UNIVERSITÀ DEGLI STUDI DI FIRENZE
Facoltà di Ingegneria - Dipartimento di Sistemi e Informatica

Tesi di laurea in Ingegneria Informatica

Classificazione di Elementi Audio nei Video Sportivi

Candidato
Alessio Bazzica

Relatore
Prof. Alberto Del Bimbo

Correlatori
Ing. Giuseppe Serra

Ing. Marco Bertini

Ing. Lamberto Ballan

Anno Accademico 2007-2008

alla mia famiglia
a Iris
ai miei amici

"Essere ingegneri è una cosa, essere ingegnosi è un'altra."
Horacio Pagani

Indice

Elenco delle figure

Abbreviazioni

ACA	Audio Content Analysis
ASR	Automatic Speech Recognition
BP	Back-Propagation
DCT	Discrete Cosine Transform
DBN	Deep Belief Network
EM	Expectation Maximization
GMM	Gaussian Mixture Model
HMM	Hidden Markov Model
IR	Information Retrieval
KR	Knowledge Representation
MDL	Minimum Description Length
MFCC	Mel-Frequency Cepstral Coefficient
ML	Machine Learning
PCA	Principal Component Analysis
RBF	Radial-basis Function
RBM	Restricted Boltzmann Machine
S3VM	Semi-Supervised Support Vector Machine
SSL	Semi-Supervised Learning
SVM	Support Vector Machine

Prefazione

I recenti progressi di hardware e tecnologie per le telecomunicazioni hanno portato ad un rapido aumento della quantità di informazioni multimediali liberamente fruibili. L'utilità di queste risorse è in gran parte determinata dall'accessibilità dei contenuti così è diventato necessario studiare nuove soluzioni riguardanti la memorizzazione, la trasmissione, la personalizzazione, la ricerca, l'indicizzazione e il recupero.

Un mezzo comune per avere accesso ai contenuti multimediali sono i motori di ricerca: attualmente troviamo in commercio servizi quasi esclusivamente basati sulla sola analisi di informazioni testuali (ad esempio YouTube e Google Video basano la ricerca su tag e descrizioni testuali). Questi strumenti sono in grado di fornire una soluzione solo quando esistono informazioni testuali, limitatamente alla pertinenza del testo e dipendono fortemente dall'efficienza linguistica degli strumenti di analisi. Inoltre l'annotazione manuale, ovvero l'attività svolta dall'uomo di annotare occorrenze di concetti in un oggetto multimediale tramite informazioni testuali, nonostante possa garantire un elevato livello di astrazione, soffre di soggettività nelle descrizioni. Risulta quindi un'attività estremamente costosa in termini di tempo e crea problemi di interoperabilità.

Per superare questi ostacoli sono necessarie una descrizione e una più profonda comprensione delle informazioni a livello semantico e, dato il volume delle informazioni sulle quali è necessario effettuare annotazione, è indispensabile che i processi di recupero delle informazioni vengano effettuati in modo automatizzato oppure solo con il minimo intervento da parte dell'uomo. In-

oltre sarebbe opportuno definire un insieme di regole per l'annotazione che garantisca interoperabilità e riusabilità delle informazioni estratte.

Le limitazioni della sola annotazione testuale possono essere superate con l'analisi dei contenuti correlati (ad esempio audio o video). L'analisi è realizzabile affrontando il problema del *gap semantico*. Con tale termine viene indicata la "distanza" tra caratteristiche percettive di basso livello e le caratteristiche di alto livello semantico. I primi sforzi per la riduzione di questo gap si sono concentrati sull'estrazione di descrittori numerici quanto più rappresentativi e sulla definizione di parametri di somiglianza piuttosto che sull'emulazione del concetto umano di similarità. Ma la definizione di metriche e la progettazione di tecniche di segmentazione per i soli descrittori di basso livello non risulta sufficiente per poter estrarre informazioni semantiche nei media audiovisivi. In parallelo quindi sono state studiate soluzioni per derivare features più astratte e per definire, attraverso queste, concetti.

Si è presentata quindi la necessità di rappresentare la conoscenza e di trasformare il modo in cui le applicazioni multimediali garantiscono l'accessibilità ai contenuti. Tra le possibili rappresentazioni della conoscenza troviamo le ontologie le quali presentano una serie di vantaggi. I più importanti sono la possibilità di definire conoscenza in modo che risulti processabile dalla macchina, consentire la derivazione di nuova conoscenza in modo automatico attraverso l'inferenza e la possibilità di essere condivise allo scopo di fornire interoperabilità e riusabilità. In particolare per l'annotazione semantica, le ontologie si rivelano adatte per definire concetti e relazioni tra di essi estraibili dagli oggetti multimediali.

Un generico framework che realizzi un sistema di annotazione semantica per media audiovisivi dovrebbe permettere la costruzione e l'aggregazione di sottosistemi che analizzino i vari domini di informazione reperibili: per esempio video, audio, testo in sovraimpressione ed eventualmente anche metadati già presenti negli oggetti. L'obiettivo delle analisi è riconoscere occorrenze di concetti di alto livello basandosi, in linea generale, sulle informazioni estratte

da tutti i domini. Per individuare questi concetti alcuni sistemi elaborano gli oggetti multimediali a differenti livelli di astrazione: i primi livelli ovviamente utilizzeranno come input caratteristiche quasi esclusivamente fisiche. Per esempio, nel caso dell'audio, è ricorrente trovare un primo livello che dal segnale definito nel dominio del tempo estragga caratteristiche percettive definite nel dominio della frequenza: questa nuova informazione è ancora lontana dall'individuare un concetto target ma è necessaria per calcolare nuovi dati che possano offrire un'astrazione ulteriore.

In questo lavoro è stata esplorata l'analisi ristretta al solo dominio audio; in particolare sono state cercate soluzioni per la realizzazione di classificatori di contenuti audio utili a generare features di medio livello. Le informazioni prodotte dai classificatori costituiscono la base per la definizione di concetti di livello più alto rappresentabili, per esempio, tramite ontologie.

L'audio in formato digitale esiste da più di trenta anni; iniziando con l'implementazione di operazioni di base (quali la memorizzazione, la modifica e la trasmissione), successivamente le energie sono state investite nella progettazione di tecniche di compressione principalmente richieste nel campo della telefonia digitale, e nella realizzazione del riconoscimento vocale per i sistemi di sorveglianza. La prima applicazione pensata per operare un'estrazione semantica è identificabile nei sistemi ASR (Automatic Speech Recognition). Dopo l'avvento di internet, e in particolare con la disponibilità di mezzi di trasmissioni a banda larga, la distribuzione attraverso la rete di contenuti audio di elevata qualità si è estesa. In questo contesto sono nate nuove possibilità in particolare nel campo dell'analisi dei contenuti audio.

Nello studio di questi sistemi, denominati *Audio Content Analysis* (ACA), è sorta quindi la necessità di gestire in modo efficace la crescente collezione di dati e di migliorare l'interazione uomo-macchina. Questi sistemi, partendo da features di basso livello, solitamente generano features intermedie (o di medio livello). Per esempio estraggono i fonemi, la prosodia[1] nel ca-

[1]La prosodia è la parte della linguistica che studia l'intonazione, il ritmo, e l'accento

so del parlato o della melodia, armonia e struttura nel caso della musica. Come già illustrato per l'annotazione semantica in generale queste features sono indispensabili per individuare istanze di concetti più vicini al pensiero dell'uomo. Per assicurare interoperabilità, features di basso e medio livello possono essere rappresentate come metadati attraverso una formalizzazione standardizzata della sintassi (es. lo standard MPEG-7, ISO/IEC 2002).

Attualmente sono presenti realizzazioni commercializzate solo per i sistemi di Speech Recognition, in quanto altri strumenti di analisi sono in stato di sviluppo e di valutazione. Tuttavia gli attuali strumenti di filtraggio dei contenuti, consentono un efficace recupero delle informazioni combinando metadati (indicatori dei contenuti) e informazioni sociali e culturali (conoscenza a priori sul contesto). Un altro fatto importante a cui assistiamo è il passaggio dall'utilizzare tassonomie prefissate all'uso di ontologie dinamiche: queste ultime hanno la capacità di comprendere metadati estratti da sorgenti fortemente eterogenee.

La sfida attuale è quella di unire le annotazioni manuali con i dati generati dai sistemi ACA in modo da migliorare robustezza e usabilità dei sistemi per l'accesso ai contenuti multimediali. Si noti che l'obiettivo non è sostituire l'annotazione manuale con sistemi automatici: informazioni soggettive potrebbero essere d'interesse e comunque, vista la quantità di contenuti multimediali in rete, non è praticabile un processo di conversione delle annotazioni nè limitarsi ad aggiungere annotazioni automatiche per tutti i contenuti.

Congiuntamente all'analisi dei contenuti correlati è possibile aggiornare l'interazione con la quale gli utenti effettuano le operazioni di ricerca. Come

nel linguaggio parlato. Le caratteristiche prosodiche di una unità di linguaggio parlato (si tratti di una sillaba, di una parola o di una frase) sono dette soprasegmentali, perché sono simultanee ai segmenti in cui può essere divisa quell'unità. Le si può infatti rappresentare idealmente come 'sovrapposte' ad essi. Alcuni di questi tratti sono, ad esempio, la lunghezza della sillaba, il tono, l'accento.

anticipato, ad oggi sono comuni sistemi *key-word based* i quali operano limitatamente alla pertinenza del testo e dipendono fortemente dall'efficienza linguistica degli strumenti di analisi. Il primo salto in avanti, limitatamente però alla ricerca musicale, è stato il passaggio ad un'interazione di tipo *query by humming* (QBH). Consiste nel fornire elementi simbolici e nel ricercarne le occorrenze; segue uno scenario d'esempio: l'utente fornisce una breve trascrizione musicale per uno strumento per effettuare una ricerca di tutti i brani musicali che contengono tale sequenza. Questa soluzione, però, opera solo con le risorse per cui esiste un'annotazione simbolica; in pratica ha avuto successo solo per la ricerca di file MIDI (in cui l'informazione è già rappresentata in forma simbolica). L'evoluzione che sembra poter riscuotere successo, e il cui processo è stato avviato recentemente, è il passaggio ad un'interazione di tipo *query by example* (QBE): l'utente fornisce un esempio nel formato di istanza multimediale e indica l'oggetto della ricerca.

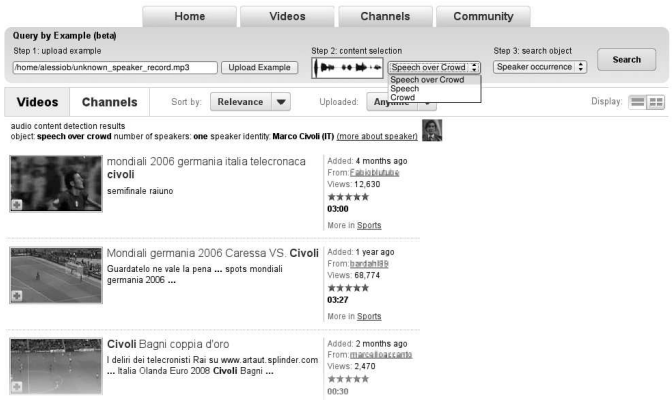

Figura 1. Possibile interfaccia *query by example*

Lo scenario riportato in fig. 1 ne è un esempio. Un utente non conosce il nome di uno speaker che commenta video sportivi ma è in possesso di una clip audio in cui interviene e desidera ricercare altri contenuti in cui occorre la sua voce. L'interazione QBE sarà la seguente: l'utente indica la clip audio come esempio e congiuntamente richiede contenuti in cui sia presente il parlato dello speaker riconosciuto nella clip.

Riguardo agli approcci utilizzati per la realizzazione dei sistemi ACA si è verificata una evoluzione importante: il passaggio dall'utilizzo di soluzioni *model based* all'utilizzo di tecniche di *machine learning* (ML). Le prime si basano sulla conoscenza dei meccanismi percettivi e dei criteri di similarità propri dell'uomo: con questa conoscenza a priori vengono modellati i processi audio da analizzare e vengono utilizzati criteri di similarità per determinare decisioni come la classificazione dei contenuti. Solitamente queste soluzioni ricorrono a modelli matematici basati su auto regressione e medie mobili; un altro approccio è quello di mappare un segnale in un vettore che lo rappresenti (per esempio una stringa di caratteri) detto *signature*: data una metrica, si implementa il concetto di similitudine come distanza tra i vettori e si operano le decisioni in base a valori di soglia prefissati per le distanze. Altri metodi di confronto si basano sul calcolo di indicatori di correlazione tra segnali. Con l'evoluzione dell'intelligenza artificiale, e vista la non sufficiente conoscenza dei complessi processi umani, i sistemi ACA si sono orientati verso l'uso di tecniche di ML.

Le applicazioni più recenti dove sono richiesti i sistemi ACA sono distinguibili in base all'area di interesse. Al momento esistono molti lavori nell'area musicale:

- sistemi di raccomandazione musicale;
- trascrizione musicale;
- monitoraggio trasmissione brani (in questa applicazione rientra anche il monitoraggio di spot pubblicitari);
- separazione delle sorgenti;
- rilevazione di plagio.

Nell'analisi del parlato, i sistemi ASR risultano affermati e le nuove frontiere riguardano le seguenti applicazioni:

- pronuncia di testo con carattere emozionale;

- indexing e retrieval di documenti contenenti parlato (`http://googleblog.blogspot.com/2008/07/in-their-own-words-political-videos.html`);
- identificazione dell'identità dello speaker.

Anche nella biomedica l'analisi audio sta riscuotendo un notevole successo; risulta infatti utile per:

- analisi non invasiva tratto vocale, in particolare:
  - analisi pianto neonatale (sviluppo apparato fonatorio, patologie neurologiche);
  - analisi vocale pre/post trattamento chirurgico (cisti, carcinoma, paralisi corde vocali);
  - analisi vocale pre/post trattamento farmacologico (distonie neurologiche o muscolari);
  - analisi vocale per individuazione malattie (disfonia cronica, tracheotomia, Parkinson, dislessia);
- valutazione della qualità:
  - supporto didattico musicale;
  - stima parametri qualità cantanti professionisti (frequenza vibrato, estensione vibrato, intonazione vocale);

L'area in cui si sviluppa questo lavoro riguarda le applicazioni di *information retrieval* (IR):

- analisi audio per sistemi di sorveglianza;
- content filtering (es. parental control);
- indicizzazione e recupero di news giornalistiche, documentari, video sportivi.

Come descritto in [13], per queste applicazioni l'architettura generale di un sistema ACA separa i contenuti musicali, il parlato e altri tipi di contenuto per poi operare classificazioni più specifiche. In base al tipo di informazione target, i sistemi si riducono all'analisi specifica del parlato, della musica o di altre categorie audio (fig. 2).

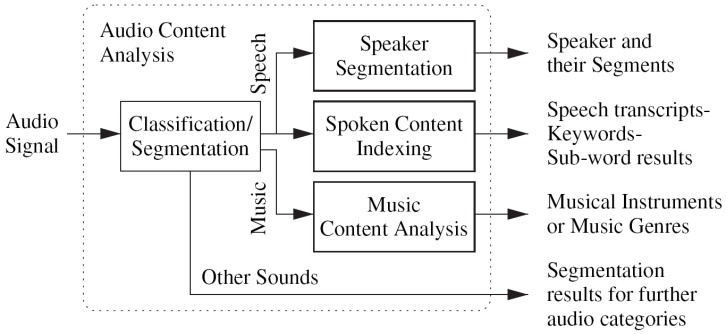

Figura 2. Architettura sistema ACA

Riassumendo i sistemi ACA sono fondamentali in quanto il suono è un'interazione naturale umana, e hanno il ruolo di implementare applicazioni come il riconoscimento e la segmentazione del parlato o l'analisi musicale; queste permettono di risolvere il gap semantico con un approccio bottom-up tra le features fisiche e le rappresentazioni della conoscenza ontologica. Allo stato dell'arte esistono solo pochi lavori in cui si utilizzi l'approccio ontologico in un contesto generalizzato di analisi audio; più frequentemente l'attenzione viene posta sull'analisi di elementi specifici come la musica (es. sistemi di raccomandazione musicale) o il parlato.

Nel lavoro qui presentato l'attenzione è stata rivolta verso i video sportivi e lo scopo è stato quello di studiare alcune soluzioni per la realizzazione di classificatori capaci di individuare un carattere emozionale (l'eccitazione) tra le principali sorgenti audio presenti in una partita di calcio (speakers e pubblico). In particolare gli esperimenti sono stati condotti su registrazioni di match trasmessi in broadcast. Questo caso particolare si rivela utile per

l'annotazione di grandi archivi audiovisivi che altrimenti dovrebbero essere visionati e commentati manualmente.

Introduzione

Questo lavoro di tesi è nato dalla necessità di integrare l'analisi del dominio audio in un sistema IR per l'annotazione semantica automatica nei video sportivi. L'obiettivo è lo studio di fattibilità e la realizzazione di un sistema che estragga informazioni di alto livello dal segnale audio presente nei video. Escludendo soluzioni per l'analisi generale, lo studio si è concentrato sulle soluzioni *context specific* (es. rilevazione scene di violenza, eventi sportivi). Le soluzioni proposte si differenziano per due aspetti: il primo riguarda i domini utilizzati per l'estrazione di informazione di alto livello e le opzioni sono la sola analisi audio o l'analisi audio e video congiunta. Il secondo aspetto è relativo agli approcci utilizzati per la realizzazione e si trovano soluzioni *model based*, basate su tecniche di *machine learning* (ML) o ibride.

Nel lavoro pubblicato da Pfeiffer [21], viene proposta una soluzione *context specific* per l'individuazione di scene di violenza nei film e per l'analisi pubblicitaria. Tale soluzione risulta utile nelle seguenti applicazioni: l'integrazione del sistema proposto nei multimedia player per i filtri di *parental control* e l'integrazione nei sistemi di video sorveglianza per la rilevazione automatica di uno scenario di allarme.
Il lavoro è stato condotto utilizzando tecniche ibride: tecniche di ML per la realizzazione di classificatori congiuntamente a tecniche *model based*. Per esempio per la rilevazione del silenzio, viene proposto un modello a media mobile per l'adattamento temporale della soglia sull'energia. Un'altra soluzione proposta riguarda l'indicizzazione e il recupero di segmenti audio presenti nelle pubblicità trasmesse in broadcast. In questo caso, una clip audio relativa ad una pubblicità da monitorare viene processata da un classificatore che

individui i tratti musicali. Questi tratti vengono rappresentati da una feature (la frequenza fondamentale[2]). Il vettore estratto viene utilizzato come signature e viene operata una ricerca confrontando le signature presenti nel database.
Il punto debole dell'approccio proposto è la costruzione della funzione di similitudine per determinare la distanza tra signatures; in particolare è stata inizialmente implementata la funzione di similitudine basandosi sulla sola correlazione; questa scelta non ha portato a buoni risultati e quindi è stata scelta una funzione basata sulla combinazione lineare di minimo, massimo, media, varianza e valore mediano delle signatures da confrontare (cinque parametri). Come descritto nell'articolo tali valori sono stati determinati in modo euristico: è difficile perciò avere garanzia di ottimalità.
Un altro aspetto dei metodi model-based riguarda i modelli in cui una decisione viene presa in base ad un valore di soglia prefissato: la scelta empirica di tale valore può richiedere molto tempo e un certo valore può risultare adatto solo sotto ipotesi restrittive.

