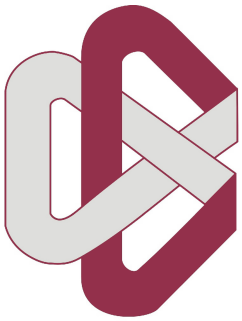
RAFAEL DE JESÚS ROBLEDO JUÁREZ

SEGMENTACIÓN DE INTERLOCUTORES A PARTIR DE SEÑALES DE AUDIO UTILIZANDO CADENAS ESCONDIDAS DE MARKOV Y TÉCNICAS DE SELECCIÓN AUTOMÁTICA DE MODELOS



CIMAT

Centro de Investigación en Matemáticas

Departamento de Ciencias de la Computación

Tesis:

SEGMENTACIÓN DE INTERLOCUTORES A PARTIR DE SEÑALES DE AUDIO UTILIZANDO CADENAS ESCONDIDAS DE MARKOV Y TÉCNICAS DE SELECCIÓN AUTOMÁTICA DE MODELOS

Que para obtener el grado de:

MAESTRO EN CIENCIAS CON ESPECIALIDAD EN COMPUTACIÓN Y MATEMÁTICAS INDUSTRIALES

Presenta:

RAFAEL DE JESÚS ROBLEDO JUÁREZ

Director de tesis:

DR. SALVADOR RUÍZ CORREA

Guanajuato, Gto. a xx de noviembre del 2013

SUPERVISORES:
Dr. Salvador Ruíz Correa
Dr. Johan Jozef Lode Van Horebeek
Dr. Rogelio Hasimoto Beltrán

LUGAR:
Guanajuato, Gto.

FECHA:
xx de noviembre del 2013

A mi familia, por su apoyo, cariño y confianza.

RESUMEN

En este trabajo de tesis se aborda el problema *Speaker Diarization*, o de segmentación automática de una señal de audio de acuerdo a los interlocutores que participan en una grabación. Las grabaciones que se utilizarán, simulan ser conversaciones entre dos o más personas, y se tratará de identificar tanto el número de personas que hablan, así como los momentos en los que participan.

Para esto resolver este problema, se propondrán múltiples modelos con Cadenas Escondidas de Markov, a partir de la cuales se estimarán posibles segmentaciones correcta. Para seleccionar cuáles son los modelos que mejor se ajustan a los datos, se propondrá primero una exploración de todas las posibles soluciones utilizando una función de penalización estilo Criterio de Información Bayesiano (BIC) con un parámetro de regularización auto-ajustable. Después, de entre las mejores candidatos se escogerá a la mejor solución utilizando simulaciones bootstrap para realizar una prueba de hipótesis.

Para probar el desempeño de la metodología propuesta, se realizaron pruebas con 6 secuencias de audio generadas sintéticamente. Se muestran los resultados tanto en la selección de modelo correcto (número de personas que participan), como con la segmentación recuperada de la señal de audio.

AGRADECIMIENTOS

Agradezco a Dios las oportunidades que me ha presentado, y la gente de la que me ha acompañado.

A mi familia, que me ha apoyado durante toda mi vida académica. A mi mamá que ha sido un muestra inmejorable de trabajo y servicio. A mi hermano, quien ha sido un buen ejemplo y que me ha cuidado. A mis tíos y tías que siempre han procurado lo mejor para mí.

Quiero agradecer también a mi asesor el Dr. Salvador Ruíz Correa; por su tiempo, su compromiso y apoyo en mi formación académica, y en especial con este trabajo de tesis.

A los sinodales Dr. Johan Jozef Lode Van Horebeek y Dr. Rogelio Hasimoto Beltrán por sus observaciones y sugerencias en la revisión de esta tesis.

También, doy gracias al Consejo Nacional de Ciencia y Tecnología (CONACYT) por el apoyo económico aportado para la realización de estos estudios de posgrado.

A la comunidad CIMAT/DEMAT, y en especial a todo al departamento de Ciencias de la Computación, por la inspiración y el conocimiento que me han brindado. Al personal de Servicios Escolares y en general a todo el personal administrativo por las facilidades y la orientación en los procesos que he requerido realizar.

Asimismo, a mis compañeros de generación: David, Carlos, Guillermo, Marcela, Cristóbal, Mauricio, Lázaro, Mario, Ángel y Fimbres; con quienes he compartido anhelos, desvelos y entusiasmo. Por toda su ayuda y palabras de aliento en estos dos años.

Por último, a todos aquellos amigos de la secundaria, la preparatoria y la universidad que me han acompañado estos años y hacen del camino uno más agradable.

ÍNDICE GENERAL

I PRELIMINARES 1

1 INTRODUCCIÓN 2
- 1.1 Definición del problema 2
- 1.2 Motivación 3
- 1.3 Contribuciones 4
- 1.4 Principales enfoques 5
- 1.5 Trabajos previos 6

II MARCO TEÓRICO 8

2 SPEAKER DIARISATION 9
- 2.1 Aplicaciones de *speaker diarisation* 9
- 2.2 Formulación matemática 10
- 2.3 Componentes del sistema 11
- 2.4 Procesamiento acústico 14
  - 2.4.1 Pre-procesamiento de señal de audio 14
  - 2.4.2 Obtención de características acústicas 17

3 MODELOS DE MÁRKOV 21
- 3.1 Cadena de Márkov 22
- 3.2 Cadena oculta de Márkov 25
- 3.3 Estimación de Máxima Verosimilitud del HMM 28
  - 3.3.1 Estimación de parámetros del HMM 28
  - 3.3.2 Algoritmo *backward-forward* 31
  - 3.3.3 Implementación 35

4 SELECCIÓN DE MODELO 37
- 4.1 Funciones de penalización 38
  - 4.1.1 BIC 38
- 4.2 Bootstrap 40
  - 4.2.1 Bootstrap paramétrico 41
- 4.3 Selección de modelo usando BIC 42
- 4.4 Selección de modelo usando bootstrap con likelihood ratio testing 43

III MARCO EXPERIMENTAL 46

5 METODOLOGÍA 47
- 5.1 Esquema general 50

6 EXPERIMENTOS Y RESULTADOS 52
- 6.1 Experimentos 53
  - 6.1.1 Secuencia 1 53
  - 6.1.2 Secuencia 2 59
  - 6.1.3 Secuencia 3 62

6.1.4 Secuencia 4 67
6.1.5 Secuencia 5 70
6.1.6 Secuencia 6 77
6.2 Resultados 81
7 DISCUSIÓN Y CONCLUSIONES 84
7.1 Trabajo futuro 85

BIBLIOGRAFÍA 87

ÍNDICE DE FIGURAS

Figura 2.1 Esquema general del sistema. Se muestran las etapas principales para la estimación de la segmentación. 15
Figura 2.2 Señal original de audio. 16
Figura 2.3 Identificación y eliminación de silencio en señal. 16
Figura 2.4 Banco de filtros triangulares en frecuencia Mel. 18
Figura 2.5 Respuesta al banco de filtros. 18
Figura 2.6 Mel Frequency Cepstrum Coefficients. 19
Figura 2.7 MFCC agrupados con k-means++. 19
Figura 3.1 Observaciones independientes e idénticamente distribuidas. 22
Figura 3.2 Modelo de Márkov de primer orden. 23
Figura 3.3 Modelo de Márkov de segundo orden. 24
Figura 3.4 Modelo oculto de Márkov. 24
Figura 3.5 Matriz de transición de modelo oculto de Márkov (grafo). 26
Figura 3.6 Matriz de transición de modelo oculto de Márkov (rejilla). 26
Figura 4.1 Esquema del proceso bootstrap. 42
Figura 5.1 Esquema general. 51
Figura 6.1 Secuencia 1: Parámetros estimados (1) 54
Figura 6.2 Secuencia 1: Parámetros estimados (2) 55
Figura 6.3 Secuencia 1: Pruebas BIC 56
Figura 6.4 Secuencia 1: Pruebas de hipótesis usando bootstrap 57
Figura 6.5 Secuencia 1: Secuencia recuperada 58
Figura 6.6 Secuencia 2: Parámetros estimados (1) 59
Figura 6.7 Secuencia 2: Parámetros estimados (2) 60
Figura 6.8 Secuencia 2: Pruebas BIC 61
Figura 6.9 Secuencia 2: Pruebas de hipótesis usando bootstrap 62
Figura 6.10 Secuencia 2: Secuencia recuperada 63
Figura 6.11 Secuencia 3: Parámetros estimados (1) 63
Figura 6.12 Secuencia 3: Parámetros estimados (2) 64
Figura 6.13 Secuencia 3: Pruebas BIC 65
Figura 6.14 Secuencia 3: Pruebas de hipótesis usando bootstrap 66
Figura 6.15 Secuencia 3: Secuencia recuperada 67

Figura 6.16 Secuencia 4: Parámetros estimados (1) 68
Figura 6.17 Secuencia 4: Parámetros estimados (2) 69
Figura 6.18 Secuencia 4: Pruebas BIC 70
Figura 6.19 Secuencia 4: Pruebas de hipótesis usando bootstrap 71
Figura 6.20 Secuencia 4: Secuencia recuperada 72
Figura 6.21 Secuencia 5: Parámetros estimados (1) 72
Figura 6.22 Secuencia 5: Parámetros estimados (2) 73
Figura 6.23 Secuencia 5: Pruebas BIC 74
Figura 6.24 Secuencia 5: Pruebas de hipótesis usando bootstrap 75
Figura 6.25 Secuencia 5: Secuencia recuperada 76
Figura 6.26 Secuencia 6: Parámetros estimados (1) 77
Figura 6.27 Secuencia 6: Parámetros estimados (2) 78
Figura 6.28 Secuencia 6: Pruebas BIC 79
Figura 6.29 Secuencia 6: Pruebas de hipótesis usando bootstrap 80
Figura 6.30 Secuencia 6: Secuencia recuperada 81

ACRÓNIMOS

NIST	Instituto Nacional de Normas y Tecnología (National Institute of Standards and Technology)
HMM	Modelo Oculto de Márkov (Hidden Markov Model)
E-HMM	Modelo Oculto de Márkov Evolutivo (Evolutive Hidden Markov Model)
GMM	Modelo de Mezcla de Gaussianas (Gaussian Mixture Model)
BIC	Criterio de Información Bayesiano (Bayesian Information Criterion)
AIC	Criterio de Información Akaike (Akaike Information Criterion)
MFCC	Coeficientes Cepstrales en la Frecuencia Mel (Mel Frequency Cepstral Coefficients)
LFCC	Coeficientes Cepstrales en Frecuencia Lineal (Linear Frequency Cepstral Coefficients)
aMFCC	Coeficientes Cepstrales en la Frecuencia anti-Mel (Anti-Mel Frequency Cepstral Coefficients)
ANN	Redes Neuronales Artificiales (Artificial Neural Network)
DTW	Deformación Dinámica del Tiempo (Dynamic Time Warping)
FT	Transformada de Fourier (Fourier Transform)
FFT	Transformada Rápida de Fourier (Fast Fourier Transform)
DCT	Transformada de Coseno Discreta (Discrete Cosine Transform)
EM	Esperanza-Maximización (Expectation-Maximization)
B-W	Baum-Welch

MLE	Estimador de Máxima Verosimilitud (Maximum-Likelihood Estimator)
LLR	Log-Likelihood Ratio
TTS	Sintetizador de voz (Text-To-Speech engine)
DER	Proporción de error de diarización (Diarization Error Ratio)

Parte I

PRELIMINARES

1

Despacio. Al fin y al cabo tenemos
toda la vida por delante.

Juan Rulfo.

INTRODUCCIÓN

1.1 DEFINICIÓN DEL PROBLEMA

En este trabajo de tesis se abordará el problema de *Speaker Diarization*, que consiste en identificar el número de interlocutores que participan en una grabación de audio, y además encontrar en qué segmentos de la grabación habla cada persona.

Para esta tarea, se considera que se tiene una señal de audio con información de nuestro interés, y se requiere segmentar de acuerdo a las personas que participan en la grabación. Este proceso involucra, por una parte, encontrar el número total de personas que hablan en la conversación; y por otro lado, para cada persona propuesta se debe identificar los momentos en los que habla.

Estos dos sub-procesos se pueden realizar en diferente orden, dependiendo de la metodología utilizada. Se puede por ejemplo, tratar de inferir primero cuántas personas hablan en la secuencia de audio, y luego segmentar la grabación de acuerdo a este número de personas; o también ir segmentando la grabación y a partir de ahí cuántos son los estados posibles para la grabación de audio.

El problema en general de *speaker diarization*es muy amplio, y puede presentarse muchos entornos que dificulten la tarea:

1. Dos personas pueden hablar al mismo tiempo, por lo que hay que considerar cierta incertidumbre adicional en el modelo.

2. En el lugar donde se realizó la grabación puede haber mucho ruido ambiental o música de fondo, por lo que es necesario realizar una etapa de pre-procesamiento para eliminar los segmentos innecesarios.

3. Para algunos tipos grabaciones, habrá momentos que serán difíciles de asignar a un interlocutor en específico: cuando tose, se ríe o estornuda alguien, por ejemplo.

4. Puede haber múltiples micrófonos presentes en la grabación, por lo que si se quiere utilizar toda la información disponible se deben considerar los posibles retrasos entre las diferentes fuentes.

5. La conversación entre los interlocutores no necesariamente está planeada, por lo que puede ser irregular el porcentaje de participación de cada persona.

6. Se pueden presentar algunas secuencias en que el cambio entre interlocutores sea muy rápido.

Por ello, el trabajo se acotó en varios de estos aspectos. Para tener un mayor control tanto en el contenido como en el medio de la grabación de audio, se utilizó un sintetizador de voz para simular de forma artificial conversaciones entre varias personas tanto hombres como mujeres, en dos idiomas: inglés y español. Esto permitió además que se pudiera obtener la secuencia original de forma mucho más sencilla.

1.2 MOTIVACIÓN

En los últimos años la tarea de *Speaker Diarization* se ha vuelto una parte importante de diferentes procesos que se realizan con las grabaciones de audio, tales como la identificación y navegación por segmentos en específico, además de la búsqueda y recuperación de información en grandes volúmenes de secuencias de audio.

La investigación desarrollada referente a *speaker diarization* se ha guiado principalmente de acuerdo a la financiación existente para proyectos específicos. Hasta principios de la década de 1990, el trabajo de investigación se concentraba en trabajar con grabaciones telefónicas. Principalmente se usaba para segmentar la conversación, así como etapa de pre-procesamiento para luego realizar reconocimiento y/o transcripción del habla.

Para la década del 2000, las aplicaciones fueron cambiando, a la par que aumentaba la capacidad disponible de almacenamiento; por lo que creció el interés de mantener un registro de forma automática de noticieros televisivos y transmisiones de radio a lo largo de todo el planeta. Entre la información más útil que se registraba de las grabaciones, era la transcripción del diálogo, meta-etiquetas referentes al contenido así como la segmentación y le orden de las personas que intervienen en la grabación.

A principios del año 2002, empezaron a surgir varios proyectos de investigación cuya principal intención era mejorar la comunicación interpersonal, y en especial la que ocurre a larga distancia y de forma multimodal. La investigación y desarrollo se enfocó en extracción del contenido y su etiquetación de acuerdo a las personas que participan, ya sea para mantener un archivo histórico, o para fácil disposición de personas interesadas en su contenido.

Debido a la creciente investigación en estos campos, el Instituto Nacional de Normas y Tecnología (NIST), ha organizado un sistema de oficial de evaluaciones que permita unificar así como dirigir el esfuerzo de los investigadores que trabajan en esta área; al tener una manera precisa de comparar las diferentes metodologías en las que trabajan. Primero, en el 2002 las pruebas consistían en grabaciones correspondientes a noticieros, y para el 2004 se empezaron a incluir pruebas con grabaciones de reuniones, que resultaban ser la principal aplicación en esos años.

Una gran diferencia entre ambos tipos de entornos, es que mientras en un noticiero se suelen tener preparados los diálogos que se dirán e incluso se leen las noticias; en el caso de las reuniones la conversación es más espontánea y puede suceder que dos personas hablen al mismo tiempo.

Otra característica diferente, es que mientras en un noticiero cada participante suele tener un micrófono de solapa o hay micrófonos ambientales, éstos suelen ser de calidad, por lo que no el audio de la grabación es mucho mejor y no hay mucho ruido presente en la señal.

Por otro lado, para el caso de las reuniones el ambiente no suele ser tan controlado, y puede haber un mayor ruido ambiental; además de que tanto al disposición de los micrófonos así como su calidad no siempre son la mejor, agregando interferencia o redundancia innecesaria a la señal.

Por eso, se considera el caso de reuniones como el escenario completo para las tareas involucradas con reconocimiento del habla; por la complejidad y los diferentes problemas que se pueden presentar.

1.3 CONTRIBUCIONES

Las principales contribuciones de este trabajo se enumeran a continuación:

1. Se implementó el algoritmo Baum-Welch (B-W) en Matlab, con funciones críticas desarrolladas en C.

2. Se propuso un método de selección de modelo en dos etapas: primero usando una versión de Criterio de Información Bayesiano (BIC) regularizada, para escoger un subconjunto de soluciones más probables a partir del conjunto de modelos propuestos; luego, como segunda etapa una prueba de hipótesis usando el estadístico Log-Likelihood Ratio (LLR), para seleccionar dentro de los modelos más probables a la solución ganadora.

3. Se implementó un algoritmo para generación, ajuste automático de modelos y selección de mejor solución de acuerdo a las métricas establecidas en el Capítulo 5.

Esencialmente, el trabajo realizado se centra en la selección adecuada del número de estados ocultos o participantes de la cadena de Márkov modelada, donde a través la metodología propuesta, y utilizando varias técnicas como: BIC regularizado, bootstrap paramétrico y prueba de hipótesis se encuentra el modelo solución correspondiente.

1.4 PRINCIPALES ENFOQUES

De acuerdo al trabajo desarrollado hasta ahora, se puede distinguir que hay dos grandes enfoques que se usan para *speaker diarization*: *bottom-up* (de abajo a arriba) y *top-down* (de arriba a abajo). De forma general, se consideran *bottom-up* las metodologías en las que se inicia la estimación con pocos clústers (e incluso un sólo segmento), y *top-down* aquellas en las que se inicia la estimación con muchos más clústers de los que se esperan encontrar.

Ambos enfoques iteraran hasta converger a un número de clústers óptimo, en que cada grupo debe corresponder a un interlocutor.

Para ambas estrategias se suelen usar Modelos Ocultos de Márkov (HMMs) donde cada estado se representa muchas veces como Modelos de Mezclas de Modelo de Mezcla de Gaussianas (GMM) y corresponde a cada interlocutor. Las matrices de transición entre estados representan el cambio entre la persona que está hablando; mientras que las matrices de una emisión corresponden a la probabilidad de los interlocutores de *decir* cada una de las *palabras* que se definan.

Es importante remarcar que con *decir una palabra* no necesariamente se refiere a la acción de que una persona pronuncie una palabra; sino que se definirá un *diccionario de palabras* que en realidad es conjunto de vectores de características que representen la señal de audio de forma discreta; y cada interlocutor tendrá una probabilidad de emisión asociada para cada uno de estos vectores.

1.5 TRABAJOS PREVIOS

En el trabajo de Anguera et al. [AMBE+12] se hace un profundo análisis sobre el estado del arte hasta hace un año, y se mencionan las principales características de cada uno de estos trabajos, que a continuación se resumen.

Dentro de los trabajos que usan un enfoque *bottom-up* se encuentran por ejemplo el de Anguera et al. [AMWP06], en donde se propone una medida de similitud entre clústers utilizando una variante de BIC para decidir qué pares de clústers agrupar, así como para establecer un criterio de paro.

En el trabajo de Wooters et al. [WH07], también proponen una métrica basada en BIC para la aglomeración de los grupos, pero se enfocan principalmente en mejorar una etapa de pre-procesamiento de la señal para detectar cuando alguien habla o no habla en la grabación. Esto es esencial cuando por ejemplo, se puede tener segmentos musicales o momentos en los que el ruido.

Por otro lado, en el trabajo de Nguyen et al. [NLS09] se presentan dos funciones objetivo que buscan maximizar la distancia intra-clase y la distancia inter-clase y que automáticamente deducen el número óptimo de grupos para esto. Después realizan un proceso de aglomeración jerárquico, pero en un subespacio espectral en que sean separables más fácilmente los interlocutores.

Mencionando algunos de los trabajos que son del tipo *top-down*, se tiene por ejemplo a Meignier et al. [MBI01] quien propone un sistema conformado por un Modelo Oculto de Márkov Evolutivo (E-HMM), en el que mediante un proceso iterativo se van detectando y agregando interlocutores al modelo propuesto. El método propuesto busca reducir las falsas detecciones al incorporar información al sistema tan pronto como sea detectada.

En el trabajo de Fredoulle et al. [FE07a] también proponen un sistema en el que se usa un E-HMM para la estimación de parámetros, pero además realizan una etapa de pre-procesamiento más extensa: primero usan un HMM de dos estados para diferenciar

los silencios de la actividad de los interlocutores y luego mediante una ventana movible evalúan cuándo es más probable que haya cambio entre los hablantes, de acuerdo al Generalized Likelihood Ratio. A partir de esa información es como delimitan más el problema, y sólo tienen que buscar cada segmento encontrado a qué interlocutor corresponde.

Además, en otro trabajo de Fredoulle et al. [FBE09] presentan el sistema *top-down* anteriormente usado pero ahora se enfocan en el uso de múltiples micrófonos a diferentes distancias para obtener más información y mejorar la etapa de segmentación. También, en este sistema descartan la etapa de pre-segmentación que anteriormente se había propuesto para encontrar los posibles cambios entre interlocutores.

Por último, en la misma línea de trabajo utilizando E-HMM, Bozonnet et al. [BEF10] muestran algunas mejoras en el ajuste de parámetros utilizados al momento de entrenar el modelo; pero además de que agregan una etapa de purificación después de la clasificación y segmentación de la secuencia, re-evaluando la pertenencia de intervalos. Esta etapa de reasignación en metodologías *top-down* muestra un buen desempeño, pues se hace la purificación con respecto a segmentos en los que un interlocutor es dominante respecto a los demás.

