Available online at www.sciencedirect.com

ScienceDirect

Procedia Technology 9 (2013) 958 – 967

Procedia Technology

CENTERIS 2013 - Conference on ENTERprise Information Systems / PRojMAN 2013 - International Conference on Project MANagement / HCIST 2013 - International Conference on Health and Social Care Information Systems and Technologies

Determining the number of beams in 3D conformal radiotherapy: a classification approach

Joana Dias[a]*, Rupa Jaganathan[b], Sanja Petrovic[b]

[a]*Faculty of Economics and Inesc-Coimbra, University of Coimbra, Av. Dias da Silva 165 3004-512 Coimbra, Portugal*
[b]*Nottingham University Business School, Nottingham, NG8 1BB, United Kingdom*

Abstract

Radiation therapy consists of the treatment of cancerous tissues using radiation. For each patient, a personalized treatment has to be planned, most of the times using a time consuming trial and error procedure. In 3D conformal treatment planning, one of the most important treatment parameter is the number of radiation incidence directions to be used during treatment. Considering an existent database with information regarding already treated patients, we explore the possibility of applying classification methodologies to be able to correctly predict the number of radiation incidence directions that should be used in the treatment of a new patient.

Keywords: radiotherapy; supervised classification; ensemble predictors; *R* packages

1. Introduction

Each year about 3.2 million EU citizens are diagnosed with cancer [1], and nearly two-thirds of all cancer patients will receive radiation therapy during their illness [2]. Radiation therapy consists of the treatment of cancerous tissues using radiation, having as goal the destruction of cancerous cells and the preservation of

* Corresponding author. Tel: +351 790 500; fax: +351 239 790 514
E-mail address: joana@fe.uc.pt

doi:10.1016/j.protcy.2013.12.107

healthy tissues. The treatment plan is based on the patient CT images, where the radiation oncologist delineates the target volume (PTV) and organs at risk (OAR). The medical physicist is then responsible for generating a treatment plan that aims to deliver the prescribed dose to the tumor while minimizing the irradiation of the surrounding healthy organs and tissues. Three-dimensional (3D) conformal radiation therapy is a technique where the beams of radiation used in treatment are shaped by a multileaf collimator to match the tumor as best as possible. To plan a given treatment, several decisions have to be made: how many incidence directions to use (number of different beams); what incidence directions to use (beam angles); the radiation intensity from each beam angle (beam intensity or weight); the use of wedges (metallic wedge-shaped blocks placed in the path of the beam to attenuate the radiation). In clinical practice, most of the times, the treatment planning is done by using an iterative and time consuming trial and error procedure. The planner changes some of the treatment plan parameters until a treatment complying with the medical prescription is obtained (*forward planning*). An alternative approach considers using *inverse planning*, where given the desired medical prescription the treatment parameters are automatically determined. Inverse planning applied to radiotherapy is a fruitful ground of research with several important unresolved issues, and is mainly based on the use of optimization mathematical models and algorithms (see, for instance, [3]).

In this paper we consider the problem of deciding how many radiation incidences (beams) should be used in a patient's 3D conformal treatment. We consider the use of an existing database with information regarding already treated patients. Assuming that similar patients are expected to receive similar treatments, the database of treated patients can be used to predict the number of beams for a new patient. The choice of the number of beams to be used in the treatment of a new patient can be helpful for both forward and inverse planning. In forward planning, the planners can consider the number of beams fixed. The decrease in the degrees of freedom available allows a faster planning (less treatment parameters have to be tried). Regarding inverse planning, fixing *a priori* the number of angles can be an important input to optimization algorithms that will then try to automatically calculate the remaining treatment parameters. Our approach consists in applying supervised classification algorithms, where the number of beams used in the treatment is the dependent variable (Fig. 1). Each new patient will be assigned to a given class, determining the number of beams to use. As far as the authors know, this is the first attempt to use classifiers to determine the number of beams to be used in 3D conformal radiotherapy treatments. We not only apply several well-known supervised classification algorithms, but we also develop an ensemble of classifiers. This ensemble is created by resorting to a library of models, using a genetic algorithm to choose the best subset of models to consider.

