Код комп’ютерної програми

## «Ієрархічне машинне навчання системи керування протезом кінцівки руки з неінвазивним зчитуванням біосигналів»

**Код IEITFrame.java**

package machine.learning.view;

import java.awt.Frame;

import java.awt.Toolkit;

import java.awt.event.WindowEvent;

import java.io.IOException;

import java.util.logging.Level;

import java.util.logging.Logger;

import machine.learning.controler.HierarchyDist;

import machine.learning.model.Parameters;

public class IEITFrame extends javax.swing.JFrame {

public IEITFrame() {

initComponents();

}

@SuppressWarnings("unchecked")

// <editor-fold defaultstate="collapsed" desc="Generated Code">//GEN-BEGIN:initComponents

private void initComponents() {

jFileChooser1 = new javax.swing.JFileChooser();

jLabel1 = new javax.swing.JLabel();

jLabel2 = new javax.swing.JLabel();

jLabel3 = new javax.swing.JLabel();

jLabel4 = new javax.swing.JLabel();

jLabel5 = new javax.swing.JLabel();

jTextClasses = new javax.swing.JTextField();

jTextRealiz = new javax.swing.JTextField();

jTextAttrib = new javax.swing.JTextField();

jTextDelta = new javax.swing.JTextField();

jScrollPane1 = new javax.swing.JScrollPane();

jTextParamsArea = new javax.swing.JTextArea();

jButton1 = new javax.swing.JButton();

jLabel6 = new javax.swing.JLabel();

jButton2 = new javax.swing.JButton();

jButton3 = new javax.swing.JButton();

jScrollPane2 = new javax.swing.JScrollPane();

jTextClassArea = new javax.swing.JTextArea();

jLabel7 = new javax.swing.JLabel();

jButton4 = new javax.swing.JButton();

jScrollPane3 = new javax.swing.JScrollPane();

jTextLearningArea = new javax.swing.JTextArea();

jSeparator1 = new javax.swing.JSeparator();

jLabel8 = new javax.swing.JLabel();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT\_ON\_CLOSE);

setTitle("IEIT Machine Learning");

setLocation(new java.awt.Point(0, 0));

addWindowListener(new java.awt.event.WindowAdapter() {

public void windowClosing(java.awt.event.WindowEvent evt) {

formWindowClosing(evt);

}

});

getContentPane().setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

jLabel1.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N

jLabel1.setText("Parameters");

getContentPane().add(jLabel1, new org.netbeans.lib.awtextra.AbsoluteConstraints(12, 13, -1, -1));

jLabel2.setText("Classes");

getContentPane().add(jLabel2, new org.netbeans.lib.awtextra.AbsoluteConstraints(12, 41, -1, -1));

jLabel3.setText("Realizations");

getContentPane().add(jLabel3, new org.netbeans.lib.awtextra.AbsoluteConstraints(12, 72, -1, -1));

jLabel4.setText("Attributes");

getContentPane().add(jLabel4, new org.netbeans.lib.awtextra.AbsoluteConstraints(12, 101, -1, -1));

jLabel5.setText("Delta");

getContentPane().add(jLabel5, new org.netbeans.lib.awtextra.AbsoluteConstraints(12, 130, -1, -1));

jTextClasses.setText("6");

getContentPane().add(jTextClasses, new org.netbeans.lib.awtextra.AbsoluteConstraints(92, 38, 54, -1));

jTextRealiz.setText("20");

jTextRealiz.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jTextRealizActionPerformed(evt);

}

});

getContentPane().add(jTextRealiz, new org.netbeans.lib.awtextra.AbsoluteConstraints(92, 69, 54, -1));

jTextAttrib.setText("1000");

getContentPane().add(jTextAttrib, new org.netbeans.lib.awtextra.AbsoluteConstraints(92, 98, 54, -1));

jTextDelta.setText("130");

jTextDelta.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jTextDeltaActionPerformed(evt);