Parte II

MARCO TEÓRICO

I thank you for your voices: thank
you: Your most sweet voices.

William Shakespeare

2

SPEAKER DIARISATION

Se conoce como *Speaker Diarization* al problema de segmentación automática de audio a partir de la identificación de las diferentes personas que participan en una grabación. Esto se realiza generalmente identificando los segmentos que son más *homogéneos* y a partir de estos, identificar el número de personas que hablan en la grabación.

Se considera que cada persona posee ciertas características propias que la diferencian de los demás. Por ejemplo, tanto la tesitura como el timbre son rasgos que varían de persona a persona.

La idea básica cuando se busca abordar este problema es el de poder segmentar la señal de audio de acuerdo a los cambios que ocurren en ésta. Es decir, detectar por ejemplo las variaciones en la voz (que usualmente implican que alguien más está hablando) y luego, con todos los segmentos obtenidos, tratar de agruparlos según características similares; pues una misma persona tiene características propias en su voz.

2.1 APLICACIONES DE *speaker diarization*

Por la misma naturaleza del problema, cuando se realiza *speaker diarization*, lo que se trata de inferir es cuántas personas hablan, y en qué momentos habla cada una de ellas; es decir, se identifica a un grupo n de personas, pero no se dice nada acerca de ellos o su identidad.

Debido a esto, las etiquetas asignadas a cada persona identificada en la grabación puede cambiar, pues no hay nada que nos permita asociar de manera no arbitraria una etiqueta a una persona en especifico. Es decir, el orden en que se asignan las etiquetas puede cambiar, aunque la segmentación obtenida sea en esencia la misma.

La segmentación y agrupación de voz, es una parte importante de la transcripción de voz, así como también del reconocimiento de voz e identificación de personas que hablan.

Es de gran ayuda para la tarea de reconocimiento de voz, puesto que éste se basa en identificar palabras completas; por lo que al lograr segmentar una señal de audio de acuerdo a las personas que hablan, se tendrá entonces muy seguramente una buena segmentación tanto de palabras como oraciones completas.

En cuanto a la identificación de personas y alguna otra modelación acústica que se quiera realizar; es importante que los modelos que se entrenan usen segmentos de audio homogéneos (en este caso, que se tenga la certeza que corresponden a la misma persona), para que en realidad se esté modelando la voz de la persona, y no algo diferente a ella.

Como último ejemplo, la transcripción automática de voz, es el proceso en el que además de lograr identificar cuándo habla cada persona, se identifica de quién se trata realmente esta persona, y qué es lo que está diciendo. Ésto sirve por ejemplo, cuando se tiene una gran cantidad de grabaciones de audio; y se desea etiquetar de qué se habla en cada una de estas grabaciones.

2.2 FORMULACIÓN MATEMÁTICA

De forma general, se esbozará cuál es el problema que trata de resolver con *speaker diarization*. El enfoque que se usará es estadístico, entonces en la formulación es necesario el cálculo de probabilidades.

Denótese por $\mathcal{A}$ la evidencia acústica o los datos a partir del cuál el modelo deberá encontrar la segmentación correcta para un fragmento de señal. Sin entrar mucho a detalle, y puesto que de alguna manera se debe digitalizar y caracterizar la señal analógica de audio; podemos pensar en $\mathcal{A}$ como la secuencia de símbolos correspondiente a un segmento de señal, y que está conformada por elementos de un alfabeto mucho más grande $\mathbb{A}$.

$$\mathcal{A} = a_1, a_2, ..., a_K \quad a_i \in \mathbb{A} \tag{2.1}$$

en donde los sub-índices de los elementos a_i hacen referencia a un intervalo de tiempo i en la secuencia de audio original, y sus valores pueden repetirse de acuerdo a la grabación.

De la misma manera, definamos

$$\mathcal{S} = s_1, s_2, ..., s_N \quad s_i \in \mathbb{S} \tag{2.2}$$

donde $\mathcal{S}$ es la secuencia que corresponde a la segmentación correcta para un intervalo del audio original. En este caso S es el conjunto de todos los interlocutores que participan en la grabación de audio y que por el momento se considerará como información conocida; y s_i de igual manera representa al interlocutor que habla en el tiempo i.

Si $P(\mathcal{S} \mid \mathcal{A})$ denota la probabilidad de que una secuencia de interlocutor $\mathcal{S}$ esté hablando dada la evidencia acústica en $\mathcal{A}$, entonces una forma de escoger cuáles son los interlocutor que hablan en ese intervalo es por ejemplo:

$$\hat{\mathcal{S}} = \arg\max_{\mathcal{S}} P(\mathcal{S} \mid \mathcal{A}) \tag{2.3}$$

Esto es, se seleccionaría la sucesión de interlocutores más probable para una secuencia de datos dada.

Por el teorema de Bayes, podemos reescribir la parte derecha de (2.3) como sigue:

$$P(\mathcal{S} \mid \mathcal{A}) \;=\; \frac{P(\mathcal{S}) \cdot P(\mathcal{A} \mid \mathcal{S})}{P(\mathcal{A})} \tag{2.4}$$

donde $P(\mathcal{S})$ es la probabilidad de que la secuencia de interlocutor $\mathcal{S}$ hable a priori en ese orden; $P(\mathcal{A} \mid \mathcal{S})$ la probabilidad de que sea observada la evidencia acústica $\mathcal{A}$ cuando los interlocutores $\mathcal{S}$ están hablando, y $P(\mathcal{A})$ la probabilidad a priori de que los datos $\mathcal{A}$ sean observados. Por probabilidad total, esto último se puede escribir como:

$$P(\mathcal{A}) = \sum_{\mathcal{S}'} P(\mathcal{S}') \cdot P(\mathcal{S}' \mid \mathcal{A}) \tag{2.5}$$

Como en (2.3) se está maximizando con respecto a $\mathcal{S}$, es decir la variable $\mathcal{A}$ permanece fija -pues es la única evidencia acústica observada-, se sigue de (2.3) y de (2.4) que es equivalente maximizar únicamente el producto $P(\mathcal{S}) \cdot P(\mathcal{A} \mid \mathcal{S})$:

$$\hat{\mathcal{S}} = \arg\max_{\mathcal{S}} P(\mathcal{S}) \cdot P(\mathcal{A} \mid \mathcal{S}) \tag{2.6}$$

2.3 COMPONENTES DEL SISTEMA

Para diseñar un sistema que sea capaz de resolver el problema de *speaker diarization*, se debe de primero plantear la formulación matemática que se abordará.

En general, todos los sistemas que involucran procesamiento de voz, tienen varias etapas esenciales, como se menciona en Jelinek [Jel98].

PROCESAMIENTO ACÚSTICO: Primero, se necesita decidir de qué forma procesará la información. Usualmente se contará con un micrófono o un arreglo de micrófonos que captarán las voces y las transformarán en impulsos eléctricos. Luego, se deberá de muestrear esta señal analógica para poder almacenarla digitalmente para su posterior procesamiento.

Después de este proceso se tendrá una representación discreta en el tiempo de la señal, que se analizará en pequeñas ventanas de tiempo. Dependiendo de la aplicación de interés, suele variar el intervalo de análisis. De la misma manera, hay diferentes formas de caracterizar estas ventanas de tiempo, estimando diferentes vectores característicos. Entre los más comunes para tareas relativas a procesamiento de voz se encuentran los MFCC, LFCC [DM80] y aMFCC [LG09].

Por último, después de obtener alguna representación paramétrica de la señal, se deberá realizar una discretización de estos vectores. A este procedimiento se le conoce como construcción del *diccionario de palabras*, pues a cada valor o clase posible en la discretización se le conocerá como *palabra*.

MODELADO ACÚSTICO: En esta segunda etapa, se considera que ya se ha construido el diccionario de palabras o la evidencia acústica $\mathbb{A}$, por lo que ahora se necesita proponer una forma de calcular las probabilidades $P(\mathcal{A} \mid \mathcal{S})$ que se refieren a la probabilidad de que la secuencia $\mathcal{A}$ sea observada dado que los interlocutores $\mathcal{S}$ hablan.

Puesto que esta probabilidad se debe de calcular para todos los pares posibles de $\mathcal{S}$ con $\mathcal{A}$: se debe de hacer este cálculo de la forma más eficiente posible, ya que el número de combinaciones existentes suele ser muy grande.

Para estimar las probabilidades $P(\mathcal{A} \mid \mathcal{S})$ se necesita un modelo estadístico que además de considerar la interacción de los interlocutores, tenga en cuenta el ambiente, la ubicación de los micrófonos y sus características, entre otras cosas.

El modelo acústico más comúnmente utilizado en tareas de procesamiento de voz, es el Modelo Oculto de Márkov (HMM), que se discutirá en el Capítulo 3. Sin embargo, no es el único modelo existente, y hay trabajos que utilizan otras

técnicas, como por ejemplo aquellos que usan Redes Neuronales Artificiales (ANN) [JRP09] [FE07b] o métodos de Deformación Dinámica del Tiempo (DTW) [HMvL11]

MODELADO DE INTERLOCUTORES: Por otro lado, se tiene que estimar también $P(\mathcal{S})$, la probabilidad de que una secuencia $\mathcal{S}$ de interlocutores participe en ese orden a priori.

De la misma manera, por el teorema de Bayes, y puesto que deseamos calcular la probabilidad $P(\mathcal{S})$ en ese orden, se puede reescribir entonces como

$$P(\mathcal{S}) = \pi_{i=1}^{K} P(s_i \mid s_1, ..., s_{i-1}) \tag{2.7}$$

De (2.7), es de donde empiezan a surgir elementos para que el usar un HMM resulte una opción natural.

Por otro lado, ahora se deberían de poder estimar las probabilidades a posteriori para cada uno de los tiempos i, $P(s_i \mid s_1, ..., s_{i-1})$. Hay que tener en cuenta que por el producto de (2.7), para cada muestra i se vendrán acarreando $i-1$ productos, por lo que el cálculo de esa probabilidad estará propenso a errores numéricos en su representación.

Por último, hay que notar también que la asunción de que el interlocutor s_i depende de la secuencia completa de interlocutores hasta ese momento $s_1, ..., s_{i-1}$ es algo rigorista e irreal; por lo que es más natural considerar solo un subconjunto $\phi(s_1, ..., s_{i-1})$ de esa secuencia.

Entonces, la (2.7) se podría escribir como sigue

$$P(\mathcal{S}) = \pi_{i=1}^{K} P(s_i \mid \phi(s_1, ..., s_{i-1})) \tag{2.8}$$

BÚSQUEDA DE HIPÓTESIS: En esta etapa, se deberá buscar de entre todos los posibles $\mathcal{S}$ cuál es el que maximiza (2.6). Como ya se mencionó, el espacio de búsqueda es realmente muy grande; por lo que no se puede hacer una búsqueda exhaustiva y deberá de seguirse una estrategia basada en la información que provee $\mathcal{A}$.

Por otra parte, y después de obtener la segmentación correspondiente para la señal de audio, se deberá ahora inferir cuál es el modelo más probable de entre todos los que se propongan. Hay que recordar que hasta ahora, se había supuesto que se conocía el número de participantes en la conversación; pero ese es un dato que también de deberá de estimar a partir de la evidencia acústica $\mathcal{A}$.

En este capítulo, se describirá a fondo la etapa de procesamiento acústico, en cuanto a todos los procesos que son necesarios para la caracterización de la señal de audio original.

En el Capítulo 3 se ahondará en las etapas tanto de modelado acústico como modelado del interlocutor, proponiendo el modelo probabilístico que se utilizará para abordar el problema.

Por último, la etapa de búsqueda de hipótesis se cubrirá tanto en la segunda parte del Capítulo 3, como en el Capítulo 4; pues hay dos momentos en los que se deben hacer selecciones: se debe de escoger el número de personas que participan en la conversación, así como la segmentación correspondiente de la señal de audio para ese número de interlocutores.

En la Figura 5.1 se muestra el esquema de los componentes principales que conforman el sistema particular utilizado para la tarea de *speaker diarization*.

2.4 PROCESAMIENTO ACÚSTICO

Antes de abordar de lleno el problema de *speaker diarization*, se tiene que realizar cierto tratamiento a la señal de audio con la que se trabajará.

Es decir, a partir de la señal de entrada (que se considerará es digital) se tratarán de obtener vectores de características cada cierto tiempo, que representen de forma adecuada los rasgos que nos interesan distinguir.

Una vez que se tienen estos vectores, se les aplica un algoritmo de aglomeración para entonces obtener un conjunto de etiquetas que se podrían considerar como posibles estados o palabras de diccionario referentes a la señal de audio.

El proceso en detalle se especifica a continuación:

2.4.1 *Pre-procesamiento de señal de audio*

Se considera que se tiene una señal digital de voz, y que a partir de ésta se identificarán a las diferentes personas que hablan durante la grabación.

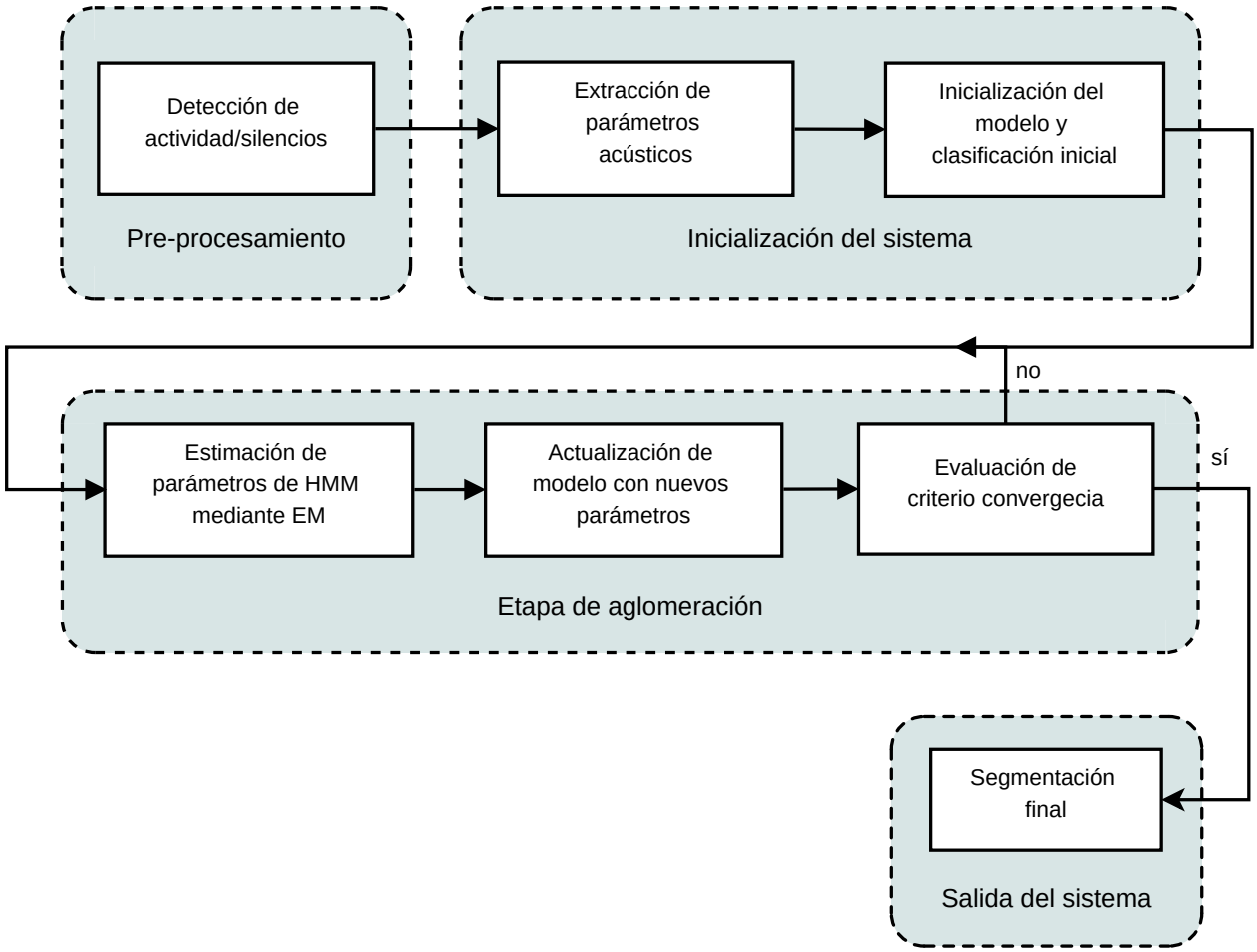


Figura 2.1: *Esquema general del sistema. Se muestran las etapas principales para la estimación de la segmentación.*

El primer paso en el procesamiento de la señal, es tanto la detección como eliminación de silencios; pues éstos realmente no nos interesan para la modelación del sistema.

Nota: Tanto en ancho de la ventana, como el umbral para definir si será silencio se pueden ajustar dependiendo del tipo de señal.

Para realizar entonces la detección de los silencios, en esta primer etapa y como un primer intento para la eliminación de silencios, se hace una detección básica de qué partes de la señal son mayormente silencios.

Para ésto, se utiliza una ventana móvil que se irá recorriendo a lo largo señal, y que irá calculando el total de energía de la señal dentro de la ventana. Se considerará entonces silencio aquellas partes de la señal cuyo total de energía esté debajo de un umbral específico.

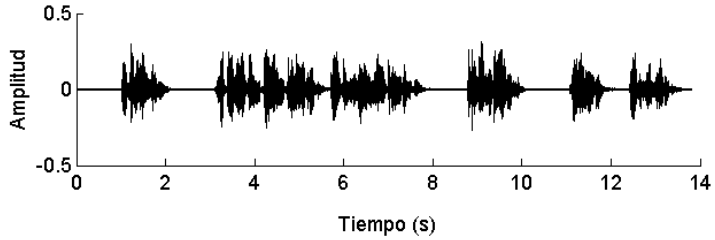
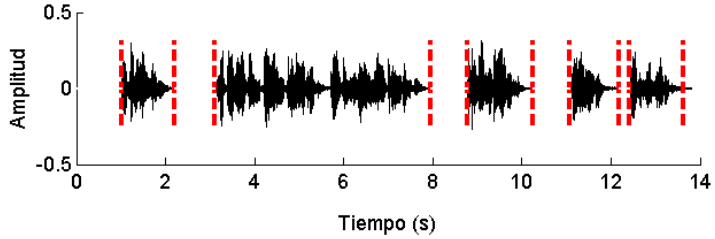
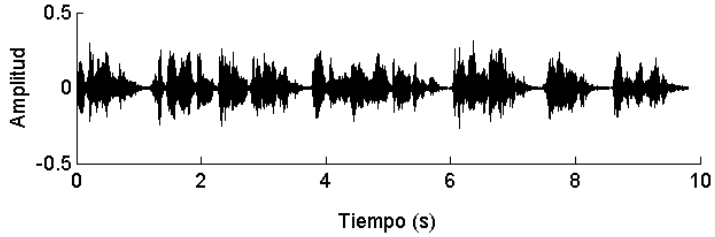


Figura 2.2: *Señal original de audio.*

A partir de esta ventana se obtienen entonces las regiones que pertenecen tanto al silencio, como a la señal que realmente se desea modelar.



(a) *Señal original con silencio identificado.*



(b) *Señal procesada y recortada.*

Figura 2.3: *Identificación y eliminación de silencio en señal.*

Como se observa en la Figura 2.3b basta entonces con eliminar los segmentos que con baja energía y reagrupar la señal restante.

Con esto, se obtiene una señal en general más pequeña, y que se podría considerar sólo contiene realmente los datos que se desea modelar.

2.4.2 *Obtención de características acústicas*

Una vez que se tiene la señal ya sin silencios, se trata de buscar características propias de la señal de audio que nos permitan identificar de buena forma los cambios de voz a través de la señal.

Tanto la extracción como selección de la mejor representación paramétrica de la señal de audio es una importante tarea en el diseño de cualquier sistema relacionado al reconocimiento o procesamiento de señales de audio.

Para la tarea de *speaker diarization*, se usarán los Coeficientes Cepstrales en la Frecuencia Mel (MFCC), que son ampliamente utilizados por ejemplo en *speaker diarization* entre otros procesos relacionados al procesamiento de voz; cuyo objetivo es comprimir la señal de audio eliminando la información que no es útil para análisis fonético.

Cabe mencionar, que originalmente los MFCC fueron diseñados para la tarea específica de reconocimiento de voz [referencia ICASSP 82], por lo que al momento de diseñarlos se trataba principalmente de que una misma palabra fuera parametrizada de la misma manera sin importar quién fuera quien la pronunciara.

Ésto va en contra del proceso requerido en *speaker diarization*, puesto que se desea identificar a las diferentes personas que hablan, sin dar tanta importancia a qué es lo que están diciendo; por lo que la tarea de segmentación de señales de audio se vuelve un poco más complicada.

Para calcular los MFCC, se usa la Escala de Frecuencia Mel, que está espaciada de forma lineal en frecuencias bajas, mientras que aumenta su separación de forma logarítmica para frecuencias más altas. Este cambio de separación se realiza comúnmente a partir de los 1000Hz.

A partir de esta escala, se diseña un banco de filtros triangulares que después se usará (Ver Figura 2.4); y esto corresponde de forma similar a la que el oído (la cóclea) captura las características importantes del habla.

Puesto que el banco de filtros de Mel trabajan en la frecuencia; a la señal de audio se le calcula la Transformada de Fourier (FT) y entonces a ésta es a quien se le aplica el banco de filtros.

