Treball final de grau

GRAU DE MATEMÀTIQUES

Facultat de Matemàtiques
Universitat de Barcelona

Màquina de vectors de suport i aplicació a un problema d'anàlisi d'àudio

---

Autor: Josep Pérez Díez

Director:	Dr. Oriol Pujol Vila
Departament:	Matemàtica Aplicada i Anàlisi
Barcelona,	18 de gener de 2016

Abstract

The present work aims to find a mathematical model for classifying musical instruments from their timbres, so that we can use it for building an Android application which is able to classify a recorded instrument efficiently. To do so, we train a support vector machine with samples of such an instrument spectra as vectors, in order to obtain a pattern that let us classify new samples of these instruments.

Resum

Aquest treball té l'objectiu de trobar un model matemàtic per classificar instruments musicals a partir dels seus timbres, de manera que el puguem usar per construir una aplicació Android que sigui capaç de classificar eficientment un instrument enregistrat. Per fer això, entrenem una màquina de vectors de suport amb mostres d'aquest tipus d'espectres com a vectors, per tal d'obtenir un patró que ens permeti classificar noves mostres dels mateixos instruments.

Agraïments

En primer lloc vull agrair a mons pares el seu suport moral i la seva paciència durant tot aquest temps que ha durat el treball. Gràcies a ells he tingut forces per continuar i acabar-lo, malgrat les dificultats al llarg del darrer curs i l'actual.

També vull agrair a n'Oriol Pujol que acceptés dirigir-me aquest treball, que difereix de la seva proposta inicial, doncs en documentar-me sobre aquesta vaig descobrir el mètode de la màquina de vectors de suport i vaig pensar que podria ser útil per classificar sons. L'hi vaig plantejar i va acceptar canviar el tema del treball.

Índex

1 Introducció

1.1 Motivació

La principal motivació per la qual he escollit aquest tema és la realització d'un treball de l'assignatura Projecte Integrat de Software, de la menció en Informàtica, l'objectiu del qual era la monitorització del so gravat amb el dispositiu mòbil dels usuaris geolocalitzats, per tal de realitzar un mapa de so de les zones on s'han enregistrat les gravacions. Com a complement es va intentar implementar un algorisme de classificació de sons, que no va resultar molt eficaç. Prenent com a punt de partida el tema de la classificació i fent ús de les tècniques d'aprenentatge automàtic ens disposem a trobar un mètode eficient per a classificar sons.

1.2 Objectius

Els objectius principals del treball són els següents:

- Entendre i descriure el concepte de classificació, com a part de la teoria de l'aprenentatge automàtic.
- Entendre i descriure les màquines de vectors de suport, una de les principals eines de l'anomenada Classificació Estadística.
- Descriure la base de la teoria de Fourier per tal de comprendre la Transformada Ràpida de Fourier, eina clau per al càlcul de la Transformada Discreta de Fourier.
- Construir un model matemàtic, a partir de les eines anteriors, que ens porti a la implementació d'una aplicació capaç de classificar dos tipus de sons de forma eficient.

Descripció del projecte

En aquest projecte final de Grau s'ha volgut combinar l'anàlisi i matemàtica aplicada apresa al llarg de la carrera, juntament amb les eines informàtiques adquirides a les assignatures de la menció en Informàtica, durant l'últim curs. Així, la part pràctica ha tingut un pes important en el treball.

S'ha fet servir una eina d'aprenentatge supervisat com és la màquina de vectors de suport, utilitzant espectres de dos instruments diferents com a vectors. Prèviament s'ha realitzat la tasca de gravació d'un nombre determinat d'àudios de cada instrument.

Per a l'entrenament i testeig del model matemàtic s'ha utilitzat el llenguatge Python, que incorpora moltes llibreries de "machine learning". Per tal de visualitzar el resultat amb un exemple també s'ha elaborat una aplicació Android que classifica dos tipus d'instruments, la flauta dolça i el saxofon.

Estructura de la Memòria

El treball està estructurat en quatre grans blocs. El primer consisteix en descriure l'aprenentatge automàtic com a disciplina de la intel·ligència artificial, centrant-se en la classificació com a aplicació. Seguidament es descriu el model matemàtic de classificació, juntament amb un breu resum sobre la teoria de Fourier, per entendre bé la principal eina del treball, que és l'epectre d'un senyal acústic. Després s'exposa com aplicar el model descrit a un problema concret d'anàlisi d'àudio, tot mostrant els resultats obtinguts a partir de les gravacions d'àudio realitzades. Finalment s'explica en què consisteix l'aplicació desenvolupada per a poder classificar dos instruments musicals a partir del seu timbre.

2 Introducció a l'aprenentatge automàtic

L'aprenentatge automàtic ("Machine learning") és una disciplina de la Intel·ligència artificial que consisteix en desenvolupar models matemàtics per a l'aprenentatge a base de reconeixer patrons preestablerts. A mesura que afegim mostres d'entrenament al nostre model, aquest va eixamplant el seu coneixement, de cara a un millor reconeixement de futures mostres. Així, l'aprenentatge va estrictament lligat a l'estadística.

L'objecte d'estudi d'aquest treball se centra en la classificació, un tipus particular d'aprenentatge automàtic supervisat, on les mostres d'entrenament són parelles de dades, la primera representada normalment per un vector i la segona per un valor numèric que identifica la classe a la qual pertany la primera. A la primera dada se l'anomena vector característic i a la segona etiqueta. Així, si disposem d'un conjunt de n mostres i cada vector característic té dimensió p, podem construir la matriu característica i el vector d'etiquetes, que representen tots els vectors característics i totes les etiquetes respectivament:

Matriu característica: $\mathbf{X} = \begin{bmatrix} x_{11} & x_{12} & \cdot & \cdot & \cdot & x_{1d} \\ \cdot & & & & & \\ \cdot & & & & & \\ \cdot & & & & & \\ x_{n1} & x_{n2} & \cdot & \cdot & \cdot & x_{np} \end{bmatrix}$

Vector d'etiquetes: $\mathbf{Y} = [y_1 \cdots y_n]$

La mesura més immediata de l'eficàcia del nostre classificador és l'*exactitud*, que mesura la proporció de mostres classificades correctament

$$e = \frac{\text{Nombre de prediccions correctes}}{n}$$

No obstant, hi ha casos en que aquesta mètrica no és suficient per a una bona classificació, com per exemple en els casos en que el nombre de mostres de les diferents classes no és similar.

En el cas d'un problema binari, és a dir que $\mathbf{Y} = \{-1, 1\}$, podem definir una nova mètrica a partir del que anomenem matriu de confusió, que es construeix a partir de les quatre combinacions lògiques que el classificador ens aporta:

- **Vertader positiu (TP):** Quan el classificador prediu una mostra com a positiva i és positiva.
- **Fals positiu (FP):** Quan el classificador prediu una mostra com a positiva i és negativa.
- **Vertader negatiu (TN):** Quan el classificador prediu una mostra com a negativa i és negativa.

- **Fals negatiu(FN):** Quan el classificador prediu una mostra com a negativa i és positiva.

La matriu de confusió es defineix de la següent manera:

$$MC = \begin{bmatrix} \mathbf{TP} & \mathbf{FP} \\ \mathbf{FN} & \mathbf{TN} \end{bmatrix}$$

Observem que a partir d'aquestes quatre definicions podem recuperar la mètrica que hem definit com exactitud de la següent manera:

$$e = \frac{\mathbf{TP} + \mathbf{TN}}{\mathbf{TP} + \mathbf{FP} + \mathbf{TN} + \mathbf{FN}}$$

A més a més la combinació d'aquestes també ens permet definir quatre noves mètriques:

$$\text{sensibilitat} = \frac{\mathbf{TP}}{\text{Positius reals}} = \frac{\mathbf{TP}}{\mathbf{TP} + \mathbf{FN}}$$

$$\text{especificitat} = \frac{\mathbf{TN}}{\text{Negatius reals}} = \frac{\mathbf{TN}}{\mathbf{TN} + \mathbf{FP}}$$

$$\text{precisió} = \frac{\mathbf{TP}}{\text{Positius predits}} = \frac{\mathbf{TP}}{\mathbf{TP} + \mathbf{FP}}$$

$$\text{Valor predictiu negatiu} \equiv \text{NPV} = \frac{\mathbf{TN}}{\text{Negatius predits}} = \frac{\mathbf{TN}}{\mathbf{TN} + \mathbf{FN}}$$

L'eficàcia i fiabilitat del nostre classificador dependrà de la quantitat de falsos positius i negatius que obtinguem en l'etapa d'entrenament i del nombre de mostres usades, respectivament. El que fem és subdividir les mostres en dos conjunts, el d'entrenament i el de test. Això ens permet crear el patró de classificació a partir del conjunt d'entrenament, el que seria la fase d'aprenentatge, i seguidament testejem aquest model amb les mostres del conjunt de test, que són noves i per tant l'encert o error en aquesta etapa ens dóna una idea de la qualitat del nostre classificador. Formalment, podem definir dos errors:

- E_{in}**:** Error en el conjunt de mostres d'entrenament, és a dir mitjana mostral de l'error. Formalment

  $$\frac{1}{N} \sum_{i=1}^{N} e(x_i, y_i)$$

  on $e(x_i, y_i)$ representa, en el nostre cas, 1 si encertem ó 0 si errem.

- E_{out}**:** Error generalitzat, que representa l'error esperat sobre mostres desconegudes. El que es fa és estimar aquest error poblacional amb la mitjana de l'error en el test.

Una bona manera de triar un classificador per al nostre model és repetir iterativament el procés d'entrenament i testeig per a diferents paràmetres del nostre classificador o per a diferents classificadors i escollir aquells paràmetres o classificador que minimitzi E_{out}.