}

});

getContentPane().add(jTextDelta, new org.netbeans.lib.awtextra.AbsoluteConstraints(92, 127, 54, -1));

jTextParamsArea.setColumns(20);

jTextParamsArea.setFont(new java.awt.Font("Times New Roman", 0, 12)); // NOI18N

jTextParamsArea.setRows(5);

jScrollPane1.setViewportView(jTextParamsArea);

getContentPane().add(jScrollPane1, new org.netbeans.lib.awtextra.AbsoluteConstraints(310, 40, 560, 112));

jButton1.setText("Set Parameters");

jButton1.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButton1ActionPerformed(evt);

}

});

getContentPane().add(jButton1, new org.netbeans.lib.awtextra.AbsoluteConstraints(170, 40, -1, -1));

jLabel6.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N

jLabel6.setText("Classes");

getContentPane().add(jLabel6, new org.netbeans.lib.awtextra.AbsoluteConstraints(12, 167, -1, -1));

jButton2.setText("Select Files");

jButton2.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButton2ActionPerformed(evt);

}

});

getContentPane().add(jButton2, new org.netbeans.lib.awtextra.AbsoluteConstraints(12, 202, 111, -1));

jButton3.setText("Print Selected");

jButton3.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButton3ActionPerformed(evt);

}

});

getContentPane().add(jButton3, new org.netbeans.lib.awtextra.AbsoluteConstraints(12, 240, -1, -1));

jTextClassArea.setColumns(20);

jTextClassArea.setFont(new java.awt.Font("Times New Roman", 0, 12)); // NOI18N

jTextClassArea.setRows(5);

jScrollPane2.setViewportView(jTextClassArea);

getContentPane().add(jScrollPane2, new org.netbeans.lib.awtextra.AbsoluteConstraints(182, 167, 689, 116));

jLabel7.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N

jLabel7.setText("Learning");

getContentPane().add(jLabel7, new org.netbeans.lib.awtextra.AbsoluteConstraints(12, 308, -1, -1));

jButton4.setText("Start Learning");

jButton4.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButton4ActionPerformed(evt);

}

});

getContentPane().add(jButton4, new org.netbeans.lib.awtextra.AbsoluteConstraints(12, 338, -1, -1));

jTextLearningArea.setColumns(20);

jTextLearningArea.setFont(new java.awt.Font("Times New Roman", 0, 12)); // NOI18N

jTextLearningArea.setRows(5);

jScrollPane3.setViewportView(jTextLearningArea);

getContentPane().add(jScrollPane3, new org.netbeans.lib.awtextra.AbsoluteConstraints(182, 308, 689, 167));

getContentPane().add(jSeparator1, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 480, 900, -1));

jLabel8.setFont(new java.awt.Font("Segoe UI", 2, 13)); // NOI18N

jLabel8.setText("Hierarchial IEIT machine learning");

getContentPane().add(jLabel8, new org.netbeans.lib.awtextra.AbsoluteConstraints(700, 480, -1, -1));

getAccessibleContext().setAccessibleName("IEITMachineLearning");

pack();

}// </editor-fold>//GEN-END:initComponents

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_jButton1ActionPerformed

try{

int amclass=Integer.valueOf(jTextClasses.getText());

int amrealiz=Integer.valueOf(jTextRealiz.getText());

int amattr=Integer.valueOf(jTextAttrib.getText());

int delta=Integer.valueOf(jTextDelta.getText());

if(amclass<=0||amrealiz<=0||amattr<=0||delta<=0){

amclass=6;

amrealiz=20;

amattr=1000;

delta=130;

jTextParamsArea.setText("Unsupposed values.\nParameters must be positive integers.\nSaved default values.\n"

+ "Amount of classes= "+amclass

+"\nAmount of realizations= "+amrealiz

+"\nAmount of attributes= "+amattr

+"\nDelta= "+delta);

