 | N2 | N3 | N4 | MD | S | S1 | S2 | S3 |
|---|---|---|---|---|---|---|---|---|---|---|
| RU | 0 | 87.5 | 87.5 | 100 | 87.5 | 100 | 87.5 | 37.5 | 50 | 75 |
| N1 | 12.5 | 0 | 50 | 62.5 | 87.5 | 62.5 | 37.5 | 12.5 | 12.5 | 37.5 |
| N2 | 12.5 | 12.5 | 0 | 62.5 | 87.5 | 62.5 | 37.5 | 12.5 | 12.5 | 37.5 |
| N3 | 0 | 37.5 | 37.5 | 0 | 37.5 | 37.5 | 12.5 | 25 | 12.5 | 37.5 |
| N4 | 12.5 | 12.5 | 12.5 | 62.5 | 0 | 62.5 | 37.5 | 12.5 | 12.5 | 37.5 |
| MD | 0 | 37.5 | 37.5 | 62.5 | 37.5 | 0 | 12.5 | 25 | 12.5 | 37.5 |
| S | 12.5 | 62.5 | 62.5 | 87.5 | 62.5 | 87.5 | 0 | 12.5 | 0 | 37.5 |
| S1 | 62.5 | 87.5 | 87.5 | 75 | 87.5 | 75 | 87.5 | 0 | 50 | 62.5 |
| S2 | 50 | 87.5 | 87.5 | 87.5 | 87.5 | 87.5 | 100 | 37.5 | 0 | 50 |
| S3 | 25 | 62.5 | 62.5 | 62.5 | 62.5 | 62.5 | 62.5 | 25 | 37.5 | 0 |

Figure 4.29 At left, by dataset the averaged overlapping between different classes caused by each resampling method. At right, the matrix of probabilities for a given resampling method (line) to create more overlap than other method (column) along datasets.

In figure 4.29 (right), the row average in probability matrix which allows the highest overlapping, in order is: 79.16 % (RU), 75.00% (S1) 75.00% (S2) 51.38% (S3), 47.22% (S), 41.66% (N1), 37.50% (N2), 29.16% (N4), 29.16% (MD) and 26.38% (N3). However, average class overlap has its drawbacks, e.g. as decreasing artificially the overlap. For instance, considering three classes as the problem depicted in figure 4.10, the overlapping is all concentrated between two classes while the third class (the minor one), become distant, the average will shown one high overlap between major classes and two low overlaps between minor and remaining major classes. Instead of average class overlap by one-against-one let's use one-against-all average class overlap. Assessing the same matrix, for one-against-all average overlap, in order we have: 84.72% (S1), 79.16% (S2), 72.22% (RU), 55.55% (S3), 55.55% (S), 40.27% (MD), 33.33% (N1), 26.38% (N3,) 25.00% (N2) and 19.44% (N4). In both situations, informed undersampling achieves less overlap. Also, N4 algorithm obtain a reasonable separation between classes and higher precision than imbalanced setup.

In figure 4.30 the probability matrices calculated along the dataset comparison shows the probability of one method provides higher precision than other.

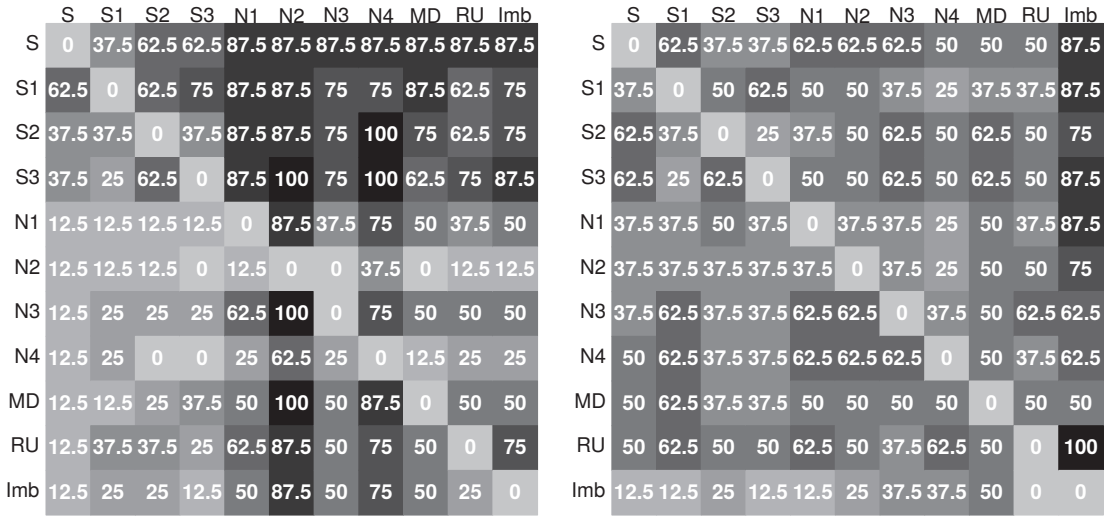



Figure 4.30 Probability matrices for the sampling methods precision compared to the random tie-break kNN ("Imb"). At left, 2-fold CV, at right LOOCV.

It is noticeable that oversampling techniques achieves higher precision for 2-fold CV. For LOOCV the RU achieves higher precision than use only kNN without imbalance aids. The difference between sampling methods is evident in the probability matrix for 2-fold CV precision. The line average probability matrix without "Imb" shows that SMOTE's family have the higher probability to provide a higher precision than remaining sampling methods. Whereas RU presents higher line average than informed undersampling. For LOOCV probability matrix, evidence of the best strategy is tight between oversampling and informed undersampling, but oversampling techniques surpasses. Regarding the improvement compared to the imbalanced problem for LOOCV all methods are effective (except from MD that has the same probability of "Imb").

Table 4.4 compares 10-fold CV and LOOCV precision for kNN with sampling methods, metric and normalization searches against others classifiers from R language packages, namely: SVM from *e1071* package; conditional inference trees

(ctree) from *party* package; bootstrap aggregating (bagging) from *ipred* package; learning vector quantization (lvq) from *class* package; wkNN from *kknn* package; and kNN tuned with experiments described in past sections. Respectively by dataset in table 4.4 (upper), the kNN highest LOOCV precision was achieved by the following sampling methods: RU (plasma), S (psyconvert), N3 (bloodabeta), MD (adni), S3 (proteom), MD (neuropsy), S3 (neuroimag), S1 (parkinson).

Table 4.4 Classifier comparison for LOOCV (upper table) and 2-fold CV (bottom table).

| dataset | SVM (%) | ctree (%) | bagging (%) | lvq (%) | wkNN (%) | kNN (%) |
|---|---|---|---|---|---|---|
| plasma | 36.45 | 37.20 | 37.88 | 29.81 | 36.14 | **43.15** |
| psyconvert | 34.14 | 37.08 | 37.92 | 32.96 | 38.03 | **45.84 ± 0.16** |
| bloodabeta | 16.21 | 14.29 | 14.23 | 14.72 | 15.04 | **16.52 ± 0.14** |
| adni | 71.13 | 94.10 | 94.02 | 58.23 | 85.87 | **94.59 ± 0.53** |
| proteom | 34.86 | 33.33 | 38.58 | 38.77 | 33.07 | **50.74 ± 0.74** |
| neuropsy | 92.30 | 94.10 | 93.71 | 67.65 | 92.31 | **95.81** |
| neuroimag | 42.20 | 47.25 | 43.10 | 44.62 | 46.67 | **58.01 ± 0.41** |
| parkinson | 65.62 | 74.02 | 85.80 | 69.92 | 93.43 | **97.26** |