Un cop triat el classificador adient, procedim a l'estudi de l'error comès en la classificació mitjançant el nostre model. Per a fer-ho hem triat un mètode anomenat validació creuada de Monte Carlo. Consisteix en fer diferents particions aleatòries del conjunt de mostres entre dades d'entrenament i test. Es procedeix a entrenar i testejar el model iterativament per a cada partició, per tal de garantir que el model no depengui d'una partició determinada, i finalment s'estima E_{out} fent la mitjana aritmètica de les mesures de cada iteració. La figura 1 il·lustra una validació creuada de Monte Carlo per a un problema de classificació binària. S'hi poden veure les diferents particions aleatòries del conjunt de mostres.

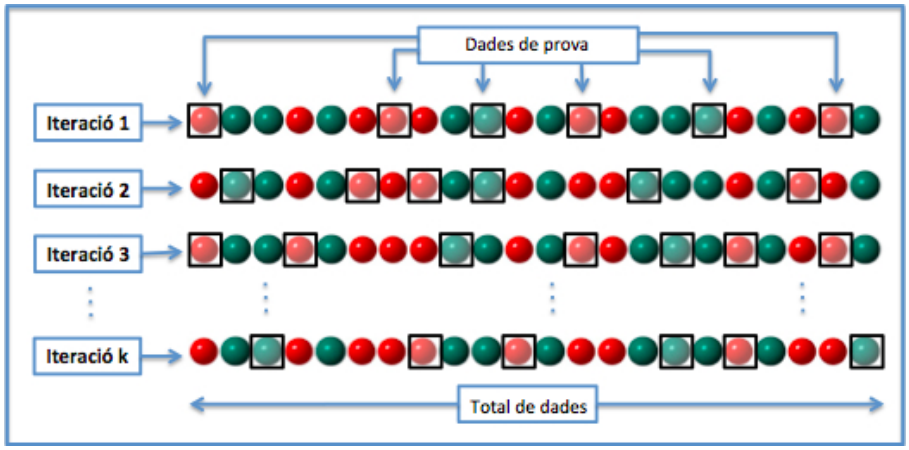


Figura 1: Exemple de les diferents particions d'una validació creuada per a l'entrenament d'un conjunt de mostres que conformen un problema de classificació binària [1]

En el següent capítol exposem l'algorisme de classificació utilitzat en aquest treball.

3 Màquina de vectors de suport (SVM)

3.1 SVM lineal

3.1.1 SVM amb marge rígid

Suposem que, en base a un criteri previ, tenim un conjunt de dos tipus de dades. L'objectiu del mètode dels vectors de suport (SVM) és establir, a partir d'una mostra d'aquest conjunt, un criteri que ens permeti decidir a quin dels dos tipus pertany una nova dada del conjunt que definim formalment com $X \subset \mathbb{R}^p$, on $\mathbb{R}^p$ representa l'espai vectorial euclidià de dimensió $p \in \mathbb{N}$, amb el producte escalar usual. Per tal de fer això, suposem que existeixen dos subespais disjunts convexes $C, D \subset \mathbb{R}^p$, que contenen tots els punts de X, de manera que C només contingui punts d'un tipus i D de l'altre. Llavors, pel teorema de l'hiperplà separador[2, 2.5.1], existeixen $\mathbf{a} \in \mathbb{R}^p$ i $b \in \mathbb{R}$ tals que $\mathbf{a} \cdot \mathbf{x} \leq b$ per a qualsevol $\mathbf{x} \in C$ i $\mathbf{a} \cdot \mathbf{x} > b$ per a qualsevol $\mathbf{x} \in D$. L'hiperplà $\{\mathbf{x} \in \mathbb{R}^p \mid \mathbf{a} \cdot \mathbf{x} = b\}$ s'anomena hiperplà separador, ja que separa $\mathbb{R}^p$ en dos subespais, que contenen C i D respectivament. Això ens permet definir una classe d'equivalència, de manera que dos punts de $\mathbb{R}^p$ són equivalents quan ambdós pertanyen a un dels següents subespais:

$$H^- : \{\mathbf{x} \in \mathbb{R}^p \mid \mathbf{a} \cdot \mathbf{x} \leq b\} \tag{3.1}$$

$$H^+ : \{\mathbf{x} \in \mathbb{R}^p \mid \mathbf{a} \cdot \mathbf{x} > b\} \tag{3.2}$$

Per tant es compleix que $C \subset H^-$ i $D \subset H^+$.

Si fem l'extrapolació que tots els punts continguts a H^- i H^+ són del mateix tipus que els continguts a C i D respectivament, llavors podem separar noves dades en funció de la classe d'equivalència a la qual pertanyen. Dit d'una altra forma, les podem separar en funció de si estan a un costat o altre de l'hiperplà. A la figura 2 es pot observar una representació gràfica del problema, per al cas de dues dimensions.

Tanmateix, no hem d'oblidar que estem intentant establir un model matemàtic que descrigui el nostre problema real de la manera més precisa possible i que, físicament, sempre hi haurà dades que caiguin tant a prop de la frontera que no sigui correcte classificar-les mitjançant el model establert. La física, doncs, ens està imposant una certa precisió a l'hora de poder classificar les dades.

En aquest sentit, la idea és que, enlloc d'agafar un únic hiperplà, prenem dos hiperplans paral·lels a l'hiperplà separador i que segueixin separant les dades en dues classes de punts. Aquests hiperplans delimiten el marge on idealment no ha de pertànyer cap punt i els definim de la següent forma

$$\mathbf{a} \cdot \mathbf{x} - b = 1 \tag{3.3}$$

$$\mathbf{a} \cdot \mathbf{x} - b = -1 \tag{3.4}$$

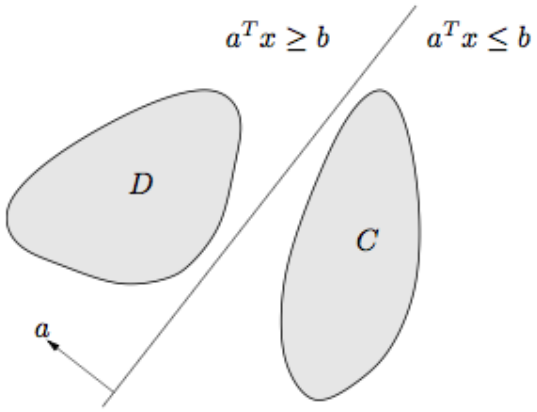


Figura 2: Il·lustració gràfica en dues dimensions del teorema de l'hiperplà separador.[2, Figura 2.19]

escollint **a** i b de manera que la distància entre els hiperplans sigui màxima.

La distància entre els dos hiperplans[3] vé donada per la fórmula $\frac{2}{\|\mathbf{a}\|}$. El que volem és maximitzar aquesta distància, sense que cap punt quedi fora de la zona que li pertoca. Per tant, suposant que el tamany de la mostra de dades és n, ens enfrontem al següent problema d'optimització:

$$\begin{aligned} &\textbf{minimitzar} \quad \|\mathbf{a}\| \\ &\textbf{subjecte a} \quad 1 - y_i(\mathbf{a} \cdot \mathbf{x}_i - b) \leq 0 \quad \forall i = 1, ..., n \end{aligned} \tag{3.5}$$

on $y_i = \{-1, 1\}$.

Sigui $h(a_1, ..., a_p) = \|\mathbf{a}\| = \sqrt{\sum_i a_i^2}$ la funció objectiu del problema 3.5, llavors tenim que

$$\nabla h = \frac{\nabla(\sum a_i^2)}{2h} = \frac{\nabla(\|\mathbf{a}\|^2)}{2h} \tag{3.6}$$

per tant

$$\nabla\|\mathbf{a}\| = 0 \iff \nabla(\|\mathbf{a}\|^2) = 0 \tag{3.7}$$

És a dir que el nostre problema d'optimització és anàleg al següent, que és un problema d'optimització convexa[1]:

[1]Un problema d'optimització convexa és aquell en el que la funció objectiu i les condicions són convexes.

$$\begin{aligned} &\textbf{minimitzar} \quad f_0(\mathbf{a}) := \frac{1}{2}\|\mathbf{a}\|^2 \\ &\textbf{subjecte a} \quad f_i(\mathbf{a}, b) := 1 - y_i(\mathbf{a} \cdot \mathbf{x}_i - b) \leq 0 \end{aligned} \tag{3.8}$$

Definim el Lagrangià $L : \mathbb{R}^p \times \mathbb{R} \times \mathbb{R}^n \longrightarrow \mathbb{R}$ associat al problema 3.8 com

$$L(\mathbf{a}, b, \boldsymbol{\alpha}) = f_0(\mathbf{a}) + \sum_{i=1}^{n} \alpha_i f_i(\mathbf{a}, b) \tag{3.9}$$

on $\boldsymbol{\alpha} = (\alpha_1, ..., \alpha_n)$ és el vector dels multiplicadors de Lagrange associats a les desigualtats lligam.

Prenent el mínim valor de L respecte $\mathbf{a}$ i b obtenim la següent funció

$$g(\boldsymbol{\alpha}) = \inf_{\mathbf{a}\in\mathbb{R}^p, b\in\mathbb{R}} L(\mathbf{a}, b, \boldsymbol{\alpha}) = \inf_{\mathbf{a}\in\mathbb{R}^p, b\in\mathbb{R}} (f_0(\mathbf{a}) + \boldsymbol{\alpha} \cdot \mathbf{f}(\mathbf{a}, b)) \tag{3.10}$$

on $\mathbf{f}(\mathbf{a}, b) = (f_1(\mathbf{a}, b), ..., f_n(\mathbf{a}, b))$.