}

else{

jTextParamsArea.setText("Parameters saved.\nAmount of classes= "+amclass

+"\nAmount of realizations= "+amrealiz

+"\nAmount of attributes= "+amattr

+"\nDelta= "+delta);}

Parameters.setK\_klass(amclass);

Parameters.setK\_realiz(amrealiz);

Parameters.setK\_oznaka(amattr);

Parameters.setDelta(delta);

Parameters.CreatePath();

}catch(NumberFormatException ex){

jTextParamsArea.setText("Unsupposed values. \nParameters must be positive integers.");

}

}//GEN-LAST:event\_jButton1ActionPerformed

private void jTextDeltaActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_jTextDeltaActionPerformed

// TODO add your handling code here:

}//GEN-LAST:event\_jTextDeltaActionPerformed

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_jButton2ActionPerformed

if(Parameters.valid()){

int result=jFileChooser1.showOpenDialog(jLabel1);

if(result==0){

Parameters.setInPath(jFileChooser1.getSelectedFile().getAbsoluteFile());

jTextClassArea.append("Added "+jFileChooser1.getName(jFileChooser1.getSelectedFile())+"\n");

}

}

else{jTextClassArea.setText("Class filling limit reached.");

}

}//GEN-LAST:event\_jButton2ActionPerformed

private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_jButton3ActionPerformed

jTextClassArea.setText("List of Recognition Classes:\n");

String[] temp=Parameters.getPath();

for(int i=0;i<Parameters.getPath().length;i++){

jTextClassArea.append((i+1)+". "+temp[i]+"\n");

}

}//GEN-LAST:event\_jButton3ActionPerformed

private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_jButton4ActionPerformed

String output="";

if(Parameters.allowLearning()){

try {

output= HierarchyDist.learning();

} catch (IOException ex) {

Logger.getLogger(IEITFrame.class.getName()).log(Level.SEVERE, null, ex);

}

}

else{jTextLearningArea.append("Set parameters and select classes first");

}

jTextLearningArea.append(Parameters.getLog()+"\n");

jTextLearningArea.append(output);

}//GEN-LAST:event\_jButton4ActionPerformed

private void jTextRealizActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_jTextRealizActionPerformed

// TODO add your handling code here:

}//GEN-LAST:event\_jTextRealizActionPerformed

private void formWindowClosing(java.awt.event.WindowEvent evt) {//GEN-FIRST:event\_formWindowClosing

// TODO add your handling code here:

}//GEN-LAST:event\_formWindowClosing

/\*\*

\* @param args the command line arguments

\*/

public static void main(String args[]) {

/\* Set the Nimbus look and feel \*/

//<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">

/\* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.

\* For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html

\*/

try {

for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {

if ("Windows".equals(info.getName())) {

javax.swing.UIManager.setLookAndFeel(info.getClassName());

break;

}

}

} catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(IEITFrame.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(IEITFrame.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(IEITFrame.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(IEITFrame.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

}

//</editor-fold>

/\* Create and display the form \*/

java.awt.EventQueue.invokeLater(new Runnable() {

public void run() {

new IEITFrame().setVisible(true);

}

});

}

// Variables declaration - do not modify//GEN-BEGIN:variables

private javax.swing.JButton jButton1;

private javax.swing.JButton jButton2;

private javax.swing.JButton jButton3;

private javax.swing.JButton jButton4;

private javax.swing.JFileChooser jFileChooser1;

private javax.swing.JLabel jLabel1;

private javax.swing.JLabel jLabel2;

private javax.swing.JLabel jLabel3;

private javax.swing.JLabel jLabel4;

private javax.swing.JLabel jLabel5;

private javax.swing.JLabel jLabel6;

private javax.swing.JLabel jLabel7;

private javax.swing.JLabel jLabel8;

private javax.swing.JScrollPane jScrollPane1;

private javax.swing.JScrollPane jScrollPane2;

private javax.swing.JScrollPane jScrollPane3;

private javax.swing.JSeparator jSeparator1;

private javax.swing.JTextField jTextAttrib;

private javax.swing.JTextArea jTextClassArea;

private javax.swing.JTextField jTextClasses;

private javax.swing.JTextField jTextDelta;

public static javax.swing.JTextArea jTextLearningArea;

private javax.swing.JTextArea jTextParamsArea;

private javax.swing.JTextField jTextRealiz;