Sea pues Figura 2.3b la señal procesada sin los silencios, la respuesta que se obtiene al aplicar la Transformada Rápida de Fou-

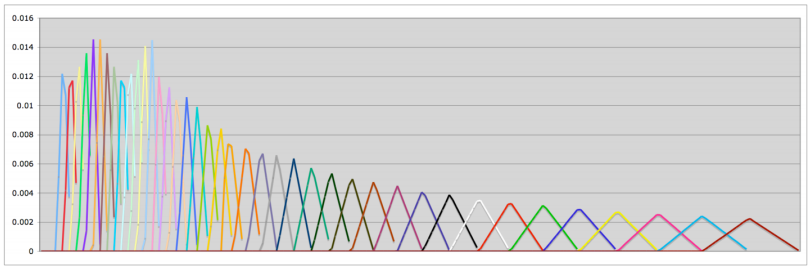


Figura 2.4: *Banco de filtros triangulares en frecuencia Mel.*

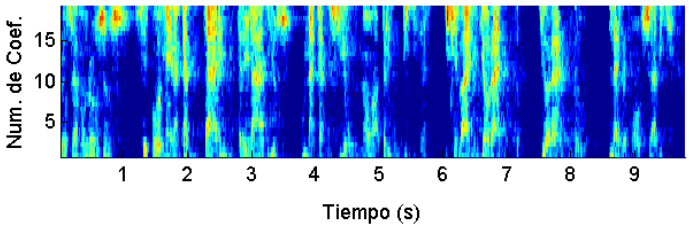


Figura 2.5: *Respuesta al banco de filtros.*

rier (FFT) y luego el banco de filtros de Figura 2.4 se puede observar en la figura Figura 2.5.

La dimensión de la respuesta al banco de filtros dependerá de la misma construcción del banco de filtros (tanto el número de canales que se usarán, como el tamaño de ventana que se usará para las convoluciones); pero en general se tendrá que es muy alta; por lo que es conveniente tratar de disminuir la dimensionalidad de estos datos.

Para esto, la respuesta obtenida del banco de filtros se le aplicará la Transformada de Coseno Discreta (DCT), para tratar de concentrar la energía en ciertos componentes (los primeros n coeficientes), y descartar los restantes.

Después de este proceso, nuestros datos se podrían representar de la siguiente manera (Figura 2.6) que son los MFCC que antes habíamos mencionado.

Por último, para los modelos que usaremos, se necesitan que las características estén de cierta forma *discretizadas*, es decir, no nos es útil el tener para cada observación en el tiempo un vector de características; sino que necesitamos una etiqueta o clase para cada observación.

Para esto, podemos utilizar diferentes métodos tanto de reducción de dimensionalidad como de agrupación/clasificación. Como

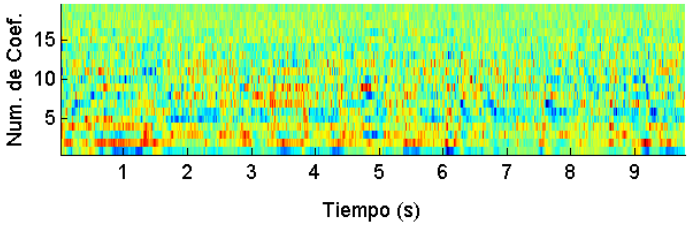


Figura 2.6: *Mel Frequency Cepstrum Coefficients.*

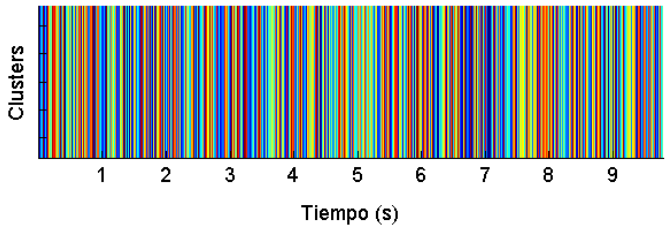


Figura 2.7: *MFCC agrupados con k-means++.*

primer idea, utilizaremos el método de *k-means* para agrupar los vectores de acuerdo a su cercanía en el espacio euclidiano.

Se tendría entonces el siguiente resultado para nuestra matriz de MFCC obtenida: Cabe aclarar que usamos una variante del algoritmo original de k-means, que se llama *k-means++* [AV07] y que propone una mejor inicialización para que el algoritmo converja más rápido. En Algoritmo 1 se describe mejor esta etapa inicial.

Algoritmo 1 *k-means++*

Input:
Conjunto de datos $\{x_n\}_1^N$, número de grupos k
Iniciar $\mu_{1:k} = sample(x_{1:N}, k)$
$t = 0$
$l_{1:N}^{(t)} = 0$
repeat
  $t = t + 1$
  $l_n^{(t)} = \arg\min_k \|x_n - \mu_k\|^2$
  $r_{nk} = \mathbb{1}_k\left(l_n^{(t)}\right)$
  $\mu_k = \left\{\sum_{n=1}^N r_{nk} x_n\right\} / \left\{\sum_{n=1}^N r_{nk}\right\}$
until $l^{(t)} == l^{(t+1)}$

Ahora sí, con nuestro vector de etiquetas correspondiente a la señal de audio, se podrá aplicar un modelo y tratar de inferir los parámetros que le correspondan.

3

Essentially, all models are wrong, but some are useful.

George E. P. Box

MODELOS DE MÁRKOV

Como ya se mencionó en el Capítulo 2, se debe proponer un modelo acústico o generativo, que será la forma en que entenderemos y trataremos de abstraer la conversación o diálogo, de acuerdo a las características que nos interesan recuperar.

Puesto que lo que nos interesa principalmente es identificar a las diferentes personas que participan en una conversación, nuestro modelo se deberá parametrizar de forma que logre capturar las características esenciales en la conversación; así como no tomar en cuenta información que no nos sea de utilidad para esta tarea, como por ejemplo, qué es lo que se está diciendo.

En aprendizaje máquina, por modelo generativo se entiende un modelo probabilístico para generar datos aleatoriamente que correspondan a la naturaleza de un cierto conjunto de datos que tengamos.

Al proponer un modelo generativo, se busca representar y de forma concisa poder parametrizar un fenómeno o situación de la que se obtuvieron los datos. Cuando los datos son secuenciales, se suelen agrupar en dos tipos, de acuerdo a las características de la distribución que los generó: distribuciones estacionarias y distribuciones no estacionarias.

En el caso de secuencias de datos estacionarias, se considera que los datos *evolucionan* a través del tiempo, pero la distribución a partir de la cuál fueron generados permanece igual; mientras que para el caso de datos no estacionarios, su distribución generativa varía también según pasa el tiempo.

Para el problema de *speaker diarization*, consideraremos que los datos son estacionarios; es decir, supondremos que su distribución generativa no cambia. Para resolver nuestro problema entonces trataremos a partir de los datos ajustar una distribución inicial e inferir sus parámetros correspondientes.

Este tipo de abstracción, permiten modelar una gran cantidad de situaciones, en las que se tienen observaciones de acuerdo al

Figura 3.1: *Observaciones independientes e idénticamente distribuidas.*

tiempo y se quiere predecir el siguiente valor en la serie, dadas la observaciones que se tienen hasta el momento.

Comúnmente se suelen considerar únicamente las observaciones más recientes, pues son las que pudiéramos pensar son más informativas para la predicción; además de que al asumir que las predicciones de nuevos datos sólo dependen de las últimas observaciones, nos permite simplificar mucho el modelo de la distribución generativa.

Para trabajar con este tipo de datos, se puede usar entonces un modelo de Márkov que es un proceso aleatorio muy trabajado en la teoría de probabilidad, que incorpora una cantidad mínima de memoria, sin necesariamente llegar a ser un proceso sin memoria.

Se puede pensar como una red bayesiana en la que se asume la independencia de las variables aleatorias que corresponden a las predicciones futuras con todas las observaciones excepto las últimas (i.e. el futuro es independiente del pasado, dado el presente).

3.1 CADENA DE MÁRKOV

Se entiende por v.a. i.i.d aquellas que son independientes entre sí y que fueron muestreadas de la misma distribución

El modelo generativo más sencillo que se podría pensar, es considerar que todas las observaciones son variables aleatorias independientes e idénticas distribuidas (v.a. i.i.d) cuyo modelo gráfico se muestra en la Figura 3.1.

Al asumir que no hay ninguna dependencia entre los datos, sin embargo, se pierde la información relativa al orden en que se fueron dando estas observaciones; situación que en muchos casos nos interesa conservar.

Si se tiene un conjunto N de observaciones $\mathbf{X} = \{x_1, ..., x_N\}$, y se asume que son i.i.d, la distribución conjunta de la secuencia de datos se puede escribir como sigue:

$$p(x_1, ..., x_N) = \prod_{n=1}^{N} p(x_n) \tag{3.1}$$

En cambio, para expresar la dependencia entre un grupo de observaciones secuenciales, se puede utilizar un modelo probabilísti-

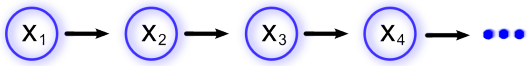


Figura 3.2: *Modelo de Márkov de primer orden.*

co llamado *modelo o cadena de Márkov*. Si por ejemplo, se considera que cada variable depende de todas las observaciones anteriores, entonces usando el teorema de Bayes se puede escribir la distribución conjunta de las observaciones de la siguiente manera:

Del teorema de Bayes se tiene la siguiente expresión

$$P(A \mid B) = \frac{P(B \mid A)P(A)}{P(B)}$$

$$p(x_1, ..., x_N) = \prod_{n=1}^{N} p(x_n \mid x_1, ..., x_{n-1}). \tag{3.2}$$

Ahora, si se asume que cada x_n es independiente de las observaciones anteriores excepto de la observación anterior x_{n-1}, se tiene que la distribución conjunta se puede escribir de forma más sencilla utilizando el teorema de separación D [1], puesto que

$$p(x_n \mid x_1, ..., x_{n-1}) = p(x_n \mid x_{n-1}) \tag{3.3}$$

A este modelo probabilístico se le conoce como cadena de Márkov de primer orden y la probabilidad conjunta de las observaciones está dada por:

$$p(x_1, ..., x_N) = p(x_1) \prod_{n=2}^{N} p(x_n \mid x_{n-1}). \tag{3.4}$$

mientras que su modelo gráfico corresponde a la Figura 3.2

Aunque el modelo es mucho más general, puede que se necesite representar de forma más fuerte la dependencia con las observaciones anteriores, por lo que se pueden usar cadenas de Márkov de órdenes superiores.

Al caso en el que x_n dependa de las dos observaciones previas se le conoce como cadena de Márkov de segundo orden; y siguiendo el mismo proceso, la distribución conjunta de las observaciones se puede escribir como

$$p(x_1, ..., x_N) = p(x_1) \cdot p(x_2 \mid x_1) \prod_{n=3}^{N} p(x_n \mid x_{n-1}, x_{n-2}) \tag{3.5}$$

puesto que se tiene que $x_n \perp x_1, ..., x_{n-3} \mid x_{n-1}, x_{n-2}$.

En este caso su modelo gráfico correspondiente es la Figura 3.3, en el que se puede observar la dependencia de una x_n en específico únicamente con sus dos observaciones anteriores.

1 Ver anexo

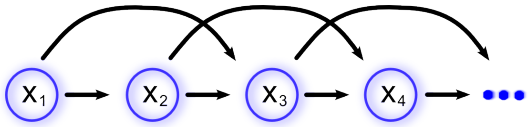


Figura 3.3: *Modelo de Márkov de segundo orden.*

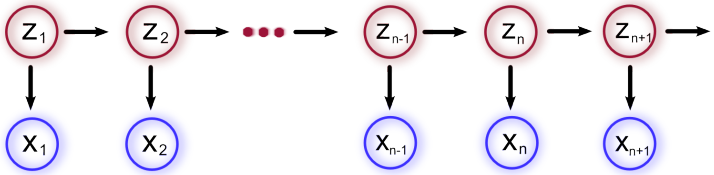


Figura 3.4: *Modelo oculto de Márkov.*

Se puede generalizar entonces para cualquier M un modelo de Márkov de M-ésimo orden, aunque al considerar demasiados estados previos se puede complicar de más el modelo; pues el número de parámetros implicados aumenta de forma exponencial al orden del modelo de Márkov.

Para evitar que el modelo se vuelva demasiado complejo, así como para no hacer alguna suposición a priori sobre cuál es el orden es del modelo generativo, se puede introducir una variable oculta, que permita cambiar el planteamiento del modelo.

Se considera entonces una variable latente z_n correspondiente a cada observación x_n, y entonces el conjunto de variables latentes forman una cadena de Márkov, como se muestra en la Figura 3.4.

Se asumirá entonces

$$z_{n+1} \perp z_{n-1} \,|\, z_n \tag{3.6}$$

Y entonces, usando la regla de la cadena, la distribución conjunta para este modelo es la siguiente:

$$p(x_1, ..., x_N, z_1, ..., z_N) \;=\; p(z_1) \left[\prod_{n=2}^{N} p(z_n \,|\, z_{n-1}) \right] \prod_{n=1}^{N} p(x_n \,|\, z_n). \tag{3.7}$$

Para éste último modelo gráfico, si las variables latentes son discretas, entonces se le conoce como Modelo Oculto de Márkov (HMM), y es justo el modelo que se utilizará para resolver la tarea de *speaker diarization*.

3.2 CADENA OCULTA DE MÁRKOV

Un Modelo Oculto de Márkov , sigue siendo entonces un modelo para datos secuenciales, en los que además se introduce el concepto de una variable oculta de la cual dependen las observaciones que se tienen; y de forma más especifica, esta variable oculta es discreta.

Usando un HMM entonces podemos modelar un proceso bivariado discreto en el tiempo, con ciertas propiedades interesantes que se mencionarán más adelante.

Nuestro modelo HMM, se puede pensar como una mezcla de distribuciones en la que la densidad tiene un distribución dada por $p(x|z)$, es decir, la mezcla de los componentes está dada por las observaciones previas.

Se puede considerar entonces, que cada variable latente z_n tendrá una distribución multinomial discreta, que indicará cuál componente de la mezcla de distribuciones es la que ha generado la observación x_n. Para ésto, se usará la notación 1-de-K, que corresponde a un conjunto de variables indicador $z_{nk} \in \{0, 1\}$, donde $k = 1, ..., K$ señalando cuál de las K distribuciones generó a la variable x_n, i. e., si el componente generador de x_n fue la k-ésima mezcla, entonces $z_{nk} = 1$ y $z_{nj} = 0$ para todo $j \neq k$.

Ahora, se tiene que cada variable z_n depende únicamente de z_{n-1}, y puesto que las variables latente son vectores binarios de K dimensiones, entonces se tendría que la probabilidad condicional de $z_n \,|\, z_{n-1}$ se puede representar mediante una tabla o matriz, que se denotará como $\mathbf{A}$, y será referida como *matriz de probabilidades de transición*.

Los componentes de la matriz de transición se definen tal que $A_{jk} \equiv p(z_{nk} = 1 \,|\, z_{n-1,j} = 1)$, y puesto que son probabilidades, se cumple que $0 \leqslant A_{jk} \leqslant 1$ además de que $\sum_k A_{jk} = 1$. Considerando estas restricciones, entonces la matriz $\mathbf{A}$ tiene $K(K-1)$ parámetros independientes.

Se puede escribir entonces la probabilidad condicional de una variable latente z_n dada la anterior variable latente z_{n-1} de la siguiente forma:

$$p(z_n \,|\, z_{n-1}, \mathbf{A}) = \prod_{k=1}^{K} \prod_{j=1}^{K} A_{jk}^{z_{n-1,j} \cdot z_{nk}} \tag{3.8}$$

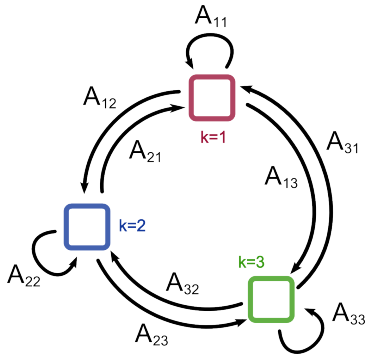


Figura 3.5: *Matriz de transición de modelo oculto de Márkov (grafo).*

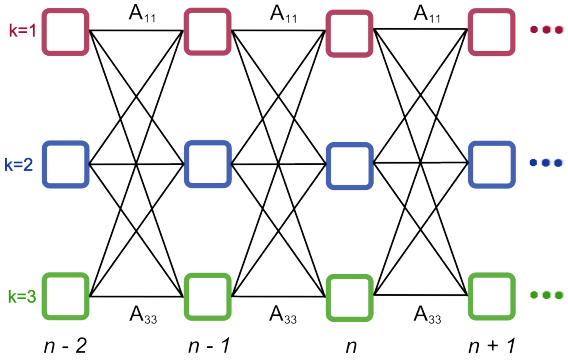


Figura 3.6: *Matriz de transición de modelo oculto de Márkov (rejilla).*

Además, se tiene que considerar la variable latente inicial z_1, puesto que ésta no tiene una variable latente anterior, la Ecuación 3.7 no aplica; por lo que entonces la probabilidad marginal de z_1 está representada por un vector de probabilidades π tal que $\pi_k \equiv p(z_{1k})$, por lo que se puede reescribir como sigue:

$$p(z_1 \mid \pi) = \prod_{k=1}^{K} \pi_k^{z_{1k}} \tag{3.9}$$

y además se cumple que $\sum_k \pi_k = 1$ por definición.

La matriz de transición también se puede llegar a representar como un grafo dirigido, si se consideran las entradas de la matriz $\mathbf{A}$ como los pesos de las aristas, y los nodos son cada uno de los posibles K estados. Por ejemplo, para el caso de una variable latente de $K = 3$ estados, se tendría el grafo de la Figura 3.5.

De la misma manera, el grafo correspondiente a la matriz de transición, se puede representar como una rejilla a través del tiempo, en la que se mantienen los vértices y aristas del grafo, pero además se introduce la noción del tiempo. Como se observa en la Figura 3.6

Por último, para completar el modelo se tiene que considerar además la distribución condicional de las variables observadas, es decir $p(x_n \mid z_n, \phi)$ donde ϕ es un conjunto de parámetros específicos a la distribución de x. Se les conoce como probabilidades de emisión, y puede ser tanto una distribución discreta como una continua. Para el caso en que la probabilidad de emisión esté dada por una distribución discreta, entonces se tendrá una tabla de probabilidades.

Para calcular entonces esta probabilidad de emisión, se tiene tanto z_n como unos parámetros ϕ dados, por lo que entonces se tienen un vector de K valores para cada uno de los posibles estados del vector indicador z_n.

Se puede escribir entonces la probabilidad de emisión como sigue:

$$p(x_n \mid z_n, \phi) = \prod_{k=1}^{K} p(x_n \mid \phi_k)^{z_{nk}} \tag{3.10}$$

y entonces la probabilidad conjunta quedaría definida de la siguiente manera:

$$p(\mathbf{X}, \mathbf{Z} \mid \theta) = p(z_1 \mid \pi) \left[\prod_{n=2}^{N} p(z_n \mid z_{n-1}, \mathbf{A}) \right] \prod_{n=1}^{N} p(x_n \mid z_n, \phi) \tag{3.11}$$

donde $\mathbf{X} = \{x_1, ..., x_N\}$, $\mathbf{Z} = \{z_1, ..., z_N\}$ y $\theta = \{\pi, \mathbf{A}, \phi\}$ es el conjunto de parámetros requeridos por el modelo.

La intuición del modelo oculto de Márkov se puede entender más fácilmente si se revisa desde el punto de vista generativo. Primero, se muestrea la variable latente inicial z_1, de acuerdo a las probabilidades π_k, así como su correspondiente x_1. Luego, se debe escoger un z_2. Para esto, si se supone que z_1 es igual a algún estado j, entonces usando la matriz de transición se muestrea z_2 con probabilidades A_{jk} para $k = 1, ..., K$, y de igual manera, su correspondiente variable observada x_2. Éste mismo proceso es el que se sigue para cada iteración en el tiempo, hasta que se forma completamente el modelo oculto de Márkov y se le conoce como *muestreo ancestral* y se suele usar para modelos con grafos dirigidos.

Si la matriz de transición es predominantemente diagonal, entonces en la secuencia de datos puede que un mismo estado i sea el que genere muchos puntos x_n, pues con poca probabilidad cambiará de un estado i a j. Este fenómeno es justo el que se espera para el caso de *speaker diarization*, pues usualmente una persona p_i hablará durante mucho tiempo, y luego cuando otra persona p_j toma la palabra, sucederá lo mismo.

3.3 ESTIMACIÓN DE MÁXIMA VEROSIMILITUD DEL HMM

La tarea de aprendizaje de parámetros del HMM se refiere a encontrar, dada una secuencia de datos observados, el mejor conjunto de probabilidades de transición y emisión.

Este problema es difícil de resolver, pues no se tiene una forma analítica de resolverlo [Rab89]. De hecho, dada una secuencia finita de observaciones como conjunto de entrenamiento, no hay forma óptima de estimar los parámetros verdaderos del modelo, y lo más que se puede hacer, es escoger $\theta = (\mathbf{A}, \mathbf{B}, \pi)$ tal que $P(\mathbf{X}|\theta)$ corresponda a un máximo local usando algún algoritmo tipo Esperanza-Maximización (EM) o técnicas de descenso de gradiente.

A continuación, se presentará el algoritmo de Baum–Welch que es básicamente EM aplicado a una cadena de Márkov, y que nos permitirá tanto inferir los parámetros θ del HMM así como luego recuperar la secuencia de estados.

3.3.1 *Estimación de parámetros del HMM*

Si se tiene un conjunto de datos observados $\mathbf{X} = \{x_1, ..., x_N\}$, se pueden determinar los parámetros del HMM usando máxima verosimilitud. La función de verosimilitud se obtiene de la distribución conjunta (3.11) al marginalizar las variables latentes.

$$p(\mathbf{X}, \theta) = \sum_{Z} p(\mathbf{X}, \mathbf{Z}|\theta) \tag{3.12}$$

Sin embargo, la función obtenida presenta algunas dificultades, pues no se puede tratar z_n como si fueran variables independientes y separarlas, puesto que cada variable latente z_n depende del estado anterior. Además, no es factible separar las sumas de estas N variables, pues para cada una se tendría que considerar cada uno de sus K posibles estados, y entonces el número de términos en la suma es del orden K^N, y crece exponencialmente con el largo de la cadena. Por esto y algunas otras razones, es que se descarta el estimar los parámetros del modelo de forma directa por máxima verosimilitud.

Por esto, se usará el algoritmo de EM para maximizar la verosimilitud. Inicialmente, se hace una selección inicial de los parámetros que denotaremos como θ^{old}. Luego, en el primer paso del *expectation-maximization*-conocido como *E-step*- se toman estos

parámetros para encontrar la probabilidad a posteriori de las variables latentes $p(\mathbf{Z}|\mathbf{X},\theta^{old})$, las cuales se usarán para evaluar la esperanza de la log-verosimilitud completa; que se puede escribir como una función tanto de la primera estimación de θ^{old} así como de los nuevos parámetros θ.

