SOFIA UNIVERSITY St. KLIMENT OHRIDSKI

FACULTY OF ECONOMICS and BUSSINES ADMINISTRATION

GAME MODELS AND
TIME SERIES MODELING

ABSTRACT

For awarding the educational and scientific degree "PhD"

Field of higher education: 3. Social, economic and legal sciences
Professional direction: 3.8. Economy
Scientific direction: Analytical research on data /Data Science/

Author:
Vladislav Krasimirov Tanov

Supervisor :
Assoc. Prof. PhD Nikolay Netov

Sofia
2023

CONTENT

Introduction

Game theory is a concept that scientists can harness to predict how rational people will make decisions that help them make effective, data-driven decisions in strategic circumstances.

Data scientists can apply game theory based on the type of decision problem they are dealing with: https://www.dezyre.com/article/is-game-theory-important-for-data -scientists/139

Game modeling and big data modeling find common points of joint applications in geo-sciences and geo-data. The author (Bruce, 2013) outlines mechanisms for applying game theory models to big data analytics and decision making in geosciences and geodata. The author proposes the use of strategic, competitive game theory models for the purpose of spectral band clustering using hyperspectral imagery. The proposed system uses conflict data filtering based on mutual entropy and the interaction process of multiple group strategies in a conflict environment, which aims to maximize the benefit of multiple groups from the whole system. The proposed system uses the Nash equilibrium to find a stabilizing solution to the clustering problem and implements the model under the assumption that all players are rational. The author uses the proposed group clustering as a component in a multifunction fusion decision (MCDF) system for automatic land cover classification with hyperspectral imagery.

From this point of view, we will consider some different types of games and present methodologies for searching for equilibria by solving Riccati equations. Through the considered types of games, we show how, under different conditions, an equilibrium party can be found in a competitive environment. We use stabilizing solutions for the Riccati equations to reach the Nash equilibrium: https://www.igi-global.com/chapter/applied-game-theory-in-business-analytics/107224

The first and second chapters of the current thesis project are devoted to finding equilibrium in linear quadratic games by creating methods and algorithms for searching for stabilizing solutions of the corresponding Riccati equations. These studies could be the basis for developing game models with applications in big data analysis (machine learning.) Studies in this direction appear when applying the concept of finding optimal Nash strategies under the conditions of a classification task (Moy and co-authors, 2023;

Wang et al., 2019). In the first chapter, we consider a linear quadratic stochastic game analyzed by Zhu and Zhang, for which we construct an iterative method for finding a stabilizing solution of a system of four nonlinear matrix equations.

The first chapter uses methods, algorithms, and examples from (Ivan Ivanov, 2012; Ivelin Ivanov, 2016). At the same time, the proposed methods complement the research in (Ivelin Ivanov, 2016). In the second chapter we consider antagotistic games and game models on positive systems. Methods are proposed for finding a stabilizing non-negative solution to the corresponding Riccati equation. The results are published in three articles, two of which are indexed in Scopus.

The authors (Koziarski et al., 2020) and many others analyze the difficulties in modeling multiclass and imbalanced big data. The goal in chapter three is to develop a data-driven approach to conducting big data classification analysis. We formulate an optimization model that searches for the best training set, in a particular sense, for models performing classification analysis. To solve the optimization task, we propose an algorithm that has been applied to different sets of big data. The results were reported at an international conference and supported by two publications indexed in Scopus (https://www.scopus.com/authid/detail.uri?authorId=57208207140).

The markings in the abstract retain the numbering and citations, according to the text of the PhD Thesis.

Chapter One. Linear quadratic differential stochastic games

Linear quadratic games in which Riccati equations are solved in order to reach equilibrium parties have been widely studied in the scientific literature. An example of such studies can be cited (Azevedo-Perdicoulis, Jank, 2005), (Basar, Olsder, 1999), (Broek, Engwerda, Schumacher, 2003), (Engwerda 2005). A major role in the theory is given to stochastic differential games, in which Ito's lemma is used, applied to differential systems with disturbances in the system state and control (Yu, 2012), (Zhu, Zhang, 2013), (Zhu, Zhang, Bin, 2014). The intensive study of the generalized Riccati equations is based on their wide applicability – for example, when researching and constructing optimal financial portfolios one goes through the solution of an appropriate generalized Riccati equation – (Yao, Zhang, Zhou, 2006; Costa, de Paulo 2007; Costa, de Oliveira, 2012).

In this chapter, we introduce the concept of game theory and its use as a decision-making tool in a competitive situation among players. We search for the equilibrium point by searching for a stabilizing solution for a system of two Riccati equations and two more additional equations, as derived by the authors in (Zhu, Zhang, 2013).

The goal in this chapter is to describe and propose methods and algorithms for searching for a stabilizing solution to a system of Riccati equations, whose solutions lead to finding the equilibrium point in the considered game model.

1.2. Linear Quadratic Differential Stochastic interference games with state and control-dependent noise

We consider a linear quadratic stochastic game studied by Zhu and Zhang in their paper in (Zhu, Zhang, 2013). The existence of equilibrium in the solution terminology of a system of matrix equations is presented by the authors in Theorem 2 (Zhu, Zhang, 2013). Authors Zhu and Zhang do not present a method or algorithm for finding a solution to these equations. We will propose an approach to calculate the sought solution. The considered system of Riccati equations seeks a solution that sets the Nash equilibria in a stochastic differential game described by Zhu and Zhang. The approach proposed here allows one to consider a stochastic differential game with different numbers of players. The proposed method does not depend on the number of players.

In this section, we will describe an algorithm published by (Ivelin Ivanov and V. Tanov, 2018, (Ivelin Ivanov and V. Tanov, 2018, An Iterative Method for an Equilibrium Point of Linear Quadratic Stochastic Differential Games with State and Control-Dependent Noise) for finding of the corresponding solution According to Zhu and Zhang, the Nash equilibrium is the solution $\widetilde{X_1}, \widetilde{X_2}$ of the following system of two coupled nonlinear matrix equations:

$$R_1\,(\mathrm{X}_1, \mathrm{X}_2) := \mathrm{X}_1 \bar{A}_0 + \bar{A}_0{}^T \mathrm{X}_1 + \bar{A}_1{}^T \mathrm{X}_1 \bar{A}_1 + \bar{Q}_1 - \left(\mathrm{X}_1 \mathrm{B}_1 + \bar{A}_1{}^T \mathrm{X}_1 \mathrm{C}_1\right) \\ \times (R_{11} + C_1^T X_1 C_1)^{\{-1\}} (B_1^T X_1 + C_1^T X_1 \bar{A}_1) = 0$$

$$F_1 = -(R_{11} + C_1^T X_1 C_1)^{\{-1\}}(B_1^T X_1 + C_1^T X_1 \bar{A}_1)$$
$$(R_{11} + C_1^T X_1 C_1) > 0 \qquad (1.4)$$
$$R_2\,(\mathrm{X}_1, \mathrm{X}_2) = \mathrm{X}_2 \tilde{A}_0 + \tilde{A}_0{}^T \mathrm{X}_2 + \tilde{A}_1{}^T \mathrm{X}_2 \tilde{A}_1 + \bar{Q}_2$$
$$-\left(\mathrm{X}_2 \mathrm{B}_2 + \tilde{A}_1{}^T \mathrm{X}_2 \mathrm{C}_2\right)(R_{22} + C_2^T X_2 C_2)^{\{-1\}}\left(B_2^T X_2 + C_2^T X_2 \tilde{A}_1\right) = 0$$
$$F_2 = -(R_{22} + C_2^T X_2 C_2)^{\{-1\}}\left(B_2^T X_2 + C_2^T X_2 \tilde{A}_1\right)$$
$$(R_{22} + C_2^T X_2 C_2) > 0 \quad \text{label\{H17Eq.10\}} \quad (2.4)$$

under notations:

$$\bar{A}_0 = \mathrm{A}_0 + \mathrm{B}_2 \mathrm{F}_2, \qquad \bar{A}_1 = \mathrm{A}_1 + \mathrm{C}_2 \mathrm{F}_2\,,$$
$$\tilde{A}_0 = \mathrm{A}_0 + \mathrm{B}_1 \mathrm{F}_1, \qquad \tilde{A}_1 = \mathrm{A}_1 + \mathrm{C}_1 \mathrm{F}_1\,,$$
$$\bar{Q}_1 = \mathrm{Q}_1 + \mathrm{F}_2^{\mathrm{T}} R_{12} \mathrm{F}_2, \qquad \bar{Q}_2 = \mathrm{Q}_2 + \mathrm{F}_1^{\mathrm{T}} R_{21} \mathrm{F}_1.$$

Moreover, A_0 and A_1 are real nxn matrices, Q_1 and Q_2 are real symmetric nxn matrices, B_1 and C_1 are real nxm_1 matrices , B_2 and C_2 are real nxm_2 matrices, R_{11} and R_{21} are real m_1xm_1 матрици, and R_{12} and R_{22} are real m_2xm_2 matrices.

The following definitions are known. Matrix A is said to be stable if all its eigenvalues lie in the left half-plane, about the y-axis. We will use the notation X > Y or X≥Y if X – Y is a positive definite matrix, i.e., with positive eigenvalues or X – Y is a positive semidefinite matrix, i.e. with non-negative eigenvalues.

We present experimental results that show that the proposed game-theoretic-algorithmic approach significantly leads to the desired equilibrium.

1.3. An iterative method

We construct an iterative algorithm for solving the defined system of nonlinear matrix equations and inequalities (1.4). A matrix series of solutions of this system is constructed, which converges to a stabilizing solution for two Riccati equations. This stabilizing solution leads to finding a Nash equilibrium for the game under consideration. We present experimental results that show that the proposed game-theoretic-algorithmic approach leads to the desired equilibrium.

We write the Riccati equations $R_1\,(\mathrm{X}_1, \mathrm{X}_2) = 0$ and $R_2\,(\mathrm{X}_1, \mathrm{X}_2) = 0$ as a general Riccati equation but with a larger dimension:

$$\mathcal{R}(\boldsymbol{X}) = \mathcal{A}_0{}^T \mathbf{X} + \mathbf{X}\,\mathcal{A}_0 + \Pi_1(\mathbf{X}) + \mathcal{Q}$$
$$-\,\mathcal{S}\,(\mathbf{X})\,[\mathfrak{R}\,(\mathbf{X})]^{\{-1\}}[\mathcal{S}\,(\mathbf{X})]^{\mathrm{T}} = \mathbf{0} \qquad (1.5)$$

where

$$\mathfrak{R}\,(\mathbf{X}) = R + \mathcal{C}^T\,\mathbf{X}\,\mathcal{C} = diag(R_{11} + C_1^T\,X_1 C_1, R_{22} + C_2^T\,X_2 C_2)$$
$$\mathcal{S}\,(\mathbf{X}) = \mathbf{X}\,\mathcal{B} + \mathcal{A}_1{}^T\,\mathbf{X}\,\mathcal{C}$$
$$= \mathrm{diag}\,(\mathrm{X}_1 \mathrm{B}_1 + \bar{A}_1^{\mathrm{T}} \mathrm{X}_1 \mathrm{C}_1, \mathrm{X}_2 \mathrm{B}_2 + \tilde{A}_1^{\mathrm{T}} \mathrm{X}_2 \mathrm{C}_2)$$
$$\Pi_1(\mathbf{X}) = \mathcal{A}_1{}^T\,\mathbf{X}\,\mathcal{A}_1 = \mathrm{diag}\,(\bar{A}_1^{\mathrm{T}} \mathrm{X}_1 \bar{A}_1\,, \tilde{A}_1^{\mathrm{T}} \mathrm{X}_2\,\tilde{A}_1\,)$$
$$\mathcal{A}_0 = \mathrm{diag}\,(\overline{\mathrm{A}}_0\,, \widetilde{\mathrm{A}}_0)\,; \qquad \mathcal{A}_1 = \mathrm{diag}\,(\overline{\mathrm{A}}_1\,, \widetilde{\mathrm{A}}_1)\,;$$
$$\mathcal{B} = \mathrm{diag}\,(\mathrm{B}_1\,, B_2)\;; \qquad \mathcal{C} = \mathrm{diag}\,(\mathrm{C}_1\,, C_2)\;;$$
$$R = \mathrm{diag}\,(\mathrm{R}_{11}\,, \mathrm{R}_{22})\;; \qquad \mathcal{Q} = \mathrm{diag}\,(\bar{Q}_1\,, \bar{Q}_2)\;;$$
$$\mathbf{X} = \mathrm{diag}\,(\mathrm{X}_1\,, X_2)\;;$$