This paper is organized as follows: In section 2 the database is described. Section 3 presents the classification methods. Section 4 shows the computational results. In section 5 conclusions are drawn.

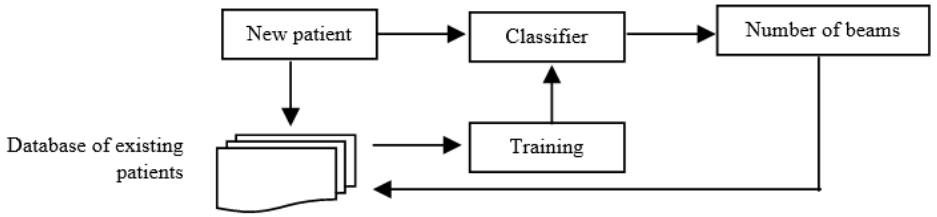

Fig. 1. The database of existing patients is used to train classifiers that are then used to determine the number of beams for a new patient. The new patient is then included in the existing database.

2. Data

Our database consists of data regarding brain cancer patients treated at the Nottingham University Hospitals NHS Trust, City Hospital Campus, UK. For each patient, a set of CT scan images is available in the form of DICOM-RT (a standard for digital image communications in medicine-radiation therapy) files and also information regarding the treatment planning parameters that were used. Observing that forward planning is basically done considering the geometrical location of PTVs and the OARs, DICOM files have, potentially, all the information that is needed. Since treatment planning aims to focus the prescribed radiation on the PTV while avoiding the OAR, the treatment plan parameters are largely determined based on the location and dimensions of the PTV and the spatial relationship between the PTV and the OAR [4]. The chosen attributes try to capture the geometry of a patient and are computed using the 3D coordinates representing the outlines of the PTV and OAR structures extracted from the DICOM files (see [5] for a more detailed description).

The available database has a total of 86 brain cancer patients, 24 of them treated with 3 angles, 61 treated with 4 angles and only one patient treated with 2 angles. For each patient a single PTV was delineated. The attributes that were considered define the OARs that were delineated for each patient, the angles between the PTV and the OARs, the edge to edge distance connecting the outline of the PTV and each OAR, the volume of the PTV, the ratio of the tumor volume to the volume of the entire patient body, the minimum edge to edge distance between the outline of the PTV and the outline of the body, and the relative position of the PTV with respect to each OAR (given by a set of binary attributes that define whether the PTV is placed to the left or to the right of the OAR, in an inferior or superior, posterior or anterior position) [5–7]. In the available dataset, there is incomplete information regarding OARs. Although there are very different ways of imputing missing values, considering that imputation is a complex and inaccurate procedure, we have chosen to consider only a subset of all the available attributes, minimizing the number of records with missing values: body to PTV volume ratio; angles to Left and Right Lens, edge to edge distance between the PTV and Left and Right Lens; PTV volume, PTV to body distance; binary variables indicating the existence of a given OAR or not. We end up with a total of 77 patients, treated with either 3 or 4 beams. Considering a series of smaller databases composed of patients with the same OARs, instead of just one database, would allow the consideration of more attributes, but the number of records available in the dataset does not allow this split (Table 1). A natural logarithm transformation was performed to attributes that represent the body to PTV volume ratio and angles between PTV and Left and Right Lens, because these attributes were strongly skewed.

Table 1. Characterization of the patients according to their OARs

Number of patients	Left Lens (LL)	Right Lens (RL)	Spinal Cord (SC)	Left Optic Nerve (LON)	Right Optic Nerve (RON)	Brainstem (BS)	Optic Chiasm (OC)
1	X	X	X		X	X	X
1	X	X		X	X		
1	X	X				X	X
3	X	X				X	
4	X	X	X				
5	X	X		X		X	
5	X	X					X
5	X	X		X	X	X	
26	X	X		X	X	X	X
27	X	X					

3. Classification methods

The problem of determining the number of beams to use in the treatment of a new patient, based on a database of patients, can be seen as a classification learning problem. The available dataset presents two serious problems: the fact that it is a small dataset, and the fact that there are many more patients treated with 4 angles (about 70%) than with 3 angles. Throughout the rest of the paper, we will interchangeably refer to existing patients, records, examples or samples. We resort to *off-the-shelf* implementations of well-known methods, available as *R* packages. We will briefly describe the methodologies applied.