La funció 3.10 s'anomena funció dual del problema. Observem que aquesta funció és el mínim, punt a punt, d'una família de funcions afins i, per tant, és una funció còncava. Això és així ja que per a cada $\boldsymbol{\alpha} \in \mathbb{R}^n$ la funció $g(\boldsymbol{\alpha})$ pren el mínim valor de totes les imatges de la família de funcions afins que defineix el lagrangià L per a cada possible valor de $\mathbf{a}$. A continuació demostrem aquesta propietat suposant que estem en el cas $n = 1, p = 2$, és a dir que tenim un únic punt x_1 que viu a l'espai $\mathbb{R}^2$. Llavors

$$g(\alpha) = \inf_{\mathbf{a}\in\mathbb{R}^2, b\in\mathbb{R}} (f_0(\mathbf{a}) + \alpha f(\mathbf{a}, b)) \tag{3.11}$$

Com que $f = ctant$ és la família de rectes que passen pel punt x_1, podem entendre la funció g com una funció definida a trossos que per a cada α pren el valor mínim de tots els possibles, que són tots els valors que pren f en funció de $\mathbf{a}$ i b. Per tant la seva gràfica seria com la de la figura 3

D'altra banda, sigui p^* el valor òptim del problema 3.8, llavors es compleix que per a qualsevol α tal que $\alpha_i \geq 0 \ \forall i$

$$g(\boldsymbol{\alpha}) \leq p^* \tag{3.12}$$

Demostració: Siguin $\tilde{\mathbf{a}}$ i $\tilde{b}$ tals que es compleixen les desigualtats del problema 3.8, és a dir que $f_i(\tilde{\mathbf{a}}, \tilde{b}) \leq 0$. Llavors

$$\sum_{i=1}^{n} \alpha_i f_i(\tilde{\mathbf{a}}) \leq 0$$

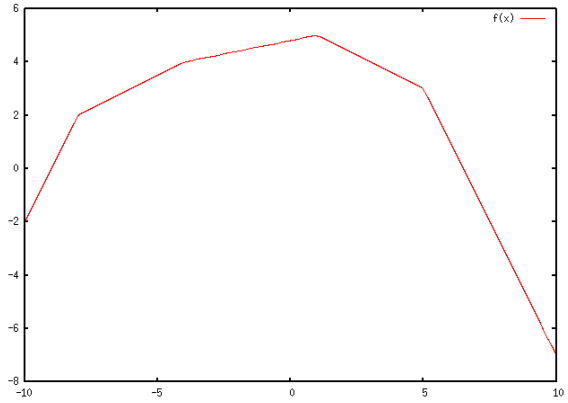


Figura 3: Exemple de gràfica d'una funció dual en dues dimensions.

com que cada terme és no positiu resulta que

$$L(\tilde{\mathbf{a}}, \tilde{b}, \boldsymbol{\alpha}) = f_0(\tilde{\mathbf{a}}) + \sum_{i=1}^{n} \alpha_i f_i(\tilde{\mathbf{a}}) \leq f_0(\tilde{\mathbf{a}})$$

Aleshores

$$g(\boldsymbol{\alpha}) = \inf_{\mathbf{a}\in\mathbb{R}^p, b\in\mathbf{R}} L(\mathbf{a}, b, \boldsymbol{\alpha}) \leq L(\tilde{\mathbf{a}}, \tilde{b}, \boldsymbol{\alpha}) \leq f_0(\tilde{\mathbf{a}})$$

□

La funció dual $g(\boldsymbol{\alpha})$ és, per tant, una cota inferior per al valor òptim del problema 3.8. La pregunta que cal fer-se és sota quines condicions es compleix la igualtat ja que, en aquest cas, només cal maximitzar la funció $g(\boldsymbol{\alpha})$ per obtenir la mateixa solució que al problema 3.8. La resposta és que la igualtat es dóna quan el problema és convex, com en el nostre cas, i a més a més es compleix la condició de Slater[2, 5.2.3]

$$\exists \mathbf{a} \in \mathbb{R}^p \ i \ \exists b \in \mathbb{R} \ : f_i(\mathbf{a}, b) < 0 \ \forall \ i \in 1, ..., n$$

Per definició, el nostre problema compleix la condició de Slater ja que el teorema de l'hiperplà separador ens assegura que existeix un hiperplà que separa els subespais C i D i, per tant, no passa per cap dels punts de la mostra de dades.

El teorema de Slater ens diu que si un problema és convex i compleix la condició de Slater llavors tenim dualitat forta, que vol dir que el màxim de la funció dual coincideix amb el valor òptim del problema inicial, i.e.

$$\max_{\boldsymbol{\alpha}\in\mathbb{R}^n} g(\boldsymbol{\alpha}) = p^*$$

Aquest resultat simplifica moltíssim el mètode de resolució del problema, ja que només cal que maximitzem la funció $g(\boldsymbol{\alpha})$, el que s'anomena problema dual.

Derivant L respecte $\boldsymbol{a}$ i b i igualant a zero, per tal de trobar els valors que el minimitzen, obtenim

$$\mathbf{a} - \sum_{i=1}^{n} \alpha_i y_i \mathbf{x_i} = 0 \tag{3.13}$$

$$\sum_{i=1}^{n} \alpha_i y_i = 0 \tag{3.14}$$

Substituint això al lagrangià obtenim

$$g(\boldsymbol{\alpha}) = \inf_{\mathbf{a}\in\mathbb{R}^p, b\in\mathbf{R}} L(\mathbf{a}, b, \boldsymbol{\alpha}) =$$

$$\frac{1}{2}\left(\sum_{i=1}^{n} \alpha_i y_i x_i\right)^2 + \sum_i \alpha_i \left(1 - y_i(\mathbf{a}\cdot\mathbf{x_i} - b)\right) =$$

$$\frac{1}{2}\left(\sum_{i=1}^{n} \alpha_i y_i x_i\right)^2 + \sum_i \alpha_i - \sum_{i=1}^{n} \alpha_i y_i \mathbf{a}\cdot\mathbf{x_i} + \sum_i \alpha_i y_i b =$$

$$\sum_i \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j \mathbf{x_i}\cdot\mathbf{x_j} \tag{3.15}$$

La qual cosa ens porta al següent problema dual

$$\begin{aligned} \textbf{maximitzar} \quad & g(\boldsymbol{\alpha}) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j \mathbf{x_i}\cdot\mathbf{x_j} \\ \textbf{subjecte a} \quad & \sum_i \alpha_i y_i = 0 \quad i \quad \alpha_i \geq 0 \end{aligned} \tag{3.16}$$

que és més fàcil de resoldre que el problema principal.

Observació 3.1.1. El vector normal $\mathbf{a}$ solució del problema 3.8 es pot escriure com a combinació lineal dels punts de la mostra que estan continguts als hiperplans separadors obtinguts.

Demostració: Siguin $\mathbf{a}^*$ i b^* valors òptims del problema 3.8 i $\boldsymbol{\alpha^*}$ el corresponent valor òptim dual, llavors

$$p^* = f_0(\mathbf{a}^*) = g_0(\boldsymbol{\alpha}^*) = min_{\mathbf{a}\in\mathbb{R}^p, b\in\mathbb{R}} \left(\frac{1}{2}\|\mathbf{a}\|^2 + \sum_{i=1}^{n} \alpha_i^* \left[1 - y_i(\mathbf{a}^T\cdot\mathbf{x_i} - b)\right]\right)$$

$$\leq f_0(\mathbf{a}^*) + \sum_{i=1}^{n} \alpha_i^* \left[1 - y_i(\mathbf{a}^{*T}\cdot\mathbf{x_i} - b^*)\right] \tag{3.17}$$

$$\leq f_0(\mathbf{a}^*) \tag{3.18}$$

La primera desigualtat és deguda a que l'ínfim del Lagrangià respecte $\mathbf{a}$ és més petit o igual al seu valor quan $\mathbf{a} = \mathbf{a}^*$. I la darrera desigualtat es dedueix de les condicions imposades pel problema, i.e.

$$\alpha_i^* \geq 0 \tag{3.19}$$

$$1 - y_i(\mathbf{a}^{*T} \cdot \mathbf{x_i} - b) \leq 0 \tag{3.20}$$

De les desigualtats 3.17 i 3.18 es dedueix que

$$\sum_{i=1}^{n} \alpha_i^* \left[1 - y_i(\mathbf{a}^{*T} \cdot \mathbf{x_i} - b)\right] = 0 \tag{3.21}$$

I com que cada terme de la suma anterior és no positiu llavors tenim que

$$\alpha_i^* \left[1 - y_i(\mathbf{a}^{*T} \cdot \mathbf{x_i} - b)\right] = 0 \tag{3.22}$$

El que s'anomena "*complementary slackness*". La conseqüència immediata d'aquesta igualtat és que si $1 - y_i(\mathbf{a}^{*T} \cdot \mathbf{x_i} - b) < 0$ llavors necessàriament $\alpha_i = 0$. Per tant, només aquells punts que facin que la desigualtat anterior sigui una igualtat (i.e. els punts que pertanyen als hiperplans de separació) tindran un corresponent valor no nul α_i a l'hora d'escriure la solució $\mathbf{a}$ tal i com indica l'equació 3.13 □

3.1.2 SVM amb marge suau

Per molt bé que optimitzem el marge, el nostre problema real sempre tindrà punts que el violin. Per tant la solució al problema haurà de posar en una balança tant la quantitat de punts que violen el marge de separació com la reducció d'aquest per tal de minimitzar aquells. S'haurà de buscar doncs un equilibri entre ambdós efectes a l'hora d'optimitzar el problema.

La representació d'aquesta flexibilitat del marge de separació es fa mitjançant la introducció de les anomenades variables *slack* ξ_i, que permeten augmentar o disminuir el marge en funció del punt de la mostra. Dit d'una altra manera, aquestes variables fan possible la violació del marge establert per a certes variables, i.e.

$$1 - y_i(\mathbf{a} \cdot \mathbf{x_i} - b) + \xi_i \leq 0$$

on, per coherència amb la definició convexa del problema 3.8, hem escollit $\xi_i \leq 0$. De manera que el problema d'optimització ens quedaria

$$\begin{aligned} \textbf{minimitzar} \quad & \frac{1}{2}\|\mathbf{a}\|^2 - C\sum_i \xi_i \\ \textbf{subjecte a} \quad & 1 - y_i(\mathbf{a}\cdot\mathbf{x_i} - b) + \xi_i \leq 0 \\ & \xi_i \leq 0 \end{aligned} \tag{3.23}$$

L'objectiu és que els ξ_i siguin el més propers a zero possible, essent $\xi_i = 0$ per a aquells punts que no estan dintre del marge definit pels hiperplans, és a dir aquells que no violen el marge. C és una constant que controla l'equilibri entre les violacions de marge i l'amplada d'aquest.