The considered Riccati equation (1.5) is of the same form as equation (1.1). The idea implemented here is to use the Lyapunov iteration method (1.3) to the system of equations (1.5). For initial matrix we choose $\mathrm{X}^{(0)} = \left[\, \mathrm{X}_1^{(0)}, \mathrm{X}_2^{(0)} \right]$ and calculate

$$\begin{aligned}
&F_1^{(0)} = -\left(R_{11} + C_1^T X_1^{(0)} C_1\right)^{-1} \left(B_1^T X_1^{(0)} + C_1^T X_1^{(0)} \bar{A}_1\right) \\
&\tilde{A}_1 = \mathrm{A}_1 + \mathrm{C}_1 F_1^{(0)} \\
&F_2^{(0)} = -\left(R_{22} + C_2^T \mathrm{X}_2^{(0)} C_2\right)^{-1} \left(B_2^T \mathrm{X}_2^{(0)} + C_2^T \mathrm{X}_2^{(0)} \tilde{A}_1\right) \qquad (1.6) \\
&\bar{A}_1 = \mathrm{A}_1 + \mathrm{C}_2 F_2^{(0)} \\
&F_1^{(0)} = -\left(R_{11} + C_1^T X_1^{(0)} C_1\right)^{-1} \left(B_1^T X_1^{(0)} + C_1^T X_1^{(0)} \bar{A}_1\right)
\end{aligned}$$

We construct the following matrix series редица $\{\mathbf{X}^{(\mathbf{k})}\}_{\mathbf{0}}^{\infty}$ as follows. We assume that the matrix $\mathbf{X}^{(\mathbf{k})}$ is known. We will calculate $\mathbf{X}^{(\mathbf{k+1})}$ by successively calculating:

$$\begin{aligned}
&\tilde{A}_1 = \mathrm{A}_1 + \mathrm{C}_1 F_1^{(k-1)} ; \qquad \bar{A}_1 = \mathrm{A}_1 + \mathrm{C}_2 F_2^{(k-1)} \\
&\mathcal{A}_1 = \mathrm{diag}\,(\, \overline{\mathrm{A}}_1 \,, \widetilde{\mathrm{A}}_1) \\
&\mathcal{S}\left(\mathbf{X}^{(\mathbf{k})}\right) = \mathbf{X}^{(\mathbf{k})} \mathcal{B} + \mathcal{A}_1{}^T \mathbf{X}^{(\mathbf{k})} \mathcal{C} \\
&\mathcal{F}_{\mathbf{X}^{(\mathbf{k})}} = \left[\Re\left(\mathbf{X}^{(\mathbf{k})}\right)\right]^{-1} \left[\mathcal{S}\left(\mathbf{X}^{(\mathbf{k})}\right)\right]^{\mathrm{T}} = \mathrm{diag}\left(F_1^{(k)}, F_2^{(k)}\right) \\
&\qquad\qquad = \mathrm{diag}\left(F_1\left(\mathbf{X}^{(\mathbf{k})}\right), F_2\left(\mathbf{X}^{(\mathbf{k})}\right)\right) \\
&\tilde{A}_0 = \mathrm{A}_0 + \mathrm{B}_1 F_1^{(k)} \;\; \bar{A}_0 = \mathrm{A}_0 + \mathrm{C}_2 F_2^{(k)}, \qquad (1.7) \\
&\mathcal{A}_0 = \mathrm{diag}\,(\, \overline{\mathrm{A}}_0 \,, \widetilde{\mathrm{A}}_0) \\
&\bar{Q}_1 = \mathrm{Q}_1 + \left(F_2^{(k)}\right)^T R_{12} F_2^{(k)} \;\; ; \quad \bar{Q}_2 = \mathrm{Q}_2 + \left(F_1^{(k)}\right)^T R_{21} F_1^{(k)} \\
&\mathcal{Q} = \mathrm{diag}\,(\, \overline{\mathrm{Q}}_1 \,, \overline{\mathrm{Q}}_2)
\end{aligned}$$

After these notations, we apply the following iteration method:

$$\begin{aligned}
&M(\mathbf{X}^{(\mathbf{k})}) = \left(\mathcal{A}_0 + \mathcal{B}\mathcal{F}_{\mathbf{X}^{(\mathbf{k})}}\right)^{\mathrm{T}} \mathbf{X}^{(\mathbf{k+1})} + \mathbf{X}^{(\mathbf{k+1})} \left(\mathcal{A}_0 + \mathcal{B}\,\mathcal{F}_{\mathbf{X}^{(\mathbf{k})}}\right) \\
&+ \mathbf{T}_{(\mathbf{X}^{(\mathbf{k})})} + \Pi_{(\mathbf{X}^{(\mathbf{k})})}\left(\mathbf{X}^{(\mathbf{k})}\right) = \mathbf{0} \qquad (1.8)
\end{aligned}$$

under notations

$$\mathbf{T}_{(\mathbf{Z})} = \begin{pmatrix} \boldsymbol{I} \\ \mathbf{F}_{(\mathbf{X}^{(\mathbf{k})})} \end{pmatrix}^{\boldsymbol{T}} \begin{pmatrix} \boldsymbol{Q} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{R} \end{pmatrix} \begin{pmatrix} \boldsymbol{I} \\ \mathbf{F}_{(\mathbf{X}^{(\mathbf{k})})} \end{pmatrix}$$

$$\Pi_{(\mathbf{X}^{(\mathbf{k})})}\left(\mathbf{X}^{(\mathbf{k})}\right) = \begin{pmatrix} \boldsymbol{I} \\ \mathcal{F}_{\mathbf{X}^{(\mathbf{k})}} \end{pmatrix}^{\boldsymbol{T}} \begin{pmatrix} \mathcal{Q} & \mathbf{0} \\ \mathbf{0} & R \end{pmatrix} \begin{pmatrix} \boldsymbol{I} \\ \mathcal{F}_{\mathbf{X}^{(\mathbf{k})}} \end{pmatrix}.$$

At the suggestion that $\mathcal{A}_0$, $\mathcal{A}_1$, $\mathcal{Q}$ are given matrices with corresponding properties, that the convergence of iteration (1.8) and tjeir properties are derived in the nest theorem:

Theorem 1.3 (a generalization of Theorem 1.2) Assume there exist symmetric matrices $\widehat{\mathbf{X}}$ and $\mathbf{X}^{(0)}$, for which $\mathcal{R}(\widehat{\mathbf{X}}) \geq \mathbf{0}$ and $\mathbf{X}^{(0)} > \widehat{\mathbf{X}}$, $\mathcal{R}(\mathbf{X}^{(0)}) < \mathbf{0}$ and $\mathcal{A}_0 + \mathbf{B}\,\mathbf{F}_{(\mathbf{X}^{(0)})}$ is a stable matrix, where $\mathbf{F}_{(\mathbf{X}^{(0)})} = \left[\mathbf{R}\left(\mathbf{X}^{(0)}\right)\right]^{\{-1\}}\left[\mathbf{S}\left(\mathbf{X}^{(0)}\right)\right]^{\mathrm{T}}$. Under the above conditions, matrix sequence $\{\mathbf{X}^{(\mathbf{k})}\}_{\mathbf{0}}^{\infty}$, constructed by (1.8) satisfies the properties:

(i) $\mathbf{X}^{(\mathbf{s})} > \mathbf{X}^{(\mathbf{s+1})}$, $\mathbf{X}^{(\mathbf{s})} > \widehat{\mathbf{X}}$ and $\mathcal{R}(\mathbf{X}^{(\mathbf{s})}) < 0$, s=0,1,2, ...

(ii) $\mathcal{A}_0 + \mathbf{B}\,\mathbf{F}_{(\mathbf{X}^{(\mathbf{s})})}$ is a stable matrix for s=0,1,2, ...

(iii) $\mathbf{lim}_{\{\mathbf{s}\to\infty\}}\, \mathbf{X}^{(\mathbf{s})} = \widetilde{\mathbf{X}}$ is a solution of Riccati equation $\mathcal{R}(\mathbf{X}) = \mathbf{0}$ with the property $\widetilde{\mathbf{X}} \geq \widehat{\mathbf{X}}$. Moreover, if $\mathbf{X}^{(0)} \geq \mathbf{X}$ for all solutions $\mathbf{X}$ to $\mathcal{R}(\mathbf{X}) = \mathbf{0}$, then $\widetilde{\mathbf{X}}$ is a maximal solution.

(iv) The eigenvalues to matrix $\mathcal{A}_0 + \mathbf{B}\,\mathbf{F}_{(\widetilde{\mathbf{X}})}$ are in the closed left half-plane relative to the ordinate axis (the ordinate axis is included). If $\mathcal{R}(\widehat{\mathbf{X}}) > \mathbf{0}$, then the eigenvalues of $\mathcal{A}_0 + \mathbf{B}\,\mathbf{F}_{(\widetilde{\mathbf{X}})}$ are in the left open half-plane relative to the y-axis.

We will conduct experiments with several examples of finding the stabilizing solution of the Riccati equation (1.5), using for this purpose the iterative method introduced through the formulas (1.6) -(1.8). In the experimental part, we use the Anaconda environment with Python 3.7. We present formula (1.8) in a more convenient form for programming and execution of the iteration:

$$\begin{aligned}&\left(\mathcal{A}_0 + \mathbf{B}\,\mathbf{F}_{(\mathbf{X}^{(\mathbf{k})})}\right)^{\mathrm{T}} \mathbf{X}^{(\mathbf{k+1})} + \mathbf{X}^{(\mathbf{k+1})}\left(\mathcal{A}_0 + \mathbf{B}\,\mathrm{F}_{(\mathrm{X}^{(\mathrm{k})})}\right) + \mathcal{Q} \qquad (1.9)\\ &+\left(\mathbf{F}_{(\mathbf{X}^{(\mathbf{k})})}\right)^{\mathrm{T}} \mathbf{R}\,\mathbf{F}_{(\mathbf{X}^{(\mathbf{k})})} + \left(\mathcal{A}_1 + \mathrm{C}\,\mathbf{F}_{(\mathbf{X}^{(\mathbf{k})})}\right)^{\mathrm{T}} \mathbf{X}^{(\mathbf{k})}\left(\mathcal{A}_1 + \mathrm{C}\,\mathbf{F}_{(\mathbf{X}^{(\mathbf{k})})}\right) = 0\end{aligned}$$

The iterative formula (1.9) is called the Lyapunov formula because at each step a Lyapunov matrix equation about the unknown $\mathbf{X}^{(\mathbf{k+1})}$ is solved. The dissertation presents an algorithm for the implementation of iteration formula (1.9).

We will present an example realized by this algorithm. The matrix coefficients of system (1.4) are presented in Python terminology for each example.

Example 1.1.

```
import numpy as np
n=3
m1=2
m2=3
A0 = np.matrix([[-1.5, 0.17,-0.049],[0.07, -1.42, -0.027],[0.04, -0.11,-1.47]])
A1 = np.matrix([[0.7, 0.19,-0.04],[0.24, 0.9,0.9],[0.3, 0.1,0.15]])
Q1=0.3*np.matlib.identity(n)
Q2=0.025*np.matlib.identity(n)
B1= np.matrix([[0.0, 0.],[0.05, 0.1],[0.04, 0.15]]);
C1= np.matrix([[0., 0.1],[1.1, 0],[0., 0.02]]);
B2= np.matrix([[0.1, 0.5 , 0.4],[0., 0, 0.08],[0., 0., 2.2]])
C2 = np.matrix([[0.1, 0. , 0.],[0., 1.5, 0.0],[0.1, 0.05, 0.0]])
```

```
R11 = np.matlib.identity(m1);
R11[0,0]=4.0
R11[m1-1,m1-1]=5.0
R21 = np.matlib.identity(m1)/2.
R21[1,1]=10.
R22 = np.matlib.identity(m2)
R22[0,0]=2.
R22[m2-1,m2-1]=8.
R12 = np.matlib.identity(m2)/2.
R12[1,1]=2.
R12[m2-1,m2-1]=3.
```

When executing Example 1.1, the specific values are n=3, tol=1.0e-8. We choose $X_1^{(0)}$ = diag [6,6,6], $X_2^{(0)}$ = diag [9,9,9]. At these values, we check whether the conditions of the Theorem are fulfilled, namely $R_1\left(X_1^{(0)}, X_2^{(0)}\right) < 0$, $R_2\left(X_1^{(0)}, X_2^{(0)}\right) < 0$.

When executing Example 1.1, the specific values are n=3, tol=1.0e-8. We choose $X_1^{(0)}$ = diag [6,6,6], $X_2^{(0)}$ = diag [9,9,9]. At these values, we check whether the conditions of the Theorem are fulfilled, namely $R_1\left(X_1^{(0)}, X_2^{(0)}\right) < 0$, $R_2\left(X_1^{(0)}, X_2^{(0)}\right) < 0$. For the lower bound of the matrix sequence we choose $\hat{X}_1 = \hat{X}_2 = \text{diag}\,[0.0002, 0.0002, 0.0002]$, and $R_1\left(\hat{X}_1, \hat{X}_2\right) > 0$, $R_2\left(\hat{X}_1, \hat{X}_2\right) > 0$. Moreover, the matrix $\mathcal{A}_0, + \mathbf{B}\,\mathbf{F}_{(\mathbf{X}^{(0)})}$ is stable, i.e., it has eigenvalues with negative real parts. The theorem's conditions are satisfied and we can apply iteration (1.9) for chosen matirces $\left(X_1^{(0)}, X_2^{(0)}\right)$.