| dataset | SVM (%) | ctree (%) | bagging (%) | lvq (%) | wkNN (%) | kNN (%) |
|---|---|---|---|---|---|---|
| plasma | 35.81 ± 2.65 | 37.23 ± 0.90 | 38.03 ± 1.10 | 32.32 ± 0.38 | 34.58 ± 0.93 | **44.47 ± 1.31** |
| psyconvert | 34.63 ± 1.26 | 37.42 ± 0.32 | 38.90 ± 0.35 | 38.46 ± 0.06 | 36.62 ± 0.38 | **47.94 ± 8.99** |
| bloodabeta | 14.85 ± 0.04 | 14.29 | 15.89 ± 0.06 | 14.99 ± 2.23 | 16.09 ± 0.19 | **23.24 ± 10.41** |
| adni | 61.48 ± 0.73 | 93.96 ± 1.12 | 94.36 ± 4.73 | 66.23 ± 4.84 | 83.76 ± 2.21 | **94.49 ± 2.45** |
| proteom | 36.42 ± 1.46 | 33.33 | 38.42 ± 1.88 | 42.61 ± 4.24 | 35.01 ± 5.18 | **54.29 ± 0.02** |
| neuropsy | 90.30 ± 1.33 | 92.48 ± 6.70 | 95.95 ± 0.36 | 72.23 ± 3.22 | 90.78 ± 0.29 | **96.02 ± 0.78** |
| neuroimag | 42.50 ± 1.62 | 41.10 ± 1.30 | 45.56 ± 3.12 | 46.61 ± 5.39 | 43.66 ± 1.10 | **59.12 ± 2.64** |
| parkinson | 65.62 ± 4.42 | 77.14 ± 0.97 | 80.96 ± 2.50 | 66.68 ± 1.96 | 87.89 ± 3.45 | **94.83 ± 5.37** |

The kNN using resampling, a suitable metric and normalization, was able to surpass all classifiers in average, only without statistical significance was the *adni* dataset. Results described in table 4.4 (bottom) shows 2-fold CV precision compared to previous classifiers. Respectively by dataset in table table 4.4 (bottom), the kNN highest 2-fold CV precision was achieved by the following sampling methods: S1 (*plasma*), S (*psyconvert*), MD (*bloodabeta*), S (*adni*), S1 (*proteom*), S1 (*neuropsy*), S1 (*neuroimag*), S3 (*parkinson*). Despite kNN achieves higher precision in average, most of the result for this validation modality are from oversampling

techniques which include increased standard deviation bias implying in significance loss. Results in table 4.4 (bottom) are statistically significant only for the following dataset, *plasma*, *proteom* and *neuroimag*.

In summary, this chapter compares two sampling strategies, oversampling and undersampling, and the main drawbacks for each one. Also, two new algorithms were proposed, NM4 and S3. Whereas the S3 did not show especial attributes, the NM4 shows properties relative to reduce overlap the between classes. The kNN with modifications shows precision in average higher than popular algorithms as listed. However, overlapping induced by oversampling techniques in 2-fold CV may increase the standard deviation leaving the LOOCV a safer option. The balance aid steps can be applied to any classifier in order to improve its classification rate. Regarding the normalization, metric optimization and the possible parameter of balance aid techniques, non-greedy strategies are required to have a better impact of such factor combined. Next chapter presents feature selection techniques in order to compare AD biomarkers discriminative power.

# 5 WRAPPERS FEATURE SELECTION

## 5.1 Introduction

Biomarkers are a characteristic that is objectively measured and evaluated as an indicator of normal biological processes, pathogenic processes or pharmacological responses to a therapeutic intervention [77]. The combination of different biomarker modalities often allows an accurate diagnosis classification [25]. In AD, biomarkers are indispensable to identify cognitively normal individuals destined to develop dementia symptoms. However, using the combination of canonical AD biomarkers, studies have repeatedly shown poor classification rates to differentiating between AD, mild cognitive impairment and control individuals. Furthermore, the design of classifiers to assess multiple biomarker combinations includes issues such as imbalance classes and missing data. Since the number biomarker combinations is large then wrappers feature selection can be used to avoid multiple comparisons [46].

In this chapter is compared the ability of three wrappers feature selection methods to obtain the biomarker combinations which maximize precision. Also, as a criterion for the wrappers feature selection is used the kNN classifier with sampling techniques from the previous section. Overall, the analyses presented in this chapter shows how biomarkers combinations affect the classifier precision and how sampling strategy improve it. Is shown also, that non-defining and non-cognitive biomarkers from dataset described in table 6.1 have less precision than cognitive measures when classifying AD and among the non-cognitive which is the biomarker that provides higher precision when combined.

It is a well-established fact that combined biomarkers provide higher classification rates than single biomarkers [61]. In this regard, neuropsychological

tests associated with different biomarker modalities have been used to classify AD [78]. These studies have combined PET, MRI, fMRI as well as CSF and blood biomarkers to perform binary classifications of AD [25], e.g. healthy versus unhealthy individuals. Despite AD's classification problem being an inherently multiclass binary classification or multiclass driven by binary strategies are the rule [25]. Since some classifiers are naturally binary they must be adapted for multiclass by means of one-versus-all or one-versus-one strategies, e.g. SVM [12]. Thus, to solve an $n$-class problem using binary classifiers, $\frac{n(n-1)}{2}$ rules are required to build a multiclass classifier. As benefits, binary classifiers are well suited for the receiver ROC analysis which has been largely applied in comparative studies and model selection [24].

### 5.1.1 Related work

Recently, various approaches used for AD's identification have achieved successful results and satisfactory classification rates. For instance, Khedher *et al.* [44] were able to accurately differentiate the three clinical classes of the AD spectrum reaching the maximum sensitivity (85.11%), specificity (91.27%), and accuracy (88.49%) values by implementing binary classifier strategy and reduction of input space with SVM and PCA techniques [12]. As previously referred, Khazaee *et al.* [43] were able to perfectly differentiate between cognitively healthy and AD classes in a small dataset of 40 individuals, using graph theory applied to brain connectivity assessed with fMRI they reach an accuracy of 100% for linear SVM and 87.5% for kNN. Although the separation between extreme cases is straightforward, difficulties are expected when considering the overlapped intermediary classes as show in the first chapter. Classifiers performance can be potentially affected by data issues, such as class overlapping, feature space with high dimensionality, missing-data, class imbalance, etc [85]. Recently, the pioneer Dubey's *et al.* work [20] on imbalanced AD classification problem showed the sensitivity and sensibility [23] for binary designed

classification problems (e.g. MCI vs AD, CN vs AD) would be improved using sampling strategies.

There is a need to identify biomarker combinations that maximize the classification and understand how much they contribute to differentiate between AD classes [38] [25]. However, this goal faces multiple classifier's comparisons when assessing biomarker combinations. In order to avoid the excessive number of comparisons, feature selection techniques are able to find a set of biomarkers that meet defined criterion [30]. For a given task (e.g. classification) examples of criterion are: identify the most cost-effective biomarkers, with higher accuracy and low false-positive; find a subspace of reduced dimensionality with the same or enhanced discriminant properties; extract/build relevant features from raw data [67].

Techniques of feature selection have been largely applied to AD-related problems, intending to provide a better understanding of biomarkers relationship [38] and achieve defined criterion of usefulness [30]. Interesting applications of feature selection techniques contributed to the understanding of AD, like the construction of potential biomarkers for enhanced classification. For instance, as graph theoretical measures obtained from fMRI connectivity matrix or metabolic connectivity mapping from PET. Another instance, owing to determine preclinical biomarkers for AD, Lopez-de-Ipiña *et al.* apply feature selection techniques on spontaneous speech to extract discriminant features [53]. They also were able to correctly classify AD subjects using kNN and MLP classifiers obtaining accuracy of 87.30% and 90.90%, respectively to each classifier. Feature selection AD-related works also is found in gene microarray analysis [67] and neuroimaging both with high-dimensional feature spaces. These fields have been provoking adaptation of feature selection techniques to deal with high dimensionality (tens of thousands of features) and small sample size in the case of microarray datasets [67]. In neuroimaging, the feature selection methods in 3D matrices are able to mitigate performances issues and improve the classification accuracy [69].