3.1. Support Vector Machines

Support Vector Machines (SVM) is a linear classifier in the parameter space that uses kernels to extend data into a high dimensional feature space to improve the classification performance [8]. SVM chooses prototypes from the training examples, the support vectors, which lie on the border between two classes, and uses theses prototypes to derive the classification decision for new samples [9]. In classification, SVM separates the different classes by a hyperplane, calculated by solving a quadratic optimization problem. It is the one with the maximal margin of separation between classes [10]. Usually, SVM implementations require the user to define some parameters (the kernel function and a cost parameter used to penalize the SVM for misclassifying a training sample). Kernels usually considered are the linear kernel, the Gaussian basis function, the polynomial kernel, the Bessel function, the Laplace radial basis function. When there is no prior knowledge about the data, Gaussian, Laplace basis function and Bessel kernels are considered to be general-purpose kernels and thus an appropriate choice [10]. We have used package *Kernlab* [11].

3.2. Model-based clustering

In model-based clustering, each sample is modeled as having been generated by one of a set of probabilistic models. The aim is to learn the parameters of those models, the associated probabilities, such that each new sample is assigned to the mixture component that is most likely to have generated it [12,13]. We have used Gaussian finite mixture modeling, in which each known class is modeled by a single Gaussian term with the same covariance structure among classes [14]. Different parameterizations of the covariance matrix lead to different models with different geometric interpretations. We have used the *R* package *mclust* [14], using function *MclustDA* for training.

3.3. Stochastic Boosting

Boosting refers to a method of producing very accurate prediction rules by combining rough and moderately inaccurate rules of thumb [15]. The basic idea is to combine very simple classification rules to form an ensemble, with a significantly improved performance [16]. Different samples will have different weights, so that incorrectly classified examples have their weights increased in order to force the predictors to focus on these harder examples. The most well-known algorithm is *AdaBoost* [15]. Initially all the elements of the training set have the same weight, but at each iteration the weights of incorrectly classified examples are increased, and those of correctly classified examples decreased. The final classification is a weighted majority vote of all the trained weak learners. In stochastic boosting, a random permutation sampling strategy is employed at each iteration to obtain the current training set [16]. The most usual weak learners applied are classification trees. We have resorted to the use of the *ada R* package [16].

3.4. Random Forests

Random Forests can be interpreted as an ensemble of predictors. They are a combination of tree predictors such that each tree is constructed based on a sampling of the available training samples [17]. The final classification is calculated by majority voting, where each tree has one vote. With random forests, there is also a random selection of attributes at each node that will determine the split. This feature distinguishes random forests from bagging, where each tree uses a bootstrap sample of the data, and each node is split using the best attribute among all possible attributes. We have used the *randomForest R* package [18].

3.5. Neural Neworks

Neural networks can be seen as an extremely simplified model of the human brain, that under certain circumstances are able to approximate any function with arbitrary accuracy [19]. They are composed of neurons, organized by layers, and strongly interconnected. The output of a neuron is a function of a weighted sum of its inputs. This function is called the neuron activation function, and most of the times a sigmoid function is used. The training of the network consists in iteratively changing the weights, in order to decrease an error function. In supervised learning, the error function considers the differences between the neural network outputs and the desired outputs available in the training set. Neural networks have been successfully used in classification problems [20]. We have chosen to use the *nnet* package in *R* [21].