For both solutions $\tilde{X}_1, \tilde{X}_2$ (which are 3x3 matrices) we obtain:

$$\tilde{X}_1 = \begin{pmatrix} 0.13952043 & 0.04144027 & 0.02188102 \\ 0.04144027 & 0.15624824 & 0.03732627 \\ 0.02188102 & 0.03732627 & 0.14154421 \end{pmatrix},$$

$$\tilde{X}_2 = \begin{pmatrix} 0.0120035 & 0.003909 & 0.00222309 \\ 0.003909 & 0.01359531 & 0.0036183 \\ 0.00222309 & 0.0036183 & 0.01226303 \end{pmatrix}.$$

The solutions $\tilde{X}_1, \tilde{X}_2$ are obtained after 25 iteration steps of formula (1.9) and have the properties derived in the theorem – the matrix $\mathcal{A}_0 + \mathbf{B}\,\mathbf{F}_{(\tilde{\mathbf{X}})}$ is stable. But we are looking for the Nash equilibrium, which is obtained after calculating the matrices $F_1\ (\tilde{\mathbf{X}}), F_2\ (\tilde{\mathbf{X}})$:

$$F_1\ (\tilde{\mathbf{X}}) = \begin{pmatrix} -0.02010912 & -0.04090623 & -0.0385815 \\ -0.00398803 & -0.00569488 & -0.00583653 \end{pmatrix}$$

$$F_2\ \left(\tilde{\mathbf{X}}\right) = \begin{pmatrix} -0.00138726 & -0.00071184 & -0.00050222 \\ -0.01597642 & -0.02063644 & -0.01883039 \\ -0.00125061 & -0.00132644 & -0.00351967 \end{pmatrix}$$

The conducted experiments confirm the applicability of the proposed iteration formula (1.8) (which is equivalent to (1.9)) for finding a stabilizing solution of the Riccati equation (1.5).

Scientific contributions in the first chapter:

The proposed iteration method through formulas (1.6) -(1.8) is new and finds a solution to the nonlinear matrix system (1.4). The found solution equations leads to a Nash equilibrium for a linear quadratic stochastic game studied by (Zhu, Zhang, 2013). The proposed iterative method is published in (Ivelin Ivanov и V. Tanov, 2018, An Iterative Method for an Equilibrium Point of Linear Quadratic Stochastic Differential Games with State and Control-Dependent Noise, Ann. Acad. Rom. Sci.,2018).

Authors's publication on the first chapter

Ivelin G. Ivanov, **Vladislav Tanov**, An Iterative Method for an Equilibrium Point of Linear Quadratic Stochastic Differential Games with State and Control-Dependent Noise, *Mathematics, and its Applications / Annals of AOSR*, 10(2), 202-210, 2018. (Scopus)

Chapter two. Linear quadratic game models with two players

The author's published contributions in the second chapter are in the field of positive games. We reflect the part of the dissertation work describing these scientific contributions.

2.4. Game models for positive games

Differential systems of the type

$$dx = Ax\, dt + B_1 u_1 dt + B_2 u_2\, dt,\ \ x(0) = x_0, \qquad (2.18)$$

are called positive if, for all non-negative initial states x_0 and non-negative control functions u_1 and u_2, then the state vector x(t) is non-negative at any instant.

We introduce two functionals (i=1,2):

$$J_i(u_1, u_2) = \int_0^\infty (x^T Q_i x + \textstyle\sum_{j=1}^{2} u_j^T\, R_{ij} u_j)\, dt,\ \text{за}\ i = 1{,}2, \qquad (2.19)$$

each of them must be minimize, under the influence of the function u_i, which is the strategy of the i-th player.

Ще разгледаме матричното Рикатиево уравнение:

$$0 = -\begin{pmatrix} A^T & 0 \\ 0 & A^T \end{pmatrix}\begin{pmatrix} X_1 \\ X_2 \end{pmatrix} - \begin{pmatrix} X_1 \\ X_2 \end{pmatrix} A - \begin{pmatrix} Q_1 \\ Q_2 \end{pmatrix} + \begin{pmatrix} X_1 \\ X_2 \end{pmatrix} \begin{pmatrix} S_1 & S_2 \end{pmatrix} \begin{pmatrix} X_1 \\ X_2 \end{pmatrix}, \ (2.20)$$

where $(-A)$ is an n x n Z-матрица, $S_j = B_j R_{jj}^{-1} B_j^T$ $(S_j = {S_j}^T)$ is a nonpositive matrix for $j = 1{,}2$, and Q_j is a symmetric square nonnegative matrix on dimension n, R_{jj} is a symmetric square negative definite matrix of corresponding dimension, B_j is a non-negative matrix of dimension n x m_j, for j=1,2, and X_1 and X_2 are unknown matrices. We will use matrices of different orders.

To control positive systems of the above type, it is necessary to solve an equation of the type (2.20). Newton's method was applied to the solution of equation (2.20) and this was done by Jank, Kremer in (Jank, Kremer, 2005).

When we write A>0 (A≥0) for a matrix A with dimension n x m we mean a_{ij}>0 (a_{ij}≥0) for each 1≤i≤n and 1≤j≤m, i.e. the matrix elements are positive (non-negative). When we write A>B (A≥B) for matrices A and B of dimension n x m we mean a_{ij}>b_{ij} (a_{ij}≥b_{ij}) for each 1≤i≤n and 1≤j≤m. For the considerations, the matrices Q_k and S_k are Q_k≥0, and S_k≤0, k=1,2. In the course of reasoning, we will use the fact that the matrix equation AXB=C is equivalent to the linear system $(B^T \otimes A\,)\, vec\ X = vec\ C,$ where vec means the transformation of the corresponding matrix into a column vector, following the columns of the matrix .

A real n x n matrix A will be called a Z-matrix if there exists a real number s and a matrix C≥0 n x n matrix such that A=sI_n -C, where I_n is a unit matrix of order n. A real n x n nonsingular matrix A=(a_{ij}) is called an M-matrix if a_{ij}≤0 for i≠j and $A^{-1} \geq 0$.

We consider linear quadratic differential games for positive linear systems with the inverse information structure and two players. Newton's accelerated method for obtaining the stabilizing solution of the two Riccati equations is presented in (Jank, Kremer, 2005), where convergence properties of the method are proved. Furthermore, the Lyapunov iteration method for computing the Nash equilibrium point is presented in (Baeva, 2016). Moreover, the convergence properties of the iteration formula are derived and proved. The implementation of the algorithm is illustrated on some numerical examples. The following theorem on the properties of non-negative matrices is known:

Theorem 2.1. For a Z-matrix A the following statements are equivalent:

(i) A is an M-matrix.

(ii) $A^{-1} \geq 0$.

(iii) $Av > 0$ for any vector $v > 0$.

(iv) All eigenvalues of the matrix A have positive real parts, i.e. the matrix (-A) is a stable matrix.

The emphasis in this section on positive games is on developing fast and efficient methods for finding Nash equilibria by solving Riccati equations. We have published such studies in the following two of our publications (I. Ivanov, N. Netov, V. Tanov, Iteratively Computation the Nash Equilibrium Points in the Two-Player Positive Games, 2016), (Ivelin Ivanov, V. Tanov, Computing the Nash Equilibrium for LQ Games on Positive Systems Iteratively, 2018).

2.4.1.Newton’s method

Newton's method for solving the equation (2.20) is studied and presented in (Jank, Kremer, 2005) by an iterative formula:

$$-K_{i+1}(A - SK_i) - (D - K_iS)K_{i+1} = Q + K_iSK_i\,,$$
$$i = 0{,}1{,}2\, \dots, \tag{2.21}$$

where $D = \begin{pmatrix} A^T & 0 \\ 0 & A^T \end{pmatrix}$, $Q = \begin{pmatrix} Q_1 \\ Q_2 \end{pmatrix}$, $S = (S_1 \quad S_2)$. The convergence of the method is proved by the following theorem (Theorem 5 from Jank, Kremer, 2005).

The iteration formula we propose and for which we prove convergence is the following:

$$-A^{(k)T} X_1^{(k+1)} - X_1^{(k+1)} A^{(k)} = \tilde{Q}_1^{(k)} \quad ,$$

$$A^{(k)} = A - S_1 X_1^{(k)} - S_2 X_2^{(k)},$$

$$\tilde{Q}_1^{(k)} = Q_1 + X_1^{(k)} S_1 X_1^{(k)} + X_2^{(k)} S_{12} X_2^{(k)} \ ,$$

and

$$-A^{(k)T} X_2^{(k+1)} - X_2^{(k+1)} A^{(k)} = \tilde{\tilde{Q}}_2^{(k)} \ ,$$

$$\tilde{\tilde{Q}}_2^{(k)} = Q_2 + X_2^{(k)} S_2 X_2^{(k)} + X_1^{(k+1)} S_{21} X_1^{(k+1)} \ .$$

The last iteration process is investigated for convergence and sufficient conditions guaranteeing this convergence are derived (Theorem 2, I. Ivanov, N.Netov, V.Tanov, 2016).

2.4.3.Implicit iteration formulas

According to the research done by Ma and Lu (Ma, Lu, 2016), here we will introduce a modification of the Newton method considered by (2.21):

$$-\mathcal{Y}_k(\gamma I + A - SX_k) = (\gamma I_{2n} - D)X_k - \mathcal{Q} \qquad (2.30)$$

$$(\gamma I_{2n} + D - \mathcal{Y}_k S)X_{k+1} = \mathcal{Y}_k(\gamma I - A) - \mathcal{Q}$$

X_0= 0 , k=0,1,2, … , $\gamma < 0$.

We call the last formula (2.30) Linearized Implicit Newton Iteration and will use the abbreviation LINI. We will present two statements that introduce properties of nonnegative matrices, and these properties will be useful in the following reasoning.

We rewrite the matrix function $\mathcal{R}\,(\mathcal{X})$ of the type $\mathcal{R}\,(\mathcal{X}) = \begin{pmatrix} R_1(X_1, X_2) \\ R_2(X_1, X_2) \end{pmatrix}$,

were

$$R_1(X_1, X_2) = -\mathrm{A}^{\mathrm{T}}\,\mathrm{X}_1 - \mathrm{X}_1\,\mathrm{A} + \mathrm{X}_1\,\mathrm{S}_1\,\mathrm{X}_1 + \mathrm{X}_1\mathrm{S}_2\,\mathrm{X}_2 - \mathrm{Q}_1$$

$$R_2(X_1, X_2) = -\mathrm{A}^{\mathrm{T}}\,\mathrm{X}_2 - \mathrm{X}_2\,\mathrm{A} + \mathrm{X}_2\,\mathrm{S}_1\,\mathrm{X}_1 + \mathrm{X}_2\mathrm{S}_2\,\mathrm{X}_2 - \mathrm{Q}_2\ .$$

General equation $\mathcal{R}\,(\mathcal{X}) = 0$ is equivalent to the sum of the two equations $R_1(X_1, X_2) = 0$ and $R_2(X_1, X_2) = 0$. We can use the cell (block) structure of the matrix coefficients in (2.30) to justify the following formulas that define a new iteration method. We will call it the Alternately Linearized Implicit Decoupled Iteration (ALIDI) method. The long name reveals its qualities – it uses linear matrix equations, implicitly reaches the solution, and each iteration equation is independent of the others:

$$Y_1^{(k)}(\,\gamma I_n + A - S_1 X_1^{(k)} - S_2 X_2^{(k)}) = (\gamma I_n - A^T)\, X_1^{(k)} - Q_1 \qquad (2.31)$$

$$Y_2^{(k)}(\,\gamma I_n + A - S_1 X_1^{(k)} - S_2 X_2^{(k)}) = (\gamma I_n - A^T)\, X_2^{(k)} - Q_2 \qquad (2.32)$$

$$(\gamma I_n + A^T - Y_1^{(k)} S_1) X_1^{(k+1)} = Y_1^{(k)}(\gamma I_n - A + S_2 X_2^{(k)}) \,- Q_1 \qquad (2.33)$$

$$(\gamma I_n + A^T - Y_2^{(k)} S_2) X_2^{(k+1)} = Y_2^{(k)}(\gamma I_n - A + S_1 X_1^{(k)}) \,- Q_2 \qquad (2.34)$$

$X_1^{(0)} = X_2^{(0)} = 0,$ k=0,1,2, ……, , $\gamma < 0.$

Iterative method (2.31) - (2.34) was derived, studied and published here (Ivelin Ivanov, V. Tanov, 2018, pp. 230-244).

We will continue with the study of the proposed iterative method (2.31) -(2.34) and the derivation of its properties related to the convergence of the method. We will derive these properties under some assumptions that we will formulate in the following theorem and by using the research of Bai et al. (Bai et al., 2006).

In the following theorem, proved in our publication, we will derive sufficient conditions for the convergence of the proposed method.