Chen [5] e Leonardi [15] propongono due soluzioni simili per la rilevazione dei goal nelle partite di calcio, basate sull'analisi video e audio congiunta. Gli autori utilizzano un elemento di pre-filtering tra l'estrazione delle features (sia video che audio) e il classificatore utilizzato. Questo sistema ha l'obiettivo di eliminare dati rumorosi quindi non utili ai fini della classificazione. In particolare, in [5], l'audio viene filtrato con un approccio *rule based*: data una sequenza video classificata come tiro in porta, questa si candida come goal se i primi tre secondi e gli ultimi tre secondi della traccia audio contengono entrambi almeno un secondo di eccitazione. Il classificatore viene addestrato per determinare se la porzione video relativa ad un tiro in porta è un goal valutando le features audio e video estratte e filtrate con il metodo rule based illustrato.
I classificatori utilizzati sono differenti: in [5] viene proposto l'algoritmo PRISM[3] per la classificazione basata su regole, mentre in [15] vengono uti-

[2]questa feature viene introdotta nella sezione 1.3.8
[3]determinazione induttiva di regole, per maggiori dettagli si consulti [4].

lizzati i *Hidden Markov Models*[4] (HMMs).
In questo lavoro si individuano alcune limitazioni:

- il sistema non è estendibile per individuare altre azioni oltre al goal;
- la conoscenza acquisita sul dominio non è rappresentata in una forma che permetta di effettuare reasoning e che possa essere condivisa (a differenza, per esempio, di un'ontologia per la definizione di concetti di alto livello basata su features di basso e medio livello);
- la definizione di regole per l'eliminazione di elementi candidati in generale è un approccio non conveniente: le regole possono risultare difficili da definire, limitate a particolari contesti a causa di una conoscenza a priori ristretta e, nel caso di regole definite su più domini, non è facile trovare un metodo per effettuare la sincronizzazione temporale dei segmenti selezionati in ogni dominio (ogni dominio individua segmenti in modo indipendente da altri domini).

In [30] Wang propone un sistema di analisi video e audio per individuare gli eventi principali di una partita di calcio (goal, punizioni, calci d'angolo, etc.). Rispetto al lavoro precedente, oltre ad individuare un set più ampio di eventi, viene proposto un metodo che introduce features di medio livello, determinate tramite classificatori basati su *Support Vector Machines*[5] (SVMs), per ridurre il gap semantico tra features di basso livello e concetti. Tale informazione viene utilizzata per addestrare un classificatore basato su HMMs che determini il concetto di alto livello semantico. Questo approccio risulta vantaggioso rispetto alla definizione di regole che associno features di basso livello ai concetti da individuare: come illustrato per l'articolo precedente le regole definite su più domini richiedono tecniche di sincronizzazione temporale e solitamente risultano difficili da determinare e definire.
Più in dettaglio il sistema presentato estrae features audio e video; ne viene poi operata una classificazione per determinare le features di medio livello:

[4] presentati nella sezione 2.2.3
[5] presentate nella sezione 2.2.2

per esempio per l'audio vengono individuate le classi silenzio, parlato del telecronista, fischi ed esultanza. Vengono poi determinati i vettori relativi alle classificazioni operate nei domini audio e video (es. ⟨"inquadratura lontana del centro campo", "parlato telecronista"⟩); infine vengono collezionate le sequenze di questi vettori relativi alle diverse azioni di gioco relative a goal, calci d'angolo, tiri in porta, rigori. Tali sequenze costituiscono il training set per i classificatori HMMs.
Anche in questo lavoro la conoscenza non è rappresentata in forma simbolica. Da un lato l'utilizzo delle HMMs rappresenta un vantaggio in quanto una tecnica di ML evita le difficoltà illustrate degli approcci model o rule based; ma questa soluzione non risponde ai recenti obiettivi di condivisione della conoscenza e applicabilità del reasoning precedentemente illustrati. Inoltre è comunque necessaria una tecnica di sincronizzazione tra i diversi domini di informazione (audio e video in questo caso).

Similmente in [12], Kim propone una soluzione per rilevare l'evento goal però basata sulla sola analisi audio. Viene giustificata questa scelta osservando che l'analisi audio è computazionalmente meno complessa dell'anlisi video e questo aspetto si rivela utile per integrare sistemi come quello proposto in hardware con poche risorse come gli home recorder. Il sistema presentato si basa su tre passi:

- l'estrazione di features audio;
- la determinazione di segmenti candidati ad essere classificati come highlights;
- l'utilizzo di un pre-filtraggio per eliminare segmenti secondo un set di regole;
- la selezione dei segmenti candidati come eventi goal.

Viene sperimentato l'utilizzo di due diverse features: i *Mel-frequency Cepstral Coefficients*[6] (MFCCs) e la feature Audio Spectrum Projection[7] adotatta dallo standard MPEG-7 per il riconoscimento audio generico; sono state anche valutate differenti dimensioni d'arrivo nella riduzione della dimensionalità delle features (dimensioni di arrivo valutate: 7, 13, 23 e 30). La segmentazione dei candidati viene realizzata tramite un classificatore HMM basato sulle features estratte. Il pre-filtraggio basato su regole elimina segmenti troppo brevi (in base ad un valore di soglia sulla durata) e che non contengono elementi audio classificati come emotivamente eccitati. Infine ogni candidato filtrato viene classificato per valutare se si tratta di un evento goal tramite un classificatore HMM basato sulla sequenza delle seguenti features di medio livello: parlato eccitato del telecronista e il rumore della folla in sottofondo.

Infine, in [17], Divakaran propone una soluzione per la creazione automatica di highlights nei video sportivi basata solamente su una classificazione della traccia audio. L'idea è quella di individuare, nella traccia audio, segmenti in cui occorre un carattere emozionale eccitato nel pubblico o nel parlato relativo al telecronista. Nel lavoro viene sostenuto infatti che l'eccitazione si verifica nei momenti più importanti (es. goal nel calcio, grand slam nel baseball) e quindi viene usata come feature per determinare gli highlights in un video sportivo. In base alla classificazione dei segmenti audio viene calcolato il livello d'interesse per ogni secondo di audio: questi valori vengono salvati come metadati e possono essere utilizzati con una soglia impostata dall'utente per scorrere un video saltando i contenuti meno interessanti.

Il sistema si basa su un classificatore che individua le seguenti classi: applausi, cori, musica, parlato normale e parlato eccitato. Per realizzare questo componente vengono addestrati più classificatori binari (uno per ogni classe)

[6] presentati nella sezione 1.3.9

[7] set di features adottate nello standard MPEG-7 per la classificazione audio generica; il set è composto da features definite nel dominio della frequenza a dimensionalità ridotta e decorrelate. La riduzione della dimensionalità può essere realizzata tramite differenti tecniche; nel lavoro analizzato vengono confrontate la *Indipendent Component Analysis* (ICA), la *Principal Component Analysis* (PCA) e la *Non-negative Matrix Factorization* (NMF).

basati su *Gaussian Mixture Models*[8] (GMMs). Come feature vengono utilizzati i coefficienti MDCT[9] in quanto direttamente estraibili dallo stream audio (vantaggio per integrare il sistema proposto negli home recorder). Dato che l'obiettivo si concentra nel rilevare un carattere di eccitazione viene proposto l'utilizzo di due GMMs per distinguere il parlato eccitato dagli altri contenuti (applausi, cori, musica e parlato normale).
La soluzione proposta risponde agli obiettivi di realizzare un sistema che individui higilights ma non è adatto per indicizzare i contenuti di un video sportivo. Inoltre la scelta di utilizzare le GMMs per la classificazione andrebbe confrontata con altre tecniche di ML. Per esempio le SVMs sono una scelta comune per la realizzazione di classificatori e non necessitano del fine-tuning dei parametri come per le GMMs (es. nelle GMMs è difficile determinare il numero di componenti miste che ottimizza il modello).

Il lavoro presentato in questa tesi propone una soluzione per l'analisi audio delle partite di calcio trasmesse in broadcast da integrare in un sistema IR. In particolare, tale sistema, permette di rappresentare la conoscenza acquisita in forma ontologica. L'architettura in cui collocare l'analisi audio è quindi quella dei sistemi di *knowledge representation* KR (fig. 3).

Il lavoro si è sviluppato in due fasi: lo studio di fattibilità di un classificatore audio per video sportivi (in particolare per il gioco del calcio) e la realizzazione di alcuni classificatori, con differenti tecniche, allo scopo di valutarne le differenti prestazioni. In riferimento alla figura 3, le attività svolte si inseriscono nei blocchi di *low-level analysis* e di *machine learning*.

Differentemente dalle soluzioni presentate precedentemente sono state introdotte due novità:

[8] presentati nella sezione 2.2.4

[9] MDCT sta per *Modified Discrete Cosine Transform*; derivano da una definizione alternativa di trasformata di Fourier e sono caraterizzati da alcune proprietà che li rendono adatti alla compressione di un segnale; vengono infatti adoperati nella codifica AC3 per la compressione audio.

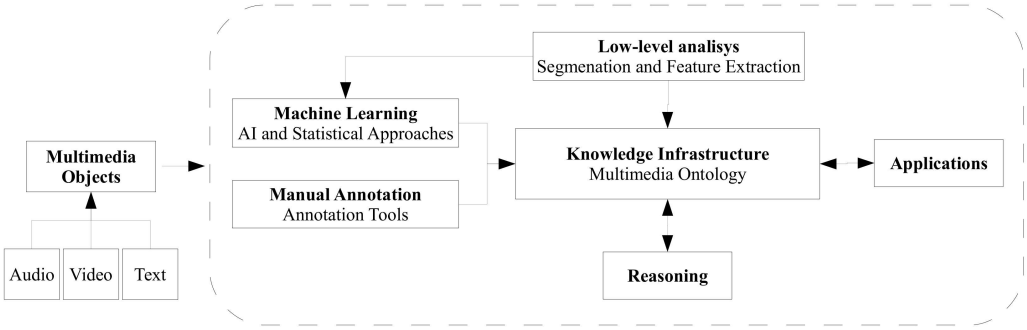

Figura 3. Architettura sistema KR

- l'uso di una recente soluzione basata su reti neurali per la realizzazione del classificatore, le *Deep Belief Networks*[10] (DBNs);
- l'uso di tecniche di *Semi-Supervised Learning*[11] (SSL).

Le DBNs sono un caso particolare di rete neurale che permettono la realizzazione di un classificatore; sono addestrabili sia in modo supervisionato che semi-supervisionato. Inoltre hanno il vantaggio di tutte le reti neurali ovvero che i parametri in gioco, come il numero di nodi o il numero di layer, sono parametri a bassa sensibilità. Ciò significa che una piccola perturbazione nei parametri non provoca un grande cambiamento nei risultati e questo rappresenta un notevole vantaggio.
I classificatori audio realizzati nei related works presentati vengono addestrati in modo supervisionato. Risultando un problema complesso, quello della classificazione audio, sono necessari molti dati etichettati (cioè per cui sono abbinati valori di classificazione determinati manualmente). Recentemente sono state proposte tecniche dette *semi-supervisionate* in modo da poter utilizzare un numero di esempi etichettati facilmente reperibile congiuntamente ad un set di esempi non supervisionati.

L'architettura generale di un classificatore audio è rappresentata nella figura 4.

[10] presentate nella sezione 2.2.5
[11] presentate nella sezione 2.1.2

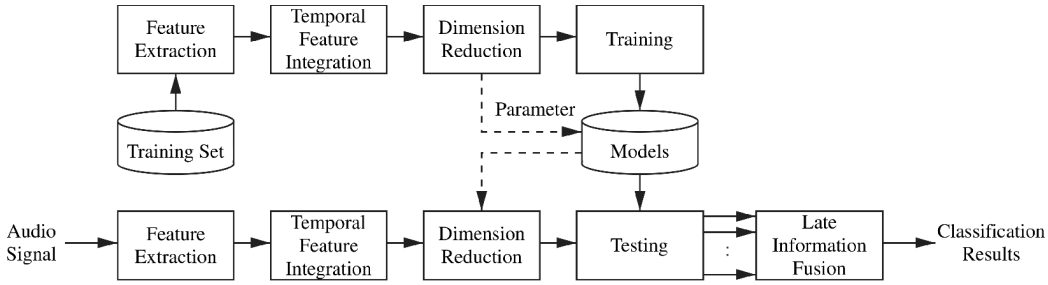

Figura 4. Diagramma a blocchi di un sistema ACA

Il classificatore viene addestrato attraverso il *training set* che contiene elementi audio; in generale per una parte di questi elementi sono presenti i rispettivi valori di classificazione assegnati manualmente.
Ogni elemento audio viene rappresentato da un vettore di *features* ed opzionalmente può essere operata un'integrazione temporale dei valori estratti da unità atomiche che costituiscono l'intero elemento (es. frame audio).
Potrebbe inoltre risultare conveniente, ai fini delle prestazioni, apportare una semplificazione ai vettori delle features utilizzando una tecnica di *riduzione della dimensionalità* dei dati.
I dati così processati vengono utilizzati con il classificatore: in fase di addestramento i dati sono relativi al training set mentre in fase di utilizzo del classificatore addestrato sopraggiungono nuovi elementi audio per i quali determinare la classe di appartenenza.

Il problema più importante, perché il classificatore possa essere utile nel contesto in cui dovrà essere applicato, è la scelta delle classi da individuare nel contenuto audio. Segue una lista di criteri utilizzati per la determinazione delle classi:

- è necessario individuare classi relative a tutti i contenuti rilevanti (dipendenza dal contesto);
- sono necessarie anche tutte le classi utili per operare una segmentazione al fine di escludere contenuti non rilevanti (es. silenzio);
- è opportuno definire il set più ampio di classi eventualmente in modo

che risulti riducibile ad un numero di classi inferiore.

Inoltre la scelta si basa sulle seguenti osservazioni: le partire di calcio trasmesse in tv contengono spot pubblicitari e rientri in studio in cui una o più persone intervengono. Quando il gioco è in corso le principali sorgenti audio sono i telecronisti, la folla e elementi audio inviati dalla regia (es. breve effetto sonoro in corrispondenza di un aggiornamento del testo sovraimpressione). Secondo i criteri e le osservazioni, sono utili classi per segmentare il flusso audio di una registrazione in gioco in corso e rientro in studio.
Durante il gioco in corso gli elementi audio della regia non sono una buona indicazione in quanto dipendono dall'emittente e hanno un'elevata variabilità nel tempo. Elementi rilevanti sono invece il parlato e il pubblico; in particolare il loro stato emozionale è d'interesse in quanto associabile ad azioni salienti.
In base a queste considerazioni le classi scelte sono: silenzio, solo parlato, parlato sopra la folla, solo folla, solo parlato eccitato, solo folla eccitata, parlato e folla eccitati. L'eccitazione è relativa allo stato emotivo. Questo set di classi può essere ridotto raggruppando le ultime tre classe in una generica classe per contenuti emotivamente eccitati.

Capitolo 1

Estrazione dell'informazione per la classificazione

I sistemi che effettuano una classificazione utilizzano come ingresso una rappresentazione dell'informazione che si vuole classificare. Nel caso di informazioni multimediali, la dimensione dei dati raw[1] risulta solitamente troppo elevata per essere direttamente classificata. Questo nella pratica viene risolto combinando le tecniche illustrate di seguito.

- L'utilizzo di *filtri* permette di enfatizzare l'informazione chiave per un problema di classificazione e può apportare ai dati una semplificazione. Per esempio risulta spesso conveniente ridurre i canali di una traccia audio ad un solo canale ed effettuarne il downsampling; o ancora, nel caso delle immagini, ridurre le profondità cromatiche ad una scala di grigi.

- La *segmentazione* dei dati ha lo scopo di definire dei confini che individuino più sottoinsiemi di dati tra quelli disponibili; per esempio potrebbe risultare conveniente segmentare spazialmente un'immagine o temporalmente un segnale audio. I sottoinsiemi individuati saranno poi utilizzati per l'estrazione di informazioni dette quindi *locali*.

- L'estrazione di *features* risulta utile per estrarre nuovi dati che rapp-

[1] dati low-level non processati, espressione diretta dell'informazione

resentino le informazioni di partenza in uno spazio preferibilmente a dimensionalità ridotta. L'obiettivo principale è quello di selezionare features dalle quali ci si possa aspettare un elevato potere di organizzazione dell'informazione in regioni. Se le regioni individuate, alle quali verranno associate le classi, risultano "facilmente separabili" allora la classificazione risulta realizzabile tramite tecniche più semplici.

- Le features relative a informazioni locali devono essere utilizzate congiuntamente per rappresentare l'informazione di partenza per la quale è stata operata una segmentazione. E' quindi necessario adottare tecniche di *information fusion* per produrre un vettore di features che rappresenti l'oggetto originale.

- E' possibile che la dimensione di un vettore di features risulti ancora troppo elevata; può quindi risultare conveniente applicare tecniche di *riduzione della dimensionalità*. Tali tecniche si basano sull'idea di selezionare un set ridotto di features o di estrarne delle nuove, con il criterio di massimizzare il potere rappresentativo nella dimensione di arrivo.

1.1 Cenni sul filtraggio audio

Per la realizzazione di classificatori audio può risultare utile applicare dei filtri per enfatizzare parte dell'informazione disponibile. Questo è possibile quando si ha la sicurezza che l'informazione ritenuta meno interessante sia indipendente dalla classificazione che si vuole realizzare. Segue un'illustrazione delle tipologie di filtri più comuni.
In molte applicazioni di analisi audio viene ridotto il numero di tracce ad una traccia mono perdendo un'informazione che potrebbe essere necessaria, per esempio, a stimare la posizione di una sorgente audio. Il vantaggio sta nella riduzione degli esempi da analizzare: nel caso di una traccia audio stereo passando ad un segnale mono i dati dimezzano.
Un'altra tipologia di filtraggio sono le tecniche di *windowing*: si applica una funzione, definita nel dominio del tempo, in un intervallo temporale finito

fuori dal quale il segnale viene annullato. Tra queste una molto utilizzata è la finestra di Hamming (si veda la figura 5).