3.6. Ensembles

An ensemble can be understood as a collection of learning algorithms, that are simultaneously used for making more reliable and accurate predictions than its individual components (for a review see [22]). It resembles the idea of combining the opinions (sometimes not coincident) of individual experts. An ensemble of classifiers should be more accurate than any of its individual members if the classifiers are accurate and diverse [23]. There are many different ways of creating ensembles: manipulating the training examples, manipulating the input features, manipulating the output targets, among others [24]. In this work, ensembles were created considering the simultaneous use of several of the aforementioned models. Due to the huge amount of combinations possible, a genetic algorithm was developed that tries to choose the best models to include in the ensemble from a set of available predictors (see, for instance, [25–27]). In this genetic algorithm, each individual represents a set of models to consider, and is represented by a chromosome of binary genes. The number of genes is equal to the number of existing models in the models library, and is equal to 1 if the corresponding model is used, 0 otherwise. The usual genetic operators are used: tournament selection, one-point crossover and mutation. Regarding mutation, each individual in the new population is changed according to a given probability, and by resorting to a binary mask randomly generated: a binary vector of the same length as the chromosome is generated so that if the vector has a value of 1 in a given position the corresponding gene will be changed. The available training samples are divided in two sets: one set that is used for training the models, another set (cross validation set) that is used to calculate the error prediction of the ensembles. The use of a cross validation set tries to prevent the possibility of overfitting. The fitness of each individual considers two different components that are weighted in order to calculate the fitness value: one that assesses the prediction capability of the ensemble (with a weight of 0.90) and another that is related to the ensemble complexity (with a weight equal to 0.10). The prediction capability is calculated as the error rate using the cross validation set. The ensemble complexity is calculated as the percentage of models available in the models library that are included in the ensemble. In this way, not only the performance of each ensemble is considered, but also its complexity (considering that the parsimony principle should be respected). Whenever a given number of generations has passed without an improvement of the fitness of the best individual in the

current population, new training and cross validation sets are created by resampling the whole available set of records. At the end, the prediction is made considering all the individual predictions of the models that belong to the ensemble, using majority vote. Summarizing:

1. Divide the available samples randomly in two sets: the cross validation set, with 20% of the samples, and the training set with all the remaining elements.
2. Generate the models library, and train these models using the training set.
3. The initial population is generated randomly.
4. While the maximum number of generations is not reached:

4.1. The best individual is passed on to the next population.

4.2. Repeat until the next population is complete

4.2.1. Apply the selection operator to select two parents.

4.2.2. Apply the one point crossover to generate two offspring.

4.2.3. Apply the mutation operator.

4.3. Calculate the fitness of all individuals in the new population using the cross validation set.

4.4. Current population ← new population

4.5. If the fitness of the best individual has not improved in the last *n* generations then randomly resample the available samples to build new training and cross validation sets; retrain all models in the library.

5. The best individual will define the models that belong to the ensemble. Apply this ensemble to predict the number of beams for the unseen patient, using the majority vote.

4. Computational results

Our goal is to be able to correctly predict the classes of unseen examples, given a training set of classified examples. All the learning methodologies briefly described in the previous section were tested, as well as the use of ensembles. For assessing the prediction capability of each of the possible models, the *leave-one-out* cross validation was chosen. This is a special case of the *k*-fold cross validation, where *k* is equal to the number of taken elements in the training set. In the *leave-one-out* cross validation, one element of the training set is chosen in each iteration, and the whole training set with the exception of the chosen element is used for training the models. Then the trained models are used to predict the class of the chosen element. The procedure is repeated for all the elements of the training set, and the accuracy of the model can be calculated as the percentage of correct predictions made. A classifier can be considered as accurate if it has an error rate better than random guessing on unseen examples [24]. Nevertheless, due to the unequal balance of elements belonging to the two possible classes, a predictor that predicts always 4 beams will be, on average, correct 70% of the times. So, more than comparing the performance of the models with random guessing, we should aim at achieving error rates less than 30%. Whenever a model depends on random components, *leave-one-out* cross validation procedure was repeated 50 times. Results are depicted in Table 2.