Theorem 2.4. (Ivelin Ivanov, V.Tanov, 2018, стр. 230-244) We assume that the matrix (–A) is an M-matrix and $Q_1 \geq 0, Q_2 \geq 0$ and $S_1 \leq 0, S_2 \leq 0, \gamma < 0$, such that (- $\gamma\, \mathrm{I_n} -$ A) is an M-matrix, and ($\gamma\, \mathrm{I_n} -$ A) is nonpositive. We assume the symmetric nonnegative matrices exist $\hat{X}_1, \hat{X}_2$, such that $R_i(\hat{X}_1, \hat{X}_2) \geq 0$, i=1,2 and $-\,\mathrm{A} +\ \mathrm{S_1}\, \hat{X}_1 +\ \mathrm{S_2}\ \hat{X}_2$ is an M-matrix. We construct matrix sequences $\left\{X_1^{(k)}\right\}, \left\{X_2^{(k)}\right\},\ k = 0, \dots, \infty,$ throught (2.29) -(2.32). The following properties are satisfied:

(i) $\tilde{X}_i \geq X_i^{(k+1)} \geq Y_i^{(k)} \geq X_i^{(k)}$, i=1,2 , k=0,1, ….;

(ii) $R_i\left(X_1^{(k)}, X_2^{(k)}\right) \leq 0,\quad R_i\left(Y_1^{(k)}, Y_2^{(k)}\right) \leq 0$, $R_i\left(X_1^{(k+1)}, X_2^{(k+1)}\right) \leq 0,$ i=1,2, k=0,1, ….;

(iii) The matrix sequences $\left\{X_1^{(k)}\right\},\left\{X_2^{(k)}\right\},\ k=0,\dots,\infty,$ are convergent to minimal nonnegative solution ($\tilde{X}_1, \tilde{X}_2$) to couple of Riccati equations $R_1(X_1,X_2)=0$ and $R_2(X_1,X_2)$ =0, for which $\tilde{X}_i \le \hat{X}_i$ and the matrix $\mathrm{A}-\ \mathrm{S}_1\,\tilde{X}_1-\ \mathrm{S}_2\ \tilde{X}_2$ is stable.

The theorem is presented without a proof.

Numerical examples

We will apply the described iterative methods for finding the equilibrium on two examples with specific data.

Example 2.5. We describe the matrix coefficients A, Bi, Qi and Rii for i=1,2 in the Matlab environment.

```
A=abs(randn(n))/99;    s=max(abs(eig(A)))+4.5;  gamma= -5.0;
for  i=1:n,    A(i,i)=-(A(i,i))-s;   end
B1  = abs(randn(n,1))/2;
B2 = eye(n,n);      B2(n,n)=n/3;
Q1=zeros(n,n);   Q1(1,1)=n/2;   Q1(n,n)=1.5;
Q2 = 2 Q1;    R11=-1;   R22 = -eye(n,n);
R22(1,1)=-50;     R_{22}(n,n) = -30;
```

We run Example 2.5 for different values of n, and 100 iterations for a fixed value of n. We choose $X_1^{(0)}=X_2^{(0)}=0$, and establish $R_i\left(X_1^{(0)},X_2^{(0)}\right)=\ -Q_i \le 0$,, i.e. the matrix is non-positive. We select the two square matrices:

$$\hat{X}_1=\begin{pmatrix} 0.3 & 0.01 & \cdots & 0.01 & 0.01 \\ 0.01 & 0.3 & \cdots & 0.01 & 0.01 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0.01 & 0.01 & \cdots & 0.3 & 0.01 \\ 0.01 & 0.01 & \cdots & 0.01 & 0.3 \end{pmatrix},$$

$$\hat{X}_2=\begin{pmatrix} 0.5 & 0.01 & \cdots & 0.01 & 0.01 \\ 0.01 & 0.5 & \cdots & 0.01 & 0.01 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0.01 & 0.01 & \cdots & 0.5 & 0.01 \\ 0.01 & 0.01 & \cdots & 0.01 & 0.5 \end{pmatrix}.$$

The conditions of Theorem 2.4 are satisfied for the matrices $\hat{X}_1,\hat{X}_2$, т.е. $\hat{X}_1 \ge X_1^{(0)},\hat{X}_2 \ge X_2^{(0)}$, $R_i\left(X_1^{(0)},X_2^{(0)}\right) \le 0,\ R_i\left(\hat{X}_1,\hat{X}_2\right) \ge 0,$ i=1,2. The matrix $(-\ \mathrm{A}+\ \mathrm{S}_1\,\hat{X}_1+\ \mathrm{S}_2\ \hat{X}_2)$ is a nonsingular M-matrix. The calculated solution $\tilde{X}_1,\tilde{X}_2$ satisfies the following inequalities $\tilde{X}_1 \le\ \hat{X}_1, \tilde{X}_2 \le\ \hat{X}_2$ and additionally $(-\ \mathrm{A}+\ \mathrm{S}_1\,\tilde{X}_1,+\ \mathrm{S}_2\ \tilde{X}_2)$ is an M-matrix. In Table 2.2. results of the experiments are described. The proposed implicit method through formulas (2.31) -(2.34) is faster although it takes a higher average number of iteration steps.

Example 2.6. We determine the matrix coefficients:

```
A=abs(randn(n))/10;        s=max(abs(eig(A)))+4.5;  gamma= -5.0;
for   i=1:n,    A(i,i)=-(A(i,i))-s;     end
B1 = abs(randn(n,1))/2;
B2 = eye(n,n);     B2(n,n)=abs(randn);
Q1=zeros(n,n);       Q1(1,1)=n/2;    Q1(n,n)=1.5;
Q2 = 2 Q1;             R11=-1;
R22 = -eye(n,n);       R22(1,1)=-80;     R22(n,n)=-90;
```

Tables 2.2 and 2.3 present the results for different values of n. 100 replicates were done for a particular value of n. The maximum number of iteration steps to reach the solution (maxIt), the average number of iteration steps (avIt), and the CPU time to complete all iterations (CPU) are reported. The results for the Newton method (2.21) and ALIDI method (2.31)-(2.34) are described.

Table 2.2.

	Newton NI (2.21)			ALIDI (2.31)-(2.34)		
N	maxIt	avIt	CPU	maxIt	avIt	CPU
8	4	3.1	0.31sec.	6	5.01	0.125 sec.
16	4	3.27	0.43 sec.	10	5.5	0.19 sec.
24	5	3.57	0.73 sec.	17	6.67	0.37 sec.
32	5	3.67	1.16 sec.	14	8.06	0.73 sec.

Table 2.3.

	Newton NI (2.21)			ALIDI (2.31)-(2.34)		
N	maxIt	avIt	CPU	maxIt	avIt	CPU
15	4	3.2	0.39 sec.	8	5.11	0.1 sec.
25	4	3.38	0.73 sec.	12	6.9	0.29 sec.
40	6	3.69	1.66 sec.	10	8.85	0.89 sec.
55	6	3.66	3.25 sec.	22	10.8	2.03 sec.

2.5.An improved iterative method

In our next publication (Ivelin Ivanov, Vladislav Tanov, 2020) we improve the iteration formula (2.31) -(2.34) and propose a new modification:

$$Y_1^{(k)}(\ \gamma I_n + A - S_1 X_1^{(k)} - S_2 X_2^{(k)}) = (\gamma I_n - A^T)\ X_1^{(k)} - Q_1 \qquad (2.35)$$

$$Y_2^{(k)}(\ \gamma I_n + A - S_1 X_1^{(k)} - S_2 X_2^{(k)}) = (\gamma I_n - A^T)\ X_2^{(k)} - Q_2 \qquad (2.36)$$

$$(\gamma I_n + A^T) X_1^{(k+1)} = Y_1^{(k)}(\gamma I_n - A + S_1\ Y_1^{(k)} + S_2 X_2^{(k)})\ - Q_1 \qquad (2.37)$$

$$(\gamma I_n + A^T)X_2^{(k+1)} = Y_2^{(k)}(\gamma I_n - A + S_1 X_1^{(k)} + S_2 X_2^{(k)}) - Q_2 \qquad (2.38)$$

$X_1^{(0)} = X_2^{(0)} = 0,$ k=0,1,2, ……, , $\gamma < 0.$

We derive the following properties for iterative process (2.35) -(2.38):

Theorem 2.5. (Ivelin Ivanov, Vladislav Tanov, 2020) We suppose the matrix (–A) is an M-matrix and $Q_1 \geq 0, Q_2 \geq 0$ and $S_1 \leq 0, S_2 \leq 0, \gamma < 0$, such that $(-\gamma \, \mathrm{I_n} - \mathrm{A})$ is an M-matrix and $(\gamma \, \mathrm{I_n} - \mathrm{A})$ is nonpositive. In addition, there exist symmetric nonnegative matrices $\hat{X}_1, \hat{X}_2$, with $R_i(\hat{X}_1, \hat{X}_2) \geq 0$, i=1,2 and $-\mathrm{A} + \mathrm{S_1}\, \hat{X}_1 + \mathrm{S_2}\, \hat{X}_2$ is an M-matrix. We construct the matrix sequences $\{X_1^{(k)}\}, \{X_2^{(k)}\},\ k = 0, \dots, \infty,$ according to iterations (2.35) -(2.38). The following properties of the matrix sequences are satisfied:

(i) $\tilde{X}_i \geq X_i^{(k+1)} \geq Y_i^{(k)} \geq X_i^{(k)}$, i=1,2 , k=0,1, ….;

(ii) $R_i\left(X_1^{(k)}, X_2^{(k)}\right) \leq 0,\quad R_i\left(Y_1^{(k)}, Y_2^{(k)}\right) \leq 0\ ,\ R_i\left(X_1^{(k+1)}, X_2^{(k+1)}\right) \leq 0,$ i=1,2, k=0,1, ….;

(iii) The matrix sequences $\{X_1^{(k)}\}, \{X_2^{(k)}\},$ k=0,…,∞ converge to the minimal nonnegative solution $\tilde{X}_1, \tilde{X}_2$ of the couple of Riccati equations $R_1(X_1, X_2) = 0$ and $R_2(X_1, X_2)$ =0, for which $\tilde{X}_1 \leq \hat{X}_i$.

(iv) If $-\mathrm{A} + \mathrm{S_1}\, \hat{X}_1 + \mathrm{S_2}\, \hat{X}_2$ is an M-matrix, and the solution $\tilde{X}_1, \tilde{X}_2$ is left-right stable solution to the couple of Riccati equations $R_1(X_1, X_2) = 0$ and $R_2(X_1, X_2)$ =0.

The proof follows similar reasoning to Lemma 2.8.

Example 2.7. We take the matrix coefficients: (description in Matlab)

```
 A=[-2.74 0.06 0.015 0.099;            B1=[0.5938; 0.2985; 0.49; 0.98];
      0.2 -2.5 0.064 0.08;             B2=[2.8  0  0  0;
      0.004 0.15 -2.56 0.09;                 0  2.9  0  0;
      0.14 0.12 0.21 -2.57];                 0   0  2.84 1.5;
                                             0   0  1.5 1.3];

Q1=eye(n,n)/2;   Q1(1,1)=n/2;    Q1(n,n)=1.5;
Q2 =.5 * Q1;        R11=-1.909;
R22 = -eye(n,n); R22(1,1)=-50; R22(n,n)=-30;
S1=B1*inv(R11)*B1';
S2=B2*inv(R22)*B2';
```

The results are given in Table 2.4.

Table 2.4.

	ALIDI (2.31) -(2.34)		(2.35) -(2.38)	
γ	avIt	CPU	avIt	CPU
-5	402	2.67sec.	431	2.7 sec.
-3	256	1.76 sec.	278	1.78 sec.
-1	112	0.82 sec.	112	0.79 sec.
-0.5	40	0.37 sec.	39	0.34 sec.
-0.25	80	0.62 sec.	77	0.57 sec.

Contributions in Chapter Two:

The contributions in the second chapter are the proposed two new iterative methods for finding a solution of a cellular Riccati equation with special coefficients:

- iteration method (2.31) -(2.34),
- iteration method (2.35) -(2.38).

For the two iteration methods, properties for their convergence are theoretically derived. Both methods are distinguished by a clear implementation scheme and easy computer implementation. Experiments demonstrate their effectiveness.

Publications of the author on the second chapter.

1.Ivan Ivanov, Nikolay Netov, Vladislav **Tanov**, Iteratively Computation the Nash Equilibrium Points in the Two-Player Positive Games. International Journal of Mathematical and Computational Methods, **1**, 378-381, 2016

2.Ivelin Ivanov, Vladislav **Tanov**, Computing the Nash Equilibrium for LQ Games on Positive Systems Iteratively, Mathematics and its Applications / Annals of AOSR, 10(2), 230-244, 2018. (Scopus)

3.Ivelin Ivanov, Vladislav **Tanov**, A Nonsymmetric Nash-Riccati Equation and Decoupled Schemes for a Stabilizing Solution, Applied Mathematics E-Notes, 20(2020), 357-366, Applied Mathematics E-Notes, 20(2020), 357-366. (Scopus)

Chapter Three. Optimization of unbalanced sets

Methods and algorithms

Imbalanced datasets, also known as class imbalance, are common in the field of machine learning (Samuel, 1959) with applications in various fields, for example, detection of cardiovascular and liver diseases, oil spills in satellite imagery, and tasks of information retrieval and filtering, and others (Raskutti and Kowalczyk, 2004) (Wu and Chang, 2003). The task of improving the performance of class imbalance problems consisting of a majority class (larger number of observations) and a minority class is a very important aspect for various optimization problems. The goal is to reach an optimized, balanced subset through majority class optimization to be used for prediction with high accuracy values.