El corresponent Lagrangià per al problema 3.23 és

$$L(\mathbf{a}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \mathbf{r}) = \frac{1}{2}\|\mathbf{a}\|^2 - C\sum_i \xi_i + \sum_i \alpha_i \left[1 - y_i(\mathbf{a}\cdot\mathbf{x_i} - b) + \xi_i\right] + \sum_i r_i\xi_i \tag{3.24}$$

amb $\alpha_i, r_i \geq 0$.

Derivant respecte $\mathbf{a}$, ξ_i, b i igualant a zero obtenim

$$\frac{\partial L(\mathbf{a}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \mathbf{r})}{\partial \mathbf{a}} = \mathbf{a} - \sum_i y_i\alpha_i\mathbf{x_i} = 0 \tag{3.25}$$

$$\frac{\partial L(\mathbf{a}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \mathbf{r})}{\partial \xi_\mathbf{i}} = -C + \alpha_i + r_i = 0 \tag{3.26}$$

$$\frac{\partial L(\mathbf{a}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \mathbf{r})}{\partial \mathbf{b}} = \sum_i y_i\alpha_i = 0 \tag{3.27}$$

Substituïnt a 3.24 obtenim la següent funció dual associada al problema 3.23

$$g(\boldsymbol{\alpha}) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i\alpha_j y_i y_j \mathbf{x_i}\cdot\mathbf{x_j}$$

que és idèntica a la funció objectiu del problema 3.16. La diferència rau en que els coeficients α_i estan acotats per la constant C. Això és degut a la condicó 3.26 que, juntament amb $r_i \geq 0$, fan que $\alpha_i \leq C$.

D'altra banda, $\xi_i \neq 0$ només si $r_i = 0$ i, llavors, $\alpha_i = C$.

Per tant, la solució al problema 3.23 vindrà donada per la solució al següent problema dual

$$\begin{aligned} \textbf{maximitzar} \quad & g_0(\alpha) = \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \mathbf{x_i} \cdot \mathbf{x_j} \\ \textbf{subjecte a} \quad & \sum_i y_i \alpha_i = 0 \\ & 0 \leq \alpha_i \leq C \end{aligned} \tag{3.28}$$

A partir de la solució al problema 3.28 trobem el vector normal solució que, segons l'equació 3.25, es pot escriure com a combinació lineal dels vectors de suport, i.e.

$$\mathbf{a} = \sum_i y_i \alpha_i \mathbf{x_i}$$

i juntament amb el valor de b obtenim l'hiperplà solució $\mathbf{a} \cdot \mathbf{x} - b = 0$.

Aleshores, diem que un punt qualsevol $\mathbf{x} \in \mathbb{R}^p$ pertany a una classe o una altra en funció de si està a un costat o a l'altre de l'hiperplà solució. Expressant-ho formalment, la funció de decisió és

$$f(\mathbf{x}) = sgn\left(\mathbf{a} \cdot \mathbf{x} - b\right) = sgn\left(\sum_i y_i \alpha_i \mathbf{x_i} \cdot \mathbf{x} - b\right) \tag{3.29}$$

3.2 El mètode del kernel

3.2.1 Explicació i motivació del mètode

Al capítol anterior hem suposat que el nostre conjunt de dues classes de dades viu en un espai vectorial euclidià, en el sentit que qualsevol dada es pot identificar amb un punt de $\mathbb{R}^p$ que està a una certa distància, en el sentit euclidià del terme, d'un cert hiperplà, la qual cosa ens permet separar l'espai en dues classes diferents de punts. En un problema real però, la majoria de les vegades aquesta suposició no és correcta, és a dir que no podem establir un relació d'equivalència que ens permeti assegurar que tots els punts a un costat d'un cert hiperplà són de la mateixa classe. Dit d'una altra forma, la distància euclidiana perd el sentit en aquest cas i no la podem fer servir per classificar els nostres punts. La figura 4 exemplifica la impossibilitat de separar punts de classes diferents amb un hiperplà per a un problema real.

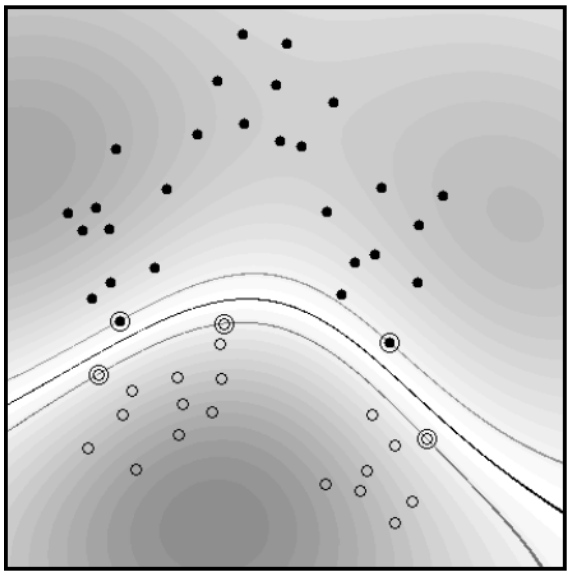


Figura 4: Exemple d'un problema real on no és possible la classificació lineal a $\mathbb{R}^2$. Les boles negres i blanques representen els dos tipus de dades.[4]

Tanmateix, el que sí podem fer és aplicar una certa transformació sobre les nostres dades a un altre espai, on sí que es pugui aplicar una classificació lineal. La figura 5 il·lustra aquesta idea.

Seguint la mateixa notació que al capítol anterior, anomenem X al nostre conjunt de dades i F l'espai d'arribada de la transformació esmentada. Llavors tenim en general

$$\phi \ : \ X \longrightarrow F \tag{3.30}$$

on F se sol anomenar *espai característic*, que no cal que tingui la mateixa dimensió que X.

Definició 1. *Un kernel és una funció K tal que, per a qualsevol $\mathbf{x}, \mathbf{z} \in X$*

$$K(\mathbf{x}, \mathbf{z}) = \langle \phi(\mathbf{x}), \phi(\mathbf{z}) \rangle$$

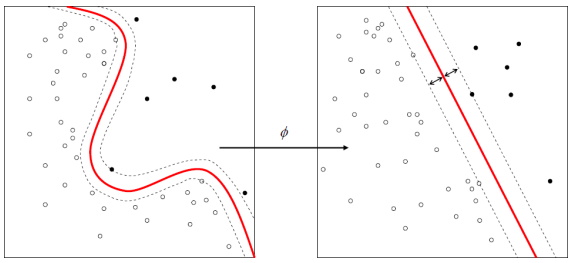


Figura 5: Il·lustració d'una transformació de l'espai de dades a un espai on la classificació lineal sí que és possible.[5]

on $\langle \cdot, \cdot \rangle$ *representa un producte escalar a* F*.*

3.2.2 Justificació del mètode i construcció de kernels

La següent proposició es dedueix trivialment de la definició 1

Proposició 3.2.1. *Sigui* $K : X \times X \longrightarrow \mathbb{R}$ *una forma bilineal simètrica, llavors* $K(\mathbf{x}, \mathbf{z})$ *és un kernel si, i només si:*

(i) la matriu $(\langle \phi(x_i), \phi(x_j) \rangle)_{i,j=1}^{n}$ *és semi definida positiva[6, Proposició 3.5] (**Cas** $\dim(\mathbf{F}) < \infty$)*

(ii) $\langle f(\mathbf{x}) \cdot f(\mathbf{z}) \rangle \geq 0 \ \forall f \in F$ *(**Cas** $\dim(\mathbf{F}) = \infty$)*

En el cas en que la dimensió de l'espai característic és infinita i suposant que $F = L^2(X)$, el teorema de Mercer[6, Teorema 3.6] ens dóna una condició necessària i suficient per a que una funció contínua i simètrica $K(\mathbf{x}, \mathbf{z})$ sigui un kernel, i.e. compleixi la proposició 3.2.1. Aquesta condició és que

$$K(\mathbf{x}, \mathbf{z}) = \sum_{i=1}^{\infty} \lambda_i \phi_i(\mathbf{x}) \phi_i(\mathbf{z}) \tag{3.31}$$

amb $\lambda_i \geq 0$ i $\phi_i \in L^2(X)$.

Una família molt important de kernels són els RBF[2] kernels. El següent teorema enuncia una condició necessària i suficient per a que un tipus de funció RBF compleixi la condició del teorema de Mercer i, per tant, sigui un kernel.

Teorema 3.2.1. $K(\mathbf{x}, \mathbf{z}) = K(\langle \mathbf{x}, \mathbf{z} \rangle)$ *és semi definit positiu si, i només si, tots el coeficients del seu desenvolupament en sèrie de Taylor són no negatius[7, Teorema 4.19]*

A continuació exposem un exemple de funció de base radial que és d'especial importància per al cas que ens ocuparà en la implementació pràctica del treball.

[2]Una funció de base radial (RBF) és una funció real del tipus $K(\mathbf{x}, \mathbf{z}) = K(\|\mathbf{x} - \mathbf{z}\|)$

Exemple 3.2.1. $K(\mathbf{x},\mathbf{z}) = e^{-\frac{\|\mathbf{x}-\mathbf{z}\|^2}{\sigma^2}}$ admet la següent expansió

$$K(\mathbf{x},\mathbf{z}) = e\left(-\frac{\|\mathbf{x}-\mathbf{z}\|^2}{\sigma^2}\right) = e-\frac{\|\mathbf{x}\|^2}{\sigma^2}e-\frac{\|\mathbf{z}\|^2}{\sigma^2}\sum_{k=0}^{\infty}\frac{(\frac{2}{\sigma^2})^k}{k!}\sum_{|\alpha|=k}C_\alpha^k\|\mathbf{x}\|^\alpha\|\mathbf{z}\|^\alpha$$

on $C_\alpha^k = \frac{k!}{\alpha_1!\ldots\alpha_n!}$. Com que tots els coeficients d'aquest desenvolupament són no negatius, el teorema 3.2.1 ens assegura que K és un kernel.