Regarding support vector machines, several combinations of parameters were tested. The best results were obtained using radial basis function (with automatic sigma estimation) as the kernel function, and a cost parameter equal to 1. Model-based clustering presents the best performance, being the most accurate of all models. After testing all available models, the multivariate mixture model used was *VVI* (allowing that both volume and shape are different for different clusters). Stochastic boosting was applied for 100 iterations (100 trees are sequentially built) and with the minimum number of observations that must exist in a node in order for a split to be attempted equal to 5. This model is the one that presents the best error considering the training set, but not good results considering the cross validation sets (Fig. 1 (a)). Nevertheless, the execution of this model and the analysis of the trees that are created in each iteration allows the estimation of the importance of each variable (Fig. 1(b), see [28], page 367). A similar analysis regarding variable importance was made considering random forests. The results obtained are similar. The attributes that seem to be more important are the ones

related to the PTV, namely PTV to body distance, PTV volume and volume ratio between body and PTV. The less important attributes are the binary attributes that represent the patient's OARs. Random Forests are composed by 500 trees. They constitute the second best model according to the average *leave-one-out* error prediction. The neural networks tested had one layer. Several different number of neurons were tested, and the best results were obtained with 5 neurons. The results obtained are not of good quality. Probably due to the unequal balance between classes in the available samples, the classifiers used have much more difficulty in correctly predicting samples with 3 beams than samples with 4 beams (see Table 3.)

Table 2. *Leave-one-out* prediction error rates

Model	Average error rate	Maximum error rate	Minimum error rate
Random Guessing	50%	61%	30%
Predicting always 4 beams	30%	30%	30%
Support Vector Machines	26.6%	27.9%	24.6%
Model-based clustering	23.4%	23.4%	23.4%
Stochastic boosting	45%	61%	30%
Random Forests	25.5%	27.3%	24.6%
Neural Networks	29.4%	32%	26.9%

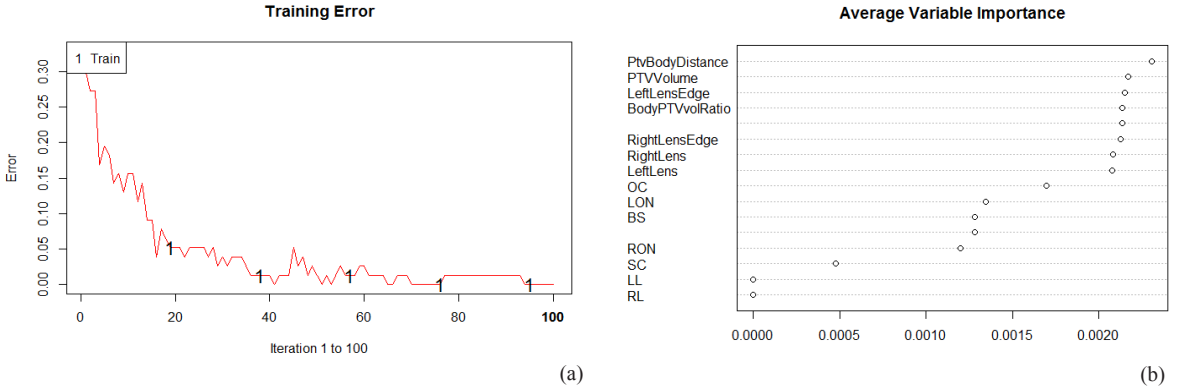

Fig. 2. Stochastic boosting (a) training error; (b) average variable importance

Table 3. Average prediction accuracy for each class

Model	3 Beams	4 Beams
Support Vector Machines	11.0%	89.0%
Model-based clustering	34.8%	94.0%
Stochastic boosting	22.4%	77.8%
Random Forests	15.7%	82.3%
Neural Networks	2.6%	96.8%

Regarding the ensemble, a library of models was created considering a total of 50 models of each one of the classifiers considered, giving in total 250 models. To increase the diversity of the models, bootstrapping was used, so that each model was trained with a randomly generated sample with replacement of the training samples. The population size was 500 individuals, evolved during 250 generations. Table 4 shows the results obtained. Fig. 3 presents an example of the evolution of the fitness value of the best individual of each generation. We can see two different behaviors. The best individual found in each generation has a good generalization capability, and the fitness does not get worse when the cross validation set is changed (Fig. 3(a)). The best individual presents an excellent behavior with the current cross validation set, but when this set is changed there is a significant increase in the error rate (Fig. 3(b)), meaning that the ensemble is overfitting. Changing periodically the cross validation set contributes to the prevention of overfitting.