Bohacik and Zabovsky studied probabilistic realization with a given controlled discretization using expertise in the field of heart disease (Bohacik and Zabovsky, 2019). The algorithmic methodology is implemented in a Waikato environment for knowledge analysis as a NaïveBayes class with the Fayyad-Irani numerical attribute discretization (Fayyad and K. B. Irani, 1993). The study was based on k-fold (k=10) cross-validation and used sensitivity, specificity, and their sum as measures. Sensitivity (True Positive Rates) represents the ability of the algorithm to identify true positive (TP) cases with respect to all positive results, with the following expression: TP/(TP+FN). False negatives (FN) will be considered negative cases when, they are positive. The specificity indicates the ability of the algorithm to identify cases of true negative rates (TN) with respect to all negative results as follows: TN/(TN+FP). False positives (FPs) will be considered positive cases when they are negative. Bohacik and Zabovsky use the sum of sensitivity and specificity as an overall scoring system for comparing algorithms (Bohacik and Zabovsky, 2019).

The experiments we conduct are on several data sets used by other authors and freely available on the Internet, shown in Table 3.1. The results of the experiments show an advantage over the above-mentioned data handling methods, such as resampling, bootstrap and selection of statistically significant variables (feature selection). We will consider the following hypothesis: The unbalanced set optimization algorithm is effective in improving the prediction accuracy with classification models. The hypothesis is tested against a specific set of measures obtained for standard evaluation of classification models, such as Accuracy, Precision, Recall and F1 Score.

Table 3.1

Datasets	Charecte-ristics	Classes and Number of observations	Source
Diabetes	8	Class 0 – 500 Class 1 - 268	link
Statlog Heart	13	Class 1 – 150 Class 2 - 120	link

Indian Liver Patient Dataset (ILPD)	10	Class 1 – 416 Class 2 - 167	links

In this chapter, we will formulate an optimization problem for conducting big data classification analysis. In the following sections, we will consider an algorithm for solving the optimization problem, i.e. an algorithm for the optimization of imbalanced datasets, and we will compare it with the results achieved by the two authors, using the same datasets and coresponding classification algorithms for RF, SVM, KNN, DT, NB and LR. The results of the research and the conducted analysis were published in Data Centric Optimization Method to Imbalanced Datasets, as part of the International Conference on Mathematical and Statistical Physics, Computational Science, Education, and Communication (Vladislav Tanov, Ivan Ivanov, 2023) and the article Data-Centric Optimization Approach for Small, Imbalanced Datasets, published in the Journal of Information and Organizational Sciences (JIOS) (Vladislav Tanov, 2023).

3.1. A model and algorithm for the optimization of imbalanced binary datasets (Data Centric Optimization - DCO)

We formulate the following model. We know a data set X in which the observations from the different classes are known. We divide into two sets Xtrain and Xtets in the ratio Xtrain: Xtets = 80:20 or Xtrain: Xtets = 70:30. In this division, observations from each class fall into both sets. We will use the Xtrain set to build a classification model, while the evaluation indicators for the model will be checked on the Xtets. We select a model for classification analysis with values of the relevant input parameters.

To maximize the function Acc(Xtest)
under conditions
1.Construction of the Xtrain set appropriately.
2.Selection of the parameters of the classification model.

The function Acc(Xtest) measures the accuracy of the classification analysis, after Xtrain and Xtets are defined, the model is built on Xtrain, and the overall accuracy (accuracy) is calculated on Xtest by the formula

$$\text{Acc(Xtest)} = \frac{\sum_{i=1}^{k} cm_{ii}}{\sum_{i,j=1}^{k} cm_{ij}} \ ,$$

where $CM = (cm_{ij})$ is a confusion matrix containing the c-class of the observations.

The solution of each optimization problem is sought empirically (at least for now), depending on the techniques for dividing a given set into two sets and the options for choosing the parameters of a selected classification model.

Here we will propose a DCO algorithm for solving the defined optimization problem (Vladislav Tanov, 2023).

The main task of DCO is to examine and divide the unbalanced binary set, or set with binary class imbalance, into a balanced subset using undersampling or the so-called sample shuffle. This idea follows the information gain example discussed by Shaltout et al. They use information gain as a methodology for selecting statistically significant variables based on entropic magnitude as a measure of disorder (Shaltout et al., 2020).

For this purpose, the unbalanced set optimization algorithm preserves the integrity of the minority class and selects an optimized subset of the majority class, so that the applied classification algorithm reaches the highest accuracy values. The optimization follows the logic of minimizing the errors that the calcification model admits during the validation of the predicted values, called model error rate (mer). In other words, DCO filters out the so-called "bad" or "noisy" rows from the subset of the majority class.

Following the classical approach, the balanced subset is divided into training and test with a ratio of 80 to 20, respectively. The process continues until mer reaches 0 or the number of positive random variables (between 0 and 100) is used up in selecting subsets from the majority class. DCO experiments were done with the following computer specifications: (RAM: 16GM, CPU: 2.6GHz 6-Core Intel Core i7) as follows:
We define the following parameters:

- $i \in \{0....100\}$
- r_i – random under-sampling integer
- n – length of the given dataset
- m – minority class length
- R – list of integers
- D_n – given dataset
- X_n – random variable (# of variables in D_n)
- Y_n – response variable (# of classes in D_n), Y = 1,…K, where K>=2
- $D_{i,m}$ – balanced, under-sampled data sub-set:
  $D_{i,m} = ([X_1,Y_1],....[X_n,Yn] \mid r_i , m)$, where [X,Y] is independent of D_n
- *mer* - model error rate, mer = 100
- T_o and V_o – optimized train and validation sub-sets

ALGORITHM: *DCO OPTIMIZATION PHASE*

```
Initialization of variables listed above.
set optimized = False
while not optimized
    draw random integer – $r_i$
    if random integer ($r_i$) not in list of integers (R)
        append random integer ($r_i$) to R
        $D_{i,m}$ = undersample($D_n \mid r_i , m$)
        split $D_{i,m}$ into train and validation sets (80/20):
        $T_i$, $V_i$ = train_test_split ($D_{i,m} \mid .20$)
        $C_i$ = build classifier
        fit train set to $C_i$ and calculate false positive error (FPE) and
        false negative error (FNE). Keep track of $C_i$ error:
        $error_{Ci} = C_i (T_i, V_i) = \Sigma [(FPE_i \mid C_i, D_{i,m}),(FNE_i \mid C_i, D_{i,m})]$
        evaluate current model error rate (mer)
        if mer is greater than $error_{Ci}$
            mer = $error_{Ci}$
            $T_o = T_i$
            $V_o = V_i$
    if lenght of R is greater than 100: optimized = True
end
```

3.1.1. Bootstrap procedure in classifying models

Diabetes is a very common disease that requires constant monitoring and control to avoid fatal consequences. Azberbg and co-authors conducted experiments to predict patients with diabetes using the algorithmic method of "random forest" (Random Forest, RF). RF is widespread and is considered one of the standard methods in the use of supervised models, supervised machine learning. Their experiment aggregates a collection of models constructed and trained based on the bootstrap procedure to divide the data into training and test subsets, using replacement, or the so-called bootstrap samples with replacemetn. Each iteration is based on a random selection of variables that build a rotation matrix, with combinations of variables. This leads to the generation of different variations of the calcification patterns defined in equation 3.4 (Azbeg et al., 2022).

In tables 3.1.1-1 and 3.1.1-2 we will present and compare the results obtained by Azberbg and co-authors algorithmic (ACA) method with the results obtained from the experiments with the application of the algorithm for optimization of imbalanced sets (Data Centric Optimization - DCO).

Table 3.1.1-1

Datasets	Classes and Number of observations	ACA Accuracy of RF(%)	DCO Accuracy of RF(%)
Diabetes 1	Class 0 – 500 Class 1 – 268	78.65	**87.04**
Diabetes 2	Class 0 – 1316 Class 1 – 684	99.5	**99.65**
Diabetes 3 (1 and 2)	Class 0 – 1816 Class 1 – 952	99.8	**100**

The experiment done by Azberbg and co-authors used an algorithmic (ACA) method by comparing the results obtained when testing with the Diabetes1 set with the following classification algorithms for machine learning, such as the model with support vectors (Support Vector Machines, SVM), the model of the nearest neighbors (K-Nearest Neighbor, KNN), the Decision Tree-badged (DT), the Adaptive Boosting (ADABoost) model, the Artificial Nearal Network (ANA) model and the Logistic regression (LR). Table 3.1.0 we will present and compare the results obtained by SVM, KNN, DT, ADABoost, ANN, LR, DL and ACA (RF) with our proposed algorithm for optimization of unbalanced sets (Data Centric Optimization - DCO) applied on the set Diabetes1 .

Azberbg and co-authors do additional experiments with the Diabetes2 sets, and a combination of Diabetes1 and Diabetes2 that they call Diabetes3 (Diabetes1 + Diabetes2). In this way, the authors create sets with a higher number of observations, which is standard for capturing the accuracy of the models under consideration (Bailly et al., 2022).

Table 3.1.1-2

Authors	Publications	Applied Algorithm	Accuracy (%)
Wei et al.	2010 г.	SVM	73
Panwar et al.	2016 г.	KNN	78
Ramezankhani et al.	2016 г.	DT	74
Mingqi, Xiaoyang and Dongdong	2020 г.	ADABoost	79.2
Pradhan et al.	2020 г.	ANN	80.4
Tigga и Garg	2021 г.	LR	75.32
Ihnaini et al.	2021 г.	DL	72.7
Azbeg et al. (ACA)	2022 г.	RF	85.9
DCO	2023г.	RF	87.04

As seen in the above table, DCO gives better results than the applied methods in the experiments of Azberbg and co-authors (ACA). As would be expected, the accuracy of the models increases with the increase of observations in the sets, which is the main idea in the experiment of Azberbg and co-authors. Here the question arises, upon a more in-depth evaluation of the results, how good these two approaches are.

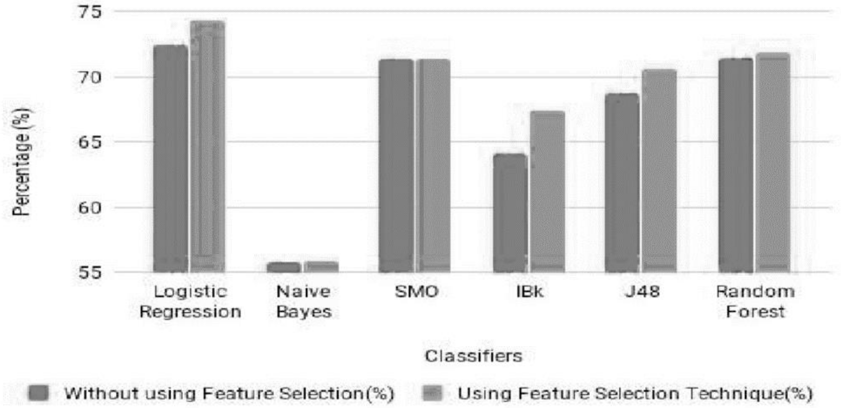
When we look at and analyze Figure 3.1.1 with the confusion matrix of the models, we notice that the high accuracy of the method of Azberbg and co-authors (ACA), in predicting Diabetes 1 multiple, is due to the greater percentage of correctly predicted values when the patients did not have diabetes, or the specificity, making errors only 15% (19 of 125) of the time. But when a patient has diabetes, or the sensitivity of the method to predict cases with diabetes, their method makes mistakes ~33% of the time (22 of 67). This means that their method will assign diabetes treatment to every 33 patients out of 100 who visit the clinic, i.e. this method actually fails to predict well enough.

This is due to the unbalanced set, which provides prerequisites for constructing biased models, and in this case the method of Azberbg and co-authors is biased towards diabetic patients. At the same time, DCO gives better results, with the specificity of the method being 87%, i.e. makes mistakes 13% (7 out of 54) of the time. At the same time, when patients have diabetes, DCO makes mistakes only 9% (5 out of 54) of the time, which is almost four (3.66) times less misdiagnoses compared to Azberbg et al.'s method.

3.1.2. Selection of statistical variables (feature selection)

Singh and co-authors experimented with the Indian Liver Patient dataset (ILPD), which is publicly available at the UCI Machine Learning Repository: ILPD (Indian Liver Patient Dataset). The ILPD set contains 416 observations with patients with liver disease (Class 1) and 167 (Class 2) observations with patients without liver disease (UCI Machine Learning Repository). On the ILPD set, two statistical variable selection methodologies are applied along with 10-fold cross-validation, namely correlation-based feature selection, which selects variables by addition or deletion, while not reach a drop in grade. Singh and co-authors used the Waikato Environment for Knowledge Analysis, WEKA (Singh, Bagga, Kaur, 2020), which is considered one of the most practical publicly available data analysis software (Written, Frank, Hall, 2022).

Figure 3.1.2



As can be seen in Table 3.1.2-1 with and without the application of statistical variable selection, logistic regression (LR) gives the highest results with and without the application of statistical variable selection, as the accuracy, accuracy, of LR is 74.36% and 72.5% respectively. The Random Forest (RF) algorithm achieved lower accuracy values of 71.87% and 71.53%, respectively (Singh, Bagga, Kaur, 2020).

Table 3.1.2-1

Set	Classes and Oservations	Singh Accuracy LR (%)		Singh Accuracy RF(%)		DCO Accuracy RF(%)
		No FS	FS	No FS	FS	
Indian Liver Patient dataset (ILPD)	Class 1 – 416 Class 2 – 167	72.5	74.4	71.53	71.9	**92.54**

This shows that the applied statistical variable selection methodology achieves poorer results compared to our proposed algorithm for the optimization of unbalanced sets (Data Centric Optimization - DCO). DCO in combination with RF reaches a much higher accuracy of 92.54%, as seen in Table 3.1.2-1.