Per a cada kernel existeix doncs una funció $\phi : X \longrightarrow F$, tal que F és un espai vectorial euclidià. De manera que l'existència d'un kernel ens permet transformar el nostre espai de dades inicial en un altre on sí que podem classificar les dades linealment. Estrictament parlant, el que estem classificant són les imatges, segons la funció ϕ, de les nostres dades. Anàlogament a 3.28 el problema a maximitzar, donat un conjunt de dades $\mathbf{x_i} \in X$, serà

$$\begin{aligned} \textbf{maximitzar} \quad & g_0(\alpha) = \sum_i \alpha_i - \frac{1}{2}\sum_{i,j}\alpha_i\alpha_j y_i y_j K(\mathbf{x_i},\mathbf{x_j}) \\ \textbf{subjecte a} \quad & \sum_i y_i\alpha_i = 0 \\ & 0 \leq \alpha_i \leq C \end{aligned} \tag{3.32}$$

La funció de decisió serà, en aquest cas

$$f(\mathbf{x}) = sgn\left(\sum_i y_i\alpha_i K(\mathbf{x_i},\mathbf{x}) - b\right) \tag{3.33}$$

4 Anàlisi de Fourier

4.1 Una mica d'història

El cas estacionari[3] en l'estudi de la difusió de la calor en un disc metàl·lic es pot modelar, en coordenades polars, mitjançant la següent equació de la calor

$$r^2\frac{\partial^2 u}{\partial r^2} + r\frac{\partial u}{\partial r} = -\frac{\partial^2 u}{\partial \theta^2} \tag{4.1}$$

on u, r, θ representen la temperatura, la coordenada radial i la coordenada angular respectivament. Mitjançant el mètode de variables separades i per linealitat de la solució, s'arriba trivialment a la següent solució general[8, 1.2]

$$u(r,\theta) = \sum_{m=-\infty}^{\infty} a_m r^{|m|} e^{im\theta} \tag{4.2}$$

Pel cas $r = 1$ obtenim doncs la següent funció

$$u(1,\theta) = \sum_{m=-\infty}^{\infty} a_m e^{im\theta} \equiv f(\theta) \tag{4.3}$$

A la vista de la solució 4.3 i degut a que la funció temperatura a la vora del disc pot ser, a priori, qualsevol funció f, Joseph Fourier va postular[4] que, donada una funció $f(\theta)$ tal que $f(0) = f(2\pi)$, existeixen coeficients a_m tals que

$$f(\theta) = \sum_{m=-\infty}^{\infty} a_m e^{im\theta} \tag{4.4}$$

Aquesta suma s'anomena sèrie de Fourier de la funció f i la convergència de la respectiva successió de sumes parcials cap a f és l'objecte principal d'estudi de l'anàlisi harmònica. La convergència no té per què donar-se en general, però en cas afirmatiu l'expressió 4.4 ens està dient que qualsevol funció $f(\theta)$ periòdica es pot expressar com una suma infinita de sinus i cosinus. Si multipliquem a banda i banda de l'expressió per $e^{-in\theta}$ i integrem respecte θ llavors tenim

$$\begin{aligned}\int_{-\pi}^{\pi} f(\theta)e^{-in\theta}d\theta &= \int_{-\pi}^{\pi}\left(\sum_{m=-\infty}^{\infty} a_m e^{im\theta}\right)e^{-in\theta} \\ &= \sum_{m=-\infty}^{m=\infty} a_m \int_{-\pi}^{\pi} e^{im\theta}e^{-in\theta} = 2\pi a_m\end{aligned} \tag{4.5}$$

i per tant

[3] $\frac{\partial u}{\partial t} = 0$

[4] La demostració complerta de quan es pot fer aquesta afirmació va arribar anys més tard.

$$a_m = \frac{1}{2\pi} \int_{-\pi}^{\pi} f(\theta) \exp^{-im\theta} \, d\theta \tag{4.6}$$

Els $\{a_m\}$ s'anomenen coeficients de la sèrie de fourier de f o bé coeficients espectrals. Si ens fixem en l'integrand de l'expressió 4.6 observem que si la funció f no oscil·la a una freqüència similar a la part real de l'exponencial, el resultat de la integral és despreciable. Per tant el coeficient a_m ens dóna informació sobre la tendència de f a oscil·lar amb la freqüència donada per l'exponencial corresponent.

Notem que el nom coeficient espectral deriva de problemes com la descomposició espectroscòpica de la llum en línies espectrals. Així, el valor de qualsevol terme de la sèrie de fourier d'una ona electromagnètica representa la quantitat d'energia lumínica a la freqüència corresponent a aquest terme. Dit d'una altra forma, l'ona electromagnètica es pot descriure com un tren d'ones, cadascuna de les quals amb una energia d'acord amb el valor del coeficient espectral que oscil·la a la mateixa freqüència.

4.2 Sèrie de fourier discreta i DFT

Suposem ara que tenim un senyal discret i periòdic representat per la seqüència $x[n]$. És a dir que existeix un nombre natural N de manera que

$$x[n] = x[n + N]$$

Per analogia amb l'equació 4.6 podem pensar que existeixen coeficients a_k tals que

$$x[n] = \sum_{k \in \mathbb{N}} a_k e^{ik(2\pi/N)n} \tag{4.7}$$

La diferència entre aquell cas i aquest és que els termes de la seqüència 4.7 són finits, doncs només hi ha N exponencials diferents. Per tant, 4.7 representa un sistema de N equacions per a N coeficients desconeguts a_k. D'una manera similar al procés seguit en el cas continu és possible obtenir aquests coeficients en funció dels valors $x[n]$. Llavors la sèrie de fourier discreta i els respectius coeficients vindran donats per les següents equacions[9, 5.2.2]

$$x[n] = \sum_{k \in 0,\ldots N-1} a_k e^{ik(2\pi/N)n} \tag{4.8}$$

$$a_k = \frac{1}{N} \sum_{n \in 0,\ldots N-1} x[n] e^{-ik(2\pi/N)n} \tag{4.9}$$

anomenades equacions de síntesi i anàlisi, respectivament.

Una de les tècniques que es deriva de l'anàlisi discret de fourier és l'anomenada Transformada de Fourier Discreta (DFT) per a un senyal complex de duració finita

$x[n]$. Aquesta tècnica consisteix en calcular els coeficients de la sèrie de fourier d'un senyal periòdic $\tilde{x}[n]$ que és igual a $x[n]$ sobre un període. Això és, si l'enter N_1 és tal que

$$x[n] = 0 \quad \text{fora de l'interval } 0 \leq n \leq N_1 - 1$$

llavors existeix $N \geq N_1$ de manera que $\tilde{x}[n]$ és periòdic amb periode N complint que

$$\tilde{x}[n] = x[n] \quad 0 \leq n \leq N - 1$$

Definició 2. *Es defineix la DFT de $x[n]$ com el conjunt de coeficients de la sèrie de fourier de $\tilde{x}[n]$ restringida a l'interval $0 \leq n \leq N$ i s'escriu:*

$$\tilde{X}(k) = \frac{1}{N} \sum_{n=0}^{N-1} x[n] e^{-ik(2\pi/N)n}, \quad k = 0, 1, ..., N-1 \tag{4.10}$$

4.3 La Transformada Ràpida de Fourier (FFT)

El càlcul directe de 4.10 suposa N^2 operacions i, per tant, té un cost computacional de $O(n^2)$. Una FFT és qualsevol algorisme de tipus *divideix i venceràs* per a calcular la DFT, amb un cost computacional de $O(n \log n)$. El més antic que es coneix va ser descrit per Karl Friedrich Gauss, per a la interpolació de l'òrbita dels asteroides *Pallas* i *Juno* a partir de les observacions realitzades. No obstant, l'algorisme FFT més conegut i usat és el proposat per James W. Cooley i John W. Tukey en un article l'any 1965[10], basat en el fet que, quan el tamany de la mostra de dades és un nombre compost, és a dir $N = r_1 \cdot ... \cdot r_m$, podem subdividir el problema i expressar 4.10 com una suma de múltiples sèries de fourier, cadascuna de tamany r_i, amb $i \in \{1, ..., m\}$. D'aquesta forma, el cost computacional total és la suma dels costos de les DFT en què hem dividit el problema, això és $Nr_1 + ... + Nr_m = N(r_1 + ... + r_m)$. Concretament, l'article se centra en el cas que $N = 2^m$ i, per tant, el cost és $2N \log_2 N$. A més a més, explica l'avantatge d'usar $N = 2^m$, a l'hora de resoldre l'algorisme mitjançant un aparell que en aquella època es va començar a posar de moda entre els científics, l'ordinador, doncs la suma 4.10 es converteix en una suma de zeros i uns, que són els valors de la respectiva representació binària de k i n. D'aquesta manera, la resolució del problema és molt eficient, ja que tot el càlcul es pot encabir en el mateix *array* d'entrada $x[n]$.

5 Aplicació a un problema d'anàlisi d'àudio

5.1 Descripció

El timbre d'un instrument musical és el que caracteritza el seu so. Físicament parlant s'obté calculant l'espectre d'un senyal acústic de l'instrument. Per tant ens dóna informació de la magnitud amb què apareixen les diferents freqüències. Així, dos instruments que estan tocant la mateixa nota sonen diferent perquè cadascun d'ells també està fent sonar altres freqüències en major o menor magnitud.

Amb l'objectiu de poder classificar dos instruments musicals a partir del seu timbre utilitzem un algorisme FFT per tal de trobar la DFT de senyals acústics dels instruments. Cada espectre representarà un vector característic de l'espai X descrit al capítol 3, la dimensió del qual serà igual al nombre de freqüències diferents que tingui la DFT usada. Amb una mostra d'espectres dels dos instruments entrenem una màquina de vectors de suport amb un kernel RBF per tal d'obtenir un hiperplà de separació que ens permeti classificar nous espectres d'aquests instruments.