La función de verosimilitud completa se define entonces como sigue:

$$Q(\theta,\theta^{old}) = \sum_{\mathbf{Z}} p(\mathbf{Z}|\mathbf{X},\theta^{old}) \ln p(\mathbf{X},\mathbf{Z}|\theta). \tag{3.13}$$

Se introduce además cierta notación para simplificar las expresiones que ahora se usarán. Se usará $\gamma(z_n)$ para denotar la distribución marginal posterior de la variable latente z_n, y $\xi(z_{n-1},z_n)$ para denotar la distribución conjunta posterior de dos variables latentes sucesivas, es decir:

$$\gamma(z_n) = p(z_n|\mathbf{X},\theta^{old}) \tag{3.14}$$

$$\xi(z_{n-1},z_n) = p(z_{n-1},z_n|\mathbf{X},\theta^{old}) \tag{3.15}$$

Considerando entonces que z_n es un vector binario, entonces se puede extender esta notación para cada uno de los componentes de la variable latente z_n, es decir, para denotar la probabilidad condicional $z_{nk} = 1$. Se tomará entonces la esperanza de tanto $\gamma(z_{nk})$ como $\xi(z_{n-1,j},z_{nk})$ y entonces

$$\gamma(z_{nk}) = \mathbb{E}[z_{nk}] = \sum_{\mathbf{Z}} \gamma(\mathbf{z}) z_{nk} \tag{3.16}$$

$$\xi(z_{n-1,j},z_{nk}) = \mathbb{E}\left[z_{n-1,j}\cdot z_{nk}\right] = \sum_{\mathbf{Z}} \gamma(\mathbf{z}) z_{n-1,j}\cdot z_{nk} \tag{3.17}$$

Si se sustituye $p(\mathbf{X},\mathbf{Z}|\theta)$ de (3.11) en (3.13), así como usando las definiciones (3.16) y (3.17), y luego desarrollando el logaritmo, se obtiene:

$$Q(\theta,\theta^{old}) = \sum_{k=1}^{K} \gamma(z_{1k}) \ln \pi_k + \sum_{n=2}^{N}\sum_{j=1}^{K}\sum_{k=1}^{K} \xi(z_{n-1,j},z_{nk}) \ln A_{jk} + \sum_{n=1}^{N}\sum_{k=1}^{K} \gamma(z_{nk}) \ln p(x_n|\phi_k) \tag{3.18}$$

Por lo que entonces en el *E-step* se debe evaluar tanto $\gamma(z_{nk})$ como $\xi(z_{n-1,j}, z_{nk})$ de forma eficiente para luego continuar con la segunda etapa del algoritmo.

El segundo paso, también conocido como *M-step*, consiste en maximizar la unción $Q(\theta, \theta^{old})$ con respecto a cada uno de los parámetros $\theta = \{\pi, \mathbf{A}, \phi\}$ mientras que ahora $\gamma(z_{nk})$ y $\xi(z_{n-1,j}, z_{nk})$ se tratan como constantes.

Para maximizar entonces π, se deriva $Q(\theta, \theta^{old})$ con respecto a π_k, usando multiplicadores de Lagrange para la restricción de $\sum_k \pi_k = 1$, por lo que entonces se tiene:

$$\frac{\partial Q(\theta, \theta^{old})}{\partial \pi_k} = \frac{\partial}{\partial \pi_k}\left[\sum_{k=1}^{K} \gamma(z_{1k}) \ln \pi_k + \lambda\left(\sum_{k=1}^{K} \pi_k - 1\right)\right]$$

y derivando y encontrando el valor de λ, para luego despejar π_k, se obtiene:

$$\pi_k = \frac{\gamma(z_{1k})}{\sum_{j=1}^{K} \gamma(z_{1j})} \tag{3.19}$$

mientras que para maximizar $\mathbf{A}$, se deriva entonces $Q(\theta, \theta^{old})$ con respecto a A_{jk}, considerando también las restricciones, es decir

$$\frac{\partial Q(\theta, \theta^{old})}{\partial \pi_k} = \frac{\partial}{\partial \pi_k}\left[\sum_{n=2}^{N}\sum_{j=1}^{K}\sum_{k=1}^{K} \xi(z_{n-1,j}, z_{nk}) \ln A_{jk} + \lambda\left(\sum_{k=1}^{K} A_{jk} - 1\right)\right]$$

y de la misma manera, resolviendo para A_{jk} se obtiene

$$A_{jk} = \sum_{n=2}^{N} \frac{\xi(z_{n-1,j}, z_{nk})}{\sum_{l=1}^{K} \xi(z_{n-1,j}, z_{nl})} \tag{3.20}$$

Entonces, el algoritmo EM, debe ser inicializado escogiendo algunos valores para π y $\mathbf{A}$, considerando las restricciones que implican cada uno, pues representan ciertas probabilidades. Por tanto, se puede inicializar tanto π como $\mathbf{A}$ con valores aleatorios, siempre y cuando cumplan con las restricciones propias de una probabilidad.

Por último, para estimar el parámetro θ, que en realidad corresponde a estimar los parámetros propios de la distribución de emisión, éstos dependen de la distribución propia de las variables observadas, aunque basta 1 observar que en (3.18) únicamente en el último término aparece θ en forma de una suma ponderada de $\ln p(x_n \mid \phi_k)$; y en el caso de que los parámetros ϕ_k sean independientes para los diferentes componentes de la mezcla o suma ponderada, entonces maximizar con respecto a esos parámetros se puede realizar de forma sencilla.

3.3.2 *Algoritmo backward-forward*

Se presenta ahora un algoritmo que nos permite estimar de forma eficiente tanto $\gamma(z_{nk})$ como $\xi(z_{n-1,j}, z_{nk})$ que se requieren para el *E-step* del algoritmo de EM.

Para deducir el algoritmo, primero se deben tener en cuenta varias propiedades de una cadena de Márkov, que nos permitirán simplificar varios cálculos (Ver apéndice). Además, se debe tener en cuenta que éste desarrollo es general, puesto que es independiente de las probabilidades de emisión $p(x|z)$, y entonces no importa si las variables observadas son discretas o continuas, pues lo único que se requiere es poder evaluar $p(x_n|z_n)$ para cada z_n, que sabemos que son discretas.

Se puede comenzar por evaluar $\gamma(z_n)$, que es la probabilidad a posteriori $p(z_n|x_1,...,x_N)$ de z_n dado un conjunto de datos $x_1,...,x_N$. Entonces, usando el teorema de Bayes se tiene que

$$\gamma(z_n) = p(\mathbf{z}_n|\mathbf{X}) = \frac{p(\mathbf{X}|\mathbf{z}_n)p(\mathbf{z}_n)}{p(\mathbf{X})} \tag{3.21}$$

y usando la regla del producto, tenemos que

$$\gamma(z_n) = \frac{p(x_1,...,x_n,z_n)p(x_{n+1},...,x_N|z_n)}{p(\mathbf{X})} = \frac{\alpha(z_n)\beta(z_n)}{p(\mathbf{X})} \tag{3.22}$$

donde usamos la siguiente notación:

$$\alpha(z_n) \equiv p(x_1,...,x_n,z_n) \tag{3.23}$$
$$\beta(z_n) \equiv p(x_{n+1},...,x_N|z_n) \tag{3.24}$$

Se puede pensar $\alpha(z_n)$ como la probabilidad conjunta de observar toda una secuencia de datos hasta el momento n, además de el valor de la variable z_n; mientras que $\beta(z_n)$ es la probabilidad condicional para una secuencia de datos desde un momento n hasta N, dado que se conoce z_n.

Primero, desarrollamos (3.23) como sigue:

$$\begin{aligned}
\alpha(z_n) &= p(x_1,...,x_n,z_n)\\
\alpha(z_n) &= p(x_1,...,x_n|z_n)p(z_n)\\
\alpha(z_n) &= p(x_1,...,x_{n-1}|z_n)p(x_n|z_n)p(z_n)\\
\alpha(z_n) &= p(x_1,...,x_{n-1},z_n)p(x_n|z_n)
\end{aligned}$$

donde se factorizo $p(z_n)$ y luego $p(x_n|z_n)$ del resto, puesto que $x_n \perp x_1,...,x_{n-1}|z_n$; para luego volver a juntar $p(z_n)$ usando el Teorema de Bayes.

Luego, marginalizando sobre z_{n-1}, podemos escribir

$$\begin{aligned}
\alpha(z_n) &= p(x_n \mid z_n) \sum_{z_{n-1}} p(x_1, ..., x_{n-1}, z_{n-1}, z_n) \\
\alpha(z_n) &= p(x_n \mid z_n) \sum_{z_{n-1}} p(x_1, ..., x_{n-1}, z_n \mid z_{n-1}) p(z_{n-1}) \\
\alpha(z_n) &= p(x_n \mid z_n) \sum_{z_{n-1}} p(x_1, ..., x_{n-1} \mid z_{n-1}) p(z_n \mid z_{n-1}) p(z_{n-1}) \\
\alpha(z_n) &= p(x_n \mid z_n) \sum_{z_{n-1}} p(x_1, ..., x_{n-1}, z_{n-1}) p(z_n \mid z_{n-1})
\end{aligned} \tag{3.25}$$

y usando que $z_{n-1} \perp x_1, ..., x_{n-1} \mid z_n$, podemos seguir el mismo procedimiento para llegar a (3.25); y por último, podemos usar la definición de (3.23) para el primer factor de la suma, llegando a que

$$\alpha(z_n) = p(x_n \mid z_n) \sum_{z_{n-1}} \alpha(z_{n-1}) p(z_n \mid z_{n-1}) \tag{3.26}$$

Con lo que se obtiene una forma recursiva para calcular $\alpha(z_n)$ a partir de $\alpha(z_{n-1})$ para cualquier n, excepto para $n = 1$; pues no se tiene definido un z_0. Por esto mismo, podemos definir el caso inicial $\alpha(z_1)$ usando (3.23) y entonces resultaría:

$$\alpha(z_1) = p(x_1, z_1) = p(z_1) p(x_1 \mid z_1) \tag{3.27}$$

De la misma manera, para el caso de $\beta(z_n)$ desarrollando a partir de (3.24)

$$\begin{aligned}
\beta(z_n) &= p(x_{n+1}, ..., x_N \mid z_n) \\
\beta(z_n) &= \sum_{z_{n+1}} p(x_{n+1}, ..., x_N, z_{n+1} \mid z_n) \\
\beta(z_n) &= \sum_{z_{n+1}} p(x_{n+1}, ..., x_N \mid z_n, z_{n+1}) p(z_{n+1} \mid z_n) \\
\beta(z_n) &= \sum_{z_{n+1}} p(x_{n+1}, ..., x_N \mid z_{n+1}) p(z_{n+1} \mid z_n)
\end{aligned}$$

donde primero agregamos la variable z_{n+1} y marginalizamos con respecto a ella, para luego factorizar $p(z_{n+1} \mid z_n)$. Después, considerando que $x_{n+1}, ..., x_N \perp z_n \mid z_{n+1}$, podemos simplificar la expresión.

A partir de ahí, podemos factorizar $p(x_{n+1} \mid z_{n+1})$ con lo que llegamos a

$$\beta(z_n) = \sum_{z_{n+1}} p(x_{n+2}, ..., x_N \mid z_{n+1}) p(x_{n+1} \mid z_{n+1}) p(z_{n+1} \mid z_n)$$

(3.28)

donde usando (3.24) obtenemos de nuevo una forma recursiva para $\beta(z_n)$

$$\beta(z_n) = \sum_{z_{n+1}} \beta(z_{n+1}) p(x_{n+1} \mid z_{n+1}) p(z_{n+1} \mid z_n) \tag{3.29}$$

que en este caso dependería del valor de $\beta(z_{n+1})$ y aplicaría para cualquier n, excepto cuando $n = N$.

Para este caso, igual partiríamos de la definición en (3.22)

$$p(z_n \mid \mathbf{X}) = \frac{\mathbf{p}(\mathbf{X}, \mathbf{z_N})_{,}(\mathbf{z_N})}{\mathbf{p}(\mathbf{X})} \tag{3.30}$$

de donde se obtiene que

$$\beta(z_N) = 1 \tag{3.31}$$

3.3.2.1 *Factor de escala*

Para la implementación, se tiene que considerar un problema común del algoritmo de backward-forward: como ya se mencionó, en el proceso recursivo para calcular cada $\alpha(z_n)$ se utilizan los previamente calculados $\alpha(z_{n-1})$, además de unas multiplicaciones por un par de probabilidades. Puesto que cada probabilidad es por definición menor o igual a 1, y que esta operación se realiza iterativamente para cada estado n, se tiene entonces que los valores de $\alpha(z_n)$ decrecerán rápidamente, y llegará un momento en el que no se puedan representar en una computadora (por los límites tanto de la notación punto flotante como del doble punto flotante).

Comúnmente, cuando se tienen problemas de precisión, se suele tomar el logaritmo, y así se amplia el rango, evitando posibles desbordamientos. Sin embargo, para el caso presentado, no es posible realizar esto, pues tanto para $\alpha(z_n)$ como $\beta(z_n)$ se manejan sumas de números pequeños, y entonces no tiene sentido usar el logaritmo; pues el logaritmo no se aplicaría directamente sobre esos valores que pudieran ser pequeños; sino sobre la suma de ellos.

Se propone entonces manejar las probabilidades de forma reescalada. Originalmente, $\alpha(z_n)$ representa la probabilidad conjunta de todas las variables observadas $x_1, ..., x_n$ junto con la variable

latente z_n. Se usará entonces $\hat{\alpha}(z_n)$ como la probabilidad condicional de la variable latente z_n dadas todas las observaciones antes mencionadas. Es decir:

$$\hat{\alpha}(z_n) = p(z_n \mid x_1, ..., x_n) = \frac{\alpha(z_n)}{p(x_1, ..., x_n)} \tag{3.32}$$

El orden entre las cantidades se mantendrá, puesto que se dividirá entre la probabilidad conjunta de las variables observadas $p(x_1, ..., x_n)$.

Como en algún momento nos será útil regresar del espacio escalado al espacio original de las variables, es importante calcular estos factores de escala para cada una de las $\hat{\alpha}(z_n)$.

$$c_n = p(x_n \mid x_1, ..., x_{n-1}) \tag{3.33}$$

y entonces podemos calcular el factor de escalamiento usando la regla del producto

$$p(x_1, ..., x_n) = \prod_{i}^{n} c_i \tag{3.34}$$

y entonces, para obtener $\alpha(z_n)$ a partir de $\hat{\alpha}(z_n)$ se procede de la siguiente manera:

$$\begin{aligned} \alpha(z_n) &= p(x_1, ..., x_n, z_n) = p(x_1, ..., x_n) p(z_n \mid x_1, ..., x_n) \\ &= \left(\prod_{i}^{n} c_i \right) \hat{\alpha}(z_n) \end{aligned} \tag{3.35}$$

De la misma manera, la forma recursiva que se había obtenido en (3.26) se puede reescribir de la siguiente manera:

$$c_n \alpha(z_n) = p(x_n \mid z_n) \sum_{z_{n-1}} \hat{\alpha}(z_{n-1}) p(z_n \mid z_{n-1}) \tag{3.36}$$

por lo que ahora, a cada paso de la recursión al calcular $\hat{\alpha}(z_{n-1})$, se debe calcular también c_n e ir almacenando los coeficientes que normalizan a $\alpha(z_n)$.

Ahora, para re-escalar $\beta(z_n)$ usando los coeficientes c_n se tendría que

$$\beta(z_n) = \left(\prod_{i}^{n} c_i \right) \hat{\beta}(z_n) \tag{3.37}$$

donde

$$\hat{\beta}(z_n) = \frac{p(x_{n+1}, ..., x_N \mid z_n)}{p(x_{n+1}, ..., x_N \mid x_1, ..., x_n)} \tag{3.38}$$

y por lo tanto tampoco presentaría problemas numéricos el cálculo de $\hat{\beta}z_n$, puesto que vuelve a ser un cociente de probabilidades. En este caso, de la condicional de las variables observadas desde $x_{n+1}, ..., x_N$ dada la variable latente z_n sobre la probabilidad condicional de las mismas variables observadas dadas todas las variables anteriores a x_{n+1}.

Entonces, para calcular recursivamente los valores de $\hat{\beta}(z_n)$, se deriva que

$$c_{n+1}\hat{\beta}(z_n) = \sum_{z_{n+1}} \hat{\beta}(z_{n+1})p(x_{n+1}\,|\,z_{n+1})p(z_{n+1}\,|\,z_n) \quad (3.39)$$

donde usamos los coeficientes de re-escalamiento que ya habíamos calculado (y almacenado) junto con los valores de $\hat{\alpha}(z_n)$.

Con las nuevas variables normalizadas, ya se pueden realizar todos los cálculos necesarios.

Por ejemplo, para calcular la verosimilitud del modelo, usando (3.34) se tiene que

$$p(\mathbf{X}) = \prod_{i}^{N} c_i \quad (3.40)$$

Así como para el algoritmo de backward-forward, se pueden reescribir tanto (**??**) como (**??**) usando las nuevas variables

$$\gamma(z_n) = \hat{\alpha}(z_n)\hat{\beta}(z_n) \quad (3.41)$$

$$\xi(z_{n-1}, z_n) = c_n\hat{\alpha}(z_{n-1})p(x_n\,|\,z_n)p(z_n\,|\,z_{n-1})\hat{\beta}(z_n) \quad (3.42)$$

3.3.3 *Implementación*

Hasta ahora, se tiene la metodología completa para estimar los parámetros de un modelo propuesto; es decir, se considera que se sabe a priori el número de interlocutores que participan en la grabación. A partir de ello se estimarán los parámetros correspondientes del modelo.

Para la estimación de los parámetros se usa el algoritmo EM, donde primero se proponen de forma aleatoria las probabilidades iniciales así como las matrices de transición y emisión correspondientes. Como condiciones de paro del método se establece un número fijo de iteraciones que depende del número de la longitud de la secuencia observada, además de usar una tolerancia mínima para la diferencia entre la log-verosimilitud de dos iteraciones contiguas.

Después de tener los parámetros refinados por el algoritmo *backward-forward*, se debe proponer una técnica para escoger la segmentación resultante, lo que equivale a encontrar la secuencia de estados *óptima* correspondiente a la secuencia observada.

Como menciona Rabiner [Rab89], hay muchas formas en las que se puede definir la secuencia óptima, de acuerdo a los intereses que se necesitan cumplir. Un criterio puede ser por ejemplo, escoger las probabilidades marginales de las variables latentes $\gamma(z_n)$ (3.19) más probables de forma individual, es decir:

$$q_k = \underset{1<=k<=K}{\arg\max}\ \gamma(z_{nk}), \qquad 1 <= n <= N \tag{3.43}$$

Otra opción, puede ser usar el algoritmo de Viterbi que nos permite obtener la secuencia de estados más probable dada una secuencia observada. Por diseño, se escogió el primer método, y se buscará el conjunto de estados que individualmente son más probables.

Choose well. Your choice is brief, and yet endless.

Johann Wolfgang von Goethe

4 SELECCIÓN DE MODELO

Puesto que el problema abarca también el detectar cuántos personas están involucrados en la grabación, y hasta ahora se ha considerado que se dispone de esta información, es necesario inferir de alguna manera cuántos interlocutores participan.

La contribución principal de este trabajo de tesis está en la selección de modelo propuesta, utilizando varias técnicas tanto bayesianas como frecuentistas para respaldar la elección realizada.

Como mencionan Claeskens y Hjort [CH10], hay varios aspectos importantes que considerar antes de abordar el problema de selección de modelos:

LOS MODELOS SON APROXIMACIONES: Cuando se usan modelos, se tiene que considerar que la realidad observada suele ser mucho más compleja que los modelos propuestos. No necesariamente existirá un modelo correcto

SESGO-VARIANZA: Se refiere a balancear tanto la simplicidad del modelo (pocos parámetros a estimar, lo que implicaría una menor variabilidad, aunque con cierto sesgo) contra la complejidad (introducir más parámetros al modelo reduciría el sesgo al modelar, pero aumentaría el grado de variabilidad). La selección estadística de modelos debe buscar un balance entre el sobre-ajuste (un modelo con muchos parámetros, más de los necesarios) o sub-ajuste (un modelo con muy pocos parámetros, no capture).

PARSIMONIA: 'El principio de parsimonia' o navaja de Ockham dice que 'en igualdad de condiciones, la explicación más sencilla suele ser la correcta'. Se puede pensar como incluir en el modelo sólo los parámetros que realmente importen y capturen la esencia del fenómeno.

CONTEXTO: Todo modelado tiene cierto propósito. Se pueden tener diferentes intereses para un mismo experimento, por lo que entonces el contexto no tiene que ser siempre el mismo para un conjunto de datos. En algunos contextos puede ser

más interesante encontrar los parámetros subyacentes del modelo e interpretarlos, mientras que en otros puede bastar con obtener respuesta a las problema planteado.

Con estas consideraciones, se presentarán varios conceptos importantes que se utilizarán al momento de seleccionar el modelo adecuado:

4.1 FUNCIONES DE PENALIZACIÓN

Una estrategia sencilla para la selección de modelo es elegir el candidato con la más grande probabilidad dados los datos.

Comparar directamente los valores máximos alcanzados de log-verosimilitud para diferentes modelos no siempre es un criterio lo suficientemente bueno para la comparación de modelos. Al incluirse más parámetros en un modelo, la máxima log-verosimilitud también aumentará, pues el modelo se sobre-ajustará; por lo que no siempre será la mejor elección.

Escoger el modelo con la mayor log-verosimilitud equivaldría a siempre elegir el modelo con más parámetros; lo que puede significar que tiene un buen poder predictivo para los datos que se usaron de entrenamiento, pero que probablemente en pruebas con otros datos diferentes no tendrá un buen desempeño.

Para evitar esto, se han diseñado funciones de penalización que permiten

Entre las más comunes, se encuentran por ejemplo el Criterio de Información Akaike (AIC) o el Criterio de Información Bayesiano (BIC), que se encargan de selección de modelo a partir de la estimación de máxima verosimilitud de los datos, así como también penalizan el número de parámetros libres que necesita el modelo.