Table 3.1.2-2

Set: Indian Liver Patient dataset (ILPD)	DCO Evaluation AUC = 92.5			
	Accuracy	Precision	Recall	F1
Клас 1 (34)	92.54	91.43	94.12	92.75
Клас 2 (33)	92.54	93.75	90.91	92.31

In Table 3.1.2-2, we provide additional test measures for overall model evaluation, namely Precision, Recall, F1 Score, and Area Under Curve (AUC).

3.1.3. A probabilistic realization with a given controlled discretization

Bohaick and Zabovsky are experimenting with a probabilistic approach to decision support for heart disease diagnosis (Bohaick, Zabovsky, 2019). The Centers for Disease Control and Prevention (CDC) associates the term "heart disease" with several types of heart disease (the most common being coronary artery disease), which is the leading cause of death in United States (CDC, 2022). To do this, Bohaick and Zabovsky used the Statlog Heart set, which is publicly available in the University of California, Irvine database. The Statlog Heart set consists of 120 observations (Class 2) diagnosed with heart disease and 150 observations (Class 1) without diagnosed heart disease (UCI Machine Learning Repository: Statlog Heart).

Experiment results with Bohaick and Zabovsky's proposed algorithm are compared with several machine learning algorithms such as NB, artificial neural networks (NN), artificial neural networks with multilayer perceptron (MLP), and the Decision Tree-badged (DT) model). They use the sum of the sensitivity and specificity of the test patterns as a measure of comparison (Bohaick, Zabovsky, 2019). In Table 3.1.3-1, we compare their published results with the results obtained from our proposed Data Centric Optimization (DCO) algorithm.

As seen in Table 3.1.3-1, Bohaick and Zabovsky's methodology achieves the highest sensitivity and specificity scores of 0.90 and 0.842 (total 1.742), respectively, compared to the other machine learning algorithms.

Table 3.1.3-1

The Set: Statlog Heart	Sensitivity	Specificity	Sum
Algorithms			
NB-Mod	0.900	0.842	1.742
NB	0.840	0.817	1.657
MLP	0.880	0.800	1.680
DT	0.840	0.692	1.532
NN	0.773	0.717	1.490
DCO	**0.96**	**1.00**	**1.96**

Our proposed imbalanced set optimization (DCO) algorithm significantly outperforms theirs, reaching test sensitivity and specificity of 0.96 and 1.0 (total 1.96). In addition to test measures for overall model evaluation, namely Precision, Recall, F1 Score, F1 as well as Area Under Curve (AUC), in Table 3.1.3- 2 we report the following results obtained by DCO.

Table 3.1.3-2

The set: Statlog Heart	DCO Evaluation AUC = 97.9			
	Accuracy	Precision	Recall	F1
Class 1 (24)	97.9	1.00	95.83	97.87
Class 2 (24)	97.9	96.0	1.00	97.96

3.1.4. A particle swarm optimization method for selecting statistically significant variables

Dubey, Sinhal, and Sharma developed experiments using a particle swarm optimization method for selection to arrive at an optimal set of categorical variables, also known as Improved Auto Categorical Particle Swarm Optimization (IACPSO). Particle Swarm Optimization (PSO) automates the approach to choosing the optimal values of the controlled parameters at each iteration. Dubey, Sinhal, and Sharma analyze the behavior and impact of optimal numerical parameters in various machine learning algorithms (Dubey, Sinhal, and Sharma, 2022).

Dubey, Sinhal and Sharma take an unusual approach to split the set into training and test subsets using a ratio of 75/25. To ensure a proper comparison between their method, IACPSO, and our presented Data Centric Optimization (DCO) algorithm, we conducted experiments using both the standard 80/20 training and test subset split ratio and and 75/25 applied to the same Statlog, Cleveland, and Hungarian sets. The following Table 3.1.4 presents the results and comparisons between the two methodologies.

Table 3.1.4

	Cleveland					**Statlog**					**Hungarian**				
MLA	AC	PR	SV	FS	MCC	AC	PR	SV	FS	MCC	AC	PR	SV	FS	MCC
LR	87	87	87	87	74	91	92	89	90	81	88	88	88	88	77
LR + IACSPO	**96**	**97**	97	**97**	89	**98**	**99**	97	**99**	95	97	**98**	98	**98**	92
LR - DCO (75/25)	94	92	97	94	89	**98**	97	**100**	98	**97**	94	93	96	95	89
LR - DCO (80/20)	**96**	93	**100**	**97**	**93**	**98**	96	**100**	98	96	**98**	95	**100**	**98**	**95**
DT	82	83	83	82	65	83	85	81	82	65	82	83	82	82	65
DT + IACSPO	92	**93**	94	93	83	93	**96**	94	**96**	83	93	93	94	93	92
DT - DCO (75/25)	93	91	94	93	86	95	94	97	95	90	96	**96**	96	96	92
DT - DCO (80/20)	**96**	**93**	**100**	**97**	**93**	**96**	92	**100**	**96**	**92**	**98**	**96**	**100**	**98**	**95**
RF	87	87	87	87	71	89	90	87	88	61	86	87	86	87	69
RF + IACSPO	97	97	**98**	97	90	97	**98**	98	**98**	89	97	98	**98**	**98**	88
RF - DCO (75/25)	96	94	97	96	91	97	97	97	97	93	92	96	89	92	85
RF - DCO (80/20)	**99**	**100**	97	**99**	**97**	**98**	97	**100**	**98**	**97**	**98**	**100**	96	**98**	**96**
SVM	87	87	87	87	74	87	87	86	87	73	87	87	87	87	74

SVM + IACSPO	97	97	98	97	90	97	98	98	**98**	89	97	98	98	98	88
SVMGS	89	89	89	89	77	89	93	86	88	78	89	89	89	89	77
SVMGS + IACSPO	98	97	96	97	90	**98**	**99**	98	97	90	**97**	**97**	97	**97**	90
SVM - DCO (75/25)	94	94	94	94	88	97	94	**100**	97	94	94	93	96	95	89
SVM - DCO (80/20)	**100**	**100**	**100**	**100**	**100**	**98**	96	**100**	**98**	**96**	95	91	**100**	95	**91**
KNN	76	76	75	76	51	76	73	75	75	50	77	76	76	76	53
KNN + IACSPO	92	92	91	92	84	92	92	93	90	81	91	91	91	89	80
KNN - DCO (75/25)	94	94	94	94	88	**97**	94	**100**	**97**	**94**	94	93	96	94	89
KNN - DCO (80/20)	**96**	**96**	**96**	**96**	**93**	96	**96**	96	96	92	**98**	**96**	**100**	**98**	**95**
NB	87	87	87	87	74	91	92	89	90	81	88	88	88	88	76
NB + IACSPO	92	93	92	91	83	94	95	94	92	85	92	92	92	91	82
NB- DCO (75/25)	93	91	94	93	86	97	94	100	97	94	**100**	**100**	**100**	**100**	**100**
NB- DCO (80/20)	**100**	**100**	**100**	**100**	**100**	**98**	**96**	**100**	**98**	**96**	**100**	**100**	**100**	**100**	**100**

Analyzing Table 3.1.4, we can see a total of 90 measures for the three datasets, Statlog, Cleveland and Hungarian, which include 5 metrics, Accuracy (AC), Precision (PR), Sensitivity (SV), f1 score (F1 Score, FS) and Matthew's correlation coefficient (MCC). This equates to 15 measures for each implemented machine learning algorithm, 6 MLA, (3 sets multiplied by 30 - 5 metrics for each algorithm). Taking a closer look at the listed measures, we can see that our proposed Data Centric Optimization (DCO) algorithm outperforms the IACPSO method 75.56% (68/90) percent of the time with an average of 5.62 points for each measure. With minor exceptions for LR, DT and SVM, where 13.33% (12/90) of the time IACPSO gave better results, and 11.11% (10/90) of the time when DCO and IACPSO gave the same results.

3.2. An algorithm for optimization of multi classes imbalanced datasets (Data Centric Multiclass Optimization - DCMO).

The classification of unbalanced data, class imbalance, manifests itself in two aspects. The first case is when we have two classes with a negative and a positive class label, or a binary class imbalance. In the second case, we have more than one class, or multiclass imbalance. For example, given a given data set of the format (Xi, yi), where Xi is the ith observation, then yi is the ith class label, as follows yi ∊ {1…K} (Aly, 2005).

Koziarski, Wozniak, and Krawczyk propose a new oversampling technique called the combined cleaning and resampling (MC-CCR) algorithm. The proposed method uses an approach to model the regions suitable for oversampling, which are less affected by disjunct definitions, small disjuncts, and deviations when applying the synthetic minority oversampling technique, SMOTE. The goal is to reduce the effect of overlapping class distributions on the performance of machine learning algorithms (Koziarski, Wozniak, Krawczyk, 2020).

Their experiments are based on 19 multiclass unbalanced sets publicly available in the Knowledge Extraction based on Evolutionary Learning database (KEEL, 2011). Koziarski, Wozniak, and Krawczyk apply the Decision Tree (DT) model, the Naïve-Bayes (NB) model, and the K-Nearest Neighbor (KNN) model. Their algorithm for combined cleaning and resampling, MC-CCR, is compared with state-of-the-art multiclass set oversampling methods such as the synthetic minority oversampling technique (SMOTE-all, S-SMOTE), SMOTE combined with an iterative-partition filter (SMOTE-IPF), the Mahalanobis Distance Oversampling (MDO) technique, as well as the Synthetic Minority Oversampling technique, SMOM (Koziarski, Wozniak, Krawczyk, 2020).

So far in Section 3.1, we have considered the proposed algorithm for the optimization of unbalanced sets with binary class imbalance. In the following sections, we will look at the multi-class variant of the unbalanced set optimization algorithm, or the multi-class unbalanced set optimization algorithm (Data Centric Multiclass Optimization - DCMO). The results obtained from the experiments with DCMO will be compared and compared with the results presented by Koziarski, Wozniak and Krawczyk, obtained from their combined cleaning and resampling (MC-CCR) algorithm. To ensure a correct comparison, we use the same sets and their applied classification algorithms – DT, KNN, and NB.

We declare the following parameters:

- $i \in \{0\ldots.100\}$
- r_i – random under-sampling integer
- $k_i \in \{1\ldots.99\}$ – odd number generator
- $or_{i,k}$ – odd random dataset-split integer
- n – length of the given dataset
- m – minority class length
- R – list of integers
- OR – list of odd integers
- D_n – given dataset
- N_c – number of calsses in the dataset
- X_n – random variable (# of variables in D_n)
- Y_n – response variable (# of classes in D_n), Y = 1,…K, where K>=2
- $D_{i,m}$ – balanced, under-sampled data sub-set:

  $D_{i,m} = ([X_1,Y_1],\ldots.[X_n,Y_n] \mid r_i , m)$, where [X,Y] is independent of D_n
- $T_{i,k}$, $V_{i,k}$ – train and validation datasets
- score = 0 – optimal multicalss clasifier score metric
- min_optimal_score:
  - if desired minimal multicalss clasifier score:
    - min_optimal_score $\in \{0\ldots.100\}$
  - otherwise, None
- T_o and V_o – optimized train and validation sub-sets

ALGORITHM: *DCMO OPTIMIZATION PHASE*

```
Initialization of variables listed above.
set optimized = False
set odd_int_end = False
while not optimized
    draw random integer – $r_i$
    if $r_i$ not in list of integers (R)
        append random integer ($r_i$) to R
        $D_{i,m}$ = undersample($D_n$ | $r_i$ , m)
        while not odd_int_end:
            draw odd random integer – $or_{i,k}$
            if $or_{i,k}$ not in list of off intergers (OR)
                append odd random integer($or_{i,k}$) to OR
                split $D_{i,m}$ into train and validation sets (80/20):
                $T_{i,k}$, $V_{i,k}$ = train_test_split ($D_{i,m}$ | $or_{i,k}$, .20)
                $C_{i,k}$ = build classifier
                fit train set to $C_{i,k}$ and calculate $F_1$ Score. Keep track:
                avg $score_{i,k}$ = $C_{i,k}$ ($T_{i,k}$, $V_{i,k}$) = $\Sigma$ [($F1_{i,k}$ | $C_{ik}$, $D_{i,m,k}$)]/Nc

                evaluate current model avg $score_{i,k}$
                if avg $score_{i,k}$ is greater than score
                    score = avg $score_{i,k}$
                    $T_o$ = $T_{i,k}$
                    $V_o$ = $V_{i,k}$
            if lenght of OR is greater or equal than 50
            OR score >= optimal_score:
                odd_int_end = True
    if lenght of R is greater than 100
    OR score >= optimal_score:
        optimized = True
end
```

In Table 3.6.1, we compare the results published by Koziarski, Wozniak, and Krawczyk with the results obtained by our presented Data Centric Multiclass Optimization (DCMO) algorithm.