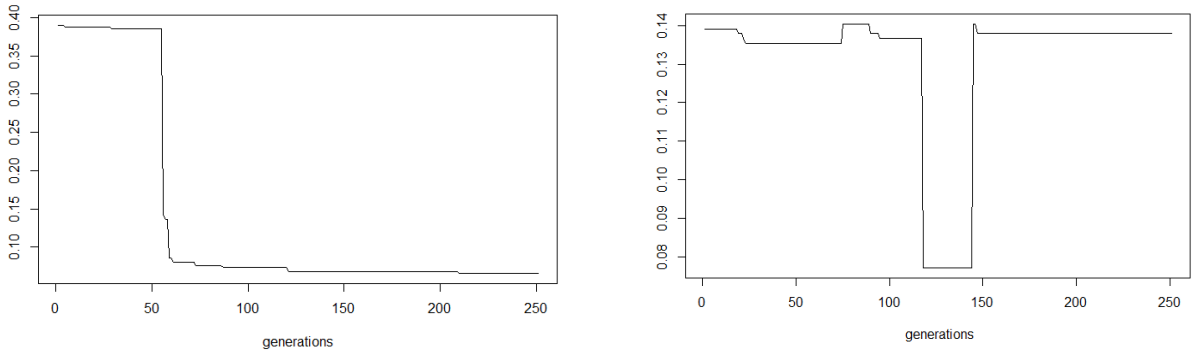

Fig. 3. Fitness of the best individual in each generation

Table 4. Ensemble: *leave-one-out* prediction error rates

Model	Average error rate	Maximum error rate	Minimum error rate
Ensemble	25.5%	27.2%	23.4%

Contrary to what was expected, the ensemble created using the genetic algorithm does not present the best average error rate. As a matter of fact, the model-based clustering approach presents better results. What can seem strange at a first sight can be easily explained. In reality, the comparison that is made considering the results depicted in Tables 2 and 4 is not fair to the ensemble. Why? In the *leave-one-out* cross validation, one prediction is made at a time, so that all the individual classifiers are trained with the whole set of available samples except for the one that will be used to make the prediction. When it comes to the genetic algorithm, only 80% of the available samples are used for training the models (actually, a little bit less than 80% since bootstrapping is used to generate the training set in order to increase the models' diversity), because an additional cross validation set is needed to calculate the fitness of the ensembles. A fairer comparison would be made if each of the individual classifiers was trained with a training set that is built by bootstrapping and considering only 80% of the available samples. The ensemble presents much better results (Table 5).

Table 5. *Leave-one-out* prediction error rates, with training sets generated by bootstrapping and using 80% of the available samples

Model	Average error rate	Maximum error rate	Minimum error rate
Support Vector Machines	29.6%	33.8%	24.7%

Model-based clustering	29.7%	40.3%	22.1%
Stochastic boosting	42.86%	50.64%	35.06%
Random Forests	34.4%	40.3%	28.6%
Neural Networks	30.1%	32.5%	28.6%

Another experiment was made, this time considering a test set that was randomly drawn from the available samples with replacement. The size of this test set is 20% of the total number of samples (but there can be repeated samples in the test set). All the other samples belong to the training set. All models were applied, and the prediction error over the test set was calculated. The procedure was repeated 100 times. Computational results are depicted in Table 6.