## 5.2 Experiment design

Here, let's find subsets of features among several feature combinations which maximize classification rates between three AD classes. Specifically, a multi-class classification problem is solved by assigning test patterns into one of following classes: CN, MCI or AD. To do that, is compared three feature selection techniques that depend on the classifier's outcome as a measure of usefulness [30]. This requirement characterizes the feature selection techniques called wrappers which select features based on the classifier's performance. However, instead of widely applied binary strategies, here let's use the all-versus-all strategy naturally achieved by the kNN classifier. The misclassification and comparison between biomarker combinations will be done by scalar measures of confusion matrices defined in chapter two. In order to observe the effect of training set size two validation processes are compared, 10-fold CV and LOOCV [12]. Analysis shows how the imbalanced dataset affects precision and shows a comparison of the feature's probability to reach higher precision. Two techniques to aid the class balances are compared with SMOTE and the random undersampling.

Here, let's use the kNN with data re-scaled by min-max normalization with Euclidean metric [54]. Importantly, since features CDRSB and MMSE are employed to define the diagnosis or are similar to ADNI categorization protocol (see ADNI site) they will be used only as comparison. The feature-wise comparison will be performed only for non-defining and non-cognitive features since it is well know that cognitive measures have more discriminative power. However, feature selection will be applied for non-defining features which include cognitive measures.

## 5.3 Wrappers

The goal of feature selection methods is to select a subset of features that is *useful* to build enhance a given classifier's measure, e.g. accuracy. Since

classifiers are induced by data with unknown underlying distributions the feature selection methods allow sub-optimal answers. There are comprehensive definitions of usefulness that would be a criterion to select relevant features, e.g. correlation and information theoretical criterion [30]. As shown in [46] the optimal choice of features does not imply the choice of relevant features. Conversely, optimality does not imply in relevancy. For instance, features that are presumably redundant may enhance the accuracy when combined with useful features [30].

Despite the lack of guarantees presented, feature selection methods were invaluable to deal with high-dimensional real-world problems. Feature selection methods were initially designed to deal with classification problems with no more than 40 features [46], now they are able to deal with thousands of features as discussed in the previous chapter. High-dimensional problems related to genetics have been received attention to uncovering the molecular mechanisms related to AD [70] and has motivated initiatives like AlzGene which is focused on providing data resources for AD genetic researches. Despite that, in this work the feature selection techniques will be applied to at least 9 features in order to observe the group-wise probability of a feature being more relevant than other.

Feature selection methods are divided into three categories due to the relation with the classifier: filter, wrapper, and embedded methods. Filters select a subset of features independently of the chosen classifier and the procedure mainly focus on ranking the features given defined criterion. Conversely, wrappers use classifiers' measures as a criterion to select subsets. Lately, embedded methods use a structured model to get the set of relevant features subject to a classifier [30]. For a complete discussion on the feature selection strategies and benefits see [30]. Others examples of feature selection techniques are feature extraction, feature construction, feature selection techniques for non-supervised learning, etc.

Three wrapper methods for feature selection using the kNN classifier combined with balance aid strategies to select the most useful subset of features are

compared. These are defined on the following search strategies: backward elimination, forward selection, and hill-climbing selection [30]. The subset obtained using the three methods will be compared to all combinations of features in order to observe if they are able to reach the optimum subset drawn from the rank with all features combinations. Additionally, a noise feature will be included in the feature selection procedure in order to compare the features to a non-significant case.

In figure 5.1 is depicted an example of search graph with all possibilities for three combinations. Regarding figure 5.1 in far left stage, no features chosen and far right all features chosen. Backward elimination moves right to left; forward selection left to right; hill climbing moves to any direction.

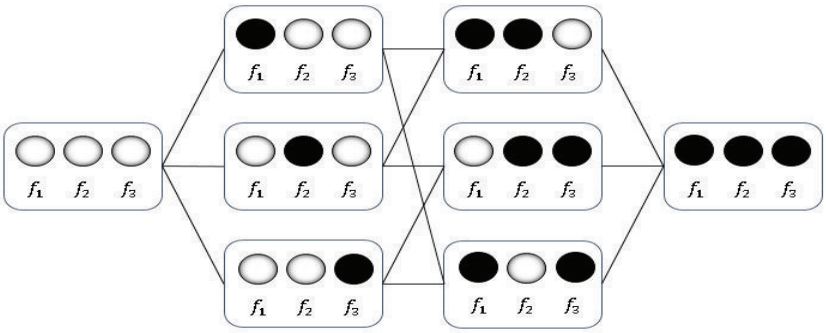



Figure 5.1 Scheme showing how to chose a subset of 3 features, using forward selection, backward elimination and hill-climbing.

The wrappers feature selection will search in a 9 (8 + noise) features graph scheme for the more useful subset. Forward selection initializes with any feature and steps up towards completing the feature subset. Iteratively it adds features to the chosen subset using the usefulness criterion that is to increase kNN classification rate. The usefulness criterion for kNN is obtained with the LOOCV, that means to classify one pattern using all remaining patterns as the training set. The backward elimination goes in the opposite direction in the searching graph. It initializes with all features and iteratively prunes features according to the highest usefulness defined by the kNN classification rate criterion. The hill-climbing selection can go

in any direction in the searching graph combining two previous approaches. Here, let's set up hill-climbing starting from the empty set feature.

All wrappers in this work are greedy algorithms and are subject to be trapped in a local maximum [46] (relative to the classification rate). A greedy algorithm can only optimize in a short distance and do not prevent that a good choice for a given iteration would lead to losing better options. There are search strategies designed to avoid this greedy drawback, for instance, the simulated annealing and the genetic algorithms [30] .

## 5.4 Results and discussion

Features for specific and general cohort studies owing to identify AD spectrum come from various sources: cognitive, genetic, neuroimaging, proteomic and others [53]. These features intend to provide insights on biological factors AD which is critical to understanding the disease progression and to early prevention strategies [25]. Features combinations provide higher classification rates than features by itself, also there are combinations more precise than others. For instance, a feature may confer a poor classification rate when many classes are included, even being highly valuable to understand AD biological processes, as depicted in past figures of ABETA and PTAU decision regions. This is, the combination $A\beta_{1-42}$ and p-tau$_{181}$, which sustain the main hypothesis for neurodegeneration [25] achieves poor results as shown in many examples in former chapters. Thus, it is necessary to find additional features or combinations to better identify AD. However, for $n$ features the number of combinations is given by $\sum_{i=1}^{n} \frac{n!}{i!(n-i)!}$, thus requiring strategies to avoid computational effort to uncover such combinations. Wrapper feature selection techniques are suitable for avoiding the comparison of all features combinations while maximizing a chosen classifier's accuracy. In case, the kNN that allows the all-versus-all strategy to observe how the classes affect each other all at once. Also

the all-versus-all strategy contrasts to the binary adapted strategies that are widely used in referenced AD researches along with ROC analysis [25]. Three techniques of wrappers feature selection are compared to the global rank of features for each sampling strategies using confusion matrices. Moreover, Gaussian noise (mean=0, sd=1) was added to feature space in order to compare an irrelevant feature to the features displayed in table 6.1, with label N standing for noise. Table 5.1 shows the precision of sorted combinations by higher classification rate among the non-defining features and by the number of features.