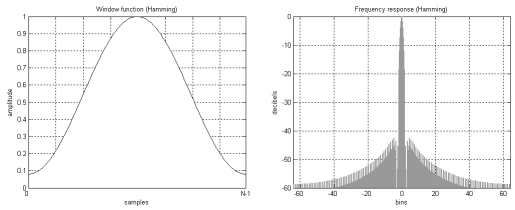

Figura 5. Hamming Function

Nell'analisi del parlato un'informazione importante è rappresentata dalla stima della *frequenza fondamentale* (F_0, presentata nella sezione 1.3.8). Supponiamo che questa feature sia sufficiente per la realizzazione di un classificatore capace di individuare se il parlato viene pronunciato da un soggetto maschile piuttosto che femminile. Dato che la F_0 si trova entro i 500 Hz si può applicare un filtro passa basso in modo da ridurre la quantità di dati.
Più in generale ricorre l'applicazione dei *filter banks*. Un *filter bank* è costituito da un array di filtri passa banda per separare il segnale in più componenti.

1.2 Tecniche di segmentazione audio

La segmentazione è un metodo per separare l'informazione e può operare a differenti livelli di astrazione. A basso livello si utilizza per individuare unità di dati atomiche da processare, mentre a livelli di astrazione più alti costituisce una vera e propria informazione. Infatti, le delimitazioni determinate da una certa tecnica, indicano le parti di segnali relative a contenuti distinti. Per esempio può essere necessario segmentare un segnale contenente parlato separando le singole parole in modo da poterle passare separatamente ad un sistema di trascrizione automatica.

1.2.1 Segmentazione di basso livello

Per processare un segnale risulta conveniente separarlo in più blocchi di informazione di dimensioni minori. Da questi è solito estrarre un set di informazioni locali da utilizzare per il task finale (es. classificazione). La segmentazione audio di basso livello definisce:

- l'estensione temporale delle unità per le quali determinare un valore di classificazione (durata delle *finestre*);
- la durata dei *frame*, che compongono le unità da classificare, da cui estrarre features[2].

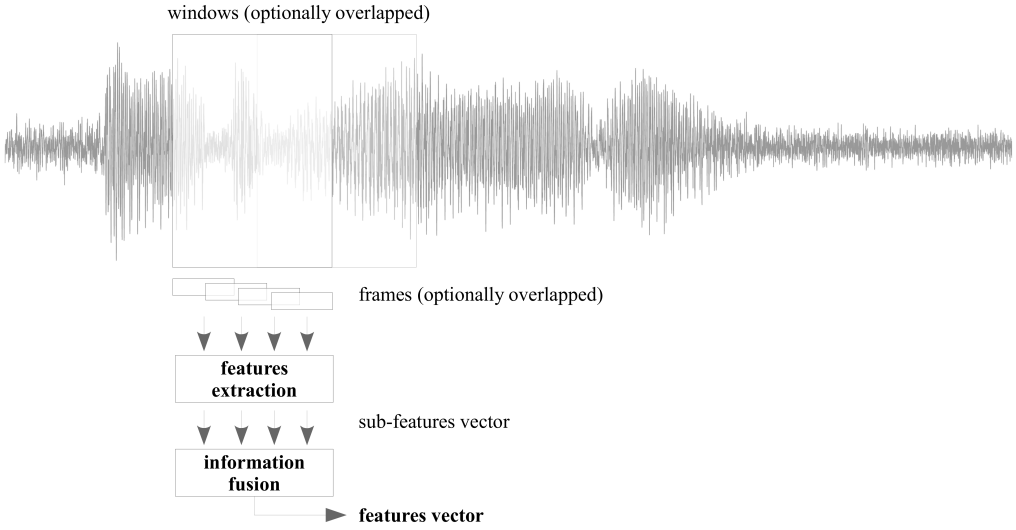

Figura 6. Segmentazione low-level

La scelta della durata delle finestre, ovvero delle unità da classificare, deve essere effettuata valutando i tempi di permanenza tipici di un valore di classificazione nei segnali trattati. Solitamente, nei segnali audio, risulta conveniente scegliere un tempo nell'ordine dei secondi.
Per quanto riguarda invece la durata dei frame, il criterio per la scelta dipende

[2]le features estratte nei vari frame dovranno essere utilizzate congiuntamente (per esempio possono essere semplicemente aggregate componendo un unico vettore di features); una breve introduzione alle tecniche di *information fusion* è presente nella sezione 1.4.

dalle proprietà statistiche del segnale. Conoscendo una stima del tempo di decorrelazione o del periodo di ciclostazionarietà dei segnali trattati si può scegliere una durata adeguata dei frame. Solitamente per segnali audio complessi, quindi contenenti segnali fortemente non stazionari come il parlato, la durata è nell'ordine delle decine dei millisecondi. Sia a livello di finestra che a livello di frame, può portare vantaggio sovrapporre i segmenti (overlap). Questo corrisponde a rafforzare le relazioni di dipendenza statistica tra unità consecutive.

1.2.2 Segmentazione di medio e alto livello

Le tecniche di segmentazione di livello più alto possono essere distinte in base all'ordine in cui vegnono applicati gli step di segmentazione e di classificazione. Le possibilità, in questo caso, sono le seguenti:

- si effettua inizialmente la segmentazione, dopodichè la classificazione assegna la classe più probabile per i segmenti;
- si determinano i limiti dei segmenti basandosi sul risultato di una classificazione precedente;
- i risultati della classificazione e della segmentazione vengono raggiunti congiuntamente.

E' possibile inoltre distinguere le tecniche di segmentazione nel modo seguente.

- segmentazione *energy-based*, solitamente utilizzata per individuare il silenzio;
- segmentazione *metric-based*, data una metrica (misura di distanza tra finestre), si selezionano, come delimitazioni tra segmenti, i massimi in quanto sono indice di un cambiamento (si veda la fig. 7);
- segmentazione *model-based*, segmentazione operata da un classificatore;
- segmentazione *ibrida* (model e metric based), si effettua una segmentazione metric-based dopodichè, tramite clustering, si individuano le possibili classi e infine si applica una segmentazione model-based.

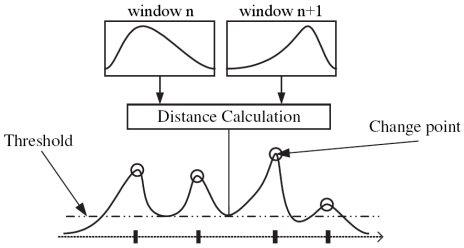

Figura 7. Esempio di segmentazione metric-based: change point detection

1.3 Estrazione di features audio

Le features di basso livello, applicabili ai segnali audio, possono essere classificate in base al dominio in cui sono definite (temporale, frequenziale o quefrenziale[3]) e distinguendo le features che effettuano un'astrazione più vicina alla percezione umana di un segnale da quelle di livello fisico. Peeters, in [19], illustra molte features comuni utilizzate per l'analisi audio.
Seguono alcune definizioni di features valutate per essere utilizzate con i classificatori da realizzare.

1.3.1 Zero-crossing rate

Lo *zero-crossing rate* (ZCR) è una feature fisica temporale che indica la frequenza di cambiamento di segno di un segnale. Questa feature ricorre spesso nelle applicazioni di speech recognition e di information retrieval in ambito musicale. Può essere definita seconda la 1.3.1:

Definizione 1.3.1 (Zero-crossing rate)

$$ZCR = \frac{1}{T}\sum_{t=0}^{T-1}\mathbb{I}\left\{s_t s_{t-1} < 0\right\}$$

con s segnale di lunghezza T e $\mathbb{I}\{A\}$ una mappa che vale 1 se A è una proposizione con valore di verità vero, altrimenti vale 0.

Nei segnali monofonici, lo ZCR può essere utilizzato come un primitivo rivelatore del pitch.

[3]la definizione di quefrenza viene illustrata nella sezione 1.3.9

1.3.2 Crest factor

Il *Crest Factor*, denominato anche *peak-to-average ratio* (PAR) o ancora *peak-to-average power ratio* (PAPR), è una feature fisica per la misura del rapporto di potenza tra i picchi e la media RMS del segnale.

Definizione 1.3.2 (Crest factor)

$$CrestFactor = \frac{|x_{\text{peak}}|}{x_{\text{rms}}}$$

1.3.3 Spectral Centroid

Lo *spectral centroid* è una misura che ha lo scopo di caratterizzare un determinato spettro audio. Indica il "centro di massa"dello spettro. Dal punto di visto della percezione, è in forte connessione con l'impressione della "brightness"[4] di un suono. La feature viene calcolata come media pesata delle frequenze di cui si compone un segnale, utilizzando le ampiezze come pesi:

Definizione 1.3.3 (Spectral centroid)

$$Centroid = \frac{\sum_{n=0}^{N-1} f(n)x(n)}{\sum_{n=0}^{N-1} x(n)}$$

$conx(n)ampiezzadell'n-esimobinef(n)frequenzacentraledelmedesimobin$

1.3.4 Spectral Slope

La *Spectral Slope* è una misura di quanto rapidamente lo spettro di un segnale audio decresce verso le frequenze più alte. Questa feature è correlata alla natura della sorgente di un suono e si è rivelata utile per caratterizzare molti segnali audio presenti in natura [7].
Un modo di calcolarne il valore è l'applicazione della regressione lineare ai

[4]brightness alla lettera si traduce in vivezza, vividezza, vivacità; questo termine, sta ad indicare la brillantezza con cui un suono è riprodotto generalmente in un ambiente vivo, riverberante.

valori di ampiezza spettrali ottenuti dalla trasformata di Fourier; tale metodo produce un singolo valore che indica la pendenza della retta che meglio approssima i dati spettrali.

1.3.5 Harmonic to Noise Ratio

L'*Harmonic to Noise Ratio* (HNR) è una feature che indica il rapporto tra l'energia delle componenti armoniche e l'energia del resto del segnale. E' molto utilizzata per l'analisi del parlato, in particolare per la stima della frequenza fondamentale (F_0). Seguono i passaggi necessari per arrivare alla definizione della feature.
L'autocorrelazione di un segnale $x(\tau)$, in funzione del ritardo τ, è definita dalla 1.3.4:

Definizione 1.3.4 (Auto correlazione)

$$r_x(\tau) = \int x(t)x(t+\tau)dt$$

Questa funzione ha massimo globale in $\tau = 0$. Sia T_0 un valore tale per cui la 1.3.4 assume il massimo successivo a $r_x(0)$. Si dimostra che tutti i massimi della 1.3.4 si trovano in $n \times T_0$. In questo caso T_0 è detto periodo fondamentale della funzione di autocorrelazione a breve termine e si definisce la frequenza fondamentale $F_0 = 1/T_0$. Un segnale $x(t)$ per cui esiste T_0 può essere scomposto in un segnale periodico $H(t)$ di periodo T_0 e in una componente di rumore $N(t)$.
Sia $x(t)$ una finestra estratta da un segnale contenente parlato per la quale esiste T_0. L'energia di tale segnale coincide con l'autocorrelazione calcolata in $\tau = 0$ e, per la linearità dell'autocorrelazione, vale che:

Definizione 1.3.5 (Equivalenza energia-autocorrelazione)

$$r_x(0) = r_H(0) + r_N(0)$$

Se $N(t)$ è un rumore bianco (vale quindi che $N(t)$ è un segnale scorrelato), esiste un massimo locale della 1.3.5 in $\tau_{max} = T_0$:

Definizione 1.3.6 (Ritardo massima energia)

$$r_x(\tau_{max}) = r_H(T_0) + r_N(0)$$

Conseguentemente, la 1.3.6 normalizzata (r'_x) rappresenta l'energia relativa tra la parte periodica del segnale ($r'_x(\tau_{max})$) e la parte complementare ($1 - r'_x(\tau_{max})$, detta rumore).
Si definisce quindi il rapporto logaritmico HNR nel modo seguente (1.3.7):

Definizione 1.3.7 (Harmonic to Noise Ratio)

$$HNR = 10 \log \frac{r'_x(\tau_{max})}{1-r'_x(\tau_{max})}$$

L'HNR rappresenta il grado di periodicità acustico. Per segnali perfettamente periodici, l'HNR è infinito.

1.3.6 Onset

L'*onset* è una feature percettiva e temporale. Indica l'istante in cui ha inizio un suono o una nota. Esistono differenti approcci, distinti in base al dominio in cui operano. Le proprietà note sulle quali basare gli algoritmi di determinazione sono:

- l'aumento dell'energia spettrale;
- cambiamenti nella distribuzione dell'energia spettrale (flusso spettrale) o nella fase;
- cambiamenti nel pitch rilevato, ad esempio usando un algoritmo di polyphonic pitch detection;
- pattern spettrali riconoscibili attraverso tecniche di ML.

Le tecniche più semplici, basate per esempio sulla valutazione dell'incremento dell'ampiezza nel dominio del tempo, possono portare a risultati non soddisfacenti. Un caso ristretto di determinazione della feature occorre quando la ricerca è limitata ad una sola sorgente audio (es. suoni percussivi); in questa eventualità la ricerca può essere più semplice.

1.3.7 Tempo

Il *tempo*, in musica, ha diverse accezioni. Nell'ambito dell'estrazione di features percettive, è utile riferirsi alle seguente definizione: il tempo indica la divisione metrica di una partitura.
In [20] viene proposto un sistema per la rilevazione del tempo al fine di estrarre un'altra informazione percettiva (il beat[5]). La rilevazione viene effettuata in due step:

- si determina l'onset: lo scopo è individuare le periodicità dominanti nel segnale;
- viene operata una ricerca tramite un criterio di somiglianza ricercando ogni possibile suddivisione temporale.

L'estrazione di questa informazione, analogamente all'estrazione dell'onset, può essere vista sia come un'estrazione di feature sia come una tecnica di segmentazione.

1.3.8 Frequenza fondamentale

La *frequenza fondamentale* (F_0) è la frequenza più bassa presente in una serie armonica[6]. Questo valore può essere stimato per qualsiasi segnale che abbia caratteristiche armoniche (solitamente la componente armonica ricercata è sommata ad altri segnali). In base al tipo di segnale da trattare esistono differenti tecniche di stima. Nel caso del parlato la stima della F_0 è utile, per esempio, nei problemi di identificazione dello speaker.
In generale la stima si basa sulla ricerca della periodicità del segnale. Esistono numerose tecniche:

- nel tempo: autocorrelazione del segnale, autocorrelazione dell'errore di stima di un modello, ZCR, AMDF[7];

[5]unità atomica temporale di un brano musicale; coincide con gli intervalli utilizzati dal metronomo per la generazione dei tick.

[6]serie in cui gli elementi sono tutti i primi n multipli interi della F_0.

[7]Average Magnitude Difference Function, è una funzione analoga alla correlazione im-

- in frequenza: prima armonica, distanza media fra le armoniche;
- in altri domini: cepstrum, wavelets;
- uso di pre-filtraggi per smussare il segnale ed eliminare le frequenze indesiderate (es. clipping, filtri passa-basso).

Nel caso del parlato ha senso ricercare la F_0 nei tratti vocalici (la pronuncia di consonanti non avviene per vibrazione delle corde vocali).

1.3.9 Mel-frequency Cepstral Coefficients

Prima di introdurre i *Mel-frequency Cepstral Coefficients* è necessario definire la *scala Mel* e il *cepstrum*.

La *scala Mel* [26] è una scala di frequenze alternativa che corrisponde ad una approssimazione della sensazione psicologica del suono. Segue la definizione analitica di Fant (1968):

Definizione 1.3.8 (Mel-scale)

$$mel(f) = 1000 \log_2(1 + \tfrac{f}{1000})$$

Altre definizioni sono state pubblicate da Beranek (1949), Lindsay & Norman (1977), e O'Shaughnessy (1987).

Il *cepstrum* [2] è il risultato della trasformata di Fourier applicata allo spettro in decibel di un segnale. Il suo nome deriva dal capovolgimento delle prime quattro lettere della parola "spectrum". La definizione originale del cepstrum di un segnale è la trasformata di Fourier del logaritmo della trasformata di Fourier del segnale; ma la definizione comunemente utilizzata è indicata nella 1.3.9.

Definizione 1.3.9 (Cepstrum)

plementabile in modo da risultare computazionalmente più performante, buona robustezza al rumore.

$$X(T) = F^{-1}\left[\ln(F(x(t)))\right]$$

La variabile indipendente del cepstrum è chiamata *quefrenza*. La quefrenza è una misura di tempo, ma non nel senso proprio di segnale che evolve nel dominio del tempo. Per esempio se la frequenza di campionamento di un segnale audio è di 44100 Hz e c'è un alto picco nel cepstrum la cui quefrenza è di 100 campioni, il picco indica la presenza di un pitch (altezza di una nota) alla frequenza di $44100/100 = 441$ Hz. Questo picco appare nel cepstrum perché le armoniche nello spettro sono periodiche e il periodo corrisponde all'altezza (pitch) della nota.
Il grafico del cepstrum serve ad analizzare le velocità di cambiamento del contenuto spettrale di un segnale. Originariamente venne inventato per analizzare terremoti ed esplosioni oltre che analizzare le risposte ai segnali radar. Attualmente è una feature molto efficace per discriminare la voce umana e recentemente è preso in considerazione per ricerche di music retrieval. Un risultato del cepstrum è separare l'energia che viene dalle corde vocali dal resto dell'energia proveniente dal tratto che percorre l'aria dalla gola all'esterno per produrre la voce.

I *Mel-frequency Cepstral Coefficients* (MFCCs) si propongono come feature percettiva standard per le applicazioni di Speech Recognition. Derivano dall'applicazione della scala Mel nel dominio della quefrenza (deriva che gli MFCCs sono una feature cepstrale).
Nel calcolo dei coefficienti MFCs lo spettro del segnale viene diviso in bande. Un algoritmo efficiente può essere realizzato operando direttamente sulla DTFT del segnale applicando un filter bank triangolare nel dominio della frequenza (si veda la figura 8). I filtri triangolari si sovrappongono parzialmente e, per effetto della scala Mel, risultano più ampi (in termini di banda) alle alte frequenze. In questo modo la risoluzione alle basse frequenze è maggiore: questa proprietà risulta particolarmente utile nelle applicazioni di Speech Recognition.

Applicando la trasformata inversa di Fourier al logaritmo dei valori in uscita del filter bank elevati al quadrato si ottengono i coefficienti MFC. In

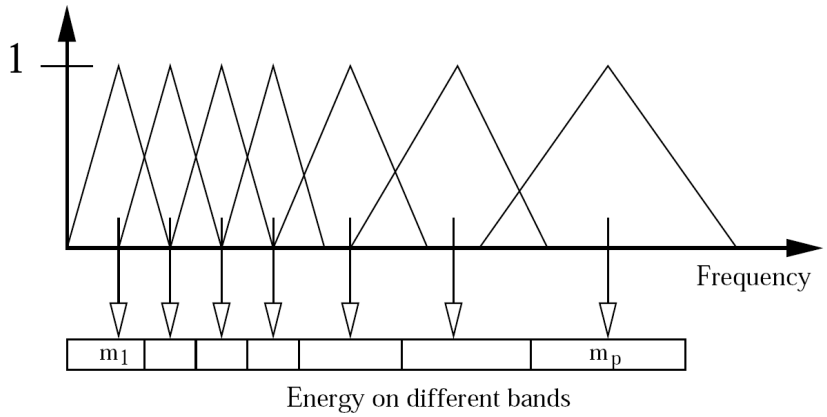

Figura 8. Mel filter bank

questo caso la trasformata inversa di Fourier è equivalente alla *Discerte Cosine Transform* (DCT), la quale ha un costo computazionale inferiore.
Esistono più definizioni per il calcolo dei coefficienti MFCs che variano in base alle caratteristiche del filter bank, dei parametri che descrivono la scala di frequenze percettiva e alla definizione di Cepstrum utilizzata; in [6] vengono confrontate alcune differenti implementazioni di MFCCs e viene valutato l'impatto delle differenti definizioni.