4.1.1 *BIC*

Cuando hay varios modelos candidatos, una estategia bayesiana se encargaría de seleccionar el modelo que a posteriori sea más probable. Este modelo puede ser identificado calculando la probabilidad posterior de cada uno de los modelos y luego seleccionando aquél modelo cuya probabilidad sea la mayor.

Sean $\mathcal{M}_1, ..., \mathcal{M}_k$ los modelos propuestos, y sea $\mathbf{X} = \{x_1, ..., x_n\}$ el vector de datos observados. La probabilidad a posteriori para cada modelo se puede calcular como sigue:

$$P(\mathcal{M}_j \,|\, \mathbf{X}) \equiv \frac{P(\mathcal{M}_j)}{f(\mathbf{X})} \int_{\Theta_j} f(\mathbf{X} \,|\, \mathcal{M}_j, \theta_j) \pi(\theta_j \,|\, \mathcal{M}_j) d\theta_j \tag{4.1}$$

donde Θ_j es el espacio de parámetros al que pertenece θ_j. Además, $f(\mathbf{X} \,|\, \mathcal{M}_j, \theta_j)$ es la verosimilitud $\mathcal{L}_j(\theta_j)$ de los datos, dado al modelo j y sus parámetros; mientras que $\pi(\theta_j \,|\, \mathcal{M}_j) d\theta_j$ representa la densidad a priori de θ_j dado el modelo $\mathcal{M}_j$; $P(\mathcal{M}_j)$ es la probabilidad a priori para el modelo j-ésimo y $f(\mathbf{X})$ es la verosimilitud de los datos.

La verosimilitud incondicional de los datos se puede calcular como sigue:

$$f(\mathbf{X}) = \sum_{j=1}^{k} P(\mathcal{M}_j) \lambda_{n,j}(y) \tag{4.2}$$

donde

$$\lambda_{n,j} = \int_{\Theta_j} \mathcal{L}_{n,j}(\theta_j) \pi(\theta_j \,|\, \mathcal{M}_j) d\theta_j. \tag{4.3}$$

La ecuación (4.3) representa la verosimilitud marginal de los datos para el modelo $\mathcal{M}_j$ integrada con respecto a θ_j sobre es espacio de parámetros Θ_j correspondiente.

Al comparar las probabilidades posteriores $P(\mathcal{M}_j \,|\, \mathbf{X})$ de los distintos modelos, $f(\mathbf{X})$ se mantiene constante para todos los modelos, por lo que se puede descartar en la comparación.

Ahora, si se define

$$BIC_{n,j}^{exact} \equiv 2log(\lambda_{n,j}(\mathbf{X})) \tag{4.4}$$

por lo que (4.1) se podría reescribir como sigue:

$$P(\mathcal{M}_j \,|\, \mathbf{X}) = \frac{P(\mathcal{M}_j) exp(\frac{1}{2} BIC_{n,j}^{exact})}{\sum_{i=1}^{k} P(\mathcal{M}_i) exp(\frac{1}{2} BIC_{n,i}^{exact})} \tag{4.5}$$

Sin embargo, el cálculo de los diferentes $BIC_{n,j}^{exact}$ es difícil de estimar numéricamente, además de que la expresión necesita las probabilidades a priori para todos los modelos y todos los parámetros; por lo que se buscará una expresión similar que sea práctica y mucho más eficiente.

Para esto, primero hay que considerar el método de Laplace, que es usado para aproximar integrales de la forma $\int_a^b e^{Mf(x)}dx$, por lo que (4.3) debe escribirse de esa manera:

$$\lambda_{n,j}(\mathbf{X}) = \int_{Theta} exp\left\{nh_{n,j}(\theta)\cdot\pi(\theta\,|\,\mathcal{M}_j)\right\}\ d\theta \tag{4.6}$$

donde $h_{n,j}(\theta) = n^{-1}\ell_{n,j}(\theta)$ y $p = \#\theta$ es la cardinalidad del θ; por lo que ahora usando la aproximación básica de Laplace, se tiene:

$$\begin{aligned}\int_\Theta exp\left\{nh_{n,j}(\theta)\cdot\pi(\theta\,|\,\mathcal{M}_j)\right\}\ d\theta = \\ \left(\frac{2\pi}{n}\right)^{\frac{p}{2}}\cdot exp\{nh(\theta_0)\}\cdot\left\{g(\theta_0)\cdot|J(\theta_0)|^{-\frac{1}{2}}+O(n^{-1})\right\}\end{aligned} \tag{4.7}$$

donde θ_0 es el valor que maximiza la función $h(\cdot)$ y $J(\theta_0)$ es la matriz Hessiana

$$J(\theta) \equiv -\frac{\partial^2 h(\theta)}{\partial\theta\cdot\partial\theta^T} \tag{4.8}$$

evaluada en θ_0.

Hay que notar que esta aproximación es equivalente a (4.4) sólo cuando h es una forma cuadrática negativa (como lo es una log-verosimilitud gaussiana) y cuando $g(\cdot)$ es constante.

En este caso, tenemos que $h(\theta) = n^{-1}\ell_{n,j}(\theta)$ y que se maximiza con el Estimador de Máxima Verosimilitud (MLE) $\hat{\theta}_j$ para el modelo $\mathcal{M}_j$. Entonces, con

4.2 BOOTSTRAP

Bootstrap es una técnica estadística que nos permite tener noción sobre qué tan precisa es alguna medida muestral estimada. Este método permite aproximar la distribución de muestreo de casi cualquier estadístico, usando métodos simples aunque computacionalmente intensivos.

Como menciona Persi et.al [DE83], esta técnica fue desarrollada en 1978 por Efron [Efr78], quien generalizó el método de *Jackknife*; siendo ambos métodos estadísticos altamente demandantes en cuanto a procesamiento computacional; y que empezaron a tener más auge a finales de la década de 1970 debido al crecimiento del poder de cómputo disponible.

La técnica bootstrap se describe a continuación en un ejemplo: supongamos que se tiene que ajustar un modelo a un conjunto

de datos. Sea este conjunto de entrenamiento $\mathbf{Z} = (z_1, z_2, ..., z_N)$ donde $z_i = (x_i, y_i)$ y son independientes con distribución F. Como $\mathbf{Z}$ es una muestra finita no se conoce tal cual la distribución F, pero se puede estimar una función empírica $\hat{F}$ donde a cada observación z_i se le asigna un peso $\frac{1}{N}$ en la densidad.

Con esto, seleccionaremos de forma aleatoria y con reemplazo de $\hat{F}$ un conjunto de datos del mismo tamaño que el conjunto original. A este conjunto se le denotara $\mathbf{Z}^*_1$. Este proceso de selección se realiza B veces, produciendo B conjuntos bootstrap $\mathbf{Z}^*_\cdot = \{\mathbf{Z}^*_1, \mathbf{Z}^*_2, ..., \mathbf{Z}^*_B\}$.

Luego, para cada uno de estos conjuntos, se volvera a ajustar el modelo, y se examinará el comportamiento de los ajustes para las respuestas obtenidas, obteniendo lo que se conoce como réplica bootstrap.

En (4.9), $S(\mathbf{Z})$ representa cualquier estadístico calculado del conjunto de datos $\mathbf{Z}$. A partir de los conjuntos muestreados se puede estimar cualquier aspecto de una distribución de $S(\mathbf{Z})$, como por ejemplo, su varianza:

$$\widehat{Var}\left[S(\mathbf{Z})\right] = \frac{1}{B-1}\sum_{b=1}^{B}\left(S(\mathbf{Z}^{*b}) - \bar{S}^*\right)^2 \tag{4.9}$$

donde $\bar{S}^* = \frac{1}{B}\sum_{b=1}^{B} S(\mathbf{Z}^{*b})$. Se puede pensar en $\widehat{Var}\left[S(\mathbf{Z})\right]$ como un estimador tipo Monte-Carlo para la varianza de $S(\mathbf{Z})$ a partir del muestro de la función de distribución empírica $\widehat{F}$ de los datos $(z_1, z_2, ..., z_n)$. De esta misma manera, se puede calcular la desviación estándar de otros estimadores de interés, tales como el coeficiente de correlación, algún cuantil, etc.

En la Figura 4.1 se muestra el proceso general que se sigue para obtener las réplicas bootstrap.

Hay que observar que lo que hace bootstrap es realmente estimar la varianza muestral a partir de la distribución empírica $\hat{F}$ y conforme $B \to \infty$, $\widehat{Var}(\hat{F})$ tiende a la varianza poblacional $\widehat{Var}(\hat{F}) = Var(\hat{F})$ de la misma $\hat{F}$. Para que realmente $\widehat{Var}(\hat{F})$ converja a $Var(F)$ se necesita además que $N \to \infty$, es decir, que se tengan muestras infinitas de la distribución original.

4.2.1 *Bootstrap paramétrico*

Como ya se mencionó, el bootstrap clásico es no paramétrico, y se vale únicamente del conjunto observado para a partir de ahí

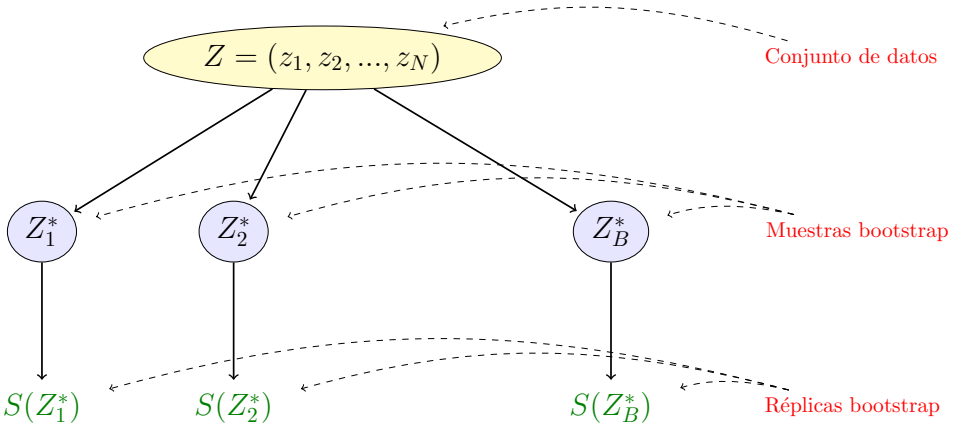


Figura 4.1: *Esquema del proceso bootstrap. Se desea estimar la precisión del estadístico* $S(\mathbf{Z})$*. Para esto, se generan* B *conjutos, cada uno muestreando con reemplazo* N *elementos del conjunto original. A cada uno de estos* $b = 1, 2, ..., B$ *elementos se le denomina* $\mathbf{Z}_b^*$*, y es a partir de estos que se calcula el estadístico de interés. Usando estas réplicas bootstrap es como se estima la varianza del* $S(\mathbf{Z})$ *previamente calculado.*

estimar la función de distribución empírica. En el caso que nos interesa, se cuenta con un modelo paramétrico que ha sido ajustado a los datos, usualmente por MLE.

Entonces, a partir de este modelo ajustado es que se muestrea. Al igual que en con la técnica no paramétrica, se suelen generan muestras de datos del mismo tamaño que el conjunto original. Luego, para cada nueva conjunto bootstrap $\mathbf{Z}_b^*$ muestreado se calcula el estadístico de nuestro interés. Éste proceso de muestreo se repite igualmente una gran cantidad de veces.

La diferencia principal con el método clásico de bootstrap, es que al usar un modelo paramétrico al momento de muestrar permite utilizar técnicas diferentes de las que comúnmente se usan para el bootstrap tradicional.

4.3 SELECCIÓN DE MODELO USANDO BIC

Como ya se mencionó al principio de este capítulo, hay muchas formas de aboradar el problema de selección de modelo. En este trabajo se usará una combinación de los dos tipos presentados.

Puesto que consideramos que dentro de nuestro espacio de modelos propuestos se encuentra el modelo solución (esto es, hay un modelo que corresponde con el número de interlocutores que participan en la grabación), resulta más natural usar BIC.

Esta función de penalización se calcula de la siguiente manera:

$$\mathrm{BIC}(\mathcal{M}) = 2\mathcal{L}_{max}(\mathcal{M}) - (\log N)\dim(M) \tag{4.10}$$

para cada propuesta de modelo $\mathcal{M}$, donde $\dim(M)$ es el número estimado de parámetros libres que le corresponden, y N es el tamaño de nuestra muestra de datos. Por otra parte, $\mathcal{L}_{max}(\mathcal{M})$ es la máxima log-verosimilitud obtenida para el modelo $\mathcal{M}$ después de realizar un número K se simulaciones, para evitar que en algún caso el algoritmo de estimación se quede atorado en un máximo local.

Para estimar el número de parámetros libres de nuestro modelo, se consideran todas las probabilidades que rigen al HMM, que en este caso son: la matriz a priori o inicial, la matriz de transiciones entre los interlocutores, así como la matriz de emisión de cada persona para todo el diccionario de palabras.

Este tipo de función de penalización nos permite seleccionar de entre un conjunto de modelos propuestos (que pueden ser muchos) al modelo o los modelos con mayor probabilidad de ser los correctos.

Se enfoca en escoger el modelo candidato con la mayor probabilidad dados los datos, pero penalizando la complejidad de cada propuesta. Se preferirán entonces los modelos con una mayor verosimilitud que involucren la menor cantidad de parámetros posibles.

En caso de que dos o más modelos tengan una puntaje similar en BIC, se buscará hacer un análisis más extenuante para revisar cuál modelo es más conveniente.

4.4 SELECCIÓN DE MODELO USANDO BOOTSTRAP CON LIKELIHOOD RATIO TESTING

Para esta otro propuesta, se utilizará la técnica bootstrap paramétrico, pues mediante el algoritmo EM es fácil obtener el modelo parametrizado. El estadístico a evaluar será el LLR, que nos permitirá comparar entre dos modelos propuestos.

Usualmente se compararán dos modelos adyacentes, es decir, el modelo $\mathcal{M}_d$ de d estados contra el modelo $\mathcal{M}_{d+1}$ de $d+1$ estados ocultos.

La prueba que se realiza es comparando el MLE $\hat{\theta}^{(d)}$ y $\hat{\theta}^{(d+1)}$ de los modelos de d y $d+1$ estados respectivamente. Para la com-

Algoritmo 2 Muestreo ancestral para un HMM

Input:
núm. de estados N, núm. de muestras en el tiempo T
núm. de posibles valores en el diccionario K,
$\pi \in \mathcal{R}^N$; $\pi_j = p(z_{1j})$
$\mathbf{A} \in \mathcal{R}^{N\times N}$; $\mathbf{A}_{jk} = p(z_{nk}\,|\,z_{n-1,j})$
$\mathbf{E} \in \mathcal{R}^{K\times N}$; $\mathbf{E}_{jk} = p(x_{nk}\,|\,z_{n,j})$

$z_1 \sim Multinomial(\pi)$
$x_1 \sim Multinomial(\mathbf{E}_{[z_1,\,:]})$
for $i = 1 \to T$ **do**
  $z_t \sim Multinomial(\mathbf{A}_{[z_{t-1},\,:]})$
  $x_t \sim Multinomial(\mathbf{E}_{[z_t,\,:]})$
end for

paración se usa la estadística LLR que corresponde a la diferencia de las log-verosimilitudes mencionadas

$$LLR_{obs}^{(d)} = \log\frac{L(\hat{\theta}^{(d+1)};y_{1:n})}{L(\hat{\theta}^{(d)};y_{1:n})} = \log L(\hat{\theta}^{(d+1)};y_{1:n}) - \log L(\hat{\theta}^{(d)};y_{1:n}) \tag{4.11}$$

Para calcular el MLE de un modelo, se estimó la verosimilitud varias veces con diferentes parámetros iniciales aleatorios. Se iteró el algoritmo EM hasta convergencia, un numero $iter_{hmm}$ fijo de iteraciones, esto para evitar el estancamiento del algoritmo en un máximo local, y obtener así una buena estimación de la máxima verosimilitud del modelo.

Se usó entonces como MLE del modelo la máxima verosimilitud correspondiente a los mejores parámetros estimados y que se denotó por LLR_{obs}.

Ahora, para hacer la prueba con boostrap se simularán datos de los HMM propuestos, utilizando el algoritmo 2 para el muestreo, tanto para el modelo $\mathcal{M}_d$ como para el modelo $\mathcal{M}_{d+1}$. Para ambos modelos se estimará su log-verosimilitud y se procederá con el cálculo del LLR, que es la réplica bootstrap que nos interesa.

Para cada par de modelos $\mathcal{M}_d, \mathcal{M}_{d+1}$ se realizará este procedimiento B veces: muestreando ambos modelos, estimando su verosimilitud y calculando el estadístico LLR. Para cada simulación bootstrap se generará un LLR_{boot}^{b} referente a la muestra generada.

Cuando se termine el proceso bootstrap, se tendrán $\{LLR^b_{boot}\}^B_{b=1}$ estadisticos, y se podrá generar una densidad sobre los valores de LLR^b_{boot}.

La última etapa será realizar una prueba de hipótesis para ver si el valor LLR_{obs} tiene la misma distribución que los $\{LLR^b_{boot}\}^B_{b=1}$. La hipótesis nula será que el modelo $\mathcal{M}_d$ es el modelo correcto, y entonces el valor LLR_{obs} estará en el rango mismo de los LLR_{boot}. Sin embargo, si se rechaza la hipótesis nula (de acuerdo a un nivel de significancia establecido) significará que el modelo $\mathcal{M}_{d+1}$ es mejor que el $\mathcal{M}_d$.

Parte III

MARCO EXPERIMENTAL

There was nowhere to go but
everywhere, so just keep on rolling
under the stars.

Jack Kerouac

5 METODOLOGÍA

A continuación se describe la metodología que se llevo a cabo para la tarea de *speaker diarization*.

Como ya se describió en el Capítulo 2 es necesaria una etapa de pre-procesamiento en el sistema; primero para eliminar de la señal de audio las cosas que no nos interesan (que en este caso son silencios), para lo que se usó un detector de energía y así ignorar los instantes en los que ningún interlocutor participa.

Luego, también en la etapa de pre-procesamiento se encuentra el obtener vectores de características que representen de alguna manera la señal, y que sean mucho más fáciles de manipular. Para esta etapa se usaron los MFCC, siguiendo el algoritmo **??**. Después de obtener los vectores característicos, se agruparon mediante k-means++ (Algoritmo 1) para de alguna forma discretizar todas las posibles palabras emitidas en el diálogo. Esta agrupación inicial nos permitirá reducir la dimensionalidad de los vectores, así como aglomerar aquellos que sean muy similares con respecto a los demás.

Después de esta etapa de clasificación, ya se tendrán los datos como observaciones que representarán las variables observadas del HMM. Como se desconoce tal cual el número de personas involucradas en la grabación, se propondrán varios modelos $\mathcal{M}_d$ con d estados ocultos. El valor de d variará de acuerdo a qué tan extensas se deseen hacer las pruebas, pero a priori no hay un límite pre-establecido.

Luego, corresponderá usar el algoritmo EM con cada uno de los modelos $\mathcal{M}_d$ propuestos, tanto para estimar sus parámetros como para estimar su verosimilitud. Como ya se mencionó en Capítulo 4 esta etapa se realizará varias veces, para evitar el estancamiento del método y obtener un buen ajuste del modelo. Del modelo con mayor versomilitud se calculará la segmentación correspondiente, y se almacenará.

Después, seguirá la etapa de selección de modelo. Este proceso se realizara en dos partes: primero, explorando todo el espacio de

soluciones y reduciendo los posibles modelos ganadores; y luego, seleccionando de entre los candidatos resultantes al mejor modelo.

En un primero paso, se estimará BIC para de todos los HMM propuestos para generar una curva de selección. Como se verá en las pruebas, es necesario introducir un término de regularización en BIC para que la penalización del modelo corresponda con las log-verosimilitudes obtenidas.

Esto se puede hacer de la siguiente forma:

$$\mathrm{BIC}_{\lambda}(\mathcal{M}) = 2\mathcal{L}_{max}(\mathcal{M}) - \lambda \cdot \log N \cdot \dim(M) \tag{5.1}$$

aunque presenta el inconveniente de tener que encontrar el valor adecuado para λ que penalice de buena forma los modelos.

Para escoger el valor adecuado de λ se realizará un análisis de sensibilidad, generando múltiples curvas de selección BIC con diferentes valores de λ. Tanto la discretización como los rangos de λ permitirán hacer una mejor búsqueda del parámetro. Con todas estas diferentes curvas, se tendrá una superficie, en donde en un eje variará el número de interlocutores del modelo, mientras que en el otro será el valor de regularización λ el que cambiará.

Se busca mediante este análisis encontrar la región de inflexión que divide a la superficie en dos: en la primera parte de la superficie, el valor de λ será pequeño, por lo que siempre tendrán una mayor verosimilitud los modelos con más parámetros; mientras que en la segunda parte, la penalización sera muy grande, y se escogerán siempre los modelos más sencillos, sin darle tomar en cuenta su verosimilitud.

Por esto mismo, se puede calcular el gradiente de la superficie generada por las funciones BIC, y se buscará la región el valor de λ en el que la suma de los valores absolutos sea menor.

Es importante recordar que el modelo HMM al ser resuelto usando EM buscará siempre maximizar la función de verosimilitud; por lo que es natural que mientras más parámetros tenga un modelo, mejor será su ajuste a los datos y mayor su verosimilitud.

Basándonos en esto, si se comparan distintos modelos con un λ pequeño, se espera que los modelos más complejos sean los que tengan un valor BIC mayor; por lo que habrá un gradiente en esa dirección con pendiente positiva. Por otro lado, cuando el λ sea demasiado grande, los modelos más sencillos seran los que tengan asociado un valor de BIC mayor; y entonces el gradiente estará en el otro sentido, con una pendiente negativa.

El caso que nos interesa es cuando el valor de λ penaliza de forma correcta, es decir, está en el mismo orden de magnitud que la mayoría de las verosimilitudes de los modelos. En ese momento, el gradiente de la superficie deberá tener dos direcciones, pues el valor máximo de BIC no corresponderá al modelo más simple o al más complicado. Y además, la suma de los gradientes para ese valor será menor con respecto a otros valores de λ.