Table 6. Prediction error rates considering randomly generated test sets

Model	Average error rate	Standard deviation
Support Vector Machines	25.9%	9.8%
Model-based clustering	26.3%	9.8%
Stochastic boosting	36.9%	11.3%
Random Forests	33.0%	10.0%
Neural Networks	28.6%	10.9%
Ensemble	20.6%	9.8%

Analyzing the ensembles represented by the best individuals generated by the genetic algorithm, we observe that, on average, the ensembles are composed by 30% of all the available models. We can also see that 30% of the models included in the ensemble are model-based clustering predictors, followed by support vector machines (26%). Neural networks are seldom chosen (on average represent less than 5% of the predictors used in the ensembles). Previously, a decision support system has been designed based on a superset of the same patient data base used in this work [29]. The system uses case-based reasoning to generate treatment plans for a new patient based on the treatment plans of similar archived patients. By using a novel fuzzy similarity measure with local (target case specific) weights, the authors report an error rate of 18% for beam number prediction, which is comparable with the Ensemble model presented above. However, the training set for this system consisted of a slightly larger number of cases and it is expected that as the number of training cases increases, the performance of the classifiers will improve as well.

5. Conclusions and future work

In this paper we try to elicit the knowledge implicitly available in existing clinical databases to determine the number of beams that should be used in a new patient 3D conformal treatment using a classification approach. Despite the fact that the existing database suffers from two main drawbacks (its size and the unequal proportion of cases in the classes), the results show that it is possible to apply predictive models with reasonable accuracy results. It would also be interesting to explore the possibility of including case-based reasoning in the ensemble of classifiers. We are currently planning on applying these models to a larger database of IMRT (intensity modulated radiation therapy) treated patients. For IMRT treatments, the complexity of planning is even greater than for 3D conformal therapy, due to the possibility of considering the modulation of the intensity for each beam. The definition of the number of beams to use is the first step of the planning process. We believe that increasing the size and the diversity of the database will lead to the improvement of the prediction accuracy.

Acknowledgements

This work was supported by FEDER, COMPETE, iCIS (CENTRO-07-ST24-FEDER-002003), FCT, PTDC/EIA-CCO/121450/2010, PEst-C/EEI/UI0308/2011, SFRH/BSAB/1301/2012. The work of J. Dias was partially supported by Nottingham University Business School, when staying during sabbatical.