1.4 Information Fusion

L'*information fusion* è la pratica di combinare differenti informazioni (solitamente un set di informazioni locali) per generarne una nuova rappresentazione. In [23] viene proposta la seguente classificazione per le diverse tecniche:

- *pre-mapping* (o early fusion), fusione operata *prima* della classificazione;
- *midst-mapping*, fusione operata *durante* la classificazione;
- *post-mapping* (o late fusion), fusione operata *dopo* la classificazione.

Per esempio il concatenamento e l'integrazione temporale di features rientrano tra le tecniche di pre-mapping. Le tecniche midst-mapping, invece, sono utili per risolvere il problema di gestire più features estratte con risoluzioni

temporali differenti [13].
Le tecniche di *late fusion* combinano risultati estratti da differenti classificatori per operare una decisione. Alcuni esempi di late fusion sono la votazione a maggioranza, la determinazione di una classifica o l'uso di connettivi logici. Per esempio se x è una clip audio contente silenzio allora non è necessario attivare il classificatore che individua il parlato:

$$IsSpeech'(x) = \neg IsSilence(x) \wedge IsSpeech(x)$$

1.5 Tecniche di riduzione della dimensionalità

Quando la conoscenza a priori sul problema è limitata una delle prime questioni da affrontare riguarda la scelta delle features. Escludere una classe di informazioni può apportare una perdita di informazione utile ma l'utilizzo di vettori di features di elevate dimensioni aumenta drasticamente i tempi di processamento. Un approccio possibile è quindi il seguente:

- utilizzare un buon numero di features;
- utilizzare una tecnica di *estrazione* o di *selezione* per ridurre il numero di features

Le tecniche di *selezione* sono riconducibili ad un problema ottimizzazione combinatoria in quanto hanno lo scopo di selezionare un subset di features, tra quelle disponibili, massimizzando il potere rappresentativo.
Le tecniche di *estrazione*, invece, si basano sulla determinazione di una mappa con spazio di arrivo ridotto rispetto a quello di partenza. Vengono estratte, quindi, nuove features.
Segue una breve presentazione delle principali tecniche di riduzione della dimensionalità basate sull'estrazione delle features.

La *Principal Component Analysis* (PCA) è la più comune tecnica di riduzione *lineare*. Viene determinata una mappa lineare tale per cui, nello spazio di arrivo, la varianza risulti massima. La tecnica si basa sulla decomposizione a valori singolari; è interessante ricercare la massima varianza in quanto,

in condizioni di linearità, una direzione in cui si ha più escursione nei dati restituirà una variazione più ampia nell'uscita. Questa tecnica introduce un errore notevole quando il modello che si approssima ha caratteristiche fortemente non lineari. In questa eventualità è preferibile optare per altre tecniche [18].

La *Logistic PCA* (LPCA) è un caso particolare di *Kernel PCA*. Attraverso l'utilizzo di una funzione kernel non lineare i dati vengono mappati in un nuovo spazio dove poi si applica la PCA tradizionale. Se il kernel riesce a mappare i dati in modo opportuno allora la tecnica individua uno spazio in cui lavorare in condizioni approssimabili alla linearità. Per maggiori dettagli sulla si consulti [24].

Un approccio totalmente differente per la riduzione della dimensionalità è la tecnica degli *auto-encoders*. Consiste nell'addestrare una rete neurale feed-forward basata sulle *Restricted Blotzmann Machines* allo scopo di codificare i dati di ingresso. Il numero di nodi del layer di uscita coinciderà con la dimensione dello spazio di arrivo. Il potere di rappresentazione è garantito dal fatto che questa soluzione è in grado di individuare momenti statistici di alto ordine nei dati. Negli ultimi anni sono stati pubblicati ottimi risultati [10]; inoltre la ricerca è attiva nello studio di questi metodi [29].

1.6 Soluzioni adottate

La prima scelta operata riguarda le unità da classificare. La durata fissa pari a due secondi risulta adeguata per individuare in una clip audio gli eventi da classificare. L'adeguatezza è stata confermata durante l'etichettamento manuale: la frequenza di clip in cui si verifica una transizione degli eventi (es. da stato emotivo non eccitato ad eccitato) è minima.

La seconda scelta riguarda l'eventuale segmentazione low-level. I contenuti audio trattati contengono parlato il quale ha la proprietà di essere pronunciato velocemente in una percentuale di casi non trascurabile. Il segnale può

quindi risultare fortemente non stazionario. A causa di ciò è opportuno operare una segmentazione del segnale in modo da estrarre informazioni locali in condizioni di quasi stazionarietà. Bisogna però individuare un compromesso con il fatto che una risoluzione temporale troppo elevata porta ad operare in una dimensione elevata.
Nei lavori di riferimento (in particolare [16] e [12]), viene effettuata una segmentazione di poche decine di millisecondi. E' stata mantenuta quindi tale scelta; in dettaglio la segmentazione opera con i seguenti parametri:

- due valori per la dimensione dei frame: inizialmente 32ms (scelta comune), definitivamente 64ms (prestazioni analoghe);
- overlap al 50% (quindi frame distanti 16ms);
- frame filtrati da funzione di Hamming.

Per la selezione delle features il criterio è stato quello di adoperare features di successo. Gli MFCCs sono sicuramente i più citati nel campo audio: vengono utilizzati negli affermati sistemi ASR e in tutti i lavori di riferimento costituiscono la feature principale. Inoltre, essendo un descrittore percettivo e a dimensionalità ridotta rispetto ai dati su cui viene calcolato, è ragionevole interpretarlo come una prima astrazione del segnale audio. E' stata quindi scelta come principale feature da utilizzare congiuntamente al logaritmo dell'energia dell'intera clip audio.

Per ogni frame vengono estratti tredici coefficienti MFC. La tecnica di information fusion adoperata è un semplice concatenamento (dei vettori che rappresentano ogni frame) in sequenza temporale. Il vettore finale delle features che rappresentano una clip audio è quindi composto dai vettori di MFCCs concatenati e dallo scalare relativo al logaritmo dell'energia.

Non è stata adoperata nessuna tecnica di riduzione della dimensionalità in quanto l'obiettivo è realizzare un classificatore tramite Deep Belief Network, soluzione che implicitamente opera tale riduzione. Inoltre, prevedendo tempi di addestramento non elevati, è risultato preferibile non adoperare alcuna

riduzione; infatti, l'utilizzo di tecniche di questo tipo (come ad esempio la PCA) in casi in cui non si abbia una forte conoscienza a priori del dominio comporta spesso una riduzione di prestazioni.

Capitolo 2

Metodi di classificazione

Ad oggi esistono molteplici soluzioni per la realizzazione di un classificatore. La scelta è influenzata da molti fattori dipendenti dalle proprietà del problema e dei dati da trattare. Segue un elenco di fattori determinanti per la selezione di uno o più metodi di classificazione.

- conoscenza a priori sul problema (utile per escludere modelli con basso potere rappresentativo);
- dimensionalità dei dati (in particolare comportamento alle alte dimensionalità);
- reperibilità di dati per l'addestramento, in particolare:
  - dati etichettati;
  - dati non etichettati;
- qualità dei dati (es. dati rumorosi);
- efficienza computazionale, in particolare:
  - tempi di addestramento (un metodo meno accurato può essere preferibile se risulta addestrabile in tempi brevi);
  - tempi di classificazione (es. vincoli temporali per calcolare una decisione in un sistema real-time);

- classificazione binaria piuttosto che multiclasse (la disponibilità di un'implementazione multiclasse semplifica gli esperimenti);

In questo capitolo vengono analizzati due metodi generali di apprendimento per la realizzazione di classificatori: *Supervised Learning* e *Semi-Supervised Learning*. Segue poi un'introduzione ai metodi candidati per la realizzazione di un classificatore audio: *Neural Networks*, *Support Vector Machines* (SVMs), *Hidden Markov Models* (HMMs), *Gaussian Mixture Models* (GMMs) e le recenti *Deep Belief Networks* (DBNs). Infine viene giustificata la scelta delle DBNs e delle SVMs come metodi per studiare la fattibilità e le prestazioni di un classificatore audio.

2.1 Metodi di apprendimento

Tradizionalmente i classificatori vengono realizzati con dati supervisionati, ovvero preventivamente classificati dall'uomo, a differenza di problemi di clustering per i quali sono sufficienti dati non etichettati. Per problemi ad altà dimensionalità, la quantità di dati richiesta per l'addestramento è solitamente necessario che sia elevata; quindi, in un problema di classificazione, i dati annotati richiesti sono molti. Questo comporta un grande investimento di risorse umane per poter disporre di dati classificati da utilizzare per l'addestramento.
A questo proposito sono stati sviluppati metodi di *Semi-Supervised Learning* che fanno utilizzo di dati annotati e non annotati. Segue un'introduzione ai classificatori realizzati come metodi *supervisionati* e *semi-supervisionati* allo scopo di effettuarne un confronto.

2.1.1 Supervised Learning

Nei metodi di *Supervised Learning* i dati per l'apprendimento consistono in un insieme di istanze $\langle x, c(x)\rangle$ dove $x \in X$ sono i vettori contenenti gli attributi dell'oggetto della classificazione e $c(x) : X \rightarrow \{0, 1, ...\}$ indicano i rispettivi valori del concetto target, ovvero la classificazione attribuita. L'obiettivo della classificazione è trovare un'approssimazione $h(x) \simeq c(x)$ apprendendo il

comportamento della $c(x)$ dal set di istanze $\langle x, c(x)\rangle$.
In generale un metodo di apprendimento opera minimizzando una *funzione di richio empirico*. Si assuma l'esistenza di una *loss function* $L : Y \times Y \rightarrow \mathbb{R}^+$ dove Y è il codominio della $h(x)$ e L è una mappa a valori reali non negativi. La quantità $L(z, y)$ è la perdita causata dall'aver predetto $h(x) = z$ quando il valore reale è $c(x) = y$. Si definisce quindi una funzione di rischio (2.1.1) come il valore atteso della loss function:

Definizione 2.1.1 (Risk function)

$$R(h) = \sum_{x_i \in X} L(h(x_i), c(x_i))\, p(x_i)$$

L'obiettivo è trovare una funzione $h^\star(x)$ tale per cui $R(h^\star)$ è minima. Tenendo conto però, che il comportamento della $c(x)$ è noto solo per il sottoinsieme di valori $x \in X$, la funzione di rischio reale si approssima con la funzione di *rischio empirico* 2.1.2.

Definizione 2.1.2 (Empirical risk function)

$$\tilde{R}_n(h) = \frac{1}{n} \sum_{i=1}^{n} L(h(x_i), c(x_i))$$

2.1.2 Semi-Supervised Learning

I metodi di *Semi-Supervised Learning* sono nati in quanto i dati non etichettati sono facili da reperire a differenza dei dati etichettati. Questi ultimi sono difficili da ottenere in quanto l'annotazione è un'operazione faticosa, può richiedere l'intervento di esperti o di particolari dispositivi e generalmente richiede molto tempo. L'obiettivo di questi metodi è utilizzare sia dati etichettati che non per migliorare le tecniche di addestramento esistenti e crearne nuove.
In questo caso i dati per l'addestramento sono separabili in due insiemi: l'insieme *labelled* e l'insieme *unlabelled*. Il primo, identificato da $X_{labelled} = \{x_1, ..., x_l\}$, coincide con il set di istanze $\langle x, c(x)\rangle$ definito per i metodi supervisionati; il secondo è un set di istanze $x \in X_{unlabelled} = \{x_{l+1}, ..., x_m\}$ per il quale non sono specificati i valori di classificazione. Più in generale è

possibile definire anche l'insieme di dati etichettati $X_{test} = x_{m+1}, ..., x_n$ per valutare l'accuratezza durante il training.
Solitamente vale $l \ll m$ ovvero che il numero di esempi etichettati è inferiore al numero di esempi non etichettati. L'addestramento si svolge utilizzando i dati contenuti in $X_{labelled}$ e $X_{unlabelled}$ durante la fase di training e valutando l'accuratezza sul set X_{test} (dati non disponibili durante la fase di training).
La figura 9 aiuta a dare un'idea di come i dati non etichettati potrebbero portare vantaggio all'addestramento.

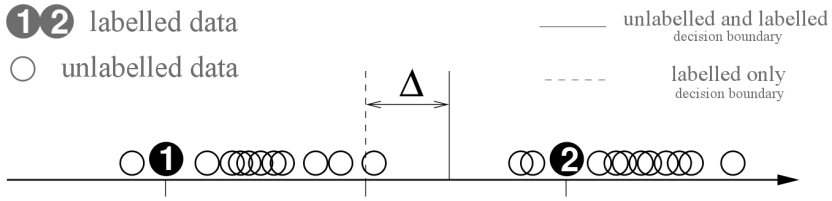

Figura 9. Esempio di utilizzo di dati unlabelled nell'apprendimento semi-supervised

In [32] vengono presentate le idee alla base dei differenti metodi di SSL e le relative problematiche, in particolare:

- self-training: si addestra in modo supervisionato, si etichettano quindi i dati non classificati e si riaddestra in modo supervised utilizzando anche i nuovi dati etichettati;
- S3VMs: semi-supervised SVMs, si basano sull'assunzione che i dati non etichettati appartenenti a classi distinte siano separati da un margine elevato;
- altri metodi: generative models, graph-based algorithms, multiview algorithms.

2.2 Soluzioni valutate

Seguono alcune delle possibili soluzioni valutate per la realizzazione di un classificatore audio. Per ogni soluzione viene fornita un'introduzione e ven-

gono messe in evidenza prorietà e problematiche utilizzate per effettuare una scelta.

2.2.1 Reti Neurali

In termini molto generali, una rete neurale è un processore distribuito, ispirato ai principi di funzionamento del sistema nervoso degli organismi evoluti, costituito dalla interconnessione di unità computazionali elementari (neuroni), con una caratteristica fondamentale: la conoscenza è acquisita dall'ambiente attraverso un processo adattativo di apprendimento ed è immagazzinata nei parametri della rete e, in particolare, nei pesi associati alle connessioni [8].
I neuroni, che si possono vedere come nodi di una rete orientata provvisti di capacità di elaborazione, ricevono in ingresso una combinazione dei segnali provenienti dall'esterno o dalle altre unità e ne effettuano una trasformazione tramite una funzione, tipicamente non lineare, detta funzione di attivazione. L'uscita di ciascun neurone viene poi inviata agli altri nodi oppure direttamente all'uscita della rete, attraverso connessioni orientate e pesate.
Una rete neurale consente di approssimare, in uno specifico contesto applicativo, la corrispondenza esistente tra un ingresso e un'uscita di natura opportuna. Nei problemi di *classificazione* l'ingresso è costituito dal vettore delle caratteristiche dell'oggetto o del fenomeno da classificare e l'uscita è una variabile a valori discreti che esprime l'appartenenza ad una delle classi prefissate. Il legame ingresso-uscita realizzato dalla rete dipende essenzialmente:

- dal tipo di unità elementari, che possono avere struttura interna più o meno complessa ed avere funzioni di attivazione caratterizzate da differenti tipi di nonlinearità;
- dall'architettura della rete, ossia dal numero di nodi, dalla struttura e dall'orientamento delle connessioni;
- dai valori dei parametri interni associati alle unità elementari e alle connessioni, che devono essere determinati attraverso tecniche di apprendimento.

Le reti neurali sono addestrabili in modo supervisionato: sono quindi adatte al problema della classificazione. Il potere rappresentativo dipende dal numero di unità e di strati utilizzati, quindi per relazioni complesse sono necessarie reti di elevate dimensioni. Questo rappresenta uno svantaggio in quanto i metodi classici di addestramento risultano computazionalmente inefficienti. Inoltre soffrono del problema degli *ottimi locali*: l'addestramento opera minimizzando l'errore di classificazione e l'ottimizzazione può terminare trovando un minimo di scarsa qualità ovvero distante dal valore più basso di errore (minimo globale).

2.2.2 Support Vector Machines

Le Support Vector Machines (SVMs) sono un insieme di metodologie, nell'ambito dell'apprendimento supervisionato, utilizzate per la classificazione e la regressione. Appartengono alla famiglia dei classificatori lineari generalizzati e possono essere considerati un caso particolare della regolarizzazione di Tikhonov; inoltre, sono note anche come classificatori a margine massimo poiché riescono simultaneamente a minimizzare l'errore di classificazione empirico e massimizzare il margine geometrico.
L'idea alla base delle SVMs è quella di far corrispondere al vettore degli ingressi un punto in uno spazio ad alta dimensionalità in cui sia facile individuare un iperpiano che riesca a separare al meglio gli esempi proposti. Vengono costruiti due iperpiani paralleli, uno per ciascuno dei lati dell'iperpiano che separa i dati: questi delimitano una porzione di spazio in cui non sono presenti esempi; l'iperpiano di separazione è quello che massimizza la distanza tra questi due iperpiani paralleli. I vettori che appartengono agli iperpiani paralleli vengono detti "vettori di supporto", da cui appunto il nome del metodo, poiché rappresentano dei punti di sostegno vincolanti. La generalizzazione del metodo consiste nel prevedere che gli esempi possano "valicare" gli iperpiani limite, ma si introduce una penalizzazione ai vincoli del problema di ottimizzazione.
Questa soluzione inoltre offre un notevole vantaggio: per superare il limitato potere discriminatorio offerto dalla separazione tramite iperpiani, è possibile

utilizzare una mappa non lineare che permetta di modellare superfici di separazione più complesse. Tale mappa viene rappresentata da una funzione detta *kernel function* la quale definisce una metrica nello spazio vettoriale in cui sono definiti i dati che rappresentano gli elementi da classificare.

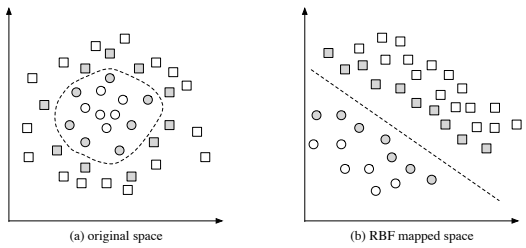

Figura 10. Esempio di insieme separabile tramite kernel Radial Basis

Un semplice esempio che mostra i vantaggi nell'utilizzare una funzione kernel è rappresentato nella figura 10: il *decision boundary* mappato dalla *Radial Basis Function* (RBF) riesce a separare i dati.
In letteratura le SVMs sono state introdotte in tempi relativamente recenti. Approfondimenti teorici riguardo la definizione e la formalizzazione possono essere trovati in [25], [3], [27] e [28].