The dataset described in table 6.1 shows the numeric label of each biomarkers. Excluding the defining features (3,5) the wrappers here were able to identify sub-optimal combinations given the accuracy of combinations available. For imbalanced dataset the combinations found are: 4,6 for hill climbing (position 2); 4,1,8 for backward elimination (position 28); 4,6,7,8 for forward selection (position 41). With random undersampling the combinations found are: 4,6 for hill climbing (position 8); 1,4,6,8 for backward elimination (position 2); 4,1 for forward selection (position 5). With oversampling (SMOTE) the combinations found are: 4,6,7,8,N for hill climbing (position 38); 1,2,4,6,7,8,N for backward elimination (position 9); 4,7,6,8 for forward selection (position 21). All strategies of sampling and wrapper feature selection found sub-optimal combinations relative to the rank position, for the complete rank lists see on-line contents. One can notice that even with hill-climbing which combine the forward and backward strategies it can be trapped in local maximum and be affected by the cross-validatory components. For instance, using oversampling with backward elimination was found position 9 and hill-climbing position 38.

Clearly, from the selected features in training phase, the combination that provides the higher classification rate in validation is the defining features (3,5). See https://github.com/yurier /TEMA-R-CODES/tree/master/PLOTS2D for all *adni* dataset features decisions regions (2D only).

Table 5.1 Rank of combinations for the three imbalanced strategies. Note the highest classification rate was bolded for each validation method.

| position | LOOCV (%) | k | 10-fold CV (%) | k | combination |
|---|---|---|---|---|---|
| | | | | | imbalanced |
| 2 | **60.37 ± 0.05** | 13 | **59.33 ± 0.35** | 6 | 4,6 |
| 1 | 58.35 | 1 | 58.37 ± 2.58 | 4 | 2,4,8 |
| 3 | 58.11 ± 0.13 | 2 | 56.81 ± 2.56 | 6 | 1,2,4,8 |
| 7 | 58.01 | 5 | 56.29 ± 4.29 | 3 | 2,4,6,8,N |
| 10 | 58.80 | 3 | 58.02 ± 1.09 | 3 | 1,4,6,7,8,N |
| 38 | 56.94 ± 0.05 | 5 | 56.74 ± 2.25 | 12 | 1 2 4 6 7 8 N |
| | | | | | undersampled |
| 5 | **70.15** | 24 | 66.78 ± 3.36 | 18 | 1,4 |
| 3 | 69.05 | 20 | 68.59 ± 3.40 | 10 | 4,6,8 |
| 2 | 69.40 | 12 | **69.15 ± 4.46** | 13 | 1,4,6,8 |
| 1 | 67.14 | 10 | 67.47 ± 4.37 | 23 | 1,4,6,8,N |
| 7 | 67.08 | 24 | 66.57 ± 2.41 | 25 | 1,2,4,6,7,8 |
| 43 | 66.39 | 22 | 66.13 ± 2.98 | 24 | 1,4,2,6,7,8,N |
| | | | | | oversampled |
| 80 | **66.98 ± 0.47** | 23 | **69.20 ± 4.21** | 24 | 4,6 |
| 41 | 63.97 ± 0.11 | 23 | 65.08 ± 5.83 | 21 | 4,7,N |
| 12 | 66.35 ± 0.05 | 19 | 67.15 ± 2.81 | 25 | 1,4,6,8 |
| 1 | 62.16 ± 0.05 | 25 | 63.80 ± 4.63 | 24 | 1,2,4,7,N |
| 3 | 64.34 ± 0.53 | 25 | 65.69 ±6.02 | 20 | 1,2,4,6,8,N |
| 9 | 62.91 ± 0.11 | 17 | 64.16± 7.89 | 24 | 1,2,4,6,7,8,N |

The wrappers can be affected by the random nature of the cross-validation process and results may vary when the random generator number for the feature N and for the sampling are unfixed. These variation ranges between the very first combinations to the middle-rank combinations. The list of 502 combinations for

each technique to aid imbalance is available on-line, also the 120 combinations rank for non-defining features and the 26 combinations rank for non-cognitive features.

Comparing the three techniques approaches (imbalanced, undersampled and oversampled) is possible to claim that there is a significant improvement over the sampling techniques and the dataset imbalance. With the rank of all training results for all combinations of eight features plus noise producing 502 different combinations, one can ask the probability that a feature belonging to a given combination have higher classification rate than other. This is done by measuring how many times combinations that have feature A and not B are more accurate than combinations that have B and not A divided by the number of comparisons between A and B. Figure 5.2 depicts this evaluation for each balance method (imbalanced, undersampling, oversampling) using only one significant digit by resolution reasons and just for the non-cognitive features (26 features combinations). For instance, one can be interested in the probability of combinations that contain the feature 2 and not 8 to provide higher accuracy than combinations containing 8 and not 2, the left matrix in figure 5.2 shown that is 38.7%. Since as argued by [25] the neuropsychological tests are more accurate and standardized measures to detect AD the figure 5.2 shows only non-cognitive features (1,2,7,8,N). Despite neuropsychological tests being cost-effective biomarkers and its combinations provide a high accuracy, they do not provide information on the biological mechanism of AD. The suggestion of how they provide a higher accuracy, aside from similarity between defining features, is due to the limited possibilities of outcomes that define the neuropsychological scores, leading to overlaying patterns. Suggesting that biomarkers with fractional values are less subject to become a cluster in comparison to neuropsychological tests. Because of that filling in a less confined region in feature space. In the left matrix of figure 5.2, the noise has a higher probability in average to increase the classification rate than proteomic biomarker 7, however, it does not mean irrelevancy [30]. As argued by [30], a feature that is supposed to be irrelevant could contribute to enhancing the classifier performance.

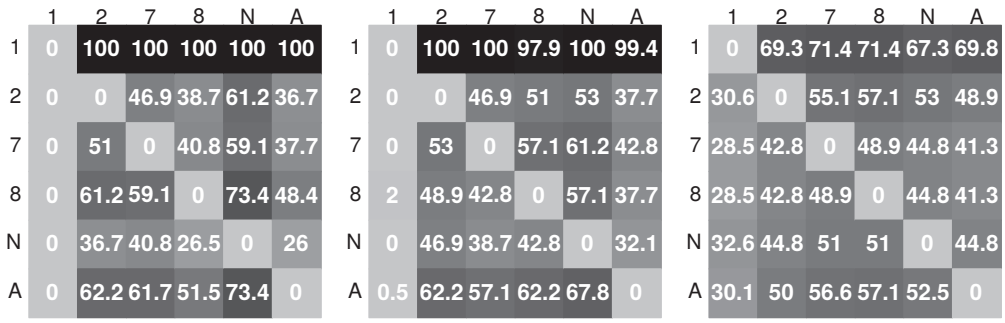



Figure 5.2 From right to left, probability matrix of a feature to provide higher classification rate than others for the imbalance, undersampled and oversampled, ranks respectively. Label "A" stands for column-wise and row-wise average.

In summary, wrappers techniques for feature selection have shown to be efficient finding the sub-optimal combinations given by the rank for all imbalance aid strategies for the proposed problem. However, adding more features to test limits of greedy search success is not ensured. Including features is challenging because it increases the number of patients who did not undergo to all examinations. This can be seen in the complete dataset available in ADNI. Fortunately, kNN inspired data missing techniques are available and would be useful to identify more accurate combinations that include interpretation benefits for AD mechanisms. However, to deal with improvements to kNN that imply in non-convex optimizations remains to be researched. This is, to include metric optimization is a matter of computational cost since it will increase the parameter space to optimize and will require improvements in computation performance. Even with the curse of dimensionality, the kNN with sampling strategies surpass many algorithms as depicted in classifier comparison the last chapter. However, the benefits of data balance would increment any classifier accuracy. The performed all-versus-all strategy requires fewer classifiers to be built than binary strategies, also one can visualize how the classes affect each other using confusion matrices. Comparisons using non-cognitive features reveal that FDG contributes more to increase the classification rate. However, more non-cognitive features are needed to observe if dominance for FDG holds, this is a challenge given mentioned data issues for missing values and data imbalance.