// End of variables declaration//GEN-END:variables

}

**Код Para.java**

package machine.learning.model;

public class Para {

int a;

int b;

public int getA() {

return a;

}

public void setA(int a) {

this.a = a;

}

public int getB() {

return b;

}

public void setB(int b) {

this.b = b;

}

public Para(int x,int y){

this.a=x;

this.b=y;

}

}

**Код Parameters.java**

package machine.learning.model;

import java.io.File;

public class Parameters {

private static int iterator=0;

private static String[] path;

private static String outdirectory="";

private static File[] recfiles;

private static int k\_realiz;

private static int k\_oznaka;

private static int k\_klass;

private static int delta;

private static int vd[];

private static int nd[];

private static boolean allow=false;

private static String log="";

public static String getLog() {

return log;

}

public static String getOutdirectory() {

return outdirectory;

}

public static void addLog(String tlog) {

Parameters.log=log.concat("\n"+tlog);

}

public static void CreatePath(){

path=new String[k\_klass];

recfiles=new File[k\_klass];

}

public static boolean valid(){

if(iterator<k\_klass){return true;}

else{return false;}

}

public static void setInPath(File fl){

if(valid()){

path[iterator]=fl.getName();

recfiles[iterator]=fl;

if(outdirectory==""){outdirectory=fl.getAbsolutePath().substring(0, fl.getAbsolutePath().length()-fl.getName().length());}

iterator++;}

if(iterator==k\_klass){allow=true;}

}

public static String[] getPath() {

return path;

}

public static File[] getFiles() {

return recfiles;

}

public static void setPath(String[] path) {

Parameters.path = path;

}

public static int getK\_realiz() {

return k\_realiz;

}

public static void setK\_realiz(int k\_realiz) {

Parameters.k\_realiz = k\_realiz;

}

public static int getK\_oznaka() {

return k\_oznaka;

}

public static void setK\_oznaka(int k\_oznaka) {

Parameters.k\_oznaka = k\_oznaka;

}

public static int getK\_klass() {

return k\_klass;

}

public static void setK\_klass(int k\_klass) {

Parameters.k\_klass = k\_klass;

}

public static int getDelta() {

return delta;

}

public static void setDelta(int delta) {

Parameters.delta = delta;

}

public static int[] getVd() {

return vd;

}

public static void setVd(int[] vd) {

Parameters.vd = vd;

}

public static int[] getNd() {

return nd;

}

public static void setNd(int[] nd) {

Parameters.nd = nd;

}

public static int getNd(int o) {

return nd[o];

}

public static int getVd(int o) {

return vd[o];

}

public static boolean allowLearning() {

return allow;

}

}

**Код RecognClass.java**

package machine.learning.model;

import java.io.File;

import java.io.IOException;

import java.util.ArrayList;

import machine.learning.controler.InputData;

import machine.learning.controler.Methods;

import machine.learning.model.Para;

public class RecognClass {

protected int id;

protected static int count = 0;

protected String name;

private String path;

private int[] avg;

protected int[] etalon = new int[Parameters.getK\_oznaka()];

private int[][] Mat;

protected int[][] BM;

private int dist\_to\_neig;

private String neig;

private int Nid;

private int radius;

private double E[]=new double[Parameters.getK\_oznaka()];

private boolean area[]=new boolean[Parameters.getK\_oznaka()];

private double d1[]=new double[Parameters.getK\_oznaka()];

private double betta[]=new double[Parameters.getK\_oznaka()];

private ArrayList<Para> distances=new ArrayList<Para>();

private int ParentA=-1;

private int ParentB=-1;

public ArrayList<Para> getDistances() {

return distances;