5.2 Metodologia

Per a l'anàlisi d'àudio del treball hem escollit com a instruments musicals la flauta dolça i el saxofon. Per a la gravació dels àudios s'ha creat una aplicació de mòvil senzilla, que grava el so per blocs d'un tamany determinat d'enregistraments. L'únic paràmetre que hem d'escollir és el nombre d'enregistraments per segon que realitza el micròfon del mòbil. Seguidament l'aplicació desa aquest conjunt d'enregistraments en un fitxer.

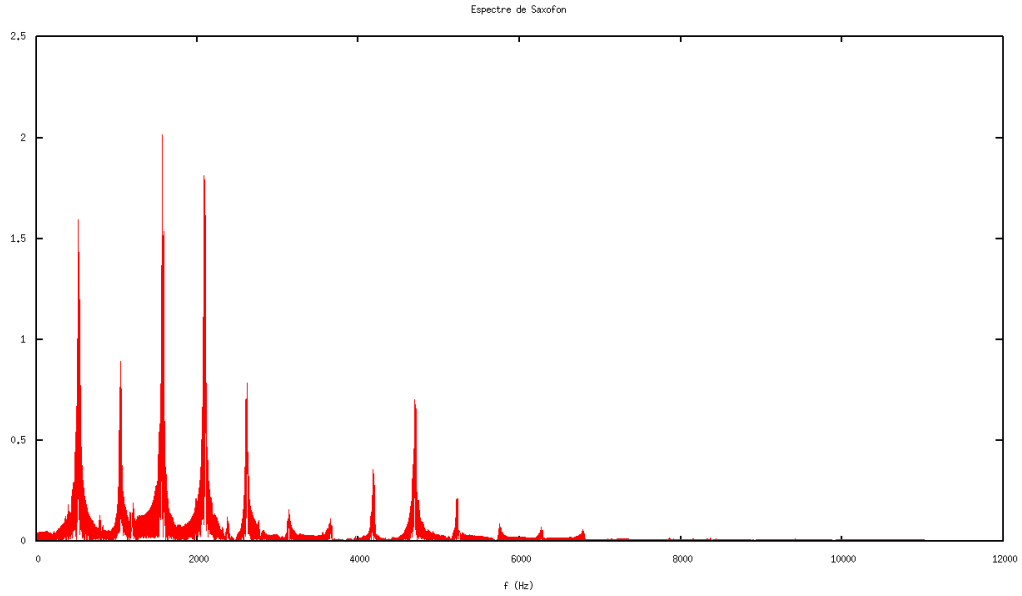
Un cop tenim gravat un fitxer d'àudio només cal calcular la DFT. En tractar-se d'un senyal real, a partir de l'equació 4.10 es dedueix que[11, Taula 8.2]

$$\tilde{X}[N-k] = \tilde{X}^*[k]$$

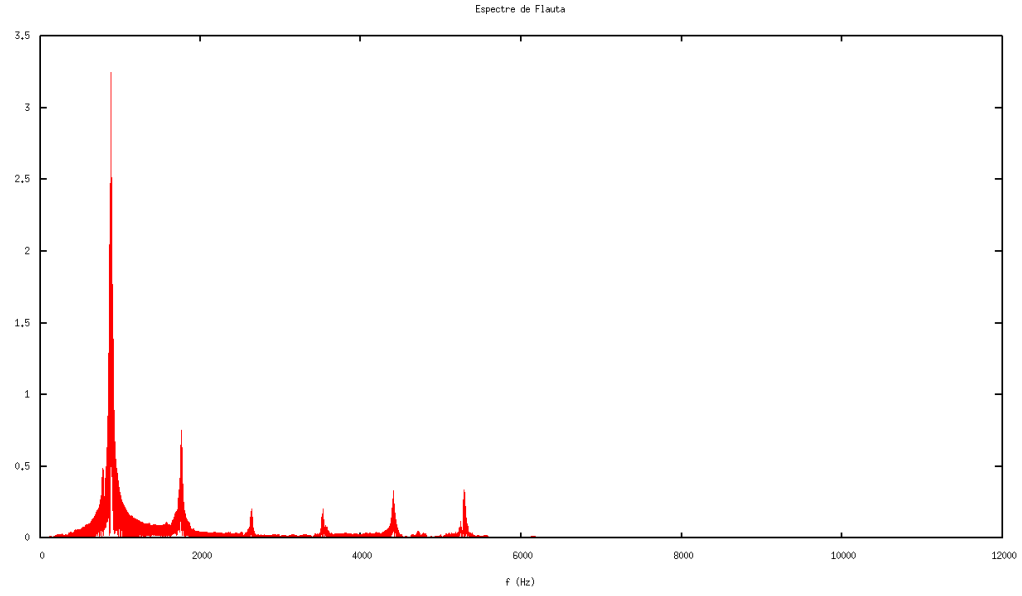
Així, si utilitzem un senyal real de mida N i construïm el corresponent senyal complexe per poder computar la DFT, obtindrem un senyal amb $N/2$ freqüències reals diferents. L'interval freqüencial Δf dependrà de la freqüència de gravació amb què hem gravat el senyal. Així, si F és la freqüència de gravació i N el nombre de punts del senyal discret llavors l'interval freqüencial serà

$$\Delta f = \frac{F}{N}$$

Per tal de computar la DFT hem usat les classes ja definides en java, FFT i Complex. La figura 6 mostra dos exemples d'espectres de saxofon i flauta obtinguts mitjançant FFT a partir de senyals d'àudio gravats a una freqüència de 22050 enregistraments/segon i prenent $N = 16384$.



(a) Exemple d'espectre d'un Saxofon.



(b) Exemple d'espectre d'una flauta.

Figura 6: Espectre exemple dels dos instruments usats per a l'anàlisi d'àudio.

Com es pot observar, el saxofon és més ric en armònics[5] que no pas la flauta. També s'observa que ambdós instruments gairebé no presenten armònics més enllà dels 5000 Hz. Així doncs, l'ús de freqüències de gravació superiors als 10000 enregistraments/segon és innecessari per a la construcció de la màquina de vectors de suport, ja que tots els valors superiors a 5000 Hz en els espectres serien pràcticament nuls i, per tant, no aportarien informació en l'aprenentatge del model alhora que augmentarien la complexitat computacional.

Hem calculat els espectres d'un conjunt d'àudios de cada instrument, per a diferents valors de N. Aquests espectres constitueixen els vectors característics del nostre espai mostral, el qual hem dividit en dos conjunts tal i com hem explicat al capítol 2, el conjunt d'entrenament i el de test. Hem entrenat una màquina de vectors de suport no lineal amb kernel RBF, per a diferents valors dels paràmetres C i $\gamma = \frac{1}{\sigma^2}$. Seguidament l'hem utilitzat per a calcular l'error en el test E_{out} corresponent a les mostres del conjunt test per tal de decidir quina és la dimensió més idònia per als vectors característics i quins són els valors dels paràmetres més adients. Cal remarcar que els errors obtinguts en aquest entrenament no són una bona estimació de l'error real que s'obtindria en un experiment repetit de classificació, ja que només hem usat una partició concreta del conjunt mostral. És a dir que si haguéssim usat uns altres espectres de la mostra per a l'entrenament i el test, ens hauria sortit un error diferent. Tanmateix, sí que ens serveix per decidir el tamany espectral que ens va bé per fer l'entrenament amb un conjunt més gran de particions i estimar d'una manera més fiable l'error de classificació en el test, mitjançant una validació creuada.

Un cop determinats quins són els paràmetres i la dimensió a utilitzar es procedeix a fer una validació creuada per tal d'estimar l'error E_{out} i alhora determinar quins seran els vectors de suport que usarem posteriorment en la funció de decisió. La construcció de la màquina de vectors de suport l'hem fet amb el llenguatge Python, que té una llibreria anomenada sklearn per a la resolució del problema d'optimització 3.32, mitjançant el paquet LIBSVM, que fa servir l'algorisme "Sequential Minimal Optimization"(SMO)[12]. De la solució al problema d'optimització ens interessen els vectors de suport i el que s'anomenen coeficients duals, que no és res més que el producte de cada $\alpha_i y_i$ a l'equació 3.33. D'aquesta manera podrem construir la funció de decisió.

Per entendre millor això que hem explicat, a continuació mostrem el codi en Python[6] de l'entrenament i testeig per a una mostra de 100 espectres de cada instrument, amb 85 per a entrenament i 15 per al test.

[5]Un armònic és qualsevol múltiple d'una freqüència determinada.

[6]Python, usat com a llenguatge de programació estructurada, està considerat pseudocodi, en el sentit que qualsevol persona amb coneixements d'algorísmica pot seguir la lògica del que s'hi programa.

```
def entrenador():

    X = []

    Yneg = [-1 for i in range(85)]
    Ypos = [1 for i in range(85)]

    Y = Yneg + Ypos
    mida = 512 # Mida de l'espectre: dimensió dels vectors característics.
               # Es pot canviar per fer un entrenament amb una altra mida
               # dels vectors.

    # Lectura dels espectres d'entrenament per a cada instrument
    for i in range(1,86):
        pfile=open('espectreSaxo'+ str(i) + '.txt','r')
        data=pfile.readlines()
        data = [each.replace('\n', '') for each in data]
        data=map(float,data)
        pfile.close()

        dataEspectre=data[:mida]
        for j in range(mida):
            dataEspectre[j] = dataEspectre[j]/max(dataEspectre)
        X.append(dataEspectre)

    for i in range(1,86):
        pfile=open('espectreFlauta'+ str(i) + '.txt','r')
        data=pfile.readlines()
        data = [each.replace('\n', '') for each in data]
        data=map(float,data)
        pfile.close()

        dataEspectre=data[:mida]
        for j in range(mida):
            dataEspectre[j] = dataEspectre[j]/max(dataEspectre)
        X.append(dataEspectre)


    #0 < C < 1  a intervals d'iteració de 0,1
    #0 < gamma < 0,5 a intervals d'iteració de 0,05

    coordenadesMax = [1,1] #Les coordenades són els valors pels
                           #quals multiplicarem els intervals d'iteració
                           #per trobar els paràmetres C i gamma que
                           #donen menys error de classificació en el test
    maxEncerts = 0 #comptador per al cas de màxim encert

    for i in range(1,11):
        for j in range(1,11):
```

```
            clf = svm.SVC(C=i*0.1,kernel='rbf',gamma=j*0.05) # SVM
            clf.fit(X,Y) #Entrenament per a la mostra (X,Y)
            encerts = 0  #Comptador d'encerts en la fase test
            for k in range(86,101):
                pfile=open('espectreSaxo'+ str(k) + '.txt','r')
                data = pfile.readlines()
                data = [each.replace('\n', '') for each in data]
                data=map(float,data)
                pfile.close()
                dataEspectre=data[:mida]
                for l in range(mida):
                    dataEspectre[l] = dataEspectre[l]/max(dataEspectre)
                if clf.predict(dataEspectre) == -1:
                    encerts+=1

                pfile=open('espectreFlauta'+ str(k) + '.txt','r')
                data = pfile.readlines()
                data = [each.replace('\n', '') for each in data]
                data=map(float,data)
                pfile.close()
                dataEspectre=data[:mida]
                for l in range(mida):
                    dataEspectre[l] = dataEspectre[l]/max(dataEspectre)
                if clf.predict(dataEspectre) == 1:
                    encerts+=1
            if(encerts > maxEncerts): # Millors valors de C i gamma
                coordenadesMax[0] = i
                coordenadesMax[1] = j
                maxEncerts = encerts

    print coordenadesMax
    print maxEncerts
```