2.2.3 Hidden Markov Models

I *modelli di Markov* sono stati introdotti da Andrei A. Markov e sono stati utilizzati inizialmente per modellare le sequenze delle lettere nella letteratura Russa. Solo successivamente sono diventati uno strumento generale di analisi statistica.
I modelli di Markov si definiscono come automi a stati finiti descritti dalle probabilità di transizione degli stati; godono inoltre della proprietà di causalità infatti uno stato dipende solo dallo stato precedenti. Tradizionalmente gli stati sono *visibili* in quanto è sempre noto lo stato del sistema. Nelle HMMs, invece, l'esatta sequenza di transizione degli stati in un sistema non è nota; viene quindi definita una funzione probabilistica che la rappresenti.
Nonostante i numerosi punti deboli, le HMMs ricorrono frequentemente nei

sistemi moderni di Speech Recognition. Questo è dovuto al fatto che durante gli anni '80 la ricerca ha compiuto molti sforzi su questi modelli. Anche i metodi alternativi più recenti frequentemente faticano a raggiungere prestazioni superiori.

Una HMM è un insieme finito di stati, ciascuno dei quali è associato ad una distribuzione di probabilità. Tale distribuzione indica la probabilità che venga osservato un certo elemento. Le transizioni sono descritte dalle probabilità di transizione tra gli stati. La figura 11 mostra una semplice struttura HMM.

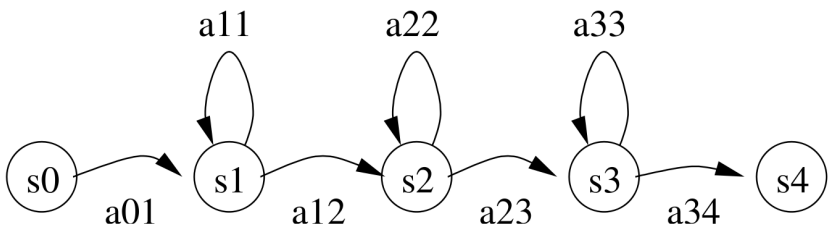

Figura 11. Semplice esempio di una HMM

Una HMM è definita dai seguenti elementi:

- il numero di stati N;
- il numero di elementi osservati M (può essere anche infinito nel caso di valori continui);
- le probabilità di transizione degli stati a_{ij} (con $i, j \in \{1, ..., N\}$ e $i \neq j$);
- una distribuzione di probabilità associata ad ogni stato $p_j(k)$ (con $j \in \{1, ..., N\}$ e $k \in \{1, ..., M\}$); nel caso continuo tale distribuzione può essere approssimata da una somma di *gaussian mixtures* (più precisamente N componenti gaussiane) ognuna delle quali è descritta dai pesi c_{jm}, dal vettore delle medie μ_{jm} e dalla matrice di covarianza Σ_{jm}.

Dati questi elementi, la probabilità di osservare un vettore o_t viene calcolata in accordo alla definizione 2.2.1:

Definizione 2.2.1 (HMM Observation vector probability)

$$b_j(k) = \sum_{m=1}^{M} c_{jm} \mathcal{N}(\mu_{jm}, \Sigma_{jm}, o_t)$$

Gli aspetti matematici riguardanti le HMMs sono abbastanza complessi e una spiegazione approfondita non rientra negli obiettivi di questo lavoro. Per approfondimenti si consultino i seguenti paper: [22] e [11].

Le HMMs hanno diversi punti deboli. In questi modelli la probabilità di trovarsi in un certo stato all'istante t dipende solamente dallo stato al tempo $t-1$; un'altra assunzione spesso non valida è l'indipendenza tra le osservazioni. In aggiunta a questi fatti, il numero di parametri necessario può crescere esponenzialmente richiedendo una grande quantità di dati per l'addestramento.

La classificazione è realizzabile addestrando una HMM per ogni classe: il training set è composto da serie temporali delle osservazioni relative ad una specifica classe; tramite un opportuno algoritmo di apprendimento viene modellata ogni HMM.
Il valore di classificazione viene determinato nel modo seguente. Ogni HMM riceverà in ingresso la serie temporale dell'osservazione da classificare e resituirà la probabilità che tale sequenza sia relativa alla classe che la HMM rappresenta. Infine si osserva quale HMM restituisce la probabilità massima

2.2.4 Gaussian Mixture Models

Gaussian Mixture Models (GMMs) sono modelli probabilistici formati dalla combinazione lineare di funzioni di distribuzione di tipo gaussiano, chiamate componenti della mistura. I parametri caratterizzanti questi modelli sono:

- la dimensione del problema d;
- il numero di componenti miste k;
- i momenti μ, k vettori di dimensione d;

- le covarianze Σ, k matrici di dimensione d^2;
- il vettore dei pesi α, di dimensione k.

Solitamente un'istanza di parametri viene indicata con la notazione $\Theta = \{\langle\alpha_1, \mu_1, \Sigma_1\rangle, ..., \langle\alpha_k, \mu_k, \Sigma_k\rangle\}$. La probabilità di osservare il campione X_i, dato il modello Θ è data dalla 2.2.2:

Definizione 2.2.2 (GMM Observation vector probability)

$$P(X_i|\Theta) = \sum_{j=1}^{k} \alpha_j \, P_j(X_i|\Theta_j)$$

con $\alpha_j = P(X_i \in j|\Theta_j)$ probabilità, nota a priori, che ogni campione X_i sia un membro della sola componente j.

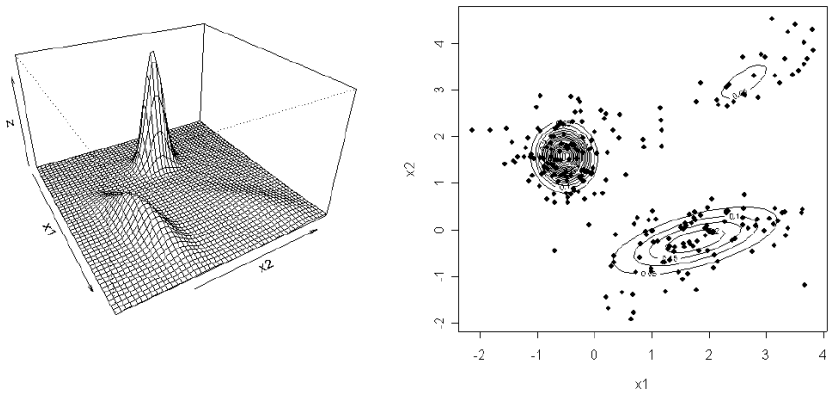

Figura 12. Esempio di GMM con $d = 2$ e $k = 3$

L'algoritmo di training per le GMMs è *Expectation Maximization* (EM). Tale algoritmo permette l'apprendimento con set di dati incompleti o con set di dati generati da misture di distribuzioni di probabilità nel caso in cui non si conoscano né i parametri delle funzioni di distribuzione, né l'appartenenza di ciascun dato ad una data funzione di distribuzione. Per una trattazione completa di EM applicato alle GMMs si consulti il capitolo 9 del testo [1].
Per realizzare un classificatore con le GMMs è necessario addestrare un numero di modelli misti pari al numero di classi da individuare. In breve, per

ogni classe, una GMM realizza un classificatore binario addestrato modellando la distribuzione degli attributi di tutti gli esempi relativi ad una certa classe. Come illustrato k denota il numero di componenti miste del modello; tradizionalmente questo valore viene scelto tramite cross-validation e in caso di classificazione multipla è costante per ogni classificatore. Questa soluzione porta al seguente problema: un certo valore di k potrebbe non essere adeguato per tutte le classi; infatti è usuale riscontrare overfitting per alcune classi e underfitting per altre. In [14] e [31] viene illustrata una soluzione a questo problema: in entrambi i lavori viene abbinato alle GMMs il criterio *Minimum Description Length* (MDL) per scegliere un valore adeguato di k determinato analiticamente.
Un altro problema delle GMMs sta nell'algoritmo di addestramento (EM) in quanto soffre del problema degli ottimi locali nella fase di massimizzazione.

2.2.5 Deep Belief Networks

Le *Deep Belief Networks* (DBNs) sono reti neurali basate sul modello generativo[1] caratterizzate da un numero elevato di layer; sono state introdotte da G. Hinton (University of Toronto) nel 2006 e in seguito si è aggiunto alle attività di ricerca Y. Bengio (University of Montreal). Sono state concepite per tentare di risolvere i seguenti problemi:

- la procedura di back-propagation (BP) nelle reti neurali richiede dati etichettati (la maggior parte dei dati non lo sono);

[1]Nell'approccio generativo le osservazioni empiriche vengono "spiegate"mediante un modello che descrive probabilisticamente le interazioni tra le quantità variabili. Tale sistema probabilistico viene specificato tramite due componenti: una lista di variabili che quantificano gli stati osservati e presunti del modello, e una probabilità congiunta definita su tutte queste variabili. L'alternativa è la scelta di metodi discriminativi: questi affrontano direttamente il problema di trovare i criteri che permettono di raggruppare ottimamente le osservazioni empiriche. Ciò viene solitamente ottenuto estraendo da queste ultime alcune caratteristiche (feature) intrinseche degli oggetti osservati, trasformando ogni osservazione in un punto in uno spazio multidimensionale e poi trovando le relazioni che governano le similarità esistenti tra punti dello stesso oggetto o le differenze tra punti di oggetti differenti.

- i tempi di esecuzione della BP sono lenti specie nei casi in cui il numero di layer nascosti sia elevato
- la BP soffre il problema dei minimi locali di scarsa qualità
- soluzioni basate sull'utilizzo delle Support Vector Machines (SVMs) necessitano una "codifica" dei dati (es. estrazione delle features); non è sempre facile individuare features utili per la corretta determinazione di un modello
- metodi di riduzione della dimensionalità come la Principal Component Analysis (PCA) richiedono troppe assunzioni sulla statistica dei dati

Queste reti derivano dall'unione delle Sigmoid Belief Networks[2] (Adford Neal, 1992) e delle Boltzman Machines[3] (Hinton e Sejnowski, 1983). Le prime permettono di implementare un modello in cui siano presenti unità stocastiche nascoste da interpretare come cause di effetti visibili; mentre le BM definiscono una struttura simmetrica capace di individuare regolarità anche complesse nei dati. Questa unione permette di realizzare un modello generativo di elevata potenza [9].
La prima differenza dalle classiche reti neurali riguarda l'addestramento: l'inizializzazione della rete non viene effettuata attribuendo pesi random ma tramite una procedura di pre-training. Questa procedura è unsupervised e risulta efficiente in quanto opera layer-by-layer (natura greedy); lo scopo è quello di determinare la dipendenza delle unità di un layer dalle unità dei layer superiori.
L'addestramento layer-by-layer si basa sull'idea di utilizzare i valori delle unità di un layer come dati per il training del layer successivo. In questo modo ogni layer tenta di modellare la distribuzione dei propri dati in ingresso. Questa procedura viene realizzata operando sulla rete vedendola come composizione di semplici moduli di apprendimento ciascuno dei quali è una Restricted Boltzman Machine[4] (RBM). Ogni modulo dunque contiene un

[2]Sigmoid Belief Network: grafico aciclico di unità stocastiche binarie

[3]Boltzmann Machine: connessioni simmetriche tra unità stocastiche binarie

[4]Una Boltzmann Machine (BM) è una rete neurale che permette di scoprire interessanti caratteristiche che rappresentano regolarità anche complesse nei dati. La principale

layer di unità visibili che rappresentano i dati e un layer di unità nascoste addestrate per rappresentare caratteristiche che "assorbono"correlazioni di alto ordine nei dati.

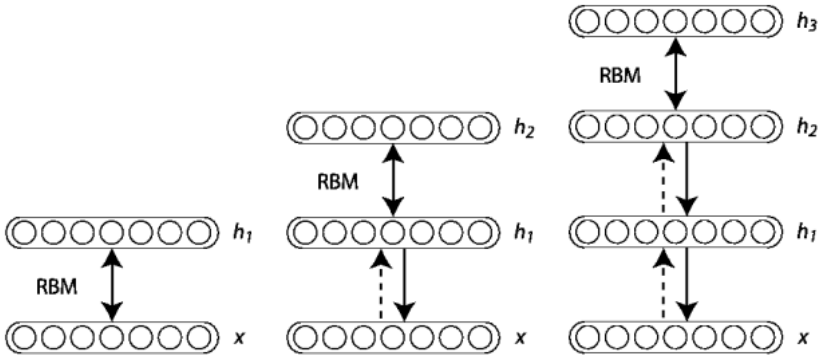

Figura 13. Addestramento layer-by-layer di una DBN tramite RBMs

Questo metodo efficiente può essere seguito da (oppure utilizzato in combinazione con) altre procedure di apprendimento allo scopo di realizzare un fine-tuning dei pesi per migliorare il potere generativo o discriminatorio della rete. Una procedura di fine-tuning discriminatoria può essere realizzata con l'aggiunta di un ultimo strato di variabili che rappresentano le uscite desiderate (es. realizzazione di classificatori) e con l'ausilio della back-propagation (BP). Hinton sostiene che, nelle reti con molti layer nascosti utilizzate per dati di ingresso altamente strutturati (es. immagini), la BP funzioni molto meglio se i pesi della rete vengono inizializzati dall'addestramento di una DBN che modelli la struttura dei dati in ingresso [9]. Globalmente quindi, le DBNs risultano un metodo semi-supervised in quanto il pre-training è unsupervised e il fine-training è supervised. Questa proprietà permette la

proprietà delle BMs è che le connessioni sono bidirezionali ovvero non c'è distinzione tra unità di ingresso e unità di uscita. Risultano però molto lente nelle reti con molti layer; si ricorre quindi all'utilizzo delle RBMs (ristrette ad avere un solo layer) in modo da ottenere un apprendimento veloce. Questo è possibile addestrando due livelli adiacenti alla volta trattandoli come layer di ingresso e uscita.

realizzazione di classificatori e auto-encoders[5].

L'addestramento di un classificatore si realizza in due step in modo semi-supervised:

- si inizializza la rete tramite procedura di pre-training utilizzando dati non etichettati
- si effettua il fine-training in modo supervisionato per esempio tramite back-propagation

Questo metodo risulta migliore rispetto alla semplice BP per i motivi illustrati di seguito:

- la procedura di pre-training operando layer-by-layer offre una buona scalabilità per reti di elevate dimensioni
- il fine-training non viene avviato finché i pesi della rete non determinano un buon comportamento della rete
  - in questo modo per esempio la BP effettuerà una ricerca locale per minimizzare l'errore
- la maggior parte dell'informazione contenuta nei pesi finali proviene dall'aver modellato la distribuzione dei vettori di input
  - gli esempi etichettati vengono utilizzati solo per il fine-tuning; quindi, essendo solitamente in numero molto minore rispetto ai dati non etichettati, permettono di velocizzare procedure come la BP
  - i dati non etichettati sono utilissimi per individuare buone features

L'architettura, intesa come numero di layer e di unità, può essere determinata con i criteri comuni utilizzati con le reti neurali classiche. Il layer di ingresso sarà caratterizzato da un numero di nodi pari alla lunghezza del

[5]Un auto-encoder è una rete neurale che apprende una rappresentazione compatta (encoding) di un set di dati per realizzare una riduzione della dimensionalità.

vettore di dati raw che vogliamo classificare. Per esempio nel caso di un'immagine di 28x28 pixel otteniamo un layer di input composto da 784 unità. Il layer di uscita dovrà avere tante unità quante sono le classi; in questo modo un esempio supervisionato può essere rappresentato in codifica *one-hot*.

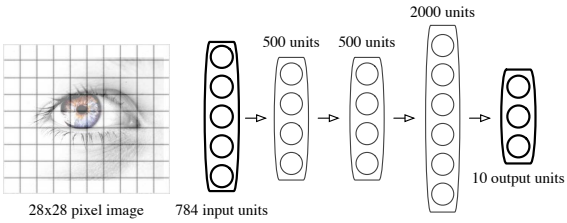

Figura 14. Esempio architettura classificatore DBN multiclasse per immagini 28x28 pixel

In [9] Hinton suggerisce alcune modifiche in base ad alcune proprietà del dataset. In particolare è importante sapere che, nel caso in cui i dati non etichettati siano in quantità elevata rispetto a quelli etichettati, la sovrapposizione di un processo gaussiano nei layer più profondi può apportare un miglioramento. Altrettanto importante è scegliere il tipo di unità visibili per la rete: nel caso di dati quantizzati sono adatte le tipiche unità sigmoidali mentre per dati continui è spiegato come risultino più adatte unità gaussiane.

2.3 Soluzioni adottate

Nel capitolo 1 è stata illustrata la scelta delle features; data l'elevata dimensionalità dei vettori che rappresentano gli elementi audio sono necessari molti esempi per addestrare un classificatore. Metodi supervisionati richiedono quindi elevati investimenti di risorse per l'etichettamento e il bilanciamento del training set. E' interessante valutare quindi metodi semi-supervised vista la difficoltà nel recuperare esempi etichettati.
E' preferibile utilizzare metodi di ML in cui i parametri siano pochi, che non necessitano di una conoscenza a priori e tali che, se perturbati, la differenza di prestazioni non sia elevata. Questi vincoli costituiscono le premesse per poter ottenere una fase di fine-tuning che richieda risorse minime.

I classificatori scelti per condurre i test sono classificatori DBNs e le SVMs. La prima scelta permette di addestrare sia in modo supervised che semi-supervised e ha parametri, come il numero di layer e il numero di nodi, che se leggermente perturbati non provocano grandi variazioni nelle prestazioni. La scelta si un metodo classico di successo come SVM è stata effettuata principalmente per avere delle performance di riferimento da confrontare con i classificatori DBMs. Inoltre i parametri delle SVMs sono determinabili senza una conoscenza a priori e tramite procedure basate sull'analisi del training set. In base al kernel utilizzato i parametri hanno significati e proprietà di stabilità differenti; in generale la perturbazione di un parametro per una SVM può provocare una variazione delle prestazioni elevata.

Capitolo 3

Esperimenti

Prima di illustrare gli esperimenti condotti e i relativi risultati segue un riepilogo in cui si riassumono obiettivi e scelte progettuali.

Questo lavoro è nato dall'esigenza di integrare informazioni estraibili dal segnale audio presente negli audiovisivi in un sistema di Knowledge Representation basato su ontologie il quale, al momento, prevede la sola analisi video. Gli studi si sono concentrati sulla parte di estrazione di informazioni di basso e medio livello semantico. Il sistema in questione non è un sistema di Information Retrieval generalizzato ma opera nel contesto dell'analisi di video sportivi, in particolare partite del gioco del calcio. Più precisamente quindi, l'obiettivo di questo lavoro è la valutazione della fattibilità e dell'accuratezza di un classificatore audio che individui i principali elementi ricorrenti in una partita di calcio trasmessa da un'emittente televisiva. Tali elementi sono il parlato e il pubblico; è inoltre utile distinguere il carattere emozionale di queste fonti.