Table 3.6.1

Sets	Results according to Average Accuracy (AvgAcc) [%] metric for MC-CCR and reference sampling methods with C5.0 as base classifier.									
	MC-CCR	SMOTE-all	S-SMOTE	MDO	SMOM	SMOTE-IPF	DCMO			
							AvgAcc	DT	KNN	NB
Automobile	76.98	80.12	73.53	78.13	79.04	75.32	**90.66**	96	88	88
Balance	82.87	55.06	55.01	57.70	59.52	54.26	**91.33**	90	87	97
Car	**97.12**	89.84	90.13	93.36	95.18	90.96	95.33	98	88	100
Cleveland	37.88	28.92	27.18	28.92	28.01	24.98	**87.33**	100	85	77
Contraceptive	53.18	50.63	46.92	53.27	55.09	52.88	**64.66**	70	65	59
Dermatology	94.29	95.72	96.1	97.48	99.31	92.18	**100**	100	100	100
Ecoli	**74.07**	64.68	67.54	61.16	61.16	60.43	N/A– not enough observations			
Flare	68.92	71.86	71.52	68.72	70.64	68.55	**79.66**	87	79	73
Hayes-Roth	92.11	86.45	88.04	87.33	90.06	89.74	**98.33**	100	95	100

Led7digit	70.48	72.39	72.55	75.03	75.94	71.35	**88.67**	91	91	84
Lymphography	**79.60**	73.02	62.67	76.54	74.72	74.20	N/A– not enough observations			
Newthyroid	96.18	94.7	93.48	92.06	90.24	93.05	**100**	100	100	100
Pageblocks	83.71	75.83	75.25	78.47	77.56	74.20	**95**	96	93	93
Thyroid	80.52	80.02	85.34	79.14	80.96	78.91	**90**	100	80	90
Vehicle	72.71	73.49	73.71	70.85	70.85	71.02	**78.67**	88	85	63
Wine	95.28	92.53	90.80	93.41	93.41	90.16	**97.00**	97	97	97
Winequality-red	46.93	37.41	35.79	40.05	42.78	36.28	**75.00**	75	75	92
Yeast	**58.39**	51.03	52.42	54.55	56.37	53.77	48.67	60	43	43
Zoo	**85.92**	82.61	68.69	79.09	79.09	67.30	N/A– not enough observations			

In the above Table 3.6.1, the results published by Koziarski, Wozniak, and Krawczyk are based on the arithmetic average accuracy value (AvgAcc) of the MC-CCR method, but it is not clear what exactly it is based on. MC-CCR results are compared with their considered oversampling methods with a basic classification algorithm DT (C5.0), with MC-CCR showing better accuracy values than SMOTE, SMOTE-all, S-SMOTE, SMOTE-IPF, MDO and SMOM.

To provide a proper comparison between their method, MC-CCR, and our presented Data Centric Multiclass Optimization (DCMO) algorithm, we present the results achieved by the listed machine learning algorithms DT, KNN, and NB as well as the Arithmetic Average Accuracy (AvgAcc). Here, it is important to note the essential difference between MC-CCR and DCMO, which consists in the treatment of the considered sets. MC-CCR applies the oversampling technique, while DCMO uses the undersampling technique. This limits the application of DCMO to three of the considered sets, Ecoli, Lymphography, and Zoo, because one or more of the classes have too low several rows (observations). In Table 4.6.1, these sets are designated as not applicable, or N/A– not enough observations, as follows:

- Ecoli - class imS,imL and oL have 2, 2 and 5 observations, respectively;
- Lymphography – class normal and fibrosis have 2 and 4 observations, respectively;
- Zoo – classes 5,3 and 6 have 4,5 and 8 sightings, respectively.

Taking a closer look at the listed measures for the remaining sets, we can notice that our proposed DCMO method outperforms the MC-CCR method 73.7% (14/19) percent of the time based on AvgAcc. On the other hand, MC-CCR outperformed DCMO 26.3% (5/19) percent of the time. As we have already explained above, in three of the cases DCMO cannot be applied due to a very low number of rows (observations), i.e. here it cannot be stated with certainty which of the two methods would have an advantage. In fact, there remain two cases where MC-CCR outperforms DCMO 10.5% (2/19) percent of the time where AvgAcc is lower. This is not at all the case if one considers the individual results of the prolog machine learning algorithms. For example, Table 4.6.1 shows better results achieved by DCMO for Car and Yeast sets when applying DT as follows:

- Car:
    - DCMO – **98**
    - MC-CCR – 97.12

- Yeast:
  - DCMO – **60**
  - MC-CCR – 58.29.

3.2.1 Test measures for comprehensive evaluation of the multi-class unbalanced set optimization algorithm (Data Centric Multiclass Optimization - DCMO)

In addition to test measures for overall model evaluation, namely Precision, Recall, F1 Score, Matthew's Correlation Coefficient (MCC), as well as the model error matrix, confusion matrix, in the following sections we report the test results obtained by DCMO when applying DT, KNN and NB algorithms for classification analysis.

Looking at Table 3.2.1, there are two cases where DCMO gave low results when applying the Yeast and Contraceptive sets. The Yeast set contains 9 classes, of which the minority class has only 20 observations. This means that applying the standard training and test subset split ratio of 80/20, DCMO will optimize and balance the Yeast set to 4 observations for each of the 9 classes in the test subset (Appendix – 3.2.1.16). DCMO achieves an unsatisfactory result of only 60 percent accuracy (MC-CCR accuracy - 58.39). Here the question arises, are only 16 observations for each class sufficient to reach optimal accuracy in machine learning?

Table 3.2.1 Results obtained by DCMO for the considered sets.

Sets	A minority class	Algorithm	DCMO (macro average)				
			Accuracy	Precision	Recall	F1	MCC
Automobile	Class 4	DT	96	97	96	96	95
		KNN	88	91	88	88	86
		NB	88	91	88	88	86
Balance	Class 1	DT	90	91	90	90	85
		KNN	87	87	87	87	89
		NB	97	97	97	97	95
Car	Class 3	DT	98	98	98	98	97
		KNN	88	89	88	88	85
		NB	100	100	100	100	100
Cleveland	Class 4	DT	100	100	100	100	100
		KNN	85	88	83	85	82
		NB	77	72	73	77	72
Contraceptive	Class 1	DT	70	69	70	70	55
		KNN	65	64	64	64	47
		NB	59	62	59	59	41
Dermatology	Class 5	DT	100	100	100	100	100
		KNN	100	100	100	100	100
		NB	100	100	100	100	100
Flare	Class 4	DT	87	86	86	85	84
		KNN	79	81	79	79	75

		NB	73	78	73	71	69
Hayes-Roth	Class 2	DT	100	100	100	100	100
		KNN	95	96	94	95	92
		NB	100	100	100	100	100
Led7digit	Class 1	DT	91	92	91	91	90
		KNN	91	92	91	91	90
		NB	84	85	84	83	82
Newthyroid	Class 2	DT	100	100	100	100	100
		KNN	100	100	100	100	100
		NB	100	100	100	100	100
Pageblocks	Class 2	DT	96	97	97	96	96
		KNN	93	94	93	93	91
		NB	93	95	93	93	92
Thyroid	Class 0	DT	100	100	100	100	100
		KNN	80	81	80	80	70
		NB	90	91	90	90	86
Vehicle	Class 0	DT	88	89	88	88	84
		KNN	85	85	85	85	80
		NB	63	68	63	62	53
Wine	Class 2	DT	97	97	97	97	95
		KNN	97	97	97	96	95
		NB	97	97	96	96	95
Winequality-red	Class 0	DT	75	64	75	75	72
		KNN	75	75	75	72	72
		NB	92	94	92	91	91
Yeast	Class 0	DT	60	62	60	59	55
		KNN	43	32	44	36	37
		NB	43	52	43	41	38

Rather, the answer lies in the quality of the data sets considered. For example, as seen in the above Table 3.2.1, the Contraceptive set has 3 classes, of which the minority class has 333 observations, therefore, the Contraceptive test subset consists of 67 (20% of 333) observations for each class (Appendix - 3.2 .1.5). DCMO achieves the disappointing result of 70 percent accuracy, which is only 10 percent better than the Yeast set experiment, despite the large difference in training observations of ~17 times.

On the other hand, the Dermatology set, like Yeast, contains 6 classes, of which the minority class has 20 observations, therefore, the Dermatology test subset consists of 4 observations for each class (Appendix - 3.2.1.6). DCMO achieves 100 percent accuracy for all three machine learning algorithms considered - the decision tree (DT) model, the Naïve-Bayes (NB) model, and the nearest-neighbor (K- Nearest Neighbor, KNN).

Similar results were seen in the Newthyroid set. Newthyroid contains 3 classes, of which the minority class has 30 observations, therefore, the Newthyroid test subset consists of 6 (20% of 30) observations for each class (Appendix - 3.2.1.10). DCMO achieves 100 percent accuracy for all three machine learning algorithms considered—

DT, NB, and KNN. These experiments show that the large size of data sets does not always lead to satisfactory accuracy in machine learning. Data-centric optimization leads to improved performance of class imbalance problems consisting of a majority class (larger number of observations) and a minority class.

Major scientific contributions

In the present dissertation, problems are investigated in two main directions:

• Derivation and study of iterative methods for searching for Nash equilibrium in dynamic games. The studies are described in Chapters One and Two.

• Creation of models and algorithms for conducting classification analysis of specific sets. The research is described in the third chapter.

Publications on Vladislav Tanov's PhD Thesis:

1. Ivan Ivanov, Nikolay Netov, **Vladislav Tanov**, Iteratively Computation the Nash Equilibrium Points in the Two-Player Positive Games. *International Journal of Mathematical and Computational Methods*, **1**, 378-381, 2016.
2. Ivelin G. Ivanov, **Vladislav Tanov**, An Iterative Method for an Equilibrium Point of Linear Quadratic Stochastic Differential Games with State and Control-Dependent Noise, *Mathematics, and its Applications / Annals of AOSR*, 10(2), 202-210, 2018. (Scopus)
3. Ivelin G. Ivanov, **Vladislav Tanov**, Computing the Nash Equilibrium for LQ Games on Positive Systems Iteratively, *Mathematics and its Applications / Annals of AOSR*, 10(2), 230-244, 2018. (Scopus)
4. Ivelin G. Ivanov, **Vladislav Tanov**, A Nonsymmetric Nash-Riccati Equation and Decoupled Schemes for a Stabilizing Solution, *Applied Mathematics E - Notes*, 2020, 20, pp. 357–366. (Scopus)
5. **Vladislav Tanov**, Ivan Ivanov, Data Centric Optimization Method to Imbalanced Datasets, *Proceedings of SPIE - The International Society for Optical Engineering*, 12616, 1261602, 2023. (Scopus), International Conference on Mathematical and Statistical Physics, Computational Science, Education, and Communication (ICMSCE 2022), 2022, Istanbul, Turkey. (Scopus) https://doi.org/10.1117/12.2674455 (Scopus)
6. **Vladislav Tanov**, Data-Centric Optimization Approach for Small, Imbalanced Datasets, *Journal of Information and Organizational Sciences (JIOS)*, Vol 47, No 1, 2023. https://jios.foi.hr/index.php/jios/article/view/1875 (Scopus)

Acknowledgement

I thank my supervisor, Assoc. Prof. Dr. Nikolay Netov, for the shared knowledge on the topics of the issertation work and the invaluable help while training and writing the dissertation work.

Refference

1 Ivan Ivanov, Generelized Riccati equation in stochastic economic modeling, Sofia University St. Kl. Ohridski Press, Sofia 2012. (In Bulgarian)

2 Ivelin Ivanov, Information thecnologies for processes (models) management. PhD Thesis, Shoumen University, 2016. (In Bulgarian).

3 M**.** Aly**,** Survey on multiclass classification methods. *Technical Report, Caltech*, 2005

4 K. Azbeg, M. Boudhane, O. Ouchetto, Diabetes emergency cases identification based on a statistical predictive model, *Journal of Big Data*, 2022

5 T.Azevedo-Perdicoulis, G.Jank, Linear Quadratic Nash Games on Positive Linear Systems, European Journal of Control, 11, 1-13, 2005.

6 B. Basar, G.J. Olsder. Dynamic Noncooperative Game Theory. SIAM, Philadelphia, 1999.

7 Z.-Z. Bai, X.-X. Guo, S.-F. Xu, alternately linearized implicit iteration methods for the minimal nonnegative solutions of the nonsymmetric algebraic Riccati equations, Numer. Linear Algebra Appl., 13, 655-674, 2006.

8 N. Baeva, The Nash Equilibrium in Open Loop Linear Quadratic Games for Positive Systems, Mathematics and its Applications / Annals of AOSR, 9(1), 17-27, 2017.

9 T. Bayes. An essay towards solving a Problem in the Doctrine of Chances. Bayes's essay as published in the Philosophical Transactions of the Royal Society of London, Vol. 53, p. 370. 1763.

10 J. Bohacik and M. Zabovsky, Discretization for Naive Bayes Taking the Specifics of Heart Data into Account. Journal of International and Organizational Science, vol. 43, no. 1, 2019

11 D. Berrar. Bayes' Theorem and Naive Bayes Classifier. *Research Gate*. DOI: 10.1016/B978-0-12-809633-8.20473-1. 2018.

12 P. Branco, L. Torgo, R.P. Ribeiro. Relevance-based evaluation metrics for multi-class imbalanced domains. *Advances in Knowledge Discovery and Data Mining - 21st Pacific-Asia Conference*. Proceedings, Part I, 2017, pp. 698–710. 2017.