# 6 DATASETS

## 6.1 Introduction

AD [40] is pathophysiologically characterized by the gradual brain deposition of amyloid plaques, neurofibrillary tangles, and eventual neuronal depletion [40]. The AD spectrum can be composed by preclinical (CN), MCI and AD dementia phases [40]. There are other categorizations which include more stratified phases of AD and MCI classes. Preclinical AD individuals are those cognitively normal with amyloid plaques and tangles, individuals with MCI have cognitive symptoms without meeting clinical criteria for dementia, and AD dementia individuals present severely compromised cognitive faculties [75]. In recent years, a plethora of biomarkers has been developed in order to track AD progression, such as biomarkers for beta peptide 1-42 ($A\beta_{1-42}$) and tau proteins that indicate the presence of hallmark pathological features of AD, amyloid plaques and neurofibrillary tangles, respectively [40] [25]. Due to its economic importance, many public and private databases focusing on Alzheimer's research were started in last two decades, few instances of featured studies are: *Alzheimer Disease Neuroimaging Initiative* (ADNI), Human Connectome Project (HCP), Department of Defense ADNI (DOD-ADNI), Australian Imaging, Biomarker & Lifestyle Flagship Study of Ageing (AIBL). The ADNI was launched in 2003 as a public-private partnership, led by Principal Investigator Michael W. Weiner, MD. The goal of ADNI has been to test whether serial MRI, PET, other biological markers, and clinical and neuropsychological assessment can be combined to measure the progression of MCI and early AD. For up-to-date information, see www.adni-info.org. For review on papers published using ADNI database see [83]. From UCI repository, the dataset *parkinson* is composed of speech signals recordings from healthy and parkinsonians patients [51].

Due to the fast changing in the field and recent discoveries relating Alzheimer's disease and the biomarkers discussed here this text will not, at least in detail, many aspects of biomarkers.

## 6.2 ADNI and UCI datasets

The data-wise comparison uses seven datasets from ADNI (*Alzheimer Disease Neuroimaging Initiative*) database and one from the popular UCI (University California Irvine) machine learning repository. Further is described how the AD dataset was obtained. The UCI dataset used is the *parkison* dataset in which attributes information and class distribution are found in https://archive.ics.uci.edu/ml/datasets/Parkinsons. This dataset contains missing value and has 195 patterns distributed in 2 classes and 22 attributes. The major class has 147 patterns and the minor one 48. Let's define the nomenclature used though text for the dataset. The *adni* dataset consist of two neuroimaging biomarkers (labels 1,2), four neuropsychological tests (labels 3,4,5,6) and two proteomic biomarkers (labels 7,8) [77], respectively: 2-[18F]fluoro-2-Deoxy-D-glucose (FDG) PET, florbetapir-fluorine-18 (18F-AV-45) PET, clinical dementia rating sum of boxes (CDRSB), Alzheimer's disease assessment scale-cognitive with 11 items (ADAS11), mini-mental state examination (MMSE), Ray auditory verbal learning test percent forgetting (RAVLT), $A\beta_{1-42}$ CSF, phosphorylated tau protein (p-$tau_{181}$) CSF.

From *adni* dataset let's define other three: the *proteom* dataset formed by features $A\beta_{1-42}$ and p-$tau_{181}$, in the text, ABETA and PTAU; the *neuropsy* dataset formed by features CDRSB, ADAS11, MMSE, RAVLT; the *neuroimag* dataset formed by features 18-F-AV-45 and FDG. Indeed the original dataset available in ADNI site have 80 features and 11.149 patterns (from baseline and follow-ups), there are incomplete patterns with 37.44% from features information being NA's values. Table 6.1 depicts dataset demographics.

Table 6.1 Dataset *adni* demographics described by mean and standard deviation.

| Feature | CN | MCI | AD | Label |
|---|---|---|---|---|
| Male | 73 | 227 | 59 | - |
| Female | 79 | 187 | 41 | - |
| Age | 73.31 ± 6.35 | 71.39 ± 7.44 | 74.88 ± 8.19 | - |
| Education | 16.53 ± 2.50 | 16.18 ± 2.65 | 15.72 ± 2.55 | - |
| FDG [1] | 6.59 ± 0.51 | 6.33 ± 0.65 | 5.28 ± 0.76 | 1 |
| 18-F-AV-45 [2] | 1.10 ± 0.17 | 1.20 ± 0.22 | 1.39 ± 0.20 | 2 |
| CDRSB | 0 | 1.44 ± 0.86 | 4.70 ± 1.63 | 3 |
| ADAS11 | 5.85 ± 3.13 | 9.25 ± 4.45 | 20.96 ± 7.13 | 4 |
| MMSE | 29.05 ± 1.18 | 28.07 ± 1.73 | 22.96 ± 1.98 | 5 |
| RAVLT | 35.22 ± 26.69 | 55.41 ± 31.37 | 89.17 ± 20.72 | 6 |
| $A\beta_{1-42}$ pg/mL | 196.67 ± 49.96 | 174.79 ± 51.55 | 133.20 ± 35.84 | 7 |
| p-$tau_{181}$ pg/mL | 33.52 ± 16.40 | 41.26 ± 24.30 | 58.06 ± 29.39 | 8 |

Dataset *adni* was processed to remove NA's while conserving the maximum number of features. Other possibilities for dataset were available however it includes genetic data which strongly correlates with AD diagnosis. Implying in increased difficulty to analyze the discriminant power of other features. Next dataset from ADNI database is the *plasma* dataset, it consists of one neuroimaging biomarker, FDG; neuropsychological tests, CDRSB, ADAS11, ADAS13, MMSE, RAVLT; and two blood-based biomarkers AB40 and AB42. It was separated in the following datasets: the *psyconvert* dataset, formed by CDRSB, ADAS11, ADAS13, MMSE, and RAVLT; the *bloodabeta* dataset, formed by $A\beta_{1-40}$ and $A\beta_{1-42}$ proteins plasma sampled. Similarly, the original *plasma* dataset has 2.442 (1057 complete) entries with 5.80% missing data. Table 6.2 describes dataset *plasma* demographics.

[1] Average of FDG-PET of angular, temporal, and posterior cingulated with pons as reference region [80]

[2] Average of standardized 18-F-AV45 uptake value ratio (SUVR) of frontal, anterior cingulate, precuneus, and parietal cortex relative to whole cerebellum as reference region [80]

Table 6.2 Dataset *plasma* demographics described by mean and standard deviation.

| Feature | CN | CN to MCI | CN to AD | MCI | MCI to CN | MCI to AD | AD |
|---|---|---|---|---|---|---|---|
| Male | 205 | 7 | 1 | 296 | 4 | 34 | 142 |
| Female | 103 | 0 | 1 | 148 | 0 | 17 | 99 |
| Age | 75.1 ± 5.1 | 76.6 ± 7.3 | 81.5 ± 2.1 | 75.4 ± 7.2 | 66.3 ± 4.6 | 74.3 ± 6.5 | 75.8 ± 6.6 |
| Education | 16.0 ± 3.1 | 16.1 ± 2.4 | 16 ± 5.6 | 15.8 ± 2.7 | 17 ± 3.4 | 16 ± 2.8 | 14.9 ± 2.9 |
| $FDG^2$ | 6.4 ± 0.6 | 5.7 ± 0.4 | 5.8 ± 0.05 | 6.0 ± 0.6 | 6.5 ± 0.9 | 5.5 ± 0.6 | 5.3 ± 0.6 |
| CDRSB | 0.12 ± 0.3 | 1.14 ± 0.47 | 5.0 ± 0.7 | 1.74 ± 0.99 | 0.62 ± 0.25 | 3.90 ± 1.58 | 5.4 ± 2.3 |
| ADAS11 | 5.8 ± 2.9 | 7.4 ± 4.2 | 11.8 ± 4.4 | 10.9 ± 4.7 | 5.5 ± 2.88 | 17.1 ± 5.4 | 20.2 ± 7.2 |
| ADAS13 | 9.2 ± 4.5 | 13.4 ± 5.8 | 16.8 ± 4.4 | 17.9 ± 6.8 | 7.5 ± 3.4 | 27.1 ± 6.6 | 31.1 ± 8.5 |
| MMSE | 29.10 ± 1.11 | 28.7 ± 1.49 | 27.0 ± 1.4 | 27.1 ± 2.0 | 29.00 ± 0.8 | 23.8 ± 3.1 | 22.2 ± 3.5 |
| RAVLT | 32.10 ± 28.4 | 65.96 ± 36.3 | 75.0 ± 35.3 | 66.68 ± 40.1 | 16.30 ± 7.5 | 90.02 ± 21.9 | 92.46 ± 17.0 |
| $A\beta_{1-40}$ pg/mL | 160.8 ± 42.7 | 179.1 ± 26.0 | 185.5 ± 6.9 | 163.2 ± 50.9 | 181.6 ± 41.5 | 175.5 ± 45.4 | 159.3 ± 46.7 |
| $A\beta_{1-42}$ pg/mL | 40.9 ± 11.1 | 46.9 ± 14.6 | 42.2 ± 4.38 | 38.7 ± 12.2 | 46.4 ± 12.4 | 41.2 ± 12.4 | 38.6 ± 10.2 |