El codi per a la validació creuada és anàleg, però amb els valors dels paràmetres ja fixats i repetint l'entrenament i test tantes vegades com particions del conjunt mostral tinguem.

6 Resultats

Hem enregistrat 100 mostres d'àudio de cada instrument a dues freqüències diferents, concretament a 8000 enregistraments/segon i a 22050 enregistraments/segon, per constatar que no s'observen canvis significatius en els resultats quan usem una freqüència d'enregistrament superior als 10000 enregistraments/segon. Tal i com hem comentat al capítol anterior, hem utilitzat una màquina de vectors de suport amb kenel RBF. Hem fet una partició de les 100 mostres de cada instrument en 85 per a l'entrenament i 15 per al test. Hem usat només una partició per a aquest primer entrenament ja que només ens serveix per decidir la dimensió dels vectors caraterístics que utilitzarem i els paràmetres. De manera que tenim 170 vectors d'entrenament i 30 de test. A les taules següents mostrem els resultats obtinguts per a diferents quantitats de punts utilitzats a la FFT i diferents valors dels paràmetres C i γ. La darrera columna mostra l'error E_{out}.

N	C	γ	$E_{out}(\%)$
512	0.2	0.12	7
1024	0.4	0.43	7
2048	0.4	0.09	7
4096	0.9	0.12	3

Taula 1: Taula de resultats de l'entrenament i percentatge de l'error en el test per a una freqüència de 8000 enregistraments/segon.

N	C	γ	$E_{out}(\%)$
512	0.8	0.22	27
1024	0.1	0.46	17
2048	0.1	0.28	17
4096	0.1	0.16	7
8192	0.2	0.11	3

Taula 2: Taula de resultats de l'entrenament i percentatge de l'error en el test per a una freqüència de 22050 enregistraments/segon.

En general s'observa que a mesura que augmentem el nombre de punts N per fer la FFT augmenta l'eficàcia del mètode, és a dir que augmenta el nombre d'encerts a l'hora de classificar l'instrument. Això té bastant de sentit, ja que en augmentar N augmenta la quantitat de freqüències que tenim en compte per fer l'entrenament i, per tant, disminueix la probabilitat de no tenir en compte determinats armònics presents a ambdós instruments. D'altra banda, observem que els resultats per a una freqüènciade gravació de 22050 Hz no aporten un millora en l'eficàcia, ans al contrari. Per tant, amb una freqüència de gravació de 8000 Hz és suficient.

Un cop constatat que 8000Hz és una bona freqüència d'enregistrament d'àudio, procedim a fer un estudi més fiable de l'error E_{out} per a cada valor de N.

Apliquem doncs una validació creuada de Monte Carlo amb 1000 particions diferents per a cada cas de la taula 1. La taula 3 mostra els resultats d'aquesta validació

N	C	γ	$E_{out}(\%)$	$I_{95\%}$
512	0.2	0.12	15	[3,27]
1024	0.4	0.43	14	[3,27]
2048	0.4	0.09	10	[0,23]
4096	0.9	0.12	9	[0, 20]

Taula 3: Taula de resultats de l'entrenament i percentatge d'error E_{out} per a cada valor de N mitjançant validació creuada amb 1000 particions. La darrera columna representa l'interval de confiança per a E_{out} al 95%

creuada.

Cal destacar que, estadísticament parlant, no sabem quina distribució segueix l'error E_{out} ja que el nostre espai mostral està compost per dades que no són independents, doncs per a cada mesura de l'error fem servir les mateixes dades. L'única cosa que canvia és la partició del conjunt en la part de dades per a l'entrenament i la part per al test. En aquest sentit, E_{out} és un estimador de l'error poblacional, del qual no en coneixem la distribució. Per tant, a l'hora de calcular l'interval de confiança, d'entre els 1000 errors obtinguts a l'experiment ordenats per ordre ascendent, ens hem limitat a acotar un rang que va del 25è al 975è, per tal d'obtenir un interval de confiança al 95%.

7 ClassificadorAPP

Com a cas pràctic de l'aprenentatge automàtic dut a terme en aquest treball, he construït una aplicació per a mòbils que sigui capaç de distingir entre el so d'una flauta i el d'un saxofon. He subdividit la tasca en dues etapes, primer he obtingut els vectors de suport i els respectius coeficients duals, que s'obtenen com a atributs de la variable d'entrenament del programa escrit en Python que he descrit al capítol 5, i els he desat en dos fitxers de dades a una carpeta de l'APP, que els carrega en iniciar-se. La següent etapa ha consistit pròpiament en el disseny i desenvolupament de l'aplicació en llenguatge Java.

7.1 Breu introducció a Android

Les aplicacions Android estan escrites en llenguatge JAVA. Dit d'una altra forma, existeixen un tipus concret de classes JAVA que proporcionen una interfície d'usuari (IU) per a la interacció amb els usuaris. La varietat i tipus d'aquestes és molt extens i variat, però tot seguit val la pena destacar-ne un parell que són els que hem usat per a la implementació de la nostra aplicació.

- **Activity** Una activity és una classe que permet a l'usuari interaccionar amb l'aplicació per a realitzar una tasca concreta. El concepte d'activity se sol associar al de pantalla, però no té per què ser necessàriament cert, en el sentit que el concepte de pantalla el reservem per a un estat determinat en el qual es troba una activity i, per tant, una activity pot tenir més d'una pantalla associada. En resum, en el moment en que es crea un objecte de tipus activity, aquest va evolucionant i pot donar lloc a diferents configuracions de la pantalla, és a dir a diferents pantalles. En aquest sentit, la classe activity té un seguit de mètodes que conformen el seu cicle de vida. Les activities seran creades, resumides, pausades i destruides.
- **Intent** Un intent és una classe que permet la interacció entre activities. Dit d'una manera més formal, és una classe que ens permet especificar quina activity ha de ser executada i amb quines variables.

7.2 Disseny de l'APP

La figura 7 mostra les diferents pantalles que configuren l'APP. A la pantalla principal podem observar el botó de gravació a l'esquerra i el de pausa a la dreta. Quan pitgem el botó de gravació s'inicia l'enregistrament de so per blocs. Paral·lelament, l'aplicació va calculant els espectres d'aquests blocs i els mostra per pantalla, com podem observar a la segona pantalla. Quan pitgem el botó de pausa aturem l'enregistrament i comença la classificació del so. Finalment anem a la pantalla del resultat, la qual ens mostra el tipus de so que hem gravat.