Una vez que se logre seleccionar el λ adecuado de regularización, se procede a evaluar su curva BIC asociada y de ahí se obtiene al modelo ganador o un subconjunto de posibles ganadores (en caso de que varios modelos tengan puntuaciones similares de acuerdo a BIC).

Como ya se mencionó, en el segundo paso, se realizará un proceso de refinamiento en caso de que se tengan varios modelos posibles. Para esto, se formarán pares de modelos que se deseen comparar, y se estimará su LLR, que se denominará como $\mathrm{LLR}_{\mathrm{obs}}$. Para este mismo par de modelos, y mediante bootstrap paramétrico se hará una prueba de hipótesis para comproar cuál modelo es más adecuado para los datos.

Esto es, a partir de los dos modelos a examinar, se simularán múltiples secuencias de datos con los parámetros que se estimaron para cada modelo; y luego, con los datos simulados se estimará su máxima verosimilitud, para luego calcular el LLR, que ahora se denominará $\mathrm{LLR}_{\mathrm{boot}}$.

Como esta simulación se harán muchas réplicas, y con todos los valores $\mathrm{LLR}_{\mathrm{boot}}$ que se obtengan se formará una curva. Mediante una prueba de hipótesis se revisará si el $\mathrm{LLR}_{\mathrm{obs}}$ tiene la misma distribución que la curva de $\mathrm{LLR}_{\mathrm{boot}}$. La hipótesis nula corresponderá a que el primer modelo sea el correcto, y entonces tanto $\mathrm{LLR}_{\mathrm{boot}}$ como $\mathrm{LLR}_{\mathrm{obs}}$ tengan la misma distribución. Por otro lado, la hipótesis alternativa significaría que se rechaza $\mathrm{LLR}_{\mathrm{obs}}$ como una muestra de $\mathrm{LLR}_{\mathrm{boot}}$ bajo una significancia dada; y entonces se rechaza que el primer modelo sea el correcto.

De esta forma, se pueden realizar pruebas de hipótesis para los modelos candidatos, e ir rechazando modelos de acuerdo al análisis propuesto. En el Capítulo 6 se describe más a detalle esta selección de moledo, con las pruebas realizadas.

5.1 ESQUEMA GENERAL

El esquema general con las diferentes etapas que se realizan para estimar el número de personas así como su segmentación correspondiente.

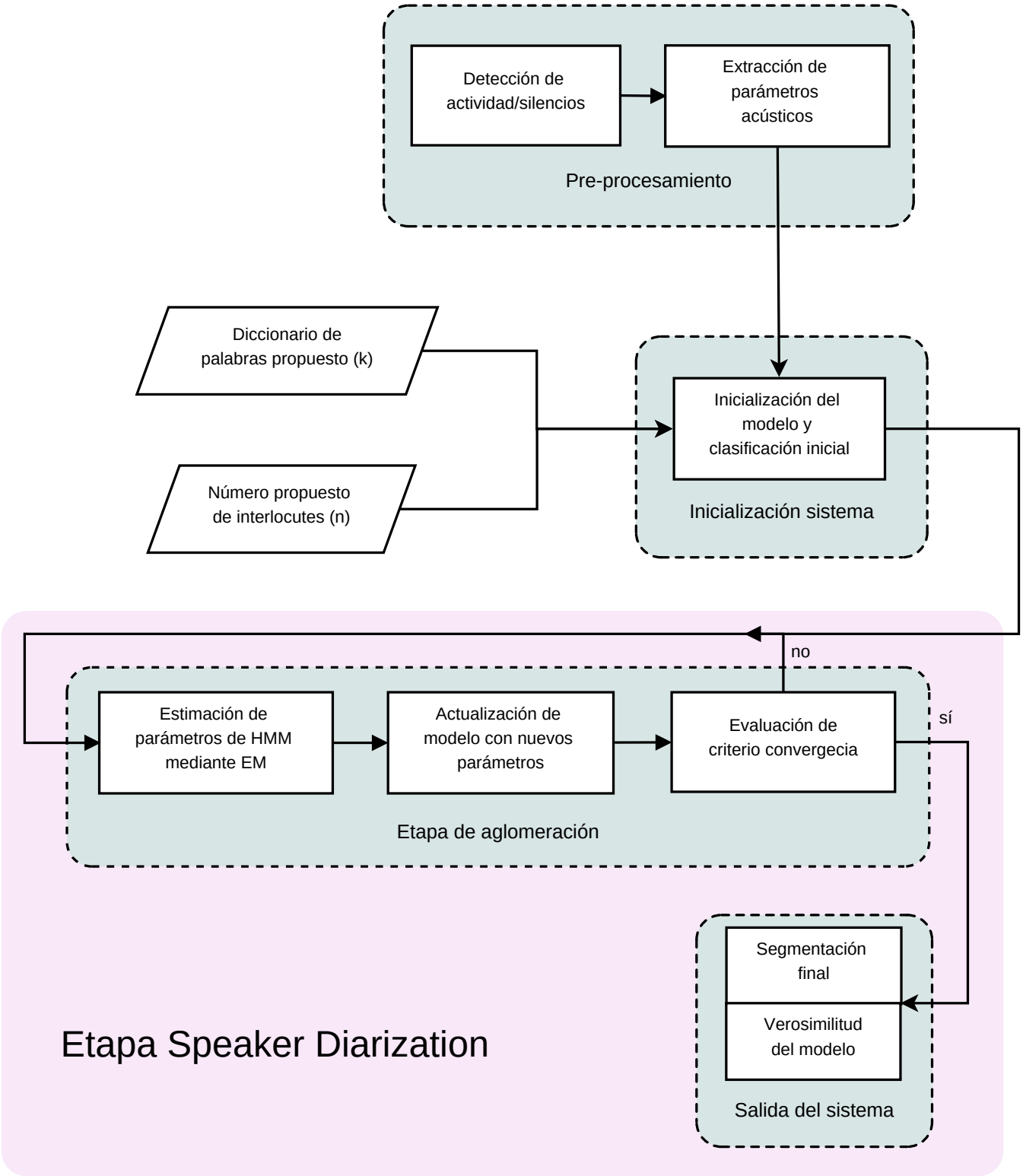


Figura 5.1: *Esquema general.*

"Data! Data! Data!" he cried impatiently. "I can't make bricks without clay."

Sherlock Holmes

6

EXPERIMENTOS Y RESULTADOS

En los capítulos anteriores se ha descrito los diferentes algoritmos que se utilizarán para realizar la tarea de *speaker diarization*, y que en esta sección se emplearán de acuerdo al marco experimental que se describe a continuación.

Inicialmente, las pruebas consistieron en usar los algoritmos presentados para selección de modelo usando datos que fueron generados aleatoriamente a partir de los parámetros de un HMM inicial; para tener una idea general de su desempeño individual.

Para estas primeras pruebas, se simuló una cadena de Márkov oculta con base en parámetros fijos, generando tanto una secuencia de datos observados, como los supuestos datos o variables ocultas que forman la cadena de Márkov. Se utilizó muestreo ancestral para la simulación de estos datos.

Para un caso en específico, se tiene lo siguiente. Realizando la inferencia de parámetros del HMM, se obtienen los siguientes resultados:

El primer algoritmo que se prueba, es el de selección de modelo usando un BIC.

Como ya se comentó, se usará una variante de BIC en donde se incorpora un término de regularización λ para que correspondan en órdenes de magnitud tanto la log-verosimilitud del modelo encontrado como su penalización respectiva.

El problema inmediato que se presenta, es cómo realizar la selección del parámetro de regularización λ que penalice de forma correcta la verosimilitud para los diferentes modelos propuestos. Si λ es demasiado pequeño, entonces la penalización realmente no tendrá efecto y dado el sobre ajuste que se presenta al usar modelos más complejos, se preferirán siempre los modelos con más parámetros. Por otro lado, si al escoger λ se da demasiado peso al término de regularización, entonces siempre se preferirán los modelos más sencillos.

Para encontrar el valor de λ adecuado, se puede entonces formar una superficie con las diferentes curvas de selección BIC de acuerdo a cómo varía λ, e inspeccionar esta superficie para encontrar una región de confianza en la que el valor de λ es el adecuado.

Por otro lado, para la segunda prueba, se procedió a usar bootstrap con la estadística LLR como ya se describió anteriormente en el Capítulo 3, y haciendo la prueba de hipótesis del modelo de n estados contra el de $n+1$ estados.

6.1 EXPERIMENTOS

Para los experimentos realizados, se generaron mediante un Sintetizador de voz (TTS) para la generación de las secuencias de prueba, lo que nos permitió tener un mayor control sobre el contenido como tal de las grabaciones, así como sobre los posibles ruidos o interferencias en la señal de audio.

Si bien, para probar el desempeño contra otras propuestas del estado del arte se suelen usar otro tipo de bases de datos (tales como NIST, ...), éstas suelen no estar disponibles de forma libre, por lo que preferimos generar nosotros un conjunto de pruebas con el sintetizador de voz.

Usando dos motores para el sintetizador de voz, uno con voces en inglés y otro con voces en español, se generaron 6 secuencias de audio (3 en cada idioma) cuya duración así como el número de interlocutores que participan varía.

6.1.1 *Secuencia 1: Edgar Allan Poe*

En esta primer secuencia se tomaron varios poemas del escritor Edgar Allan Poe, y se utilizaron 6 diferentes voces en inglés. La secuencia de audio original es de 12:06 min.

Para la etapa de agrupación de los vectores MFCC con k-means++ se usaron 140 centros iniciales, mientras que el banco de filtros fue el mismo que anteriormente se mencionó.

En las Figura 6.1 y Figura 6.2 se muestran los parámetros estimados para los diferentes modelos que se propusieron. La primer columna corresponde a los parámetros verdaderos, mientras que las demás columnas son los parámetros obtenidos para diferentes modelos estimados.

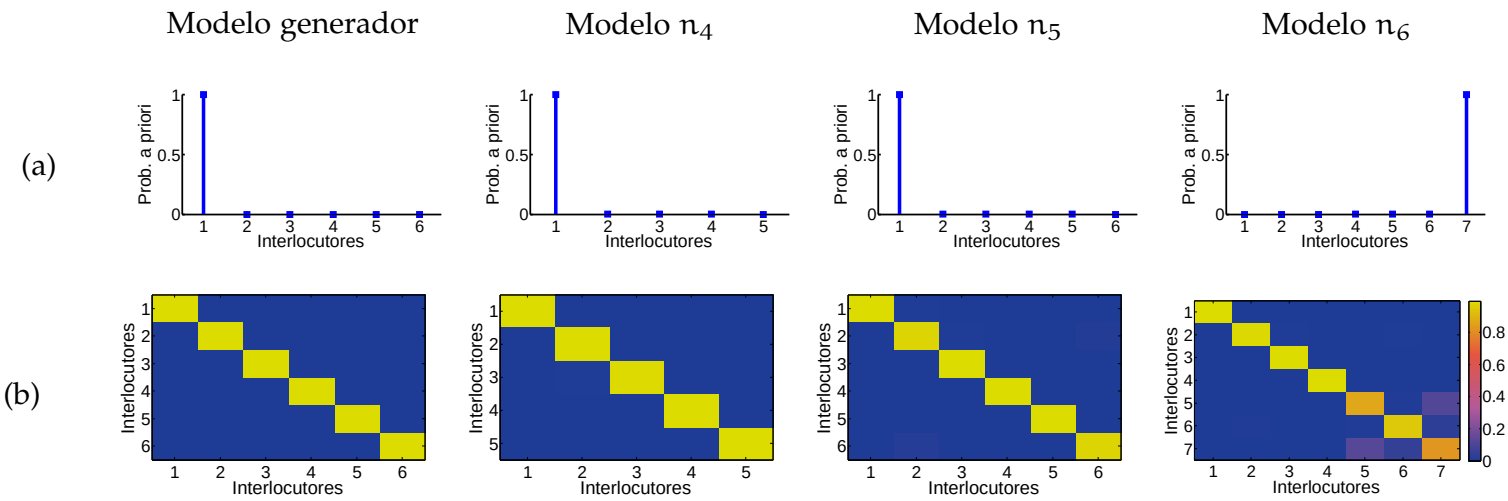


Figura 6.1: *Por columnas, se muestran los parámetros obtenidos para varios modelos propuestos. La primer columna corresponde a los parámetros verdaderos. En la primer fila* (a) *se muestran las probabilidades a priori de que cada uno de los interlocutores empiecen la conversación. En la segunda fila* (b) *se muestra en falso color la matriz de transición entre interlocutores para cada uno de los modelos mostrados.*

En la primer fila de la Figura 6.1 se muestran las probabilidades a priori de que cada uno de los interlocutores empiecen el diálogo. En la segunda fila se muestra en falso color la matriz de transición para cada uno de los modelos. Esta matriz representa la probabilidad de cambio entre las personas que participan en la conversación.

Se observa como en general para todos los modelos propuestos la matriz de transición que se recupera tiene una estructura diagonal, puesto que en este tipo de problemas, una persona suele hablar durante un periodo considerablemente largo, para que luego otra persona empiece a hablar.

En la Figura 6.2 se muestran las probabilidades de emisión para cada uno de los participantes, de acuerdo a las palabras en las que se ha discretizado la conversación, como se explicó anteriormente en el algoritmo aAlgoritmo 1. Idealmente, cada interlocutor tiene asignadas con mayor probabilidad un cierto conjunto de palabras del diccionario, lo que hace que la identificación de las personas sea mucho más fácil de realizar.

Para la selección del modelo y siguiendo la metodología propuesta, primero se efectúa una inspección por medio de BIC regularizado para encontrar cuál o cuáles son los modelos más probables.

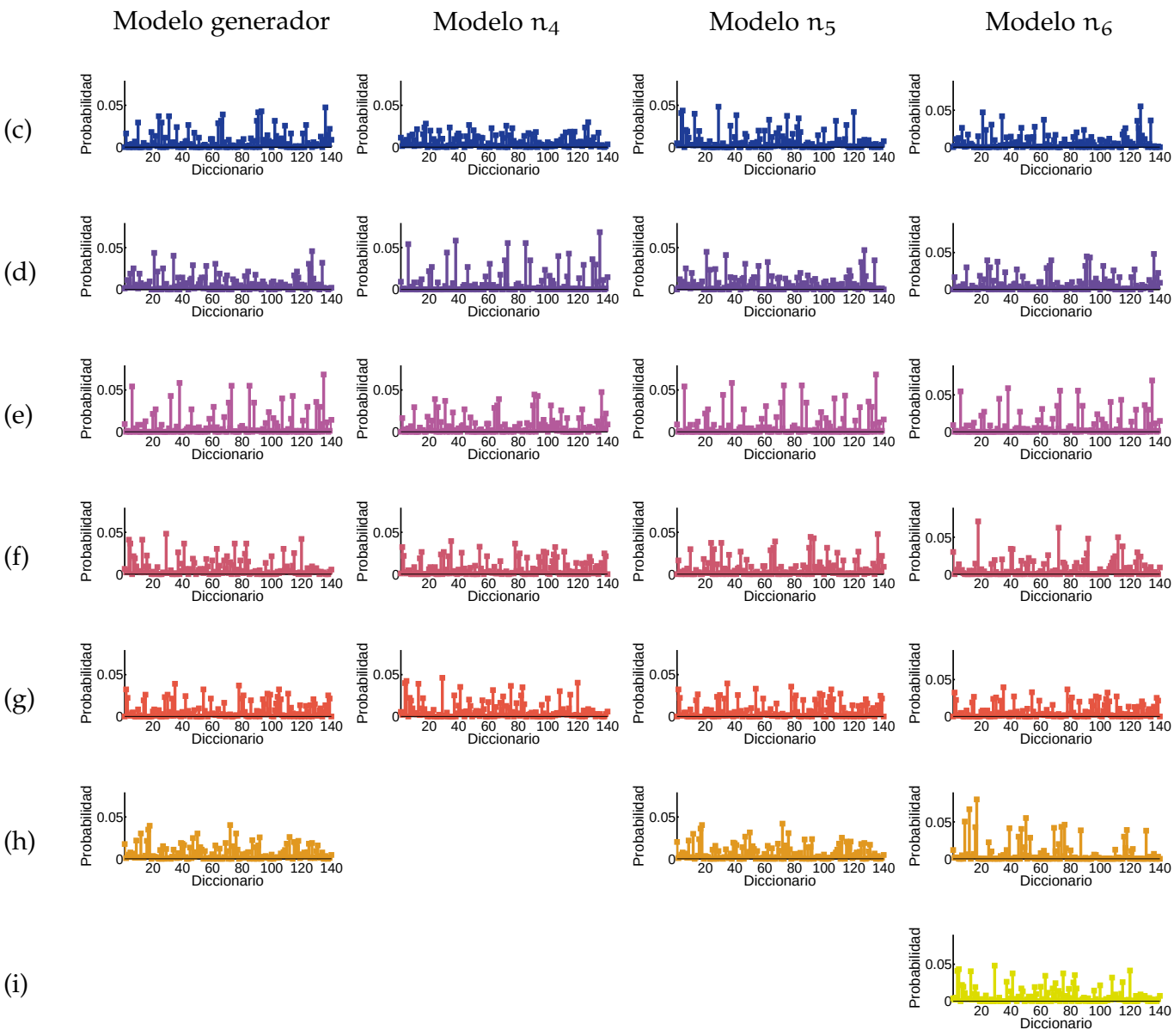


Figura 6.2: *Por columnas, se muestran las probabilidades de emisión para cada uno de los interlocutores de acuerdo al modelo. Cada fila representa las probabilidad de emisión de las palabras en el diccionario. La fila* (c) *para la primer persona, la fila* (d) *para la segunda persona y así de forma consecutiva, de acuerdo al número de personas que contempla cada modelo propuesto.*

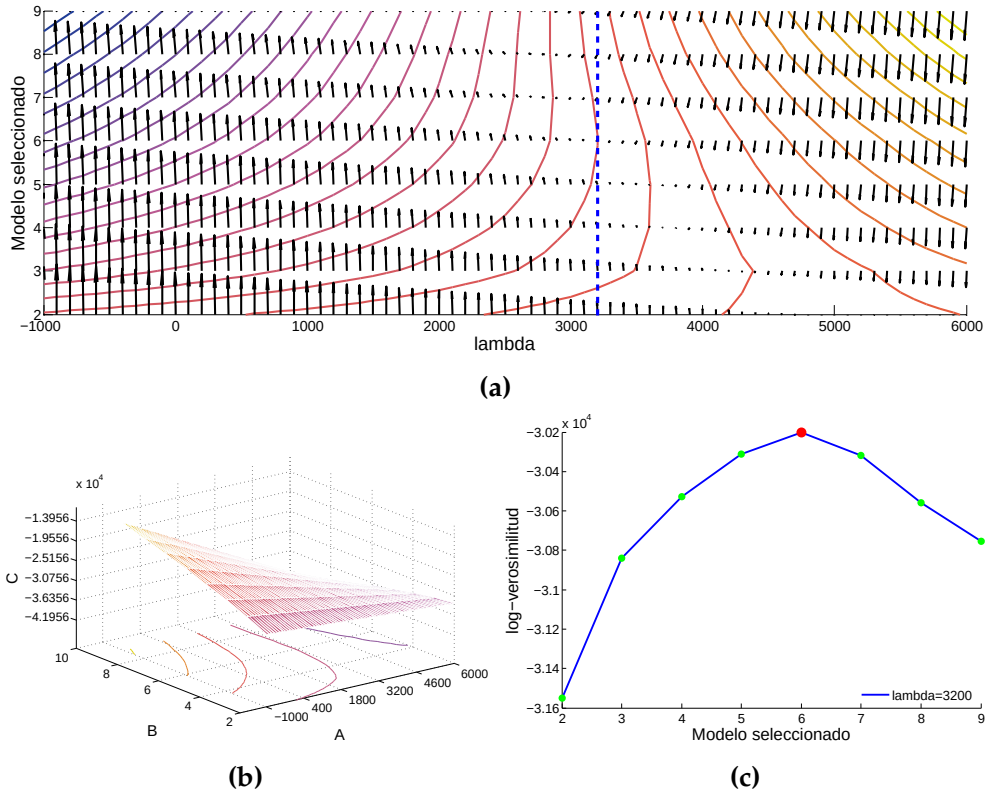


Figura 6.3: *En la Figura 6.3a, se muestra las curvas de nivel de la superficie al variar el valor de lambda para evaluar BIC, así como la dirección del gradiente en la misma. En la Figura 6.3b se muestra una perspectiva general de superficie en Figura 6.3a. En la Figura 6.3c se muestra el valor seleccionado para lambda de acuerdo al análisis de sensibilidad realizado en la superficie anterior.*

En la Figura 6.3b se muestra la superficie generada al variar el valor de λ para diferentes curvas de selección BIC. Se observa cómo al principio λ es muy pequeño, y entonces el término de penalización no funciona por lo que se prefieren los modelos más complejos y sobre ajustados. Por otro lado, cuando λ es muy grande, la penalización no permite mas que escoger los modelos más simples.

Haciendo un análisis de sensibilidad se puede determinar cuál es el parámetro λ adecuado que penaliza de buena forma la log-verosimilitud.

En la Figura 6.3a se muestran las curvas de nivel de la figura anterior, además de la dirección del gradiente de la misma. Así mismo, en falso color se resaltan las zonas en las que el gradiente es menor.