# 7 CONCLUSION

In chapter two was presented the difficulty of classifying the three class AD problem compared to the binary approach. The binary approach is more frequent in literature than multiclass since it prevents between-class interactions and overlap which leads to classification measures deterioration. Also, the extreme case for AD classification has been solved already and biomarkers to achieve perfect accuracy in AD vs CN case are available. For instance, using graph theoretical measures from fMRI allows 100% of accuracy [43] and using electroencephalogram (EEG) allows 97.7% of accuracy [4]. Since the limit accuracy is reached for the extreme case, candidates for other research direction can be for instance in cost-effective or progress-related biomarkers able to explain AD underlying causes. For the three class case, it has been a challenge with few approaches. For instance, using one-against-all binary classification for AD, MCI and CN, in [1] was achieved the accuracy of, 41.7% 66.7% and 77.8%, respectively using SVM classifier and high-dimensional gray matter density map as biomarkers. In the Dubeys' pioneer work [20] is used sampling techniques for the MCI class divided into MCI converters and non-converters and biomarkers from MRI and proteomic ADNI data. By focusing in one-against-one binary classification study design Dubey et al. identify the *k-Medoids* algorithm as the sampling technique which achieves best overall accuracy in the binary study.

In the third chapter is presented the widely know distance-based classifier, kNN, and possibilities to improve it by means of distance functions and normalization modifications. Mainly, the kNN uses all-versus-all classification strategy which allows observing how the imbalance affect spreads through the classes. With modifications on kNN, the fourth chapter presents distance-based sampling techniques to solve AD classification in which is shown kNN can surpass popular classifiers without balance aids. Also, was proposed two new sampling algorithms

SMOTE-borderline-3 (S3) and NearMiss-4 (NM4). The last was able to reach lower average class overlapping among all sampling techniques compared. Regarding the average precision achieved in data-wise comparisons, NM4 is comparable to the informed undersampling (second higher average in LOOCV) and S3 to the oversampling methods (second higher average in LOOCV).

Data-wise comparisons show the inherent difficulty of three and seven class AD classification problems, respectively for dataset based on *adni* and *plasma*. The worse improvement was in the *bloodabeta* dataset which achieves a poor precision of 16.52 ± 0.14%. However, this is expected due to being a multiclass problem with high class overlapping, high average class contrast as depicted in figure 3.21, also $A\beta_{1-40}$ and $A\beta_{1-42}$ sampled from plasma has been shown to be a not suitable biomarker [13]. However, it is not the end-line for cost-effective blood sampling, using other 18 proteins sampled from blood as shown in [66] it is possible to achieve near 90% accuracy to differentiate AD from CN subjects (extreme case). Remaining datasets obtained from *plasma* did not achieve great improvement due to the class overlapping and reduced class size for conversion classes. Conversely, the best improvement compared to the imbalanced setup is observed in the *neuropsy* dataset which precision is raised to 95.81%. However, this improvement is expected since neuropsychological tests in this dataset are similar to the features used to define the disease AD stages. Using non-defining dataset,*neuroimag* and *proteom* achieve precision of 50.74 ± 0.74% and 58.01 ± 0.41% in LOOCV validation. Showing these neuroimaing biomarkers (FDG and 18-F-AV-45) provide higher precision when classifying AD three class problem than proteomic CSF biomarkers ($A\beta_{1-42}$ and p-$tau_{181}$). The low precision for these datasets is due to being a multiclass problem which often is approached only as a binary study design, not suitable for multiclass problems, but allows better results [25].

Chapter five explores the argument of combined biomarkers would provide increased classification rate than biomarkers alone. Owing to this wrappers

feature selection using kNN precision as usefulness criterion are used to compared all features combinations of the *adni* dataset. We are able to show that wrappers can find sub-optimal biomarkers combinations among 502 possibilities (for all features). Furthermore, regarding non-defining and non-cognitive biomarkers we showed that FDG is the classifier with higher probability to provide higher precision than remaining biomarkers. However, precision was limited to 70.15% (LOOCV) for FDG and ADAS11 combination in AD three class problem.

## 7.1 Future work

Next directions to be followed from this thesis are the missing data algorithms, since the ADNI dataset is about 30% missing values it can potentially increase overall precision. Missing data solutions would be a parametric approach as expectation maximization for data imputation, or a distance-based for instance kNN data imputation which can use metric modification as done before. For instance, as claimed by [62], the feature weighted grey kNN (FWGKNN) data imputation excels other four missing data imputation strategies. The grey relational analysis as distance function has been shown to be superior to Euclidean distance when measuring similarity between two patterns, whereas the mutual information is used as weight for data imputation.

Alternatively to the data preprocessing, there are feature extraction approaches as connectivity mapping features, which shown to be highly competitive to discriminate AD from CN, however, research remains to be done with more classes. A promising strategy which allow discriminant biomarkers is the feature extraction from metabolic connectivity mapping obtained from PET-based measures and graph theory. Such approaches are planned to be implemented in near future.

# BIBLIOGRAPHY


[1] Abdulkadir, A., Peter, J., Ronneberger, O., Brox, T., and Klöppel, S. Voxel-based multi-class classification of ad, mci, and elderly controls. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI) 2014-CADDementia Challenge* (2014).

[2] Aggarwal, C. C., Hinneburg, A., and Keim, D. A. *On the surprising behavior of distance metrics in high dimensional sapce.* Springer, 2001.

[3] Aha, D. W. *Lazy learning.* Kluwer academic publishers, 1997.

[4] Ahmadlou, M., Adeli, H., and Adeli, A. New diagnostic eeg markers of the alzheimer's disease using visibility graph. *Journal of neural transmission 117*, 9 (2010), 1099–1109.

[5] Association, A. 2016 alzheimer's disease facts and figures. *Alzheimer's & Dementia 12*, 4 (2016), 459 – 509.

[6] Bailey, T., and Jain, A. K. A Note on Distance-Weighted k-Nearest Neighbor Rules. *IEEE Transactions on Systems, Man, and Cybernetics SMC-8*, 4 (1978), 311–312.

[7] Bellman, R. E. *Dynamic Programming.* Princeton university press, 1957.

[8] Beyer, K., Goldstein, J., Ramakrishnan, R., and Shaft, U. When is "nearest neighbor" meaningful? In *Database theory-ICDT'99.* Springer, 1999, pp. 217–235.

[9] Bhatia, N., et al. Survey of nearest neighbor techniques. *arXiv preprint arXiv:1007.0085* (2010).

[10] Bhattacharya, G., Ghosh, K., and Chowdhury, A. S. An affinity-based new local distance function and similarity measure for knn algorithm. *Pattern Recognition Letters 33*, 3 (2012), 356–363.