Consultando alcune delle pubblicazioni pertinenti più recenti, è stato trovato riscontro nell'interesse ad effettuare una classificazione emozionale basandosi sul solo segnale audio estratto dai video sportivi. Dai risultati pubblicati sono state studiate features e metodi di classificazione. Come illustrato nei capitoli precedenti, la scelta è stata quella di utilizzare DBNs e SVMs per la

realizzazione dei classificatori e di provare sia addestramenti supervised che semi-supervised.
Riguardo alle features è stato scelto di utilizzare i Mel-frequency Cepstral Coefficients (MFCCs), estratti su frame della durata di alcune decine di millisecondi, e il logaritmo dell'energia calcolato sull'intera clip. Le unità da classificare consistono in clip audio della durata di 2 secondi.
Infine è stato scelto di non prevedere l'uso di filtri di pre-enfasi e di non adottare tecniche di riduzione della dimensionalità.

3.1 Dataset

Segue una descrizione di tutti gli aspetti inerenti la costruzione del dataset. In particolare viene documentata l'origine dei dati ed illustrati i processi di importazione e di etichettamento; viene poi affrontato il problema della normalizzazione e infine vengono elencate le problematiche emerse evidenziando quali limitazioni hanno apportato agli esperimenti.

3.1.1 Origine dei dati

Il dataset è stato generato utilizzando registrazioni di partite di calcio in cui il parlato è in lingua italiana. Le registrazioni sono state trasmesse da emittenti televisive italiane (tabella 3.1); la codifica audio utilizzata dal distributore è AC3, sampling rate 48kHz, 2 canali (stereo). Il distributore è SPORT SYSTEM EUROPE srl (http://www.sportsystem.com/).

3.1.2 Processo di importazione

Il dataset originale consiste in un set di video in formato DVD. Sono state estratte le tracce audio in unico file per poi essere segmentate in clip di durata fissa pari a 2,048[1] secondi. Le clip estratte risultano circa diciottomila.

[1] questo valore coincide con ottenere 32768 campioni per un canale con sampling rate pari a 16kHz; tale quantità, essendo una potenza del due, permette di ottenere prestazioni migliori nella fase di estrazione delle features

Torneo	Team 1	Team 2	Emittente	Data
Champions League	Olympique L.	Real Madrid	Rete 4	13/09/2005
Euro 2004	Italia	Bulgaria	Rai 1	22/06/2004
Euro 2004	Inghilterra	Svizzera	Rai 1	17/06/2004
Bundesliga	Shalke 04	H.Rostock	Stream	21/01/2001
Premier League	Arsenal	Chelsea	Tele+ B	13/01/2001
Champions League	Barcelon	Leeds	Stream	13/09/2000
Coppa Uefa	Lierse	Bordeaux	Eurosport	12/04/2000

Tabella 3.1. Dettagli origine del dataset

In fase di estrazione è stato effettuato il downsampling[2] a 16 kHz ed è stata applicata la riduzione da canale stereo a canale mono. L'applicazione di queste operazione riduce sensibilmente la quantità di dati senza provocare una perdita di informazione utile. Infatti, nella trasmissione televisiva di partite di calcio commentate, i due canali stereo trasmettono lo stesso flusso audio. Per quanto riguarda il downsampling, questa operazione mantiene la parte di segnale compresa tra 0 e 8kHz: l'intervallo è più che sufficiente in quanto la quasi totalità dell'energia dei segnali da analizzare è contenuta in tale banda.
L'importazione è stata implementata tramite scripting Bash[3] e i tool da riga di comando disponibili con il software MPlayer[4].

3.1.3 Labelling

Il labelling è stato effettuato tramite l'utilizzo di uno script che offre le seguenti funzionalità:

[2]operazione di riduzione della frequenza di campionamento; consiste nell'applicare un filtro anti-aliasing al segnale decodificato e nel ricodificare utilizzando un numero ridotto di campioni

[3]nota shell per sistemi operativi Unix-like

[4]applicazione media player cross-platform, free e opensource, distribuita sotto licenza GPL

- propone l'ascolto di ogni clip per un numero di volte desiderato dall'utente;
- ordina le clip da ascoltare ed etichettare in ordine casuale in modo da non permettere che l'utente sia influenzato dal contesto;
- permette di fornire un valore di classificazione o in alternativa di contrassegnare una clip come *scartata*
- permette l'etichettamento da parte di più utenti gestendo il locking delle clip audio;
- permette di interrompere l'etichettamento senza perdere il lavoro.

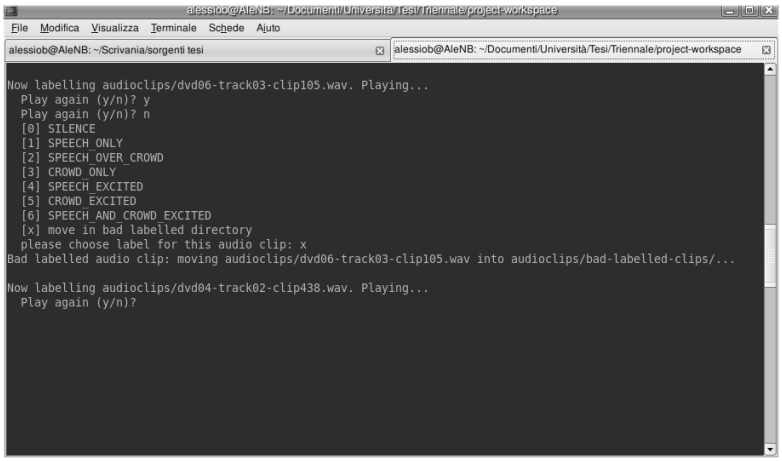

Figura 15. Interfaccia assistente etichettamento

Durante l'etichettamento sono state individuate clip relative a pubblicità e inni; tali clip sono state scartate in quanto non rientrano negli obiettivi di classificazione. Per individuare facilmente i contenuti da scartare e i contenuti di classi poco frequenti (es. silenzio), è stata adottata la seguente procedura. Inizialmente sono state etichettate circa tremila clip prese in ordine random; dopodiché, sfruttando i nomi dei file attribuiti alle clip (es. dvd01-track02-clip123.wav) e i valori di classificazione inseriti, sono state valutate manualmente le clip adiacenti a quelle per cui l'etichetta ha il valore che si ricerca.

Come illustrato in dettaglio nella sezione 3.1.5, la fase di labelling è risultata un'operazione difficile. Inizialmente è stato utilizzato un dataset contenente circa 3800 esempi (fig. 16, dataset "alpha"). Tale dataset contiene molti esempi non netti, ovvero la cui appartenenza ad una sola classe è un'operazione soggettiva e soggetta ad attribuzioni differenti anche da parte dello stesso valutatore.

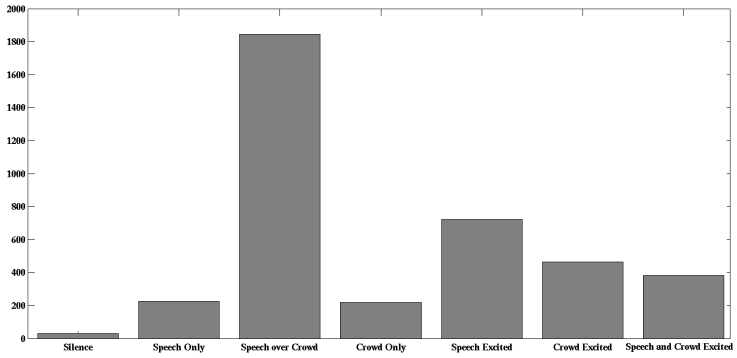

Figura 16. Distribuzione classi dataset "alpha"

Successivamente è stato creato un dataset contenente circa 1300 esempi (fig. 17, dataset "beta") relativamente obiettivi, ovvero il cui valore di classificazione è facile da determinare e sul quale si troverebbero in accordo più valutatori.

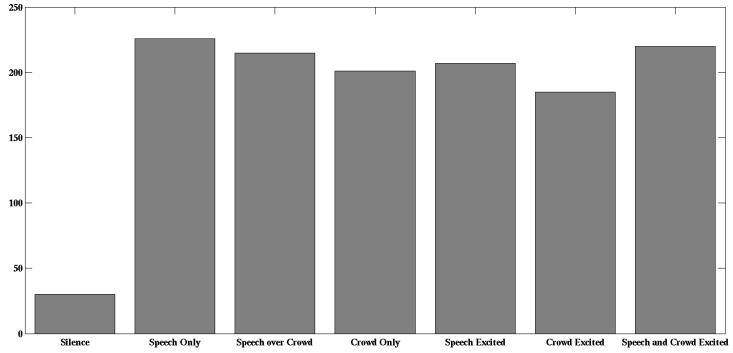

Figura 17. Distribuzione classi dataset "beta"

Infine, a causa di una bassa accuratezza riscontrata con tutti i classificatori sulla classe Crowd Only, è stato modificato il dataset alpha aggiungendo clip relative al pubblico emotivamente eccitato e non (dataset "gamma"). L'individuazione di esempi appartenenti a tale classe è difficoltosa in quanto molto più frequentemente il parlato dei telecronisti risulta sovrapposto. Sono state quindi individuate partite trasmesse in broadcast in coincidenza di scioperi della regia: è risultato più facile individuare circa 150 clip da aggiungere alle classi relative alla folla.
Questi dataset sono risultati utili per verificare il comportamento dei classificatori nei casi difficili. I dettagli sui risultati ottenuti verranno presentati nelle prossime sezioni.

3.1.4 Normalizzazione

In base alle caratteristiche delle implementazioni dei classificatori, è opportuno normalizzare i dati. Il dataset è composto dai coefficienti MFCs (valori reali); per tali dati non sono noti il massimo e il minimo assoluti. E' quindi necessario stimare questi valori e utilizzare una funzione di normalizzazione che minimizzi la probabilità di incorrere nella saturazione.
E' stato effettuato un semplice studio per realizzare una normalizzazione che permetta di accettare in input qualsiasi possibile valore. Segue una sintesi di questa analisi.

- è preferibile che la funzione di normalizzazione sia lineare in modo da non perdere le proprietà dello spazio di partenza (il rischio è introdurre una relazione non lineare non presente nel modello reale; questo porterebbe a peggiorare le prestazione del classificatore);

- un'eccessiva compressione dei dati può portare a ridurre il potere rappresentativo nello spazio di arrivo della mappa di normalizzazione in quanto, a causa di una conseguente approssimazione operata dalla macchina, punti vicini potrebbero essere mappati in uno stesso punto;

- siano v_{min} e v_{max} i valori di minimo e massimo stimati per i dati di ingresso (in generale queste coppie di valori sono distinte per ogni feature

utilizzata) e sia
$p_{outlier} = \int_{x \notin [v_{min}, v_{max}]} p(x)dx$ con $p(x)$ la distribuzione reale dei valori
la probabilità che sopraggiunga un valore fuori dall'intervallo $[v_{min}, v_{max}]$;
se tale probabilità non è trascurabile, e ampliare l'intervallo porta ad ottenere un'eccessiva compressione dei dati, allora è ragionevole provare ad utilizzare una funzione monotona crescente di classe C^1 (nota[5]).

Si è quindi reso necessario individuare una tecnica per stimare empiricamente $p_{outlier}$. Utilizzando tutte le clip audio disponibili è stato creato un piccolo set di bipartizioni di tutte le clip (rapporto tra le partizioni 5:1). Per ogni set sono stati estratti i valori v_{min} e v_{max} dalla partizione più grande e sono stati contati i valori fuori da tale intervallo presenti nella partizione più piccola. Gli outliers conteggiati sono risultati essere trascurabili ($p_{outlier} \approx 1,9 \times 10^{-6}$).
E' stata quindi scartata la scelta di una funzione di classe C^1 ed è stata selezionata una funzione che comprima i dati di ingresso attraverso una costante (valore utilizzato 0.96) per poi applicare un normalizzazione lineare e la saturazione agli estremi 0 e 1 per i valori fuori range. La compressione dei valori operata dall'applicazione della costante realizza un margine dalla saturazione che si rivela utile nell'eventualità che un valore sia fuori range.

Dato l'operatore $saturate(x)$ definito dalla 3.1.1:

Definizione 3.1.1 (Saturation Function)

$$saturate(x) = \begin{cases} 0 & for \quad x < 0 \\ 1 & for \quad x > 1 \\ x & for \quad x \in [0,1] \end{cases}$$

si definisce la funzione di normalizzazione 3.1.2:

[5] l'appartenenza alla classe C^1 permette di realizzare una saturazione progressiva ovvero esclude funzioni in cui la saturazione risiede in tratti a derivata nulla adiacenti a un tratto lineare

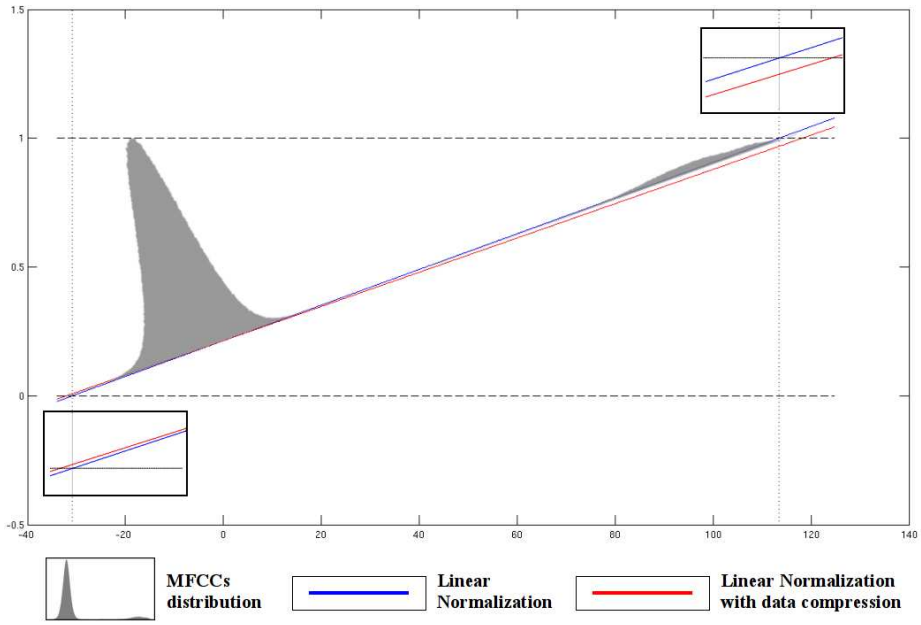

Figura 18. Normalizzazione lineare e distribuzione MFCCs

Definizione 3.1.2 (Normalization Function)

$$normalize(x) = saturate\left(\frac{compression\ factor \times x - v_{min}}{v_{max} - v_{min}}\right)$$

Analogamente alla procedura di stima di $p_{outlier}$, nella fase di addestramento v_{min} e v_{max} vengono estratti valutando solamente il training set. Tali valori vengono utilizzati come parametri della funzione di normalizzazione la quale verrà applicata a tutte le tipologie di dati: dati appartenenti al training set, al testing set o dati relativi a nuovi esempi da classificare.

3.1.5 Problematiche

Il problema principale emerso in questo lavoro riguarda la creazione del dataset. Questa fase è fondamentale in quanto determinante per l'addestramento del classificatore. La principale difficoltà risiede nel creare un dataset che risulti bilanciato, popolato di esempi sia facili che difficili e in cui le etichette attribuite siano quanto più frutto di una valutazione oggettiva.
Un dataset risulta *bilanciato* quando la distribuzione delle classi attribuite

agli esempi che lo compongono risulta approssimativamente uniforme. La limitata disponibilità di registrazioni e il fatto che in ogni partita la distribuzione delle classi non è omogenea[6], ha reso difficile il collezionamento di un numero sufficiente di esempi per creare un dataset bilanciato. In aggiunta a questo fatto, le risorse di tempo e di disponibilità di persone da assegnare alla classificazione manuale sono risultate limitate. Infine la valutazione emozionale da parte di una persona è decisamente soggettiva (in particolare nei casi difficili). E' risultato utile definire dei criteri per svolgere l'attività di etichettamento manuale:

- ascoltare le clip in sequenza random per non essere influenzati dal contesto (il classificatore lavora sotto l'implicita ipotesi di indipendenza tra clip da classificare in sequenza);
- nel caso in cui un evento audio risulti dominante su altri eventi (es. il pubblico esulta e distoglie l'attenzione dal parlato) si assegna la classe relativa all'evento dominante;
- nel caso in cui una clip audio risulti essere posizionata "a cavallo"tra due eventi audio si assegna l'evento dominante (per esempio facendo attenzione a quale evento è rimasto impresso dopo l'ascolto).

Queste problematiche potrebbero essere risolte facendo svolgere l'etichettamento ad un gruppo di persone e, data una soglia di maggioranza, selezionando solo le clip etichettate allo stesso modo da un numero di persone che supera la soglia di maggioranza.
Inoltre questa soluzione potrebbe risultare utile per determinare "l'accuratezza umana"relativamente alla classificazione emozionale. Questo risultato può essere utilizzato per valutare le prestazioni dei classificatori in maniera più precisa.

[6]in una partita di calcio gli eventi di eccitazione solitamente sono meno frequenti rispetto agli eventi emozionalmente rilassati

3.2 Risultati

Per condurre gli esperimenti le risorse software utilizzate sono Matlab e libsvm (`http://www.csie.ntu.edu.tw/~cjlin/libsvm/`). Matlab viene utilizzato per la costruzione del dataset (lettura delle clip audio, estrazione delle features, creazione dei subset necessari per l'addestramento) e per addestrare le DBNs; libsvm viene utilizzata per addestrare le SVMs. I risultati dei classificatori vengono infine analizzati tramite Matlab per la generazione di report (in particolare per generare le matrici di confusione). E' inoltre disponibile un set di script per classificare un video e generare un file di sottotitoli con le classificazioni relative al contenuto audio.
Le classi con cui il dataset è stato etichettato sono:

- silenzio (silence);
- solo parlato (speech only);
- parlato sopra la folla (speech over crowd);
- solo folla (crowd only);
- parlato eccitato (speech excited);
- folla eccitata (crowd excited);
- parlato e folla eccitati (speech and crowd excited).

Le ultime tre classi sono state ridotte in unica classe chiamata *eccitazione* (excited) in quanto l'accuratezza ottenuta senza la riduzione non è soddisfacente (60% di accuratezza globale con tutte le sette classi). Inoltre la valutazione delle prestazioni dei diversi classificatori su un problema più semplice può essere utile per individuare il classificatore con il quale realizzare un'architettura più complessa (es. classificazione gerarchica).
Per l'addestramento il dataset beta si è rivelato non adeguato: a differenza del primo dataset risulta bilanciato e contenente meno errori ma testando il classificatore su un video, non utilizzato per la costruzione del training set, l'accuratezza è risultata non soddisfacente (durante l'osservazione del video

etichettato frequentemente è stato riscontrato un valore di classificazione non valido).