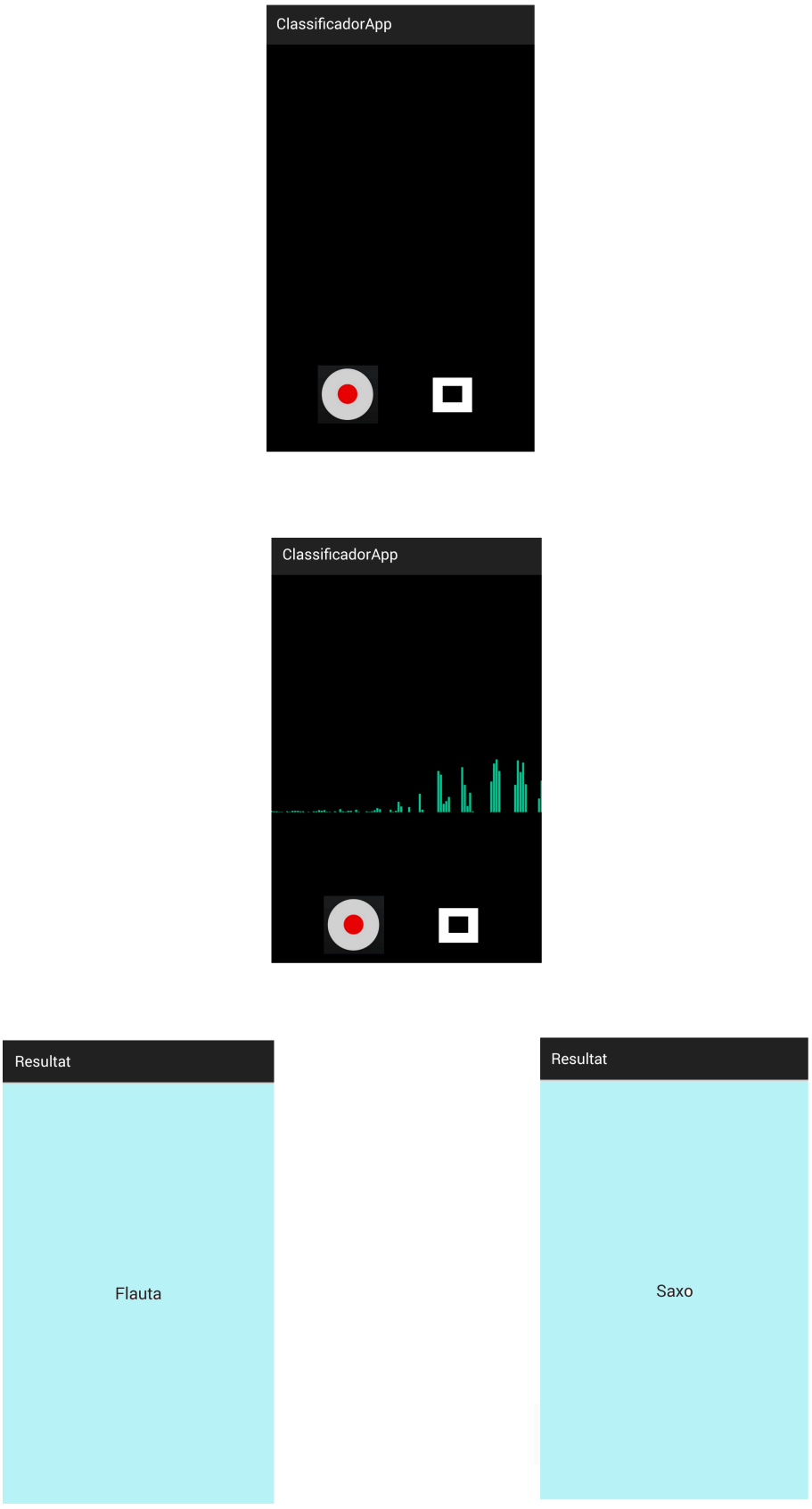


Figura 7: Navegació de l'APP: conjunt de pantalles per ordre lògic del procés pel qual està pensada. És a dir, enregistrar un so, classificar-lo i mostrar el resultat.

7.3 Procés d'implementació de l'APP

El ClassificadorAPP consta de sis classes, que citem i expliquem a continuació:

- **MainActivity** És l'activity que es crea en iniciar l'aplicació.
- **Complex** És una classe per manipular nombres complexos.
- **FourierManager** És una classe que conté mètodes de la classe FFT.java, que ens permet calcular l'espectre d'un senyal discret.
- **Gravacio** Classe que hereta de la classe AsyncTask, que ens permet gestionar un procés asíncron[7].
- **Classificació** És la classe que implementa tots els mètodes de classificació del so.
- **Resultat** És l'activity on mostrem el resultat de la classificació.

A la figura 8 mostrem el diagrama de classes de l'aplicació.

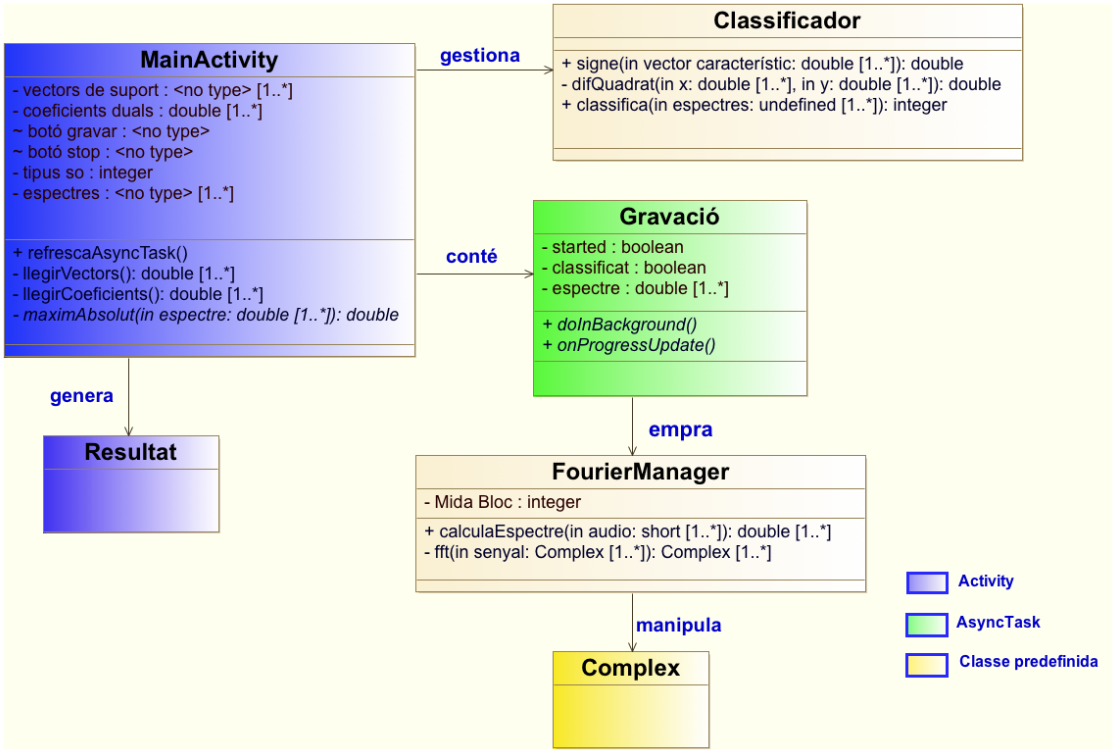


Figura 8: Diagrama de classes de l'aplicació.

De totes aquestes classes mereixen epecial dedicació i descripció les que involucren el procés de gravació del so i visualització dels respectius espectres. Abans, però, val la pena recordar el següent concepte

[7]Un procés asíncron és una forma de processament de dades input/output entre el sistema i l'exterior, que permet l'execució en paral·lel d'un altre procés.

Definició 3. *Un fil d'execució és la unitat més petita de processament que pot ser programada pels sistemes operatius. Des del punt de vista de la programació, un fil d'execució es defineix com una funció, l'execució de la qual es pot llançar en paral·lel amb d'altres. El fil primari correspon a la funció main.*

Qualsevol fil és un programa en execució que comparteix la imatge de memòria i d'altres informacions amb altres fils d'execució. És important destacar que tots els fils d'un mateix procés comparteixen el mateix espai d'adreces de memòria[13].

Per tot això els fils permeten a un procés executar diferents tasques al mateix temps. En el cas que ens ocupa necessitem enregistrar un so i paral·lelament volem anar calculant-ne els espectres i mostrar-los per pantalla. La classe AsyncTask és un tipus de classe que ens permet gestionar diferents processos en paral·lel i que han d'interactuar amb la interfície d'usuari durant un període curt de temps, idealment uns segons. És per això que és la classe idònia per dur a terme aquest procés asíncron sense que se'ns pengi l'aplicació.

8 Conclusions

Les tasques dutes a terme en aquest treball han estat, en un principi, una tasca pràctica de gravacions de diferents mostres d'àudio de dos intruments musicals, el saxofon i la flauta, i la posterior obtenció dels espectres. Paral·lelament s'ha anat entenent tota la teoria de l'aprenentatge automàtic i concretament les màquines de vectors de suport, per tal d'agrupar-ho tot en un model de classificació integrable en una aplicació Android.

Podem constatar que la màquina de vectors de suport és un mètode eficaç per a la classificació de sons. L'eficiència depèn dels instruments que vulguem classificar, doncs en augmentar el nombre d'armònics també es requereixen espectres de dimensió més elevada, cosa que augmenta la complexitat computacional. No obstant això, hem de tenir present que el nostre classificador només sap classificar el que li hem ensenyat a classificar, en el nostre cas la flauta i el saxofon i, per tant, si li mostrem un altre instrument ens dirà que és un dels dos que li hem ensenyat.

Val la pena destacar que aquest mètode es pot generalitzar per classificar n classes d'objectes diferents. Només cal subdividir el problema en n problemes de classificació binària on, per a cada problema, els objectes pertanyen a una classe o a la resta. Així, de manera iterativa anem subdividint l'espai característic en diferents regions que inclouen les diferents classes. En aquest sentit, com a línies futures de recerca es podria incloure una funcionalitat d'aprenentatge de nous instruments a l'aplicació, on l'usuari aniria enregistrant nous sons que pot desar a una base de dades amb l'etiqueta de l'instrument pertinent. D'aquesta forma, quan se'n disposes de suficients es podria entrenar una nova màquina de vectors de suport per al nou instrument. També es podria investigar si l'ús de filtratge per Fourier podria fer més eficaç el mètode.

Referències

[1] Wikimedia Commons. Random cross validation. `https://commons.wikimedia.org/wiki/File%3ARandom_cross_validation.jpg`, 2011. [Data accés: 16-01-2016].

[2] S. Boyd. *Convex Optimization.* Cambridge University Press., 2004.

[3] Catellet M.; Llerena I. *Àlgebra lineal i geometria.* Universitat Autònoma de Barcelona., 1990.

[4] Wikimedia Commons. Svm 10 perceptron. `https://commons.wikimedia.org/wiki/File:Svm_10_perceptron.JPG`. [Data accés: 16-01-2016].

[5] Wikimedia Commons. Kernel machine. `https://commons.wikimedia.org/wiki/File:Kernel_Machine.png`, 2011. [Data accés: 16-01-2016].

[6] Cristianini N.; Shawe-Taylor J. *Support vector machines and other kernel-based learning methods.* Cambridge University Press., 2000.

[7] Scholkopf A.; Smola B. *Learning with kernels.* Massachussetts Institute of Technology., 2002.

[8] Stein E. M.; Shakarchi R. *Fourier analysis: An introduction.* Princeton University Press., 2003.

[9] Oppenheim A. V.; Willsky A. S. *Signals and systems.* Prentice Hall., 1983.

[10] Cooley J. W.; Tukey J. W. An algorithm for the machine calculation of complex fourier series. *Mathematics of Computation.*, 19:297—301, 1965.

[11] Oppenheim A. V.; Schafer R. W.; Buck J. R. *Discrete-Time Signal Processing.* Pearson Education., 2006.

[12] Platt J. C. Sequential minimal optimization: A fast algorithm for training support vector machines. *Microsoft Research.*, 1998.

[13] Carretero J. *Sistemas operativos: una visión aplicada.* McGraw-Hill., 2007.