[11] Bhattacharyya, A. On a measure of divergence between two multinomial populations. *Sankhyā: the indian journal of statistics* (1946), 401–406.

[12] Bishop, C. Pattern recognition and machine learning. *Information Science and Statistics* (2006).

[13] Blasko, I., Jellinger, K., Kemmler, G., Krampla, W., Jungwirth, S., Wichart, I., Tragl, K. H., and Fischer, P. Conversion from cognitive health to mild cognitive impairment and alzheimer's disease: prediction by plasma amyloid beta 42, medial temporal lobe atrophy and homocysteine. *Neurobiology of aging 29*, 1 (2008), 1–11.

[14] Boiman, O., Shechtman, E., and Irani, M. In defense of nearest-neighbor based image classification. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on* (2008), IEEE, pp. 1–8.

[15] Boser, B. E., Guyon, I. M., and Vapnik, V. N. A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory* (1992), ACM, pp. 144–152.

[16] Chawla, N. V., Bowyer, K. W., Hall, L. O., and Kegelmeyer, W. P. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research 16* (2002), 321–357.

[17] Cover, T. M., and Hart, P. E. Nearest neighbor pattern classification. *Information Theory, IEEE Transactions on 13*, 1 (1967), 21–27.

[18] Cuingnet, R., Gerardin, E., Tessieras, J., Auzias, G., Lehéricy, S., Habert, M.-O., Chupin, M., Benali, H., Colliot, O., Ini-

TIATIVE, A. D. N., ET AL. Automatic classification of patients with alzheimer's disease from structural mri: a comparison of ten methods using the adni database. *neuroimage 56*, 2 (2011), 766–781.

[19] DEVROYE, L., GYÖRFI, L., AND LUGOSI, G. *A probabilistic theory of pattern recognition*, vol. 31. Springer Science & Business Media, 2013.

[20] DUBEY, R., ZHOU, J., WANG, Y., THOMPSON, P. M., YE, J., INITIATIVE, A. D. N., ET AL. Analysis of sampling techniques for imbalanced data: An n= 648 adni study. *NeuroImage 87* (2014), 220–241.

[21] DUCH, W., AND ADAMCZAK, R. Neural networks in non-euclidean metric spaces. In *Neural Networks, 1999. IJCNN'99. International Joint Conference on* (1999), vol. 1, IEEE, pp. 631–636.

[22] EBERT, S., FRITZ, M., AND SCHIELE, B. Active metric learning for object recognition. In *Joint DAGM (German Association for Pattern Recognition) and OAGM Symposium* (2012), Springer, pp. 327–336.

[23] FAWCETT, T. An Introduction to Rock Analysis. *Pattern Recognition Letters 27* (2005), 275–282.

[24] FAWCETT, T. An introduction to roc analysis. *Pattern recognition letters 27*, 8 (2006), 861–874.

[25] FIANDACA, M. S., ET AL. The critical need for defining preclinical biomarkers in alzheimer's disease. *Alzheimer's & Dementia 10*, 3 (2014), S196–S212.

[26] GARCÍA, V., SÁNCHEZ, J., AND MOLLINEDA, R. An empirical study of the behavior of classifiers on imbalanced and overlapped data sets. In *Iberoamerican Congress on Pattern Recognition* (2007), Springer, pp. 397–406.

[27] Goldberger, J., Hinton, G. E., Roweis, S. T., and Salakhutdinov, R. Neighbourhood components analysis. In *Advances in neural information processing systems* (2004), pp. 513–520.

[28] Golub, G. H., and Van Loan, C. F. *Matrix computations*, vol. 3. JHU press, 2012.

[29] Gou, J., Du, L., Zhang, Y., Xiong, T., et al. A new distance-weighted k-nearest neighbor classifier. *J. Inf. Comput. Sci 9*, 6 (2012), 1429–1436.

[30] Guyon, I., and Elisseeff, A. An introduction to variable and feature selection. *Journal of machine learning research 3*, Mar (2003), 1157–1182.

[31] Han, H., Wang, W.-Y., and Mao, B.-H. Borderline-smote: a new over-sampling method in imbalanced data sets learning. In *International Conference on Intelligent Computing* (2005), Springer, pp. 878–887.

[32] Hastie, T., Tibshirani, R., Friedman, J., and Franklin, J. The elements of statistical learning: data mining, inference and prediction. *The Mathematical Intelligencer 27*, 2 (2005), 83–85.

[33] Hastie, T., Tibshirani, R., Sherlock, G., Eisen, M., Brown, P., and Botstein, D. Imputing missing data for gene expression arrays, 1999.

[34] Haykin, S. Neural networks: a comprehensive foundation. *Neural Networks 2*, 2004 (2004).

[35] He, H., and Garcia, E. A. Learning from imbalanced data. *IEEE Transactions on knowledge and data engineering 21*, 9 (2009), 1263–1284.

[36] Hechenbichler, K., and Schliep, K. Weighted k-nearest-neighbor techniques and ordinal classification.

[37] Hinton, G. E., and Roweis, S. T. Stochastic neighbor embedding. In *Advances in neural information processing systems* (2002), pp. 833–840.

[38] Humpel, C. Identifying and validating biomarkers for alzheimer's disease. *Trends in biotechnology 29*, 1 (2011), 26–32.

[39] Islam, M. J., Wu, Q. J., Ahmadi, M., and Sid-Ahmed, M. A. Investigating the performance of naive-bayes classifiers and k-nearest neighbor classifiers. In *Convergence Information Technology, 2007. International Conference on* (2007), IEEE, pp. 1541–1546.

[40] Jack, C. R., et al. Hypothetical model of dynamic biomarkers of the alzheimer's pathological cascade. *The Lancet Neurology 9*, 1 (2010), 119–128.

[41] Jaggi, M. An equivalence between the lasso and support vector machines. *Regularization, Optimization, Kernels, and Support Vector Machines* (2014), 1–26.

[42] Jebara, T. *Machine learning: discriminative and generative*, vol. 755. Springer Science & Business Media, 2012.

[43] Khazaee, A., Ebrahimzadeh, A., and Babajani-Feremi, A. Identifying patients with alzheimer's disease using resting-state fmri and graph theory. *Clinical Neurophysiology 126*, 11 (2015), 2132–2141.

[44] Khedher, L., Ramírez, J., Górriz, J. M., Brahim, A., Segovia, F., Initiative, A. D. N., et al. Early diagnosis of alzheimer's disease based on partial least squares, principal component analysis and support vector machine using segmented mri images. *Neurocomputing 151* (2015), 139–150.

[45] Knill, O. Probability and stochastic processes with applications. *Havard Web-Based* (1994).

[46] Kohavi, R., and John, G. H. Wrappers for feature subset selection. *Artificial intelligence 97*, 1 (1997), 273–324.

[47] Krawczyk, B. Learning from imbalanced data: open challenges and future directions. *Progress in Artificial Intelligence* (2016), 1–12.

[48] Kuncheva, L. I. Diversity in multiple classifier systems. *Information fusion 6*, 1 (2005), 3–4.

[49] Lebedev, M. A., and Nicolelis, M. A. Brain–machine interfaces: past, present and future. *TRENDS in Neurosciences 29*, 9 (2006), 536–546.

[50] Lewicki, M. S. A review of methods for spike sorting: the detection and classification of neural action potentials. *Network: Computation in Neural Systems 9*, 4 (1998), R53–R78.

[51] Little, M. A., McSharry, P. E., Roberts, S. J., Costello, D. A., and Moroz, I. M. Exploiting nonlinear recurrence and fractal scaling properties for voice disorder detection. *BioMedical Engineering OnLine 6*, 1 (2007), 1.

[52] Liu, X.-Y., Wu, J., and Zhou, Z.-H. Exploratory undersampling for class-imbalance learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 39*, 2 (2009), 539–550.