Seguono quindi i risultati dei diversi classificatori, ottenuti sul dataset alpha utilizzando come features le sole MFCCs con frame da 64ms e overlap pari al 50% e riducendo le etichette da sette a cinque classi.

3.2.1 SVMs supervisionate

Il primo classificatore realizzato tramite SVMs è stato addestrato sul dataset alpha ed è caratterizzato dai seguenti parametri:

- SVM kernel function: Radial Basis[7];
- c = 1.41421356237;
- gamma = 1.41421356237.

L'accuratezza globale ottenuta sul testing set è pari al 74.11%, l'accuratezza tramite 3-fold cross validation è pari a 73.97%. I risultati relativi alle classi distinte sono riportati nella matrice di confusione (fig. 19).

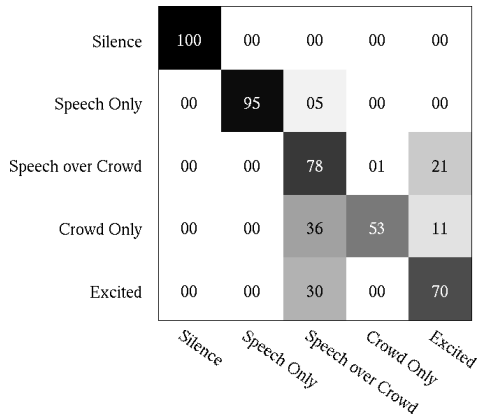

Figura 19. Matrice di confusione SVM kernel RB (dataset alpha)

[7]è stata addestrata in modo analogo una SVM utilizzando il kernel chi-square; i risultati sono analoghi a quelli ottenuti con il kernel radial basis quindi vengono omessi

La classe Crowd Only ha un'accuratezza molto bassa; in particolare il 36% delle istanze di tale classe vengono classificate come Speech over Crowd. Osservando la distribuzione delle classi nel dataset è ragionevole aspettarsi che la causa sia un numero insufficiente di esempi nel training set.
Il classificatore ha problemi nel distinguere le classi Speech over Crowd ed Excited: questo risultato potrebbe essere interpretato come necessità di aggiungere un livello di eccitazione intermedio per classificare esempi difficili.
Il silenzio viene riconosciuto sempre; la classe Speech Only ha un'elevata accuratezza (95%): questo permette di distinguere il parlato nei rientri in studio dal parlato dei telecronisti con gioco in corso (ottima informazione per operare una segmentazione).

Data la scarsa accuratezza ottenuta sulla classe Crowd Only, è stata addestrata un'altra SVM sul dataset gamma. L'accuratezza globale è rimasta praticamente invariata (accuratezza testing set 74.13%), ma dalla matrice di confusione rappresentata nella figura 20 si rileva un elevato incremento nel riconoscere la classe Crowd Only (tenere conto la distribuzione delle classi non omogenea):

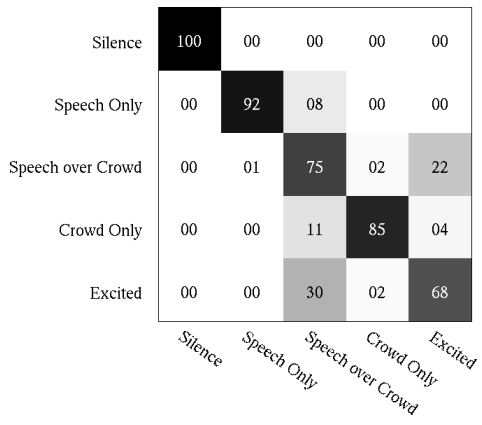

Figura 20. Matrice di confusione SVM kernel Chi-Square (dataset gamma)

I parametri individuati per il kernel utilizzato sono:

- c = 2.37841423001;

- gamma = 0.5.

3.2.2 DBNs supervisionate

Il classificatore DBN addestrato in modo supervisionato con dataset alpha è caratterizzato dai seguenti parametri:

- numero layer hidden: 3;
- numero unità layer hidden: 500 x 500 x 2000;
- tipo unità layer hidden: logistic (attivazione sigmoidale);
- pre-training:
  - numero cicli: 50;
  - learning rate connessioni: 0.1;
  - learning rate per le polarizzazione delle unità visibili: 0.1;
  - learning rate per le polarizzazione delle unità nascoste: 0.1;
  - valore iniziale dei pesi: 0.0002;
  - momento iniziale: 0.5;
  - momento finale: 0.9;
- epoche fine-training (BP): 200.

L'accuratezza globale, valutata sul testing set, è pari al 70.31% in questo caso, quindi inferiore di circa 4 punti percentuali rispetto al classificatore precedente (SVM supervisionato). Nella figura 21 viene riportata la relativa matrice di confusione.

Anche in questo caso è stata addestrata un'ulteriore DBN utilizzando il dataset gamma; l'accuratezza sulla classe Crowd Only è incrementata ma in misura decisamente minore rispetto all'incremento riscontrato con il rispettivo classificare SVM. Nella figura 22 è riportata la matrice di confusione del classificatore DBN in questione.

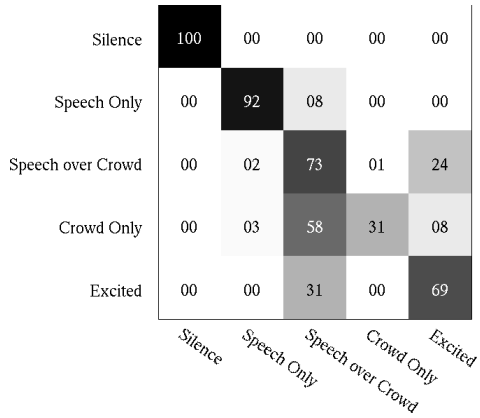

Figura 21. Matrice di confusione supervised DBN (dataset alpha)

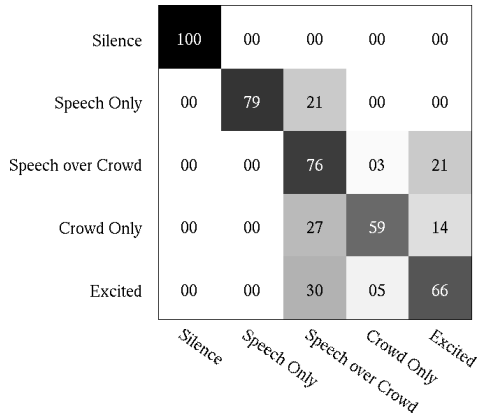

Figura 22. Matrice di confusione supervised DBN (dataset gamma)

3.2.3 DBNs semi-supervisionate

E' stato addestrato un classificatore DBN in modo semi-supervisionato analogo al classificatore DBN supervisionato presentato precedentemente. I risultati ottenuti sono leggermente inferiori a quelli ottenuti dall'addestramento supervisionato. In particolare utilizzando il dataset alpha, l'accuratezza globale valutata sul testing set risulta pari al 69.85% e in figura 23 è riportata la matrice di confusione.

Anche l'addestramento semi-supervisionato sul dataset gamma non ha prodotto un miglioramento simile a quello ottenuto con le SVMs; l'accuratezza globale sul testing set risulta pari al 69.70% e la matrice di confusione è quella

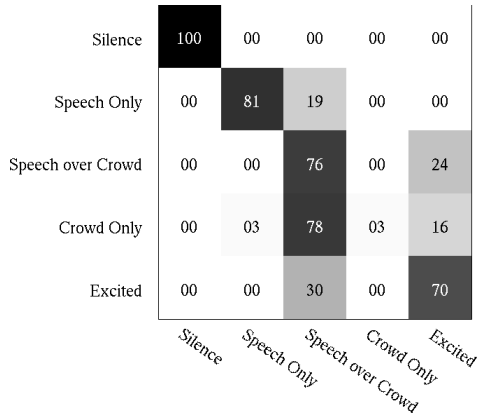

Figura 23. Matrice di confusione semi-supervised DBN (dataset alpha)

riportata in figura 24.

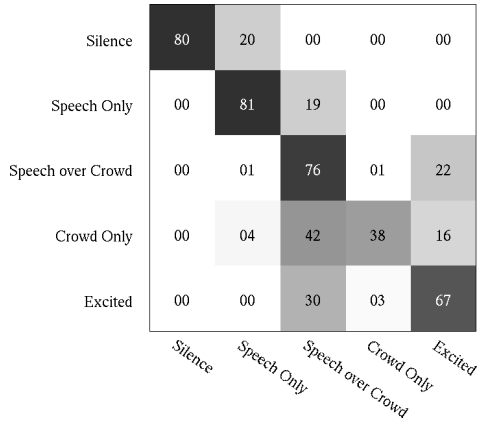

Figura 24. Matrice di confusione semi-supervised DBN (dataset gamma)

Il peggioramento delle prestazioni rispetto alla soluzione supervisionata è riconducibile alla bassa robustezza delle DBNs al rumore: utilizzare un set più ampio di clip in fase di pre-training non porta vantaggi in questa applicazione ma contribuisce solo ad osservare ulteriori dati rumorosi.

3.3 Esempi

Oltre alle valutazioni classiche è stata effettuata una valutazione dei classificatori su alcune sequenze video. I risultati illustrati sono di carattere qualitativo ma mostrano efficacemente come l'analisi audio contenga un'informazione di alto livello estraibile.
Nella figura 25 vengono mostrati alcuni frame dai quali si evince come la classificazione audio proposta sia utilizzabile per effettuare una segmentazione temporale. Per esempio possono essere distinti gli intervalli in cui la regia trasmette gli eventi dal campo dai rientri in studio o la pubblicità.

Figura 25. Frame eventi vari (classificatore SVM Radial Basis addestrato su dataset alpha)

La classe Excited si rivela decisamente efficace nell'individuare azioni salienti: nella figura 26 vengono mostrati alcuni frame relativi ad un calcio di rigore; tutti sono etichettati come Excited. Anche durante la ripresa del pubblico l'audio fornisce un'utile informazione di cosa sta accadendo.

Un esempio analogo è mostrato nella figura 27: vengono riportati alcuni frame relativi ad un calcio di rigore e al rispettivo highlight trasmesso dalla regia[8]. I frame (a) e (b) sono relativi ai momenti di attesa del calcio di rigore, il frame (c) è subito dopo il goal, i successivi frame sono relativi all'highlight in cui viene commentato il goal. La classificazione attribuita è interessante:

[8]Il video proposto non è stato utilizzato per l'addestramento ed è stato etichettato con il classificatore SVM (kernel Chi-Square, addestramento su dataset gamma).

Figura 26. Frame calcio di rigore (classificatore SVM Radial Basis addestrato su dataset alpha)

può essere utilizzata per aiutare l'analisi video a distinguere un goal dal suo highlight.

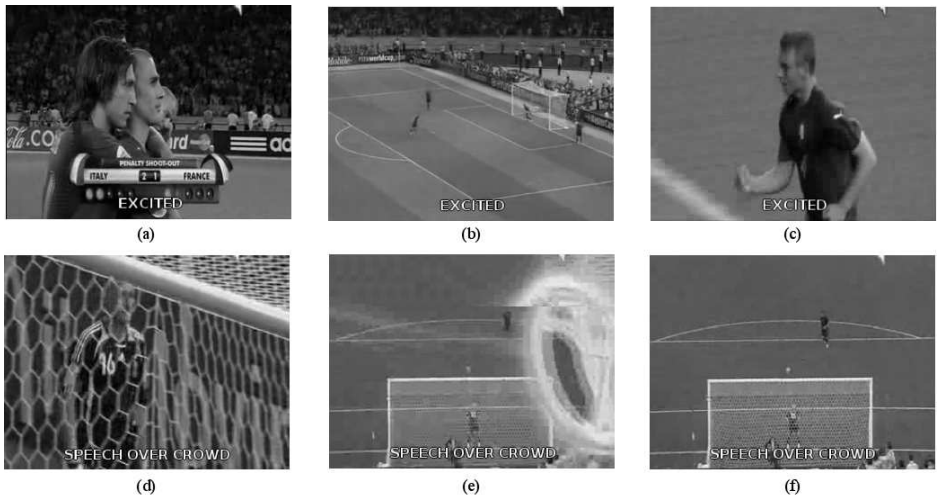

Figura 27. Frame calcio di rigore e relativo highlight (classificatore SVM Chi-Square addestrato su dataset gamma)

3.4 Valutazione classificatori

Dai risultati è evidente il vantaggio nell'utilizzare classificatori basati su SVMs. Questa soluzione è una scelta comune e l'implementazione utilizzata, libsvm, è ben documentata e consolidata. Per quanto riguarda le DBNs, i tempi di addestramento sono analoghi a quelli richiesti dalle SVMs ma non

ci sono ancora implementazioni di riferimento. Inoltre la caratteristica di individuare momenti statistici di alto ordine nei dati, propria delle DBNs, ha lo svantaggio di diminuire la robustezza al rumore. Sono quindi necessarie tecniche di denoising che però sono state proposte solo recentemente e per le quali non sono disponibili implementazioni.
Un problema comune a tutti i classificatori realizzati è il seguente: la classificazione di video esterni al dataset spesso non funziona (viene attribuita quasi esclusivamente la classe Excited). Ciò è dovuto sicuramente alla qualità del segnale audio che risulta spesso saturato: può essere interessante studiare features robuste a questo tipo di disturbo. Un'altra causa può essere un bias elevato dei classificatori dovuto all'aver utilizzato un numero non sufficiente di differenti emittenti televisive.

Conclusioni

In questo lavoro di tesi è stato condotto uno studio sui sistemi di Information Retrieval per contenuti multimediali basati sull'analisi audio; in particolare sono stati realizzati e confrontati classificatori DBNs e SVMs per la classificazione di elementi audio ricorrenti nei video sportivi trasmessi in broadcast. Il dataset utilizzato è costituito da alcune partite di calcio trasmesse da emittenti televisive italiane.

I risultati sperimentali hanno confermato la possibilità di realizzare una classificazione audio utilizzando tecniche di machine learning sia supervisionate che semi-supervisionate, consentendo l'estrazione di informazioni di alto livello semantico. Tali informazioni risultano utili per operare un'annotazione semantica degli audiovisivi congiuntamente ad altri elementi estratti da differenti domini (es. analisi video).

Sviluppi futuri includono la costruzione di un dataset bilanciato, classificato da più osservatori e di più ampio spettro (es. estendere ad emittenti anche straniere) così come lo studio di un'architettura gerarchica di classificatori per estendere le classi rilevabili e cercare di ottenere prestazioni migliori. Infine è utile quantificare i possibili vantaggi dati dall'integrazione dell'analisi audio in un sistema completo di Multimedia Information Retrieval.

Appendice A: features audio

Segue una lista delle feature audio più comuni.

Temporal Features

Global Temporal Features

Log Attack Time

Temporal Increase

Temporal Decrease

Temporal Centroid

Effective Duration

Instantaneous Temporal Features

Signal Auto-correlation function

Zero-corssing rate

Energy Features

Total energy

Total energy Modulation (frequency, amplitude)

Total harmonic energy

Total noise energy

Spectral Features

Spectral Shape

Spectral centroid

Spectral spread

Spectral skewness

Spectral kurtosis

Spectral slope

Spectral decrease

- Spectral rolloff
- Spectral variation

Harmonic Features

- Fundamental frequency
- Fundamental frequency Modulation (frequency, amplitude)
- Noisiness
- Inharmonicity
- Harmonic Spectral Deviation
- Odd to Even Harmonic Ratio
- Harmonic Tristimulus
- *Harmonic Spectral Shape*
  - Harmonic Spectral centroid
  - Harmonic Spectral spread
  - Harmonic Spectral skewness
  - Harmonic Spectral kurtosis
  - Harmonic Spectral slope
  - Harmonic Spectral decrease
  - Harmonic Spectral rolloff
  - Harmonic Spectral variation

Perceptual Features

- MFCCs
- Loudness
- Relative Specific Loudness
- Sharpness
- Spread
- *Perceptual Spectral Envelope Shape*
  - Perceptual Spectral centroid
  - Perceptual Spectral spread
  - Perceptual Spectral skewness
  - Perceptual Spectral kurtosis
  - Perceptual Spectral Slope
  - Perceptual Spectral Decrease
  - Perceptual Spectral Rolloff

Perceptual Spectral Variation
Odd to Even Band Ratio
Band Spectral Deviation
Band Tristimulus

Various features

Spectral flatness
Spectral crest

Appendice B: risorse software

Segue una lista di risorse software per l'analisi audio.

MoCA: Movie Content Analysis Libreria per sviluppare sistemi IR per contenuti audio visivi. In particolare per l'audio venongo analizzati musica, parlato ed elementi chiave per un determinato contesto (es. elementi audio tipici di scene di violenza).
`http://www.informatik.uni-mannheim.de/pi4/projects/moca/Project-automaticAudioContentAnalysis.html`

CoMIRVA: Collection of Music Information Retrieval and Visualization Applications L'obiettivo di questo progetto è la costruzione di un framework per l'implementazione in Java di algoritmi per trattare segnali musicali e contenuti multimediali, per realizzare sistemi IR e per effettuare data mining. Al momento consiste in una serie di packages per l'analisi audio. Il progetto è sviluppato e mantenuto da Markus Schedl; è pubblicato con licenza GPL.
`http://www.cp.jku.at/people/schedl/Research/Development/CoMIRVA/webpage/CoMIRVA.html`

CLAM: C++ Library for Audio and Music La libreria C++ e i tool ad interfaccia grafica si propongono come ambiente per la ricerca e lo sviluppo di applicazioni nel dominio audio.
`http://www.clam.iua.upf.edu/`

Beat Detection (libreria MATLAB) Implementazione Matlab di un algoritmo per la rilevazione del beat.
http://www.owlnet.rice.edu/~elec301/Projects01/beat_sync/beatalgo.html

AuditoryToolbox for Matlab Toolbox Matlab per l'analisi audio.
http://cobweb.ecn.purdue.edu/~malcolm/interval/1998-010/

Deep Belief Nets con PyPlearn Esempi di implementazione delle DBNs su framework PLearn tramite modulo PyPLearn (interfaccia Python).
http://www.iro.umontreal.ca/~lisa/twiki/bin/view.cgi/Public/DeepBeliefNetworks
http://plearn.org/

Deep Belief Nets con Matlab Implementazione di G. Hinton di un classificatore e di un auto-encoder tramite DBNs in Matlab per il dataset pubblico MNIST. Questo codice è stato riadattato per condurre gli esperimenti presentati in questo lavoro di tesi.
http://www.cs.toronto.edu/~hinton/MatlabForSciencePaper.html

CMU Sphinx - Open Source Speech Recognition Engines Progetto della Carnegie Mellon University per la realizzazione di un motore di speech recognition. L'obbiettivo è facilitare lo sviluppo di applicazioni e di tools.
http://cmusphinx.sourceforge.net/html/cmusphinx.php

VoxForge L'obiettivo del progetto è collezionare un set di modelli acustici per il riconoscimento vocale da rilasciare in licenza GPL. I modelli disponibili sono compatibili con motori di sspeech recognition come Sphinx.
http://www.voxforge.org/

Bibliografia

[1] Bishop, C. M. *Pattern Recognition and Machine Learning*, first ed. Springer, ISBN: 0387310738, 2007.