}

public RecognClass() {

}

public RecognClass(String name,int a,int b,int[][]Binar) {

this.name=name;

this.BM=Binar;

this.ParentA=a;

this.ParentB=b;

this.id=count;

count++;

this.etalon=Methods.getItEtalon(Binar);

}

@Override

public String toString(){

return("Id:"+id+" Class:"+getName());

}

public void setId(int id) {

this.id = id;

}

public int getRadius() {

return radius;

}

public void setRadius(int radius) {

this.radius = radius;

}

public RecognClass(File f) throws IOException {

this.path=f.getAbsolutePath();

this.name= f.getName().substring(0,f.getName().length()-4);

this.Mat=InputData.getMatrix(f);

this.id=count;

count++;

}

public int[] getItAvg() {

return avg;

}

public int[] getEtalon() {

return etalon;

}

public int[][] getMat() {

return Mat;

}

public int[] getVector(int i) {

int t[]=new int[Parameters.getK\_oznaka()];

for(int k=0;k<Parameters.getK\_oznaka();k++){

t[k]=BM[i][k];

}

return t;

}

public int getMat(int i,int j) {

return Mat[i][j];

}

public int[][] getBM() {

return BM;

}

public String getPath() {

return name;

}

public String getName() {

return name;

}

public void setBM(int[][] bm) throws IOException {

this.BM=bm;

}

public void setEtalon(int[] et) {

this.etalon=et;

}

public void setNeigh(String neighbourName) {

this.neig=neighbourName;

}

public void setDist\_to\_neig(int dist\_to\_neig) {

this.dist\_to\_neig = dist\_to\_neig;

}

public int getDist\_to\_neig() {

return dist\_to\_neig;

}

public String getNeig() {

return neig;

}

public void setAvg(int[] avg) {

this.avg=avg;

}

public void setNid(int Nid) {

this.Nid = Nid;

}

public int getId() {

return id;

}

public int getNid() {

return Nid;

}

public double getE(int i) {

return E[i];

}

public boolean getArea(int i) {

return area[i];

}

public double getD1(int i) {

return d1[i];

}

public int getParentA() {

return ParentA;

}

public void setParentA(int ParentA) {

this.ParentA = ParentA;

}

public int getParentB() {

return ParentB;

}

public void setParentB(int ParentB) {

this.ParentB = ParentB;

}

public double getBetta(int i) {

return betta[i];

}

public void setE(int rad, double d) {

this.E[rad]=d;

}

public void setBetta(int rad, double t\_betta) {

this.betta[rad]=t\_betta;

}

public void setD1(int rad, double t\_d1) {

this.d1[rad]=t\_d1;

}

public void setArea(int rad, boolean b) {

this.area[rad]=b;

}

public void addDistance(Para idDist) {

this.distances.add(idDist);

}

}

**Код SuperRClass.java**

package machine.learning.model;

import machine.learning.controler.Methods;

import java.io.IOException;

public class SuperRClass extends RecognClass {

private final RecognClass a;

private final RecognClass b;

public SuperRClass(RecognClass a,RecognClass b) throws IOException {

this.name=a.getName()+b.getName();

this.id=count;

count++;

this.BM=concatMatr(a.getBM(),b.getBM());

this.etalon=Methods.getItEtalon(BM);

this.a=a;

this.b=b;

a.setNid(b.id);

b.setNid(a.id);

}

public int[][] createMatr(int[][] a,int[][] b){

int[][] result = new int[Parameters.getK\_realiz()][Parameters.getK\_oznaka()];

for(int i=0;i<Parameters.getK\_realiz();i++){

for(int j=0;j<Parameters.getK\_oznaka();j++){

if(a[i][j]==b[i][j]){

result[i][j]=a[i][j];}

else{result[i][j]=0;}

}

}

return result;