[53] Lopez-de Ipiña, K., Alonso, J. B., Solé-Casals, J., Barroso, N., Henriquez, P., Faundez-Zanuy, M., Travieso, C. M., Ecay-Torres, M., Martinez-Lage, P., and Eguiraun, H. On automatic diagnosis of alzheimer's disease based on spontaneous speech analysis and emotional temperature. *Cognitive Computation 7*, 1 (2015), 44–55.

[54] Ma, C.-M., Yang, W.-S., and Cheng, B.-W. How the parameters of k-nearest neighbor algorithm impact on the best classification accuracy: In case of parkinson dataset. *Journal of Applied Sciences 14*, 2 (2014), 171.

[55] Mani, I., and Zhang, I. knn approach to unbalanced data distributions: a case study involving information extraction. In *Proceedings of workshop on learning from imbalanced datasets* (2003).

[56] Mann, C. Observational research methods. research design ii: cohort, cross sectional, and case-control studies. *Emergency Medicine Journal 20*, 1 (2003), 54–60.

[57] Marques, J. S. Reconhecimento de Padrões Métodos Estatísticos e Neuronais. *IST-Instituto Superior Tecnico* (2005).

[58] Milgram, J., Cheriet, M., and Sabourin, R. "one against one" or "one against all": Which one is better for handwriting recognition with svms? In *Tenth International Workshop on Frontiers in Handwriting Recognition* (2006), Suvisoft.

[59] Mohamad, I. B., and Usman, D. Standardization and its effects on k-means clustering algorithm. *Res. J. Appl. Sci. Eng. Technol 6*, 17 (2013), 3299–3303.

[60] Motoda, H., and Liu, H. Feature selection, extraction and construction. *Communication of IICM (Institute of Information and Computing Machinery, Taiwan) Vol 5* (2002), 67–72.

[61] Motsinger-Reif, A. A., Zhu, H., Kling, M. A., Matson, W., Sharma, S., Fiehn, O., Reif, D. M., Appleby, D. H., Doraiswamy, P. M., Trojanowski, J. Q., et al. Comparing metabolomic and pathologic biomarkers alone and in combination for dis-

criminating alzheimer's disease from normal cognitive aging. *Acta neuropathologica communications 1*, 1 (2013), 1.

[62] Pan, R., Yang, T., Cao, J., Lu, K., and Zhang, Z. Missing data imputation by k nearest neighbours based on grey relational structure and mutual information. *Applied Intelligence 43*, 3 (2015), 614–632.

[63] Park, H.-S., and Jun, C.-H. A simple and fast algorithm for k-medoids clustering. *Expert Systems with Applications 36*, 2 (2009), 3336–3341.

[64] Ramani, R. G., and Sivagami, G. Parkinson disease classification using data mining algorithms. *International journal of computer applications 32*, 9 (2011), 17–22.

[65] Ratsaby, J. On learning multicategory classification with sample queries. *Information and Computation 185*, 2 (2003), 298–327.

[66] Ray, S., Britschgi, M., Herbert, C., Takeda-Uchimura, Y., Boxer, A., Blennow, K., Friedman, L. F., Galasko, D. R., Jutel, M., Karydas, A., et al. Classification and prediction of clinical alzheimer's diagnosis based on plasma signaling proteins. *Nature medicine 13*, 11 (2007), 1359–1362.

[67] Saeys, Y., Inza, I., and Larrañaga, P. A review of feature selection techniques in bioinformatics. *bioinformatics 23*, 19 (2007), 2507–2517.

[68] Sáez, J. A., Luengo, J., Stefanowski, J., and Herrera, F. Smote–ipf: Addressing the noisy and borderline examples problem in imbalanced classification by a re-sampling method with filtering. *Information Sciences 291* (2015), 184–203.

[69] Sarica, A., Di Fatta, G., Smith, G., Cannataro, M., Saddy, J. D., et al. Advanced feature selection in multinominal dementia classification from structural mri data. In *Proc MICCAI Workshop Challenge*

*on Computer-Aided Diagnosis of Dementia Based on Structural MRI Data* (2014), pp. 82–91.

[70] Scheubert, L., et al. Tissue-based alzheimer gene expression markers–comparison of multiple machine learning approaches and investigation of redundancy in small biomarker sets. *BMC bioinformatics 13*, 1 (2012), 1.

[71] Simard, P., LeCun, Y., and Denker, J. S. Efficient pattern recognition using a new transformation distance. *Advances in neural information processing systems* (1993), 50–50.

[72] Sokolova, M., and Lapalme, G. A systematic analysis of performance measures for classification tasks. *Information Processing & Management 45*, 4 (2009), 427–437.

[73] Sola, J., and Sevilla, J. Importance of input data normalization for the application of neural networks to complex industrial problems. *IEEE Transactions on Nuclear Science 44*, 3 (1997), 1464–1468.

[74] Song, Y., Huang, J., Zhou, D., Zha, H., and Giles, C. L. Iknn: Informative k-nearest neighbor pattern classification. In *European Conference on Principles of Data Mining and Knowledge Discovery* (2007), Springer, pp. 248–264.

[75] Sperling, R. A., Aisen, P. S., Beckett, L. A., Bennett, D. A., Craft, S., Fagan, A. M., Iwatsubo, T., Jack, C. R., Kaye, J., Montine, T. J., et al. Toward defining the preclinical stages of alzheimer's disease: Recommendations from the national institute on aging-alzheimer's association workgroups on diagnostic guidelines for alzheimer's disease. *Alzheimer's & dementia 7*, 3 (2011), 280–292.

[76] Stoica, P., and Selen, Y. Model-order selection: a review of information criterion rules. *IEEE Signal Processing Magazine 21*, 4 (2004), 36–47.

[77] Tapiola, T., Alafuzoff, I., Herukka, S.-K., Parkkinen, L., Hartikainen, P., Soininen, H., and Pirttilä, T. Cerebrospinal fluid $\beta$-amyloid 42 and tau proteins as biomarkers of alzheimer-type pathologic changes in the brain. *Archives of neurology 66*, 3 (2009), 382–389.

[78] Teipel, S. J., Sabri, O., Grothe, M., Barthel, H., Prvulovic, D., Buerger, K., Bokde, A. L., Ewers, M., Hoffmann, W., and Hampel, H. Perspectives for multimodal neurochemical and imaging biomarkers in alzheimer's disease. *Journal of Alzheimer's Disease 33*, s1 (2013).

[79] Tewari, A., and Bartlett, P. L. On the consistency of multiclass classification methods. *Journal of Machine Learning Research 8*, May (2007), 1007–1025.

[80] Toga, A. W., and Crawford, K. L. The alzheimer's disease neuroimaging initiative informatics core: A decade in review. *Alzheimer's & Dementia 11*, 7 (2015), 832–839.

[81] Wang, S., and Yao, X. Multiclass imbalance problems: Analysis and potential solutions. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 42*, 4 (2012), 1119–1130.

[82] Weinberger, K. Q., Blitzer, J., and Saul, L. K. Distance metric learning for large margin nearest neighbor classification. In *Advances in neural information processing systems* (2005), pp. 1473–1480.

[83] Weiner, M. W., Veitch, D. P., Aisen, P. S., Beckett, L. A., Cairns, N. J., Green, R. C., Harvey, D., Jack, C. R., Jagust,

W., Liu, E., et al. The alzheimer's disease neuroimaging initiative: a review of papers published since its inception. *Alzheimer's & Dementia 9*, 5 (2013), e111–e194.

[84] Xing, E. P., Ng, A. Y., Jordan, M. I., and Russell, S. Distance metric learning with application to clustering with side-information. *Advances in neural information processing systems 15* (2003), 505–512.

[85] Yang, Q., and Wu, X. 10 challenging problems in data mining research. *International Journal of Information Technology & Decision Making 5*, 04 (2006), 597–604.

[86] Yen, S.-J., and Lee, Y.-S. Cluster-based under sampling approaches for imbalanced data distributions.