[2] Bogert, B. P., Healy, M. J. R., and Tukey, J. W. The quefrency alanysis of time series for echoes: cepstrum, pseudo-autocovariance, cross-cepstrum, and saphe cracking, 1963.

[3] Burges, C. J. C. A tutorial on support vector machines for pattern recognition, 1998. *Data Mining and Knowledge Discovery 2*, 2 (1998), 121–167.

[4] Cendrowska, J. Prism: An algorithm for inducing modular rules, 1987. *International Journal of Man-Machine Studies 27*, 4 (1987), 349–370.

[5] ching Chen, S., ling Shyu, M., Zhang, C., Luo, L., and Chen, M. Detection of soccer goal shots using joint multimedia features and classification rules. In *Proceedings of the Fourth International Workshop on Multimedia Data Mining (MDM/KDD)* (2003), pp. 36–44.

[6] Fang, Z., Zhang, G., and Song, Z. Comparison of different implementations of mfcc, 2001. *J. Comput. Sci. Technol. 16*, 6 (2001), 582–589.

[7] Fry, D. B. *The Physics of Speech.* Cambridge University Press, 1996.

[8] Haykin, S. *Neural Networks.* Prentice-Hall, 1999.

[9] Hinton, G. E. Tutorial on deep belief nets, 2007. Neural Information Processing Systems (NIPS).

[10] Hinton, G. E., and Salakhutdinov, R. R. Reducing the dimensionality of data with neural networks, 2006. *Science* (2006).

[11] Juang, B. H., and Rabiner, L. Hidden markov models for speech recognition, 1991.

[12] Kim, H., Roeber, S., Samour, A., and Sikora, T. Detection of goal event in soccer videos, 2005.

[13] Kompatsiaris, Y., and Hobson, P. *Semantic Multimedia and Ontologies Theory and Applications*, first ed. Springer, ISBN: 1848000758, 2008.

[14] Kyrgyzov, I. O., Kyrgyzov, O. O., Maître, H., and Campedel, M. Kernel mdl to determine the number of clusters. In *Proceedings of the International Conference on Machine Learning and Data Mining in Pattern Recognition (MLDM)* (2007), pp. 203–217.

[15] Leonardi, R., Migliorati, P., and Prandini, M. Semantic indexing of sports program sequences by audio-visual analysis. In *Proceedings of the International Conference on Image Processing (ICIP)* (2003), pp. 9–12.

[16] Otsuka, I., Nakane, K., Divakaran, A., Hatanaka, K., and Ogawa, M. A highlight scene detection and video summarization system using audio feature for a personal video recorder, 2005.

[17] Otsuka, I., Radhakrishnan, R., Siracusa, M., Divakaran, A., and Mishima, H. An enhanced video summarization system using audio features for a personal video recorder, 2006. [16].

[18] Pechenizkiy, M., Puuronen, S., and Tsymbal, A. The impact of sample reduction on pca-based feature extraction for supervised learning. In *Proceedings of the Symposium on Applied Computing (SAC)* (2006), pp. 553–558.

[19] Peeters, G. A large set of audio features for sound description, 2004.

[20] Peeters, G. Mirex 2005: Tempo detection and beat marking for perceptual tempo induction. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)* (London, UK, Septembre 2005).

[21] Pfeiffer, S., Fischer, S., and Effelsberg, W. Automatic audio content analysis. In *Proceedings of the ACM Multimedia* (1996), pp. 21–30.

[22] Rabiner, L. A tutorial on hidden markov models and selected applications in speech recognition. In *Proceedings of the Institute of Electrical and Electronics Engineers (IEEE)* (1989), IEEE, pp. 257–286.

[23] Sanderson, C., and Paliwala, K. K. Identity verification using speech and face information. In *Proceedings of the Digital Signal Processing* (2004), pp. 449–480.

[24] Schein, A. I., Popescul, A., Ungar, L., and Pennock, D. M. A generalized linear model for principal component analysis of binary data, 2003.

[25] Sciandrone, M. *Support Vector Machines – Lezioni*. Istituto di Analisi dei Sistemi ed Informatica A. Ruberti - CNR, Roma, 2006.

[26] Stevens, S., Volkman, J., and Newman, E. A scale for the measurement of the psychological magnitude of pitch, 1937. *Journal of the Acoustical Society of America* (1937).

[27] Vapnik, V. N. *The Nature of Statistical Learning Theory*. Springer, New York, NY, USA, 1995.

[28] Vapnik, V. N. *Statistical Learining Theory*. Wiley, New York, 1998.

[29] Vincent, P., Larochelle, H., Bengio, Y., and Manzagol, P.-A. Extracting and composing robust features with denoising au-

toencoders. In *Proceedings of the International Conference on Machine Learning (ICML)* (2008), pp. 1096–1103.

[30] Wang, J., Xu, C., Siong, C. E., and Tian, Q. Sports highlight detection from keyword sequences using hmm. In *Proceedings of the International Conference on Multimedia and Expo (ICME)* (2004), pp. 599–602.

[31] Xiong, Z., Radhakrishnan, R., Divakaran, A., and Huang, T. S. Effective and efficient sports highlights extraction using the minimum description length criterion in selecting gmm structures. In *Proceedings of the International Conference on Multimedia and Expo (ICME)* (2004), pp. 1947–1950.

[32] Zhu, X. Semi-supervised learning tutorial, 2007. International Conference on Machine Learning (ICML).

Ringraziamenti

Questo lavoro porta con se venti anni di passione per l'informatica: quando avevo cinque anni dicevo che da grande avrei voluto fare l'ingegnere dei computer. Ciò che documento di seguito sono i fatti che mi hanno condotto fin qui. E' doveroso documentare tutto questo perché ciò che sono oggi non è frutto dei soli miei sforzi.

Ringrazio il Signore per avermi donato questa immensa passione sin da quando ero piccolo.

Ringrazio il mio babbo che ha da subito assecondato la mia passione per l'informatica perché ha saputo leggere nei miei occhi quanta curiosità avevo per i computer. Quando avevo sei anni mi ha fatto il regalo che per me è il punto di partenza: il Commodore 64 con il corso di Basic. Ringrazio anche la mia mamma la quale, vedendomi troppo preso, me l'ha levato per un anno dalla circolazione: ho così continuato il mio ingenuo corso sui libri (che ancora conservo) e quando mi è stato reso il C64 avevo tante cose nuove da capire e provare.

Ringrazio Franco De Martis, insegnante di informatica alle scuole medie. Probabilmente non si ricorderà più di me ma io non potrò mai dimenticare come anche lui abbia assecondato la mia passione, quanto mi ha sopportato, a quante domande ha dovuto rispondere, quanta pazienza con il preside quando combinavo guai facendo danni ai computer della scuola, quanto affetto.

Ringrazio per la console SuperNintendo che mi è stata regalata per la Comunione: non per i giochi ma perché dopo alcuni anni ho fatto a cambio con un vicino di casa che mi ha dato un vecchio PC (Olivetti Prodest PC1, processore 8086 a 8MHz, 640K di RAM, floppy da 720KB, 20MB di harddisk) con il quale passavo il tempo a spippolare in DOS, con GWBASIC o con PCTOOLS.

Ringrazio ancora il Signore per quel giorno bellissimo che, annoiato, passeggiavo sotto casa e trovai un quadernino ad anelli piccolo di appunti di lezioni di DOS con la scrittura di una ragazza. Quante cose nuove da imparare!

Ringrazio Claudio Cefalà, Martina Compagnino e Stefano Chang per l'amicizia durante l'avventura delle scuole medie e per aver nascosto ai loro genitori i danni che combinavo nei loro computer.

Ringrazio lo zio Pino per avermi regalato il mio primo monitor e Francesco Sabatini (non Black, è un omonimo) per avermi venduto il mio primo vero PC (un 286 con case desktop e stampante ad aghi).

Ringrazio per i soldi ricevuti in regalo che ho messo via per acquistare un 486, una scheda audio e un lettore CDROM e poter finalmente far girare Windows 3.1 dentro casa mia.

Ringrazio i compagni del biennio alle scuole superiori Duccio, Alessio, Emanuele, Andrea e Maziar per aver spippolato e dedicato la quasi totalità delle nostre chiacchere a parlare dei computer che tanto ci appassionavano.

Ringrazio il Prof. Fiorenzo Burattin che ha condotto il gruppo di MTB delle scuole superiori per le colline e i monti toscani. Condividendo pedalate, sgrifate post-gara e risate ci ha voluti bene come fossimo suoi figli.

Ringrazio Claudio Turchetti che ha avuto fiducia in me e mi ha permesso di lavorare e imparare tante cose nuove sin da presto. Oltre alle grandi op-

portunità per cui gli sarò sempre grato, è stato come un fratellone maggiore.

Ringrazio i compagni della 5 INFO B: eravamo molto stupidi (molto) e per questo ancora adesso rido ma in qualche modo siamo cresciuti insieme e sono stati anni di grandi cambiamenti.

Ringrazio il Prof. Simone Lazzerini delle superiori che ci ha sempre guardati con stima e ci ha insegnato, per quanto possibile, ad essere un po' ingegneri e non si è mai tirato indietro di fronte alle nostre domande colme di curiosità e stupore. Ogni sua lezione mi ha letteralmente gasato.

Ringrazio la Prof. Elisabetta Guidi, insegnante di Italiano e Storia. Mi ha voluto davvero un gran bene e nel suo lavoro ha dato tanto per noi studenti. Ancora ricordo quando ci raccontò che un collega non voleva insegnare ai suoi studenti di un istituto tecnico Leopardi perché sarebbero diventati lavoratori dipendenti e che solo i liceali avevano il diritto di studiarlo. Io allora non volevo sapere nulla di Leopardi, ma oggi apprezzo la prof. che si è battutta con quel collega perché ha sempre voluto documentarci tutto ciò che di vero c'è nel mondo.

Ringrazio particolarmente il Prof. Leonardo Santoro. Con gratuità si è sempre preso cura di tutti i suoi studenti. Ci ha insegnato tantissime cose di informatica ma prima di tutto ci è stato vicino nel momento in cui le scuole superiori si avviavano verso la fine e un grande cambiamento ci attendeva. Lo ringrazio ancora per avermi dato fiducia e avermi presentato come un ottimo informatico[1] per aiutarmi a trovare qualche lavoro e poter continuare a studiare. Grazie a lui e alla stima che ha su di me sono cambiate tante cose. Caro prof. sappi che la stima è reciproca e che non ti scorderò mai. Hai fatto veramente tanto per me.

Ringrazio tutte le persone che mi hanno sostenuto e incoraggiato a continuare gli studi, in particolare la mia mamma. O quella passione che avevo dentro

[1] cosa hai rischiato prof...

nascondeva un richiamo ad una responsabilità oppure era meglio andare a lavorare perché se fosse stato solo per il gusto di studiare non ne valeva la pena fare i sacrifici che abbiamo fatto. E' stata veramente dura in certi momenti ma la mia mamma non m'ha fatto mai mancare nulla per affrontare con dignità i miei studi.

Ringrazio Beppino e Mario. Insieme ai loro amici hanno accolto me e l'allora compagno di studi Marco Buralli (che fine hai fatto disgraziato) in facoltà come fosse casa loro. Per la prima volta, ovunque mi trovassi, sentivo ogni angolo del mondo come casa mia. Mi hanno invitato a cena da gente che non conoscevo, mi hanno portato alla giornata di inizio anno di CL (e io non sapevo nemmeno cosa fosse), mi hanno invitato alle assemblee di Scuola di Comunità dove c'erano centinaia di studenti tutti attenti a uno adulto che affrontava con loro le domande che emergevano, mi hanno invitato a cene seguite da canti popolari del paese in cui vivo di cui ignoravo totalmente l'esistenza e che non avrei mai creduto che potessero essere così belli e tutt'altro che un modo noioso di stare insieme. Con cose così semplici mi hanno cambiato la vita.

Ringrazio Poste Italiane le quali smarriscono lettere destinate a Roma (non un paesino sperduto, Roma!). Mi sono così deciso a spedire una lettera importante e un po' diversa da quelle che eravamo soliti scriverci da circa tre anni io e Iris; per sicurezza l'ho spedita con raccomandata e ricevuta di ritorno. Così, Poste Italiane e Iris, mi hanno cambiato anche loro la vita.

Ringrazio Iris. Devo ammetterlo. C'ha un pazienza infinita[2]. Grazie a lei, che non lascia fare mai le cose che non vanno e che non si accontenta delle scuse, mi son trovato e tuttora mi trovo a mettere in discussione molte cose della mia vita. Questo è un bene perché ho accanto una persona che mi aiuta a scoprire quanto è immensa la vita e quanto non ti accorgi di questa immensità quando ti accomodi nella tranquillità. Ma la cosa di cui più son grato è che ha occhi per vedere quanto tutto ciò che mi circonda mi appas-

[2]altrettanta ne ho io con lei - per Iris: non ti gasare troppo quindi tesoro.

siona infinitamente. Così non mi sento giudicato per ciò che faccio e che ho, ma per ciò che sono. E' il dono più bello della mia vita perché per lei conta tutto ciò che sono io.

Ringrazio per questi anni in università. Sono stati così intensi e coinvolgono così tanti episodi di amicizia che spenderei più pagine per i ringraziamenti che per il lavoro di tesi. Non per questo però evito di soffermarmi completamente.

Ringrazio Jacopo. Ci sono mille fatti che documentano qualcosa di incredibile. Abbiamo affrontato lo studio come due bambini che insieme vogliono scoprire qualcosa e che via via che imparano vogliono mettere le mani in pasta. Più o meno ingenuamente abbiamo messo in gioco tutto, proprio tutto, ciò che abbiamo appreso in ogni esame. Ogni lezione, ogni capitolo è stata una risposta alla nostra immensa curiosità. Ogni domanda valeva la pena essere posta, a costo di sentirci dare di grulli per la scrupolosità o perché andavamo a rompere. Ma la cosa più bella è che nell'andare a fondo della nostra passione non è mai stata tagliata fuori la nostra umanità. Sino al punto che i suoi sono arrivati ad adottarmi per più di un mese. Ringrazio anche loro, Fernanda e Paolo, per il grande affetto che hanno verso di me.

Ringrazio il Prof. Modica e il Prof. Vicario per la passione e la cura che hanno per la vita in facoltà; in particolare li ringrazio per l'attenzione verso noi studenti e per la stima che ci riservano.

Ringrazio gli amici dell'amplificazione del CLU. In particolare ringrazio Franz perché con il suo invito a prendermi questa responsabilità in modo che non sia fine a se stessa né fine a una sola passione per l'audio, mi ha fatto conoscere qualcosa di più grande che ha da dire su tutto della mia vita. Ringrazio il Vannu per la cura e per l'attenzione a noi tutti che ha ogni volta che spendiamo le nostre energie per far sentire la bellezza dei canti e della musica che amplifichiamo. Ringrazio Alex e Black per la semplicità con cui donano il loro tempo e rimangono gasati da ciò che realizziamo. Ringrazio Psico-Paolo perché non lascia mai fare quando c'ha un'obiezione e ciò che facciamo non

lo soddisfa. Ringrazio James Blond per il bene che ci vuole. Ringrazio infine il Lancio, il Coppe e G.P. per averci tramandato la loro esperienza sia tecnica che umana.

Ringrazio anche i direttori, i componenti del coro e i musicisti del CLU: sono onorato di essere lo strumento perché la passione di ogni persona che canta e/o suona colpisca chiunque ascolti le loro opere. Grazie all'ampli ho il piacere di essere in prima fila per gustarmi una bellezza immensa.

Ringrazio tutti gli amici per la Scuola di Comunità. Nella mia vita non c'è lavoro più utile di questo per giudicare insieme ciò che di bello mi colpisce, ciò che pensavo mi bastasse e invece sento non bastare, e ciò che talvolta mi ferisce. In particolare per questo lavoro di tesi ringrazio il Don Gius per aiutarci a capire cos'è la fede: senza questo aiuto questa tesi non avrei potuto farla perché non mi sarei mai fidato dei tanti risultati dai quali sono ripartito per svolgere il mio lavoro.

Ringrazio Anna Giorgini per la grande fiducia e per avermi insegnato ad essere responsabile. Il lavoro che conduco da alcuni anni, grazie all'opportunità che mi ha concesso, è stato prima di ogni altro aspetto umano. Le non poche difficoltà che abbiamo incontrato io, lei e i suoi colleghi, le abbiamo superate grazie alla sua guida. Il frutto di questa collaborazione, inoltre, è stato fondamentale per la mia carriera di studi: l'esperienza acquisita e l'opportunità di guadagno sono state indispensabili in questi anni in università. Ringrazio sinceramente lei e i suoi colleghi, in particolare Rosita Stefano Simone e Fabrizio, per tutto ciò che di grande è nato da quando ho l'opportunità di mettere in gioco le mie conoscenze per le loro necessità.

Ringrazio tutte le persone che mi hanno aiutato a rimettermi in piedi in tempi record in seguito a una non piacevole frattura ai malleoli (mitici, mi avete risistemato in tempo per discutere la tesi sui miei piedi). In particolare i miei genitori che si sono fatti in quattro per le mie mille esigenze, per Iris che ha passato più tempo in treno che con me per aiutare i miei e starmi

vicina senza trascurare i suoi impegni di studio a Pisa, e Jacopo e i suoi genitori che mi hanno donato molto affetto. Ringrazio anche il prof. Buzzi e il suo staff: anche se è gente di poche parole hanno fatto veramente un bel lavoro.

Infine ringrazio immensamente chi mi ha dato l'opportunità di svolgere questo lavoro di tesi. Prima di tutti Beppone[3]: sapendo che ero interessato a lavorare con l'audio, di sua iniziativa mi ha fissato un appuntamento per chiedere questa tesi. Anche se ha più volte espresso la volontà di non essere ringraziato, deve sapere che gli sono davvero immensamente grato per tutto il suo prezioso aiuto.
Ringrazio il prof. Alberto Del Bimbo che mi ha assegnato questa tesi: grazie al suo consenso ho potuto svolgere un lavoro bellissimo studiando argomenti recenti e scoprendo tantissime novità.
Ringrazio Lamberto e Marco per avermi dedicato parte del loro tempo per seguirimi e aiutarmi, e per la pazienza che hanno avuto quanto non ho saputo individuare momenti migliori per interrompere il loro lavoro.

Tutti voi siete autori con me di questo lavoro, grazie davvero.

[3]scientificamente noto come Ing. Giuseppe Serra.