}

public int[][] concatMatr(int[][] a,int[][] b){

int aLen = a.length;

int bLen = b.length;

int[][] result = new int[aLen+bLen][];

System.arraycopy(a, 0, result, 0, aLen);

System.arraycopy(b, 0, result, aLen, bLen);

return result;

}

public String getName() {

return name;//change for imgs

}

}

**Код InputData.java**

package machine.learning.controler;

import java.awt.Toolkit;

import java.awt.event.WindowEvent;

import java.io.File;

import java.util.Scanner;

import javax.swing.JOptionPane;

import machine.learning.model.Parameters;

import machine.learning.view.IEITFrame;

public class InputData {

public static int[][] getMatrix(File f){

try {

Scanner sc = new Scanner(f);

int[][] matrix = new int [Parameters.getK\_realiz()][Parameters.getK\_oznaka()];

for(int i = 0; i< Parameters.getK\_realiz(); i++)

{

for(int k = 0; k < Parameters.getK\_oznaka(); k++)

{

matrix[i][k] =sc.nextInt();

}

}

return matrix;

} catch (Exception ex) {

JOptionPane.showMessageDialog(IEITFrame.jTextLearningArea,"File:"+f.getName()+"\nException occured:"+ex.toString()+"\nRestart program.\nCheck for correct version of data file with suitable parameters.");

}

return null;

}

}

**Код Methods.java**

package machine.learning.controler;

import java.io.BufferedWriter;

import java.io.File;

import java.io.FileWriter;

import java.io.IOException;

import java.util.ArrayList;

import machine.learning.model.Parameters;

import machine.learning.model.RecognClass;

public class Methods {

public static int[] getAvg(int[][] Mat) {

int[] avg=new int[Parameters.getK\_oznaka()];

for(int i=0;i<Parameters.getK\_oznaka();i++){

for(int j=0;j<Parameters.getK\_realiz();j++){

avg[i]=avg[i]+Mat[j][i];

}

avg[i]=(int)avg[i]/Parameters.getK\_realiz();

}

return avg;

}

public static int[][] toBinar(int[][] Mat) {

int[][] bm=new int[Parameters.getK\_realiz()][Parameters.getK\_oznaka()];

for(int r=0;r<Parameters.getK\_realiz();r++){

for(int o=0;o<Parameters.getK\_oznaka();o++){

bm[r][o]=0;

if((Mat[r][o]>Parameters.getNd(o))&&(Mat[r][o]<Parameters.getVd(o))){bm[r][o]=1;}

}

}

return bm;

}

public static int[] getItEtalon(int[][]bm,int rs){

int[] temp=new int[Parameters.getK\_oznaka()];

int uno=0;//amount of 1 in matrix

for(int o=0;o<Parameters.getK\_oznaka();o++){

uno=0;

for(int r=0;r<rs;r++){

if(bm[r][o]==1){uno++;}

}

if(uno>=Parameters.getK\_realiz()/2){temp[o]=1;}

else{temp[o]=0;}

}

return temp;

}

public static int[] getItEtalon(int[][]bm){

int[] temp=getItEtalon(bm,Parameters.getK\_realiz());

return temp;

}

public static int calculeteDistance(int[] a,int[] b){

int dist=0;

for(int o=0;o<Parameters.getK\_oznaka();o++){

if(a[o]!=b[o]){dist++;}

}

return dist;

}

public static void getNeighbour(RecognClass r,ArrayList<RecognClass> recclasses) {

int nId=-1;

int mindist=Parameters.getK\_oznaka();

int dist=-1;

for(RecognClass k:recclasses){

if(!r.getName().equalsIgnoreCase(k.getName())){

dist=calculeteDistance(r.getEtalon(),k.getEtalon());

if(mindist>=dist){

mindist=dist;

nId=k.getId();

}

}

}

r.setNeigh(recclasses.get(nId).getName());

r.setNid(nId);

r.setDist\_to\_neig(dist);