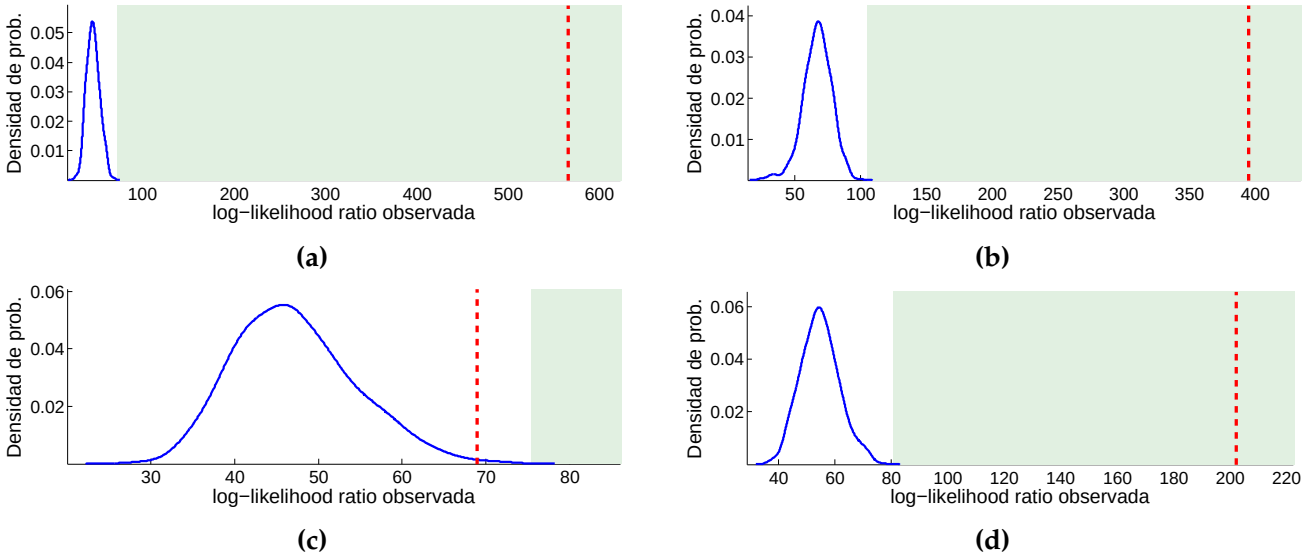


Figura 6.4: *En la Figura 6.4a se muestra la prueba de hipótesis realizada para comparar el modelo n_4 contra n_5. Como se observa, se rechaza la hipótesis de que el modelo correcto sea n_4. En la Figura 6.4b se hace la prueba del modelo n_5 contra n_6, y de la misma manera, se rechaza la hipótesis de que el modelo correcto sea n_5. Se sigue con la prueba de hipótesis del modelo n_6 contra n_7 en la Figura 6.4c, y en este caso el valor observado no cae dentro de la región de rechazo, por lo que no podemos rechazar que el modelo n_6 sea el correcto. Por último en la Figura 6.4d, se hace la prueba del modelo n_7 contra el modelo n_8, y se vuelve a rechazar la hipótesis nula.*

Lo que nos interesa encontrar en la superficie, es el valor de λ que representa el punto de inflexión entre la selección de modelos demasiado complejos y modelos más simples. Para esto, se busca la zona en la que el gradiente sea lo más cercano a cero, pues implicaría que es un punto crítico.

Debido a la escala y a la forma en la que se calculó el gradiente, aunque para algunos valores cercanos de λ no haya mucha variación en esa dirección; si BIC está penalizando mal, entonces sí habrá gran variación para los diferentes modelos. Es por esto, que sólo en la zona en que la penalización sea del mismo orden de magnitud que la verosimilitud la variación en la curva BIC con el λ adecuado no será tan grande como en otras zonas.

Por último, se muestra en Figura 6.3c la curva BIC con el valor de λ encontrado a partir del análisis de sensibilidad realizado en la Figura 6.3a. El o los modelos que tengan un mayor valor BIC serán los que se seleccionarán como posibles soluciones.

En caso de que después de utilizar BIC se presente ambigüedad para determinar un modelo ganador, o ya sea para realizar un

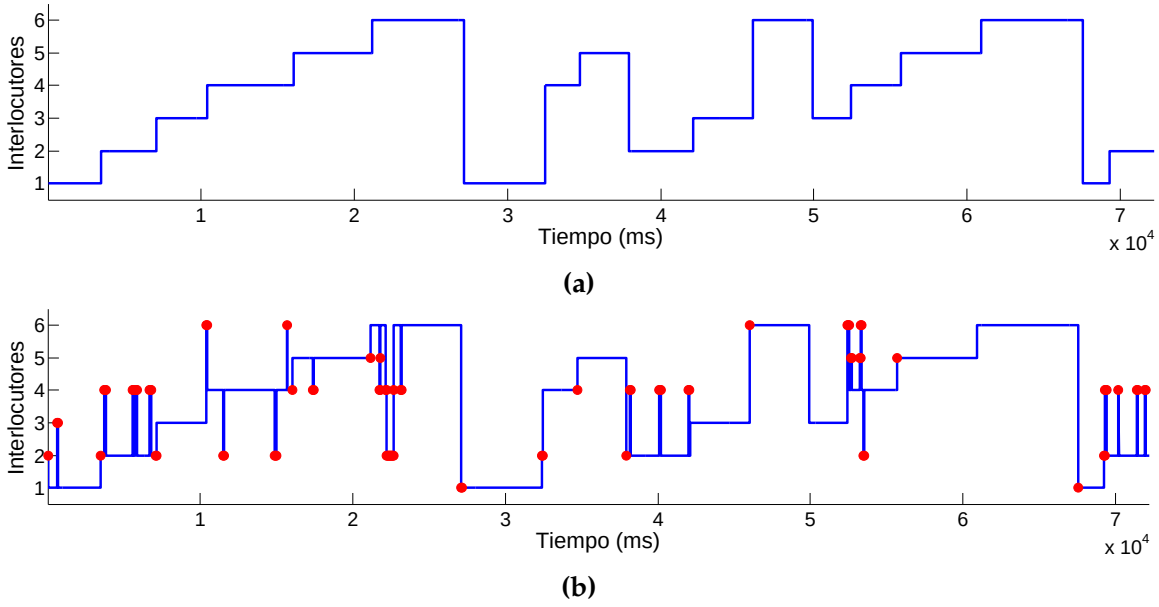


Figura 6.5: *En Figura 6.5a se muestra la secuencia original de la Subsección 6.1.1. En comparación, en Figura 6.5b se muestra la secuencia recuperada del modelo ganador y se marcan en rojo los errores cometidos respecto a la primera secuencia.*

análisis más exhaustivo, se puede propone hacer una prueba de hipótesis para determinar cuál modelo se ajusta mejor a los datos.

A diferencia de la primer etapa, en la que se usa BIC como criterio para seleccionar el mejor modelo de un conjunto no definido de modelos con diferentes parámetros, la intención de hacer pruebas de hipótesis es determinar en un pequeño conjunto de probables modelos, cuál es mejor, y qué tan bueno es un modelo respecto a otro.

Al plantear la prueba de hipótesis se harán una gran cantidad de simulaciones para ver qué tan bien se ajusta cada modelo a los datos originales, por lo que este proceso es computacionalmente intensivo y sólo se recomienda hacerlo para evitar la ambigüedad entre un par de modelos.

Por último, ya con el modelo seleccionado, se compara la segmentación recuperada con la segmentación original de la conversación:

Más a detalle, en la Figura 6.5 se observa en azul el orden en el que participan los interlocutores de acuerdo a la secuencia recuperada. En rojo se marcan tanto los falsos positivos como los falsos negativos, de acuerdo al ground truth. Se observa también que la mayoría de las veces, en la secuencia recuperada se encuentran

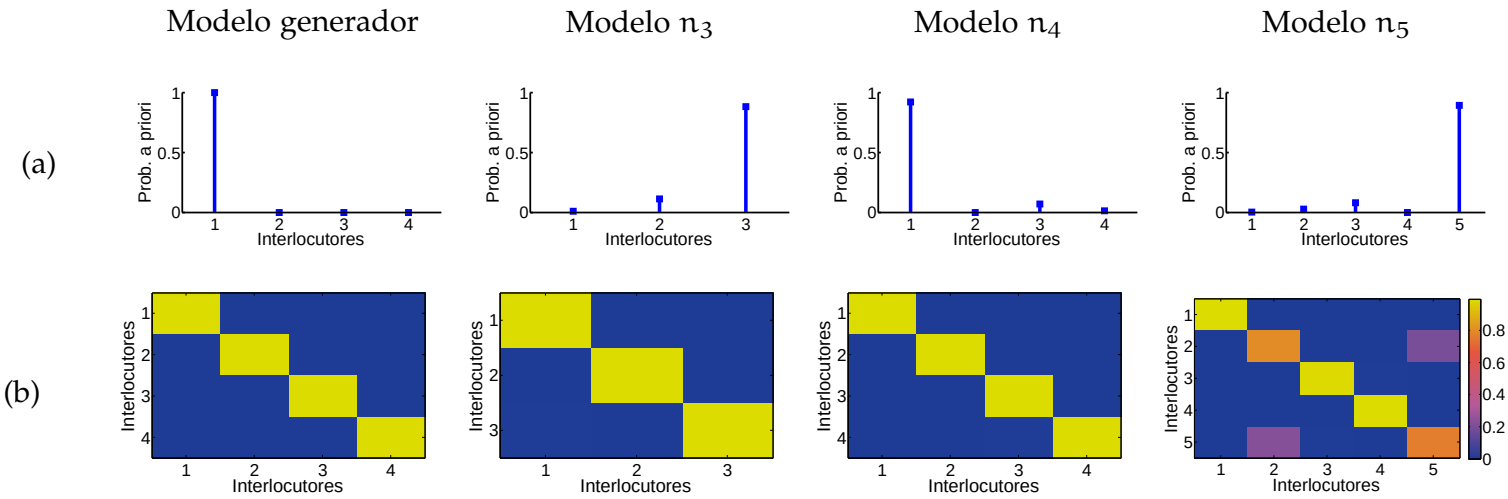


Figura 6.6: *Por columnas, se muestran los parámetros obtenidos para varios modelos propuestos. La primer columna corresponde a los parámetros verdaderos. En la primer fila* (a) *se muestran las probabilidades a priori de que cada uno de los interlocutores empiecen la conversación. En la segunda fila* (b) *se muestra en falso color la matriz de transición entre interlocutores para cada uno de los modelos mostrados.*

algunos brincos entre las personas, pero en esencia la estructura y el orden en que hablan los interlocutores es el correcto.

6.1.2 *Secuencia 2: Gabriel García Márquez*

Para la segunda secuencia se utilizaron algunos fragmentos de la novela 'El laberinto de la soledad' del escritor Gabriel García Márquez, con 4 distintas voces en español. La secuencia de audio original es de 6:55 min.

Para la etapa de agrupación de los vectores MFCC con k-means++ se usaron 90 centros iniciales, mientras que el banco de filtros fue el mismo que en la secuencia anterior.

En las Figura 6.6 y Figura 6.7 se muestran los parámetros estimados para los diferentes modelos que se propusieron. La primer columna corresponde a los parámetros verdaderos, mientras que las demás columnas son los parámetros obtenidos para diferentes modelos estimados.

En la primer fila de la Figura 6.6 se muestran las probabilidades a priori de que cada uno de los interlocutores empiecen el diálogo. En la segunda fila se muestra en falso color la matriz de transición para cada uno de los modelos. Esta matriz representa

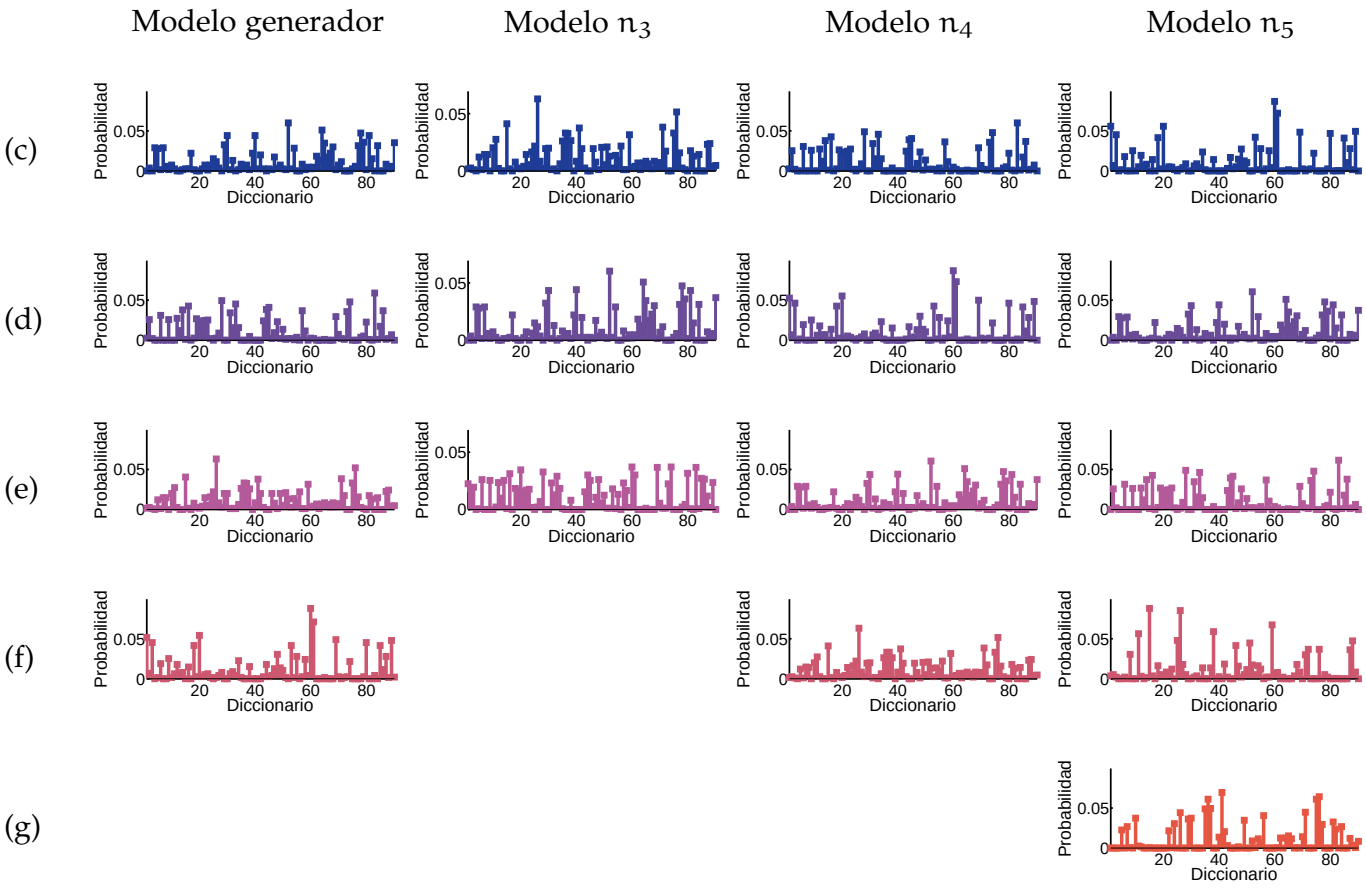


Figura 6.7: *Por columnas, se muestran las probabilidades de emisión para cada uno de los interlocutores de acuerdo al modelo. Cada fila representa las probabilidad de emisión de las palabras en el diccionario. La fila* (c) *para la primer persona, la fila* (d) *para la segunda persona y así de forma consecutiva, de acuerdo al número de personas que contempla cada modelo propuesto.*

la probabilidad de cambio entre las personas que participan en la conversación.

De la misma manera, la matriz de transición que se recupera tiene estructura diagonal, como en la primer prueba.

En la Figura 6.7 se muestran las probabilidades de emisión para cada uno de los participantes, de acuerdo a las palabras en las que se ha discretizado la conversación. Idealmente, cada interlocutor tiene asignadas con mayor probabilidad un cierto conjunto de palabras del diccionario, lo que hace que la identificación de las personas sea mucho más fácil de realizar.

Para la selección del modelo y como se explicó en el Capítulo 5, primero se realiza una exploración del conjunto de posibles

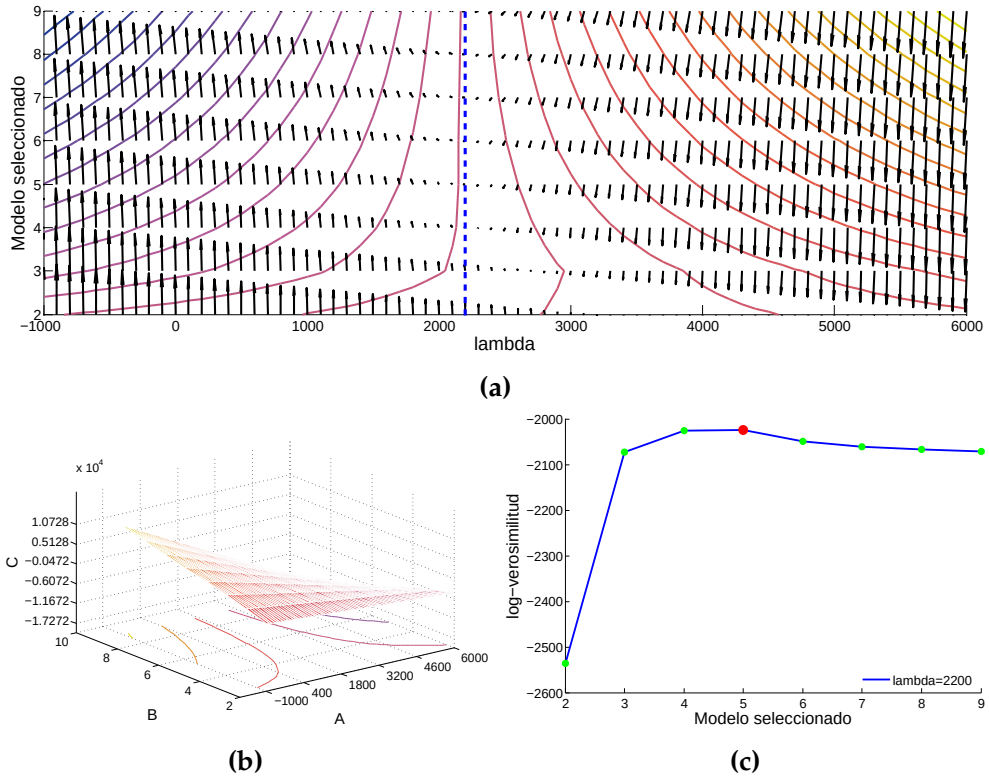


Figura 6.8: *En la Figura 6.8a, se muestra las curvas de nivel de la superficie al variar el valor de lambda para evaluar BIC, así como la dirección del gradiente en la misma. En la Figura 6.8b se muestra una perspectiva general de superficie en Figura 6.8a. En la Figura 6.8c se muestra el valor seleccionado para lambda de acuerdo al análisis de sensibilidad realizado en la superficie anterior.*

soluciones, por medio de BIC regularizado para encontrar cuál o cuáles son los modelos más probables.

En la Figura 6.8b se muestra la superficie generada al variar el valor de λ para diferentes curvas de selección BIC. Haciendo un análisis de sensibilidad se puede determinar cuál es el parámetro λ adecuado que penaliza de buena forma la log-verosimilitud. En la Figura 6.8a se muestran las curvas de nivel de la figura anterior, además de la dirección del gradiente de la misma. Así mismo, en falso color se resaltan las zonas en las que el gradiente es menor.

Por último, se muestra en Figura 6.8c la curva BIC con el valor de λ encontrado a partir del análisis de sensibilidad realizado en la Figura 6.8a. El o los modelos que tengan un mayor valor BIC serán los que se seleccionarán como posibles modelos ganadores.

En caso de que después de utilizar BIC se presente ambigüedad para determinar un modelo ganador, o ya sea para realizar un análisis más exhaustivo, se puede propone hacer una prueba de

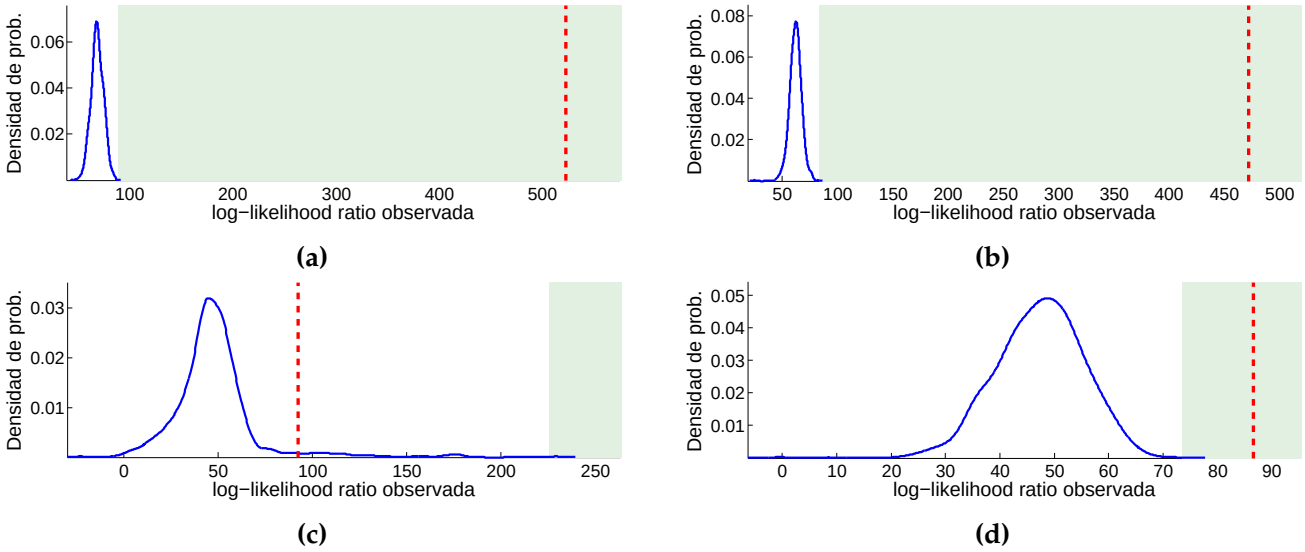


Figura 6.9: *En la Figura 6.9a se muestra la prueba de hipótesis realizada para comparar el modelo n_2 contra n_3. Como se observa, se rechaza la hipótesis de que el modelo correcto sea n_4. En la Figura 6.9b se hace la prueba del modelo n_3 contra n_4, y de la misma manera, se rechaza la hipótesis de que el modelo correcto sea n_3. Se sigue con la prueba de hipótesis del modelo n_4 contra n_5 en la Figura 6.9c, y en este caso el valor observado no cae dentro de la región de rechazo, por lo que no podemos rechazar que el modelo n_4 sea el correcto. Por último en la Figura 6.9d, se hace la prueba del modelo n_5 contra el modelo n_6, y se vuelve a rechazar la hipótesis nula.*

hipótesis para determinar cuál modelo se ajusta mejor a los datos, como ya se explicó en el Capítulo 5.