References

[1] Comission of the European Communities, "on Action Against Cancer : European Partnership," 2009.
[2] ASTRO, "About Radiation Therapy," 2013. [Online]. Available: http://www.rtanswers.org/statistics/aboutradiationtherapy.aspx. [Accessed: 21-Apr-2013].
[3] M. Ehrgott, Ç. Güler, H. W. Hamacher, and L. Shao, *Mathematical optimization in intensity modulated radiation therapy*, vol. 6, no. 3. 2008, pp. 199–262.
[4] B. Wu, F. Ricchetti, G. Sanguineti, M. Kazhdan, P. Simari, M. Chuang, R. Taylor, R. Jacques, and T. McNutt, "Patient geometry-driven information retrieval for IMRT treatment plan quality control," *Medical Physics*, vol. 36, no. 12, p. 5497, 2009.
[5] Rupa Jagannathan, "A Case-based Reasoning System for Radiotherapy Treatment Planning for Brain Cancer," University of Nottingham, 2013.
[6] R. Jagannathan, S. Petrovic, A. McKenna, and L. Newton, "A novel two phase retrieval mechanism for a clinical case based reasoning system for radiotherapy treatment planning," *International Journal on Artificial Intelligence Tools*, vol. 21, no. 4, 2012.
[7] R. Jagannathan, S. Petrovic, A. McKenna, and L. Newton, "A fuzzy non-linear similarity measure for case-based reasoning systems for radiotherapy treatment planning.," in *Artificial Intelligence Applications and Innovations*, 2010, pp. 112–119.
[8] D.-C. Li and C.-W. Liu, "A class possibility based kernel to increase classification accuracy for small data sets using support vector machines," *Expert Systems with Applications*, vol. 37, no. 4, pp. 3104–3110, Apr. 2010.
[9] A. I. Belousov, S. A. Verzakov, and J. von Frese, "A flexible classification approach with optimal generalisation performance: support vector machines," *Chemometrics and Intelligent Laboratory Systems*, vol. 64, no. 1, pp. 15–25, Oct. 2002.
[10] A. Karatzoglou, D. Meyer, and K. Hornik, "Support Vector Machines in R," vol. 15, no. 9, 2006.
[11] A. A. Karatzoglou, A. Smola, K. Hornik, and M. A. Karatzoglou, "Package ' kernlab '," 2013.
[12] M. H. C. Law, M. A. T. Figueiredo, and A. K. Jain, "Simultaneous feature selection and clustering using mixture models.," *IEEE transactions on pattern analysis and machine intelligence*, vol. 26, no. 9, pp. 1154–66, Sep. 2004.
[13] C. Fraley and A. E. Raftery, "How Many Clusters? Which Clustering Method? Answers Via Model-Based Cluster Analysis 1," *The computer journal*, vol. 41, no. 8, pp. 578–588, 1998.
[14] C. Fraley, A. E. Raftery, T. B. Murphy, and L. Scrucca, "mclust Version 4 for R : Normal Mixture Modeling for Model-Based Clustering , Classification , and Density Estimation," 2012.
[15] Y. Freund, R. E. Schapire, P. Avenue, and F. Park, "A decision-theoretic generalization of on-line learning and an application to boosting," *Journal of Computing and Systems Science*, vol. 55, no. 1, pp. 119–139, 1996.
[16] M. Culp and G. Michailidis, "ada : an R Package for Stochastic Boosting," *Journal of Statistical Software*, vol. 17, no. 2, 2006.
[17] L. E. O. Breiman, "Random Forests," *Machine Learning*, vol. 45, pp. 5–32, 2001.
[18] A. Liaw and M. Wiener, "Package ' randomForest '," 2013.
[19] K. Hornik, "Approximation Capabilities of Muitilayer Feedforward Networks," *Neural Networks*, vol. 4, pp. 251–257, 1991.
[20] G. P. Zhang, "Neural networks for classification: a survey," *IEEE Transactions on Systems, Man and Cybernetics, Part C (Applications and Reviews)*, vol. 30, no. 4, pp. 451–462, 2000.
[21] B. Ripley, "Package ' nnet '," 2013.
[22] M. Re and G. Valentini, "Ensemble methods: a review," in *Data Mining and Machine Learning for Astronomical Applications*, A. N. Srivastava, J. D. Scargle, K. Ali, and M. J. Way, Eds. Chapman & Hall, 2011, pp. 1–40.
[23] L. Hansen and P. Salamon, "Neural Networks Ensembles," *IEEE transactions on pattern analysis and machine intelligence*, vol. 12, no. 10, pp. 993–1001, 1990.
[24] T. G. Dietterich, "Ensemble Methods in Machine Learning," in *Multiple classifier systems*, Springer, 2000, pp. 1–15.
[25] E. M. Dos Santos, R. Sabourin, and P. Maupin, "Single and Multi-Objective Genetic Algorithms for the Selection of Ensemble of Classifiers," *The 2006 IEEE International Joint Conference on Neural Network Proceedings*, pp. 3070–3077, 2006.
[26] E. M. Dos Santos, R. Sabourin, and P. Maupin, "Overfitting cautious selection of classifier ensembles with genetic algorithms," *Information Fusion*, vol. 10, no. 2, pp. 150–162, Apr. 2009.
[27] D. Ruta and B. Gabrys, "Classifier selection for majority voting," *Information Fusion*, vol. 6, no. 1, pp. 63–81, Mar. 2005.
[28] T. Hastie, R. Tibshirani, and J. Friedman, *The elements of Statistical Learning: Data mining, inference and prediction*. Springer, 2001.
[29] R. Jagannathan and S. Petrovic, "A Local Rule-based Attribute Weighting Scheme for a Case-based Reasoning System for Radiotherapy Treatment Planning," in *Lecture Notes in Computer Science*, 2012, pp. 167–181.