}

static void writeOutData(RecognClass r) throws IOException{

File file = new File(Parameters.getOutdirectory()+r.getId()+"\_"+r.getName()+"\_radius.txt");

FileWriter fileWriter = new FileWriter(file);

BufferedWriter bufferedWriter = new BufferedWriter(fileWriter);

bufferedWriter.write("radius\t E\t area\t D1\t betta\t NeighbourID\t toNeighbour\n");

for(int i=0;i<Parameters.getK\_oznaka();i++){

bufferedWriter.write(i+"-"+r.getRadius()+"\t"+r.getE(i)+"\t"+r.getArea(i)+"\t"+r.getD1(i)+"\t"+r.getBetta(i)+"\t"+r.getNid()+"\t"+r.getDist\_to\_neig()+"\n");

}

bufferedWriter.close();

}

public static void limits(int avg[]){

int vtemp[]=new int[Parameters.getK\_oznaka()];

int ntemp[]=new int[Parameters.getK\_oznaka()];

for(int o=0;o<Parameters.getK\_oznaka();o++){

vtemp[o]=avg[o]+Parameters.getDelta();

ntemp[o]=avg[o]-Parameters.getDelta();

}

Parameters.setVd(vtemp);

Parameters.setNd(ntemp);

}

static int[][] createMatr(int[][] a,int[][] b){

int[][] result = new int[Parameters.getK\_realiz()][Parameters.getK\_oznaka()];

for(int i=0;i<Parameters.getK\_realiz();i++){

for(int j=0;j<Parameters.getK\_oznaka();j++){

if(a[i][j]==b[i][j]){

result[i][j]=a[i][j];}

else{result[i][j]=0;}

}

}

return result;

}

}

**Код HierarchyDist.java**

package machine.learning.controler;

import java.io.File;

import java.io.IOException;

import java.util.ArrayList;

import machine.learning.model.Para;

import machine.learning.model.Parameters;

import machine.learning.model.RecognClass;

public class HierarchyDist {

public static String learning() throws IOException{

ArrayList<RecognClass> rclas = new ArrayList<>();

for(int t=0;t<Parameters.getK\_klass();t++){

File files[]=Parameters.getFiles();

RecognClass CurrentClass= new RecognClass(files[t].getAbsoluteFile());

Parameters.addLog("Class :"+files[t].getName()+" from file "+files[t].getAbsolutePath());

rclas.add(CurrentClass);

}

for(RecognClass k:rclas){

RecognClass base =rclas.get(0);

base.setAvg(Methods.getAvg(base.getMat()));

Methods.limits(base.getItAvg());

for(RecognClass r:rclas){

if(!k.equals(r)){

r.setBM(Methods.toBinar(r.getMat()));

r.setEtalon(Methods.getItEtalon(r.getBM()));

int dist=Methods.calculeteDistance(k.getEtalon(), r.getEtalon());

Para idDist=new Para(r.getId(),dist);

k.addDistance(idDist);

}

}

}

int pcls[]=new int[2\*(Parameters.getK\_klass()-1)];

ArrayList<RecognClass> temp=rclas;

ArrayList<RecognClass> processed=new ArrayList(rclas);

ArrayList<RecognClass> all=new ArrayList(rclas);

ArrayList<RecognClass> supers=new ArrayList();

for(int i=0;i<Parameters.getK\_klass()-1;i++){

RecognClass sc=GetClosest(temp);

if(sc.getName()==null){break;}

Parameters.addLog("Closest: "+sc.getParentA()+" and "+sc.getParentB());

pcls[sc.getParentA()]=sc.getParentB();

pcls[sc.getParentB()]=sc.getParentA();

supers.add(sc);

temp.remove(getListedClass(temp,sc.getParentA()));

temp.remove(getListedClass(temp,sc.getParentB()));

if(isListed(processed,sc.getParentA()))

{

processed.remove(getListedClass(temp,sc.getParentA()));