13 L. Breiman, J.H. Friedman, C.J. Stone, R.A. Olshen. Classification and Regression Trees. *Wadsworth & Brooks*. 1984.

14 Breiman, L. Bagging predictors. Mach Learn 24. 1996.

15 Breiman L. Random Forests. Machine Learning. 2001.

16 L. M. Bruce, Game theory applied to big data analytics in geosciences and remote sensing, 2013 IEEE International Geoscience and Remote Sensing Symposium, https://ieeexplore.ieee.org/document/6723733 (2018)

17 T. Carter. An introduction to information theory and entropy. *Complex Systems*. 2011

18 R.T. Cox. Probability, Frequency, and Reasonable Expectation. American Journal of Physics. 1946.

19 L.S. Cheн, S.J. Cai, Neural-network-based resampling method for detecting diabetes mellitus, SpringerLink, 2015

20 D. Chicco, G. Jurman, The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation., *BMC Genomics*, 2020

21 O.L.V. Costa , W.L. de Paulo, Indefinite quadratic with linear costs optimal control of Markov jump with multiplicative noise systems, Automatica, Vol. 43, 587-597, 2007. (Costa , de Paulo, 2007)

22 O.L.V. Costa, A. de Oliveira, Optimal mean-variance control for discrete-time linear systems with Markovian jumps and multiplicative noises, Automatica, 48, 2012, 304-315. (Costa, de Oliveira, 2012).

23 CDC, Heart Disease | Cdc.Gov, *Centers for Disease Control and Prevention*, www.cdc.gov, 2022.

24 T. Damm, D. Hinrichsen, , Newton's Method for a Rational Matrix Equation occurring in stochastic control, Linear Algebra Appl., 332-334:81-109., 2001 (Damm and Hinrichsen 2001)

25 E. Dockner, S. Jorgensen, N. V. Long, G. Sorger, Differential games in economics and management science, Cambridge University Press, (2000).

26 A. K. Dubey, A. K. Sinhal, R. Sharma, An Improved Auto Categorical PSO with ML for Heart Disease Prediction, *Engineering, Technology and Applied Science Research*, 2022

27 V. Dragan, S. Aberkane, I. Ivanov, An iterative procedure for computing the stabilizing solution of discrete-time periodic Riccati equations with an indefinite sign, 21st International Symposium on Mathematical Theory of Networks and Systems July 7-11, 2014. Groningen.

28 V. Dragan, S. Aberkane, I. Ivanov, On computing the stabilizing solution of a class of discrete-time periodic Riccati equations, International Journal of Robust and Nonlinear Control, vol.25, 7, 2015, 1066-1093, doi: 10.1002/rnc.3131.

29 EMC. Data Science and Big Data Analytics: Discovering, Analyzing, Visualizing and Presenting Data. *Wiley*. 2015. https://www.wiley.com/en-ie/Data+Science+and+Big+Data+Analytics%3A+Discovering%2C+Analyzing%2C+Visualizing+and+Presenting+Data-p-9781118876138

30 H. J. Escalante, M. Montes, L. E. Sucar, Practicle Swarm Model Selection, *Journal of Machine Learning*, 2009

31 J. Engwerda. LQ Dynamic Optimization and Differential Games, Wiley 2005.

32 J. Grus. Data Science from Scratch. *O'Reilly Media*. ISBN: 9781491901427. 2015.

33 U. M. Fayyad and K. B. Irani, Multi-Interval discretization of continuous-valued attributes for classification learning, *International Joint Conference on Uncertainty in AI*, 1993.

34 F. Farris. The Gini Index and Measures of Inequality. *American Mathematical Monthly*. 117. 2010.

35 Y.T. Feng, B.D.O. Anderson, An iterative algorithm to solve state-perturbed stochastic algebraic Riccati equations in LQ zero-sum games, Systems & Control Letters, 59(1)(2010), 50–56.

36 B. de Finetti. Theory of Probability: A critical introductory treatment. Chichester: John Wiley & Sons Ltd. 2017.

37 G. Freiling, A.Hochhaus, On a Class of Rational Matrix Differential equations Arising in Stochastic Control. Linear Algebra Its App., 379, 43-68, 2004.

38 FastStats. *FastStats - Chronic Liver Disease or Cirrhosis*, www.cdc.gov, 2022.

39 L. Imsland, I. G. Ivanov, and S. Kostova, Linear Quadratic Differential Games and Applications, Biomath Communications, 3, 2016, N 2.

40 I. G. Ivanov, L. Imsland and B. C. Bogdanova, Iterative algorithms for computing the feedback Nash equilibrium point for positive systems, International Journal of System Science.

41 I. G.Ivanov and N. Netov, The Nash Equilibrium Point in the LQ Game on Positive Systems with Two Players, Mathematical and Computational Methods, 1, 2016, 242–246.

42 Ivan Ivanov, Nikolay Netov, Vladislav **Tanov**, Iteratively Computation the Nash Equilibrium Points in the Two-Player Positive Games. *International Journal of Mathematical and Computational Methods,* **1**, 378-381, 2016

43 Ivelin Ivanov, Vladislav **Tanov**, Computing the Nash Equilibrium for LQ Games on Positive Systems Iteratively, Mathematics and its Applications / Annals of AOSR, 10(2), 230-244, 2018.

44 Ivelin Ivanov, Vladislav **Tanov**, A Nonsymmetric Nash-Riccati Equation and Decoupled Schemes for a Stabilizing Solution, Applied Mathematics E-Notes, 20(2020), 357-366, Applied Mathematics E-Notes, 20(2020), 357-366

45 Ivelin G. Ivanov, Vladislav **Tanov**, An Iterative Method for an Equilibrium Point of Linear Quadratic Stochastic Differential Games with State and Control-Dependent Noise, Mathematics and its Applications/Annals of AOSR, 10(2), 202-210, 2018.

46 G. Jank, D. Kremer, Open loop Nash games and positive systems solvability conditions for non symmetric Riccati equations.

47 E.T. Jaynes. Bayesian Methods: General Background. In Justice, J. H. (ed.). Maximum-Entropy and Bayesian Methods in Applied Statistics. Cambridge: Cambridge University Press. 1986.

48 M.A. Hall, Correlation-based feature selection for machine learning, 1999.

49 H. He, Y. Bai, E.A. Garcia, S. Li, ADASYN: Adaptive synthetic sampling approach for imbalanced learning. *International Joint Con- ference on Neural Networks (IEEE World Congress on Computational Intelligence)*. IEEE. 2008

50 S. Jorgensen, and G. Zaccour, Differential Games in Marketing. International Series in Quantitative Marketing, Kluwer Academic Publisher in 2004.

51 M. Koziarski, M. Wozniak. CCR: A combined cleaning and resampling algorithm for imbalanced data classification. *Appl. Math. Comput. Sci*. 2017.

52 M. Koziarski**,** M. Wozniak, B. Krawczyk, Combined Cleaning and Resampling algorithm for multi-class imbalanced data with label noise, *Knowledge-Based Systems*, 2020

53 B. Krawczyk, M. Koziarski, M. Wozniak. Radial-Based Oversampling for Multiclass Imbalanced Data Classification, *IEEE Trans. Neural Netw. Learn. Syst*. 2019.

54 A. Lanzon, Feng Y., Anderson B., Rotkowitz M., Computing the positive stabilizing solution to algebraic Riccati equations with an indefinite quadratic term via a recursive method, IEEE Transactions on automatic control, 53,10(2008), 2280-2291.

55 T.H. Lee, A. Ullah, R. Wang. Bootstrap Aggregating and Random Forest. UC Riverside Department of Economics. 2019.

56 D.J.N.Limebeer, B.D.O.Anderson, B.Hendel. A Nash game approach to mixed H_2/H_{\infty} control. IEEE Transactions on Automatic Control, 1994, 39(1), 69-82, 1994.

57 C.Ma, H.Lu. Numerical Study on Nonsymmetric Algebraic Riccati Equations, Mediterranean J. of Mathematics, 13, 6, 4961-4973, 2016.

58 A. McCallum. Graphical Models, Lecture2: Bayesian Network Representation. *Archived (PDF)*. 2022.

59 T.Mitchell. Machine Learning. *McGraw-Hill Science*. ISBN: 0070428077. 1997

60 L. Metzler, Stability of multiple markets: the Hicks condition, Econometrica, 13(4), 1945, pp. 277–292

61 Z. Mo, D. Xuan, R. Shi, 2023, Robust Data Sampling in Machine Learning: A Game-Theoretic Framework for Training and Validation Data Selection, *Games* 14, no. 1: 13. https://doi.org/10.3390/g14010013

62 N. Moniz, P. Branco, L. Torgo, Resampling strategies for imbalanced time series forecasting, SpringerLink, 2017.

63 D.M.B. Powers**,** Evaluation: From Precision, Recall and F-Score to ROC, Informedness, Markedness & Correlation, *Journal of Machine Learning Technologies*, 2011.

64 Q. Qiao, A.Y. Kaltungo, R.e. Edwards, Feature selection strategy for machine learning methods in building energy consumption prediction, *Science Direct*, vol. 8, 2022.

65 J. R. Quinlan. Induction of Decision Trees. Mach. Learn. 1. 81–106. 1986.

66 J. R. Quinlan. C4.5: Programs for Machine Learning. *Morgan Kaufmann Publishers*, 1993

67 J. R. Quinlan. Improved use of continuous attributes in 4.5. Journal of Artificial Intelligence Research. 4:77-90. 1996.

68 B. Raskutti, A. Kowalczyk. Extreme rebalancing for svms: a case study, SIGKDD Explorations, 2004

69 S. Russell, P. Norvi. Artificial Intelligence: A Modern Approach. *Prentice Hall*. p. 478. 2002.

70 W. Satriaji, R. Kusumaningrum, Effect of Synthetic Minority Oversampling Technique (SMOTE), Feature Representation, and Classification Algorithm on

Imbalanced Sentiment Analysis, 2nd International Conference on Informatics and Computational Sciences (ICICoS), Semarang, Indonesia, 2018.

71 A. Samuel, Some Studies in Machine Learning Using the Game of Checkers. IBM Journal of Research and Development *3* (3). 1959.

72 D. Schum. The Evidential Foundations of Probabilistic Reasoning. Northwestern University Press. 1994.

73 B. Sjardin, L. Massaron, A. Boschetti. Large Scale Machine Learning with Python. *Packt Publishing*. 2016.

74 N. Shaltout, M. Elhefnawi, A. Rafea and A. Moustafa, Information Gain as a Feature Selection Method for the Efficient Classification of Influenza Based on Viral Hosts, Engineering and Computer Science, 2014.

75 C.E. Shannon. A Mathematical Theory of Communication. *Bell System Technical Journal*. 27. 379–423. 1948.

76 J. Singh, S. Bagga and R. Kaur, Software-based Prediction of Liver Disease with Feature Selection and Classification Techniques, Procedia Com. Science, 2020

77 UCI Machine Learning Repository: ILPD (Indian Liver Patient Dataset) Data Set. *UCI Machine Learning Repository: ILPD (Indian Liver Patient Dataset) Data Set*, archive.ics.uci.edu

78 UCI Machine Learning Repository: Statlog (Heart) Data Set. UCI Machine Learning Repository: Statlog (Heart) Data Set.

79 H.Zhu, C.Zhang, Infinite time horizon nonzero-sum linear quadratic stochastic differential games with state and control-dependent noise, J. Control Theory Appl., 11: 629-633, 2013.

80 H.N. Zhu, C.K. Zhang, N. Bin, Stochastic Nash Games for Markov Jump Linear Systems with State- and Control-Dependent Noise, Journal of the Operations Research Society of China, 2: 481-498, 2014.

81 D. Yao, S. Zhang X. Zhou, Stochastic linear quadratic control via semidefinite programming, SIAM Journal Control Optimization, 49 (3):801-823, 2001.

82 J. VanderPlas. Python Data Science Handbook: Essential Tools for Working with Data 1st Edition, O'Reilly Media; 1st edition. ISBN: 1491912057, 2017.

83 J. Wainer, G. Cawley, Empirical evaluation of resampling procedures for optimising SVM hyperparameters, J. of Machine Learning Research, 18, 2017.

84 K. Wang, W. Sun, Q.Du, A cooperative game for automated learning of elasto-plasticity knowledge graphs and models with AI-guided experimentation. Com. Mech 64, 467–499(2019). https://doi.org/10.1007/s00466-019-01723-1

85 G.M. Weiss**,** H. Hirsh, A Quantitative Study of Small Disjuncts. *In Proceedings of the Seventeenth National Conference on Artificial*. 2000

86 H. Witten, E. Frank, M.A. Hall. Data Mining: Practical machine learning tools and techniques, 3rd Edition. *Morgan Kaufmann*. p. 191. 2011.

87 H. Written**,** E. Frank, A. Hall, Data Mining: Practical Machine Learning Tools and Techniques, *Morgan Kaufmann*, 4th ed. Burlington, MA, 2016.

88 G. Wu, E. Chang. Lecture, Topic. Class-Boundary Alignment for Imbalanced Dataset Learning, ICML Workshop on Learning from Imbalanced Data Sets II, Washington, DC, 2003.