}

if(isListed(processed,sc.getParentB()))

{

processed.remove(getListedClass(temp,sc.getParentB()));

}

temp.add(sc);

all.add(sc);

for(RecognClass rc:temp){

if(!rc.equals(sc)){

Parameters.addLog(rc.getName());

int dist=Methods.calculeteDistance(rc.getEtalon(), sc.getEtalon());

Para idDist=new Para(sc.getId(),dist);

rc.addDistance(idDist);

}

}

}

Parameters.addLog("Radius optimization...");

for(int k=0;k<10;k++){

optRadius(all.get(k),all.get(pcls[k]));

Methods.writeOutData(all.get(k));

Parameters.addLog("\tClass:"+k);

}

return "Execution finished. Results printed to id\_classname\_radius.txt files.";

}

public static String optRadius(RecognClass r,RecognClass n){

n.setNid(r.getId());

n.setDist\_to\_neig(r.getDist\_to\_neig());

double d=-1,maxE=-1;

for(int rad=0;rad<Parameters.getK\_oznaka();rad++){

double k1=0,k2,k3=0,k4;

double t\_d1,t\_d2,t\_alpha, t\_betta, d1\_b, kfe = 0;

for (int i = 0; i < Parameters.getK\_realiz(); i++){

if (Methods.calculeteDistance(r.getEtalon(), r.getVector(i)) <= rad) { k1++;}

if (Methods.calculeteDistance(r.getEtalon(), n.getVector(i)) <= rad) { k3++;}

}

k4 = Parameters.getK\_realiz() - k3;

k2 = Parameters.getK\_realiz() - k1;

t\_d1 = k1 / Parameters.getK\_realiz();

t\_d2= k2 / Parameters.getK\_realiz();

t\_alpha= k4 / Parameters.getK\_realiz();

t\_betta = k3 / Parameters.getK\_realiz();

d1\_b = t\_d1 - t\_betta;

d=d1\_b \* Math.log((1.0 + d1\_b + 0.1) / (1.0 - d1\_b + 0.1)) / Math.log(2.0);

r.setE(rad,d);

r.setBetta(rad,t\_betta);

r.setD1(rad,t\_d1);

if ((t\_d1 >= 0.5)&&(t\_betta < 0.5)) {

r.setArea(rad,true);

if(d>maxE){

maxE=d;

r.setRadius(rad);

} }

else{

r.setArea(rad,false);} }

return "";

}

static RecognClass GetClosest(ArrayList<RecognClass> temp) throws IOException{

int mind=Integer.MAX\_VALUE;

RecognClass ca=new RecognClass();

RecognClass cb=new RecognClass();

for(RecognClass rc:temp){

for(Para idist:rc.getDistances()){

if(isListed(temp,idist.getA())){

if(mind>idist.getB()){

mind=idist.getB();

rc.setNid(idist.getA());

ca=rc;

cb=getListedClass(temp,idist.getA());

rc.setDist\_to\_neig(Methods.calculeteDistance(ca.getEtalon(), cb.getEtalon()));

}

}

}

}

if((ca.getName()==null)||(cb.getName()==null)){return null;}

int[][] BM=Methods.createMatr(ca.getBM(),cb.getBM());

RecognClass sp = new RecognClass(ca.getName()+cb.getName(),ca.getId(),cb.getId(),BM);

return sp;

}

static ArrayList removeByClassId(ArrayList<RecognClass> r,int num){

int out = -1;

for(RecognClass current:r){

if(current.getId()==num){

out=r.indexOf(current);

}}

r.remove(out);

return r;

}

static RecognClass getListedClass(ArrayList<RecognClass> rc, int num){

for(RecognClass current:rc){

if(current.getId()==num){

return current;

}

}

return null;

}

static boolean isListed(ArrayList<RecognClass> rc, int num) {

for(RecognClass current:rc){

if(current.getId()==num){

return true;

}

}

return false;

}

}