Por último, ya con el modelo seleccionado, se compara la segmentación recuperada con la segmentación original de la conversación:

Más a detalle, en la Figura 6.10 se observa en azul el orden en el que participan los interlocutores de acuerdo a la secuencia recuperada. En rojo se marcan tanto los falsos positivos como los falsos negativos, de acuerdo al ground truth. Se observa también que la mayoría de las veces, en la secuencia recuperada se encuentran algunos brincos entre las personas, pero en esencia la estructura y el orden en que hablan los interlocutores es el correcto.

6.1.3 *Secuencia 3: William Shakespeare*

Para esta secuencia se utilizaron algunos poemas del escritor William Shakespeare, alternando 6 voces diferentes en inglés. La secuencia de audio original tiene una duración de 12:01 min.

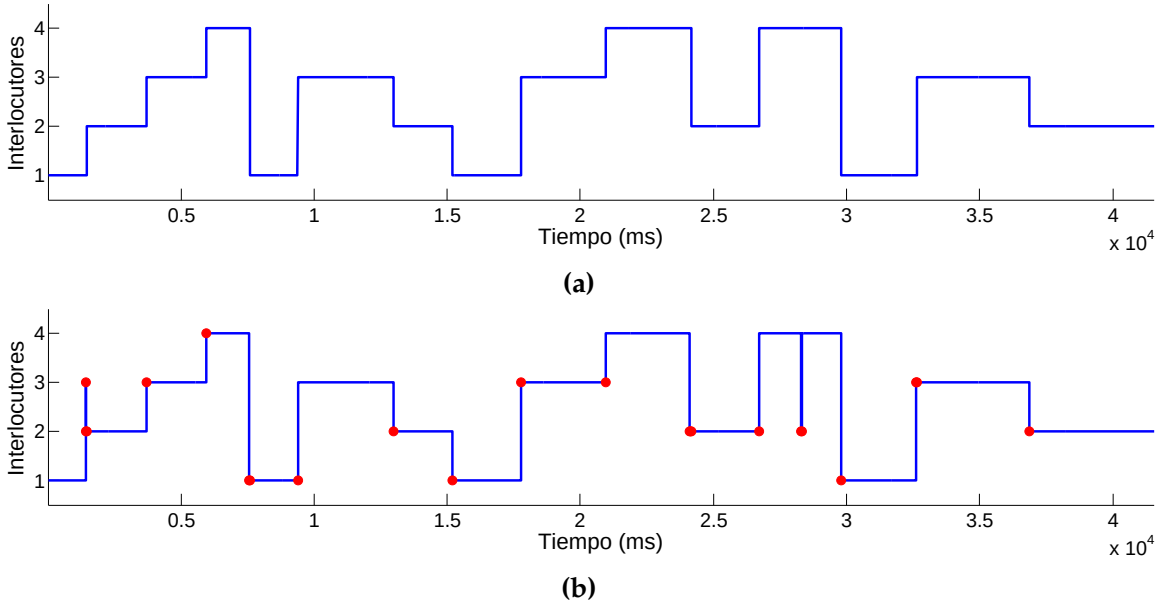


Figura 6.10: *En Figura 6.10a se muestra la secuencia original de la Subsección 6.1.2. En comparación, en Figura 6.10b se muestra la secuencia recuperada del modelo ganador y se marcan en rojo los errores cometidos respecto a la primera secuencia.*

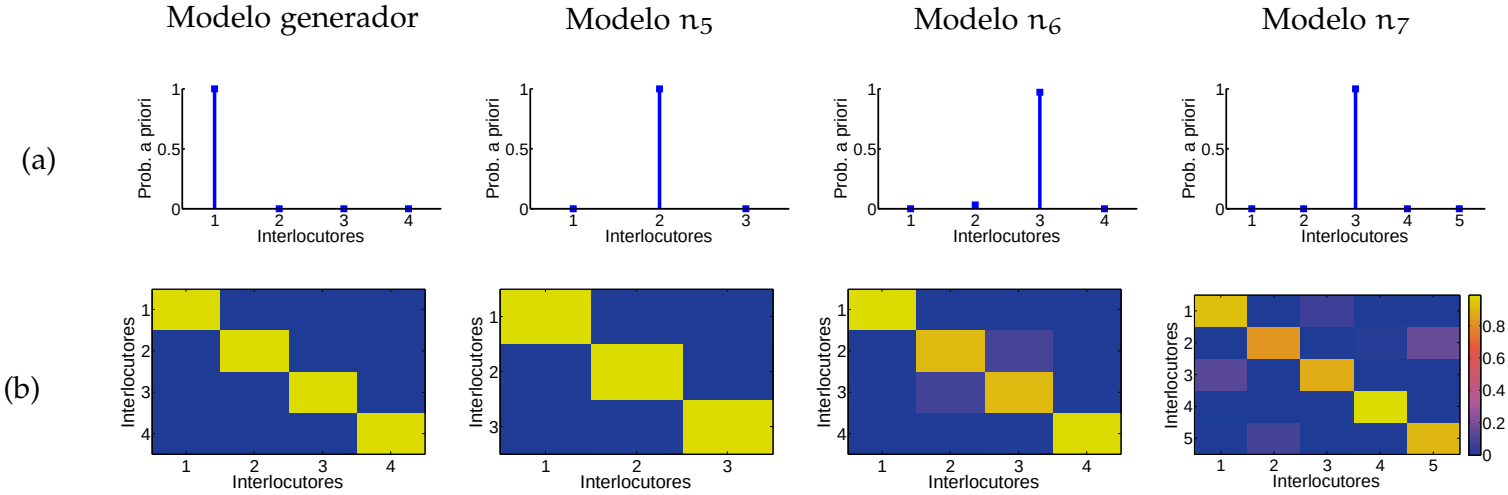


Figura 6.11: *Por columnas, se muestran los parámetros obtenidos para varios modelos propuestos. La primer columna corresponde a los parámetros verdaderos. En la primer fila* (a) *se muestran las probabilidades a priori de que cada uno de los interlocutores empiecen la conversación. En la segunda fila* (b) *se muestra en falso color la matriz de transición entre interlocutores para cada uno de los modelos mostrados.*

Para la etapa de agrupación de los vectores MFCC con k-means++ se usaron 140 centros iniciales, mientras que el banco de filtros fue el mismo que en la secuencia anterior.

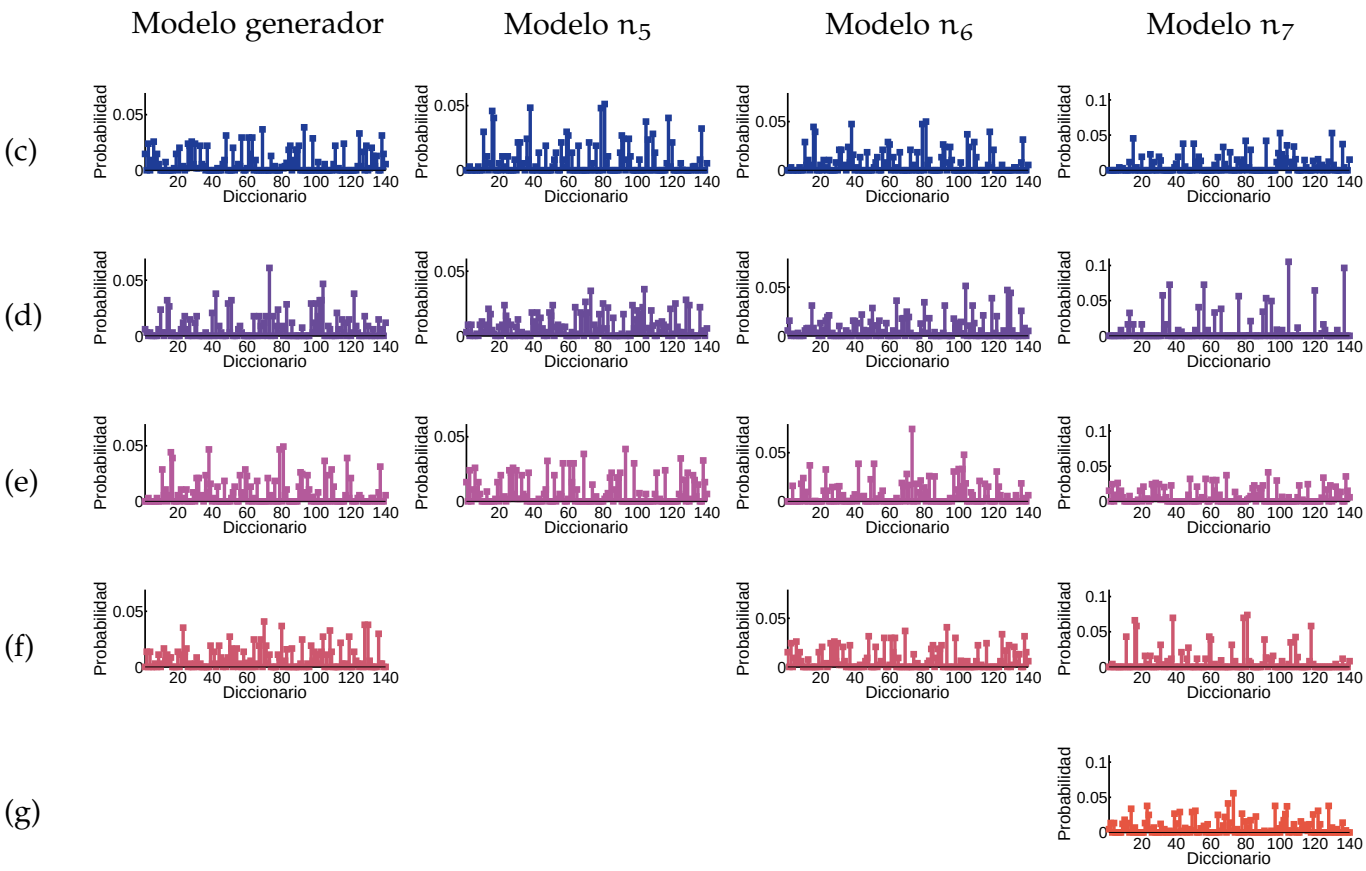


Figura 6.12: *Por columnas, se muestran las probabilidades de emisión para cada uno de los interlocutores de acuerdo al modelo. Cada fila representa las probabilidad de emisión de las palabras en el diccionario. La fila* (c) *para la primer persona, la fila* (d) *para la segunda persona y así de forma consecutiva, de acuerdo al número de personas que contempla cada modelo propuesto.*

En las Figura 6.11 y Figura 6.12 se muestran los parámetros estimados para los diferentes modelos que se propusieron. La primer columna corresponde a los parámetros verdaderos, mientras que las demás columnas son los parámetros obtenidos para diferentes modelos estimados.

En la primer fila de la Figura 6.11 se muestran las probabilidades a priori de que cada uno de los interlocutores empiecen el diálogo. En la segunda fila se muestra en falso color la matriz de transición para cada uno de los modelos. Esta matriz representa la probabilidad de cambio entre las personas que participan en la conversación.

En este caso, la matriz de transición que se recupera también tiene estructura diagonal, como en las otras prueba.

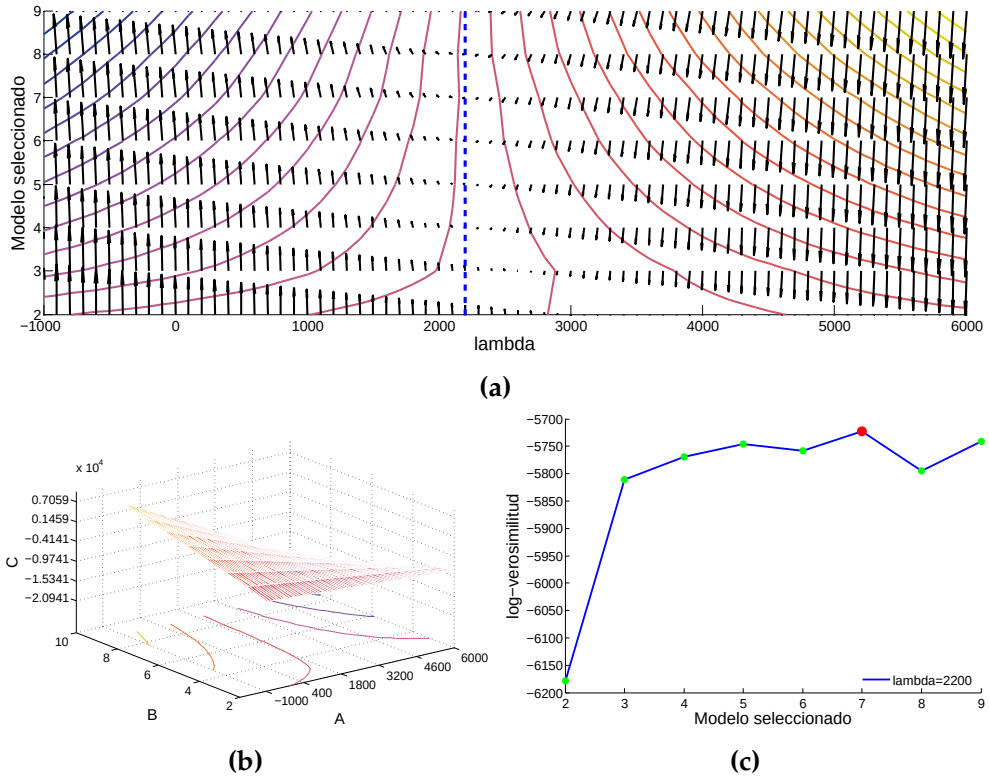


Figura 6.13: *En la Figura 6.13a, se muestra las curvas de nivel de la superficie al variar el valor de lambda para evaluar BIC, así como la dirección del gradiente en la misma. En la Figura 6.13b se muestra una perspectiva general de superficie en Figura 6.13a. En la Figura 6.13c se muestra el valor seleccionado para lambda de acuerdo al análisis de sensibilidad realizado en la superficie anterior.*

En la Figura 6.12 se muestran las probabilidades de emisión para cada uno de los participantes, de acuerdo a las palabras en las que se ha discretizado la conversación. Idealmente, cada interlocutor tiene asignadas con mayor probabilidad un cierto conjunto de palabras del diccionario, lo que hace que la identificación de las personas sea mucho más fácil de realizar.

Para la selección del modelo y siguiendo la metodología propuesta, primero se efectúa una inspección por medio de BIC regularizado para encontrar cuál o cuáles son los modelos más probables.

En la Figura 6.13b se muestra la superficie generada al variar el valor de λ para diferentes curvas de selección BIC. Haciendo un análisis de sensibilidad se puede determinar cuál es el parámetro λ adecuado que penaliza de buena forma la log-verosimilitud.

En la Figura 6.13a se muestran las curvas de nivel de la figura anterior, además de la dirección del gradiente de la misma. Así

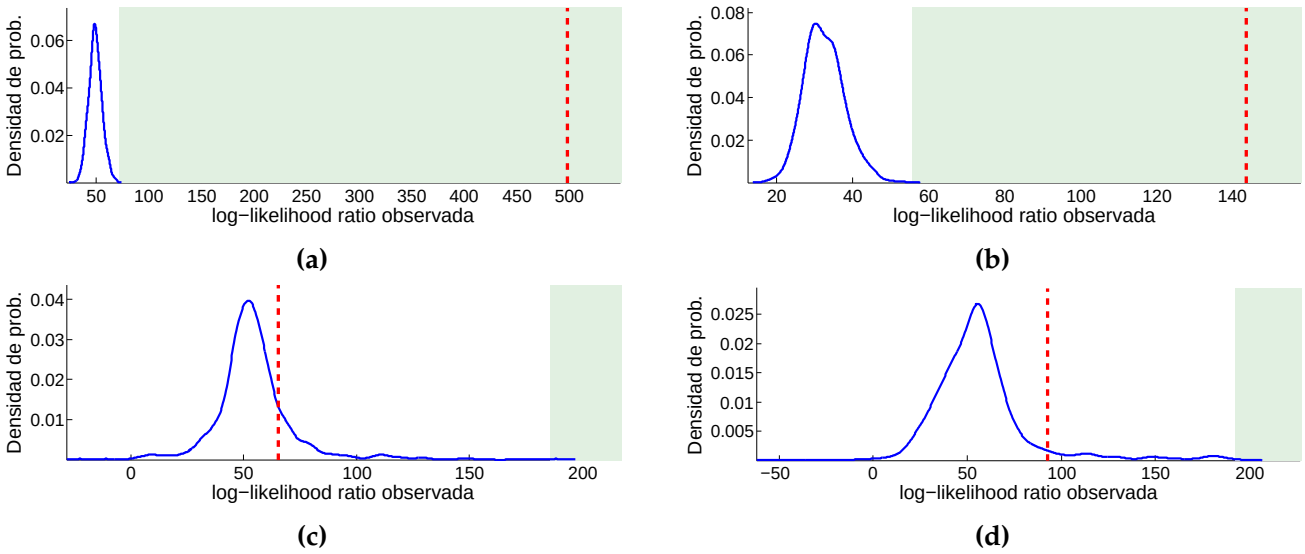


Figura 6.14: *En la Figura 6.14a se muestra la prueba de hipótesis realizada para comparar el modelo n_2 contra n_3. Como se observa, se rechaza la hipótesis de que el modelo correcto sea n_2. En la Figura 6.14b se hace la prueba del modelo n_3 contra n_4, y de la misma manera, se rechaza la hipótesis de que el modelo correcto sea n_3. Se sigue con la prueba de hipótesis del modelo n_4 contra n_5 en la Figura 6.14c, y en este caso el valor observado no cae dentro de la región de rechazo, por lo que no podemos rechazar que el modelo n_4 sea el correcto. Por último en la Figura 6.14d, se hace la prueba del modelo n_5 contra el modelo n_6, y se vuelve a rechazar la hipótesis nula.*

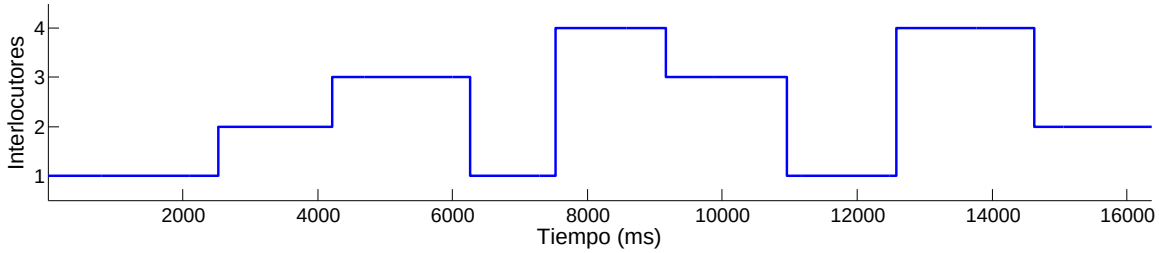
mismo, en falso color se resaltan las zonas en las que el gradiente es menor.

Por último, se muestra en Figura 6.13c la curva BIC con el valor de λ encontrado a partir del análisis de sensibilidad realizado en la Figura 6.13a. El o los modelos que tengan un mayor valor BIC serán los que se seleccionarán como posibles soluciones.

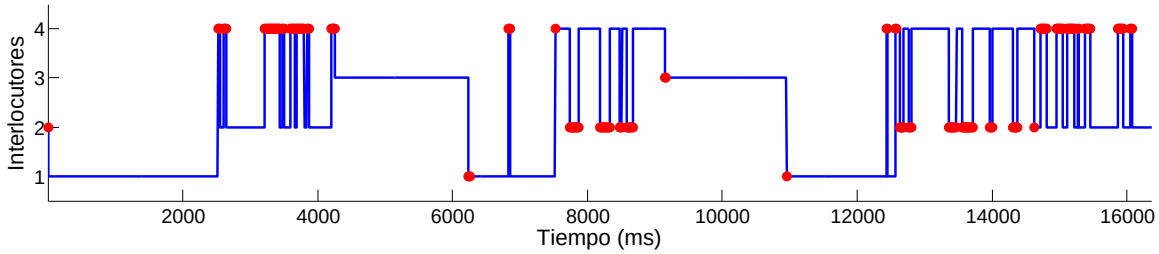
En caso de que después de utilizar BIC se presente ambigüedad para determinar un modelo ganador, o ya sea para realizar un análisis más exhaustivo, se puede propone hacer una prueba de hipótesis para determinar cuál modelo se ajusta mejor a los datos.

Por último, ya con el modelo seleccionado, se compara la segmentación recuperada con la segmentación original de la conversación:

Más a detalle, en la Figura 6.15 se observa en azul el orden en el que participan los interlocutores de acuerdo a la secuencia recuperada. En rojo se marcan tanto los falsos positivos como los falsos negativos, de acuerdo al ground truth. Se observa también que la mayoría de las veces, en la secuencia recuperada se encuentran



(a)



(b)

Figura 6.15: *En Figura 6.15a se muestra la secuencia original de la Subsección 6.1.3. En comparación, en Figura 6.15b se muestra la secuencia recuperada del modelo ganador y se marcan en rojo los errores cometidos respecto a la primera secuencia.*

algunos brincos entre personas, pero en esencia la estructura y el orden en que hablan los interlocutores es el correcto.

6.1.4 *Secuencia 4: Manuel Acuña*

Para la cuarta secuencia se utilizaron algunos poemas del escritor Manuel Acuña, alternando 6 voces diferentes en español. La secuencia de audio original tiene una duración de 10:41 min.

Para la etapa de agrupación de los vectores MFCC con k-means++ se usaron 140 centros iniciales, mientras que el banco de filtros fue el mismo que en la secuencia anterior.

En las Figura 6.16 y Figura 6.17 se muestran los parámetros estimados para los diferentes modelos que se propusieron. La primer columna corresponde a los parámetros verdaderos, mientras que las demás columnas son los parámetros obtenidos para diferentes modelos estimados.

En la primer fila de la Figura 6.16 se muestran las probabilidades a priori de que cada uno de los interlocutores empiecen el diálogo. En la segunda fila se muestra en falso color la matriz de transición para cada uno de los modelos. Esta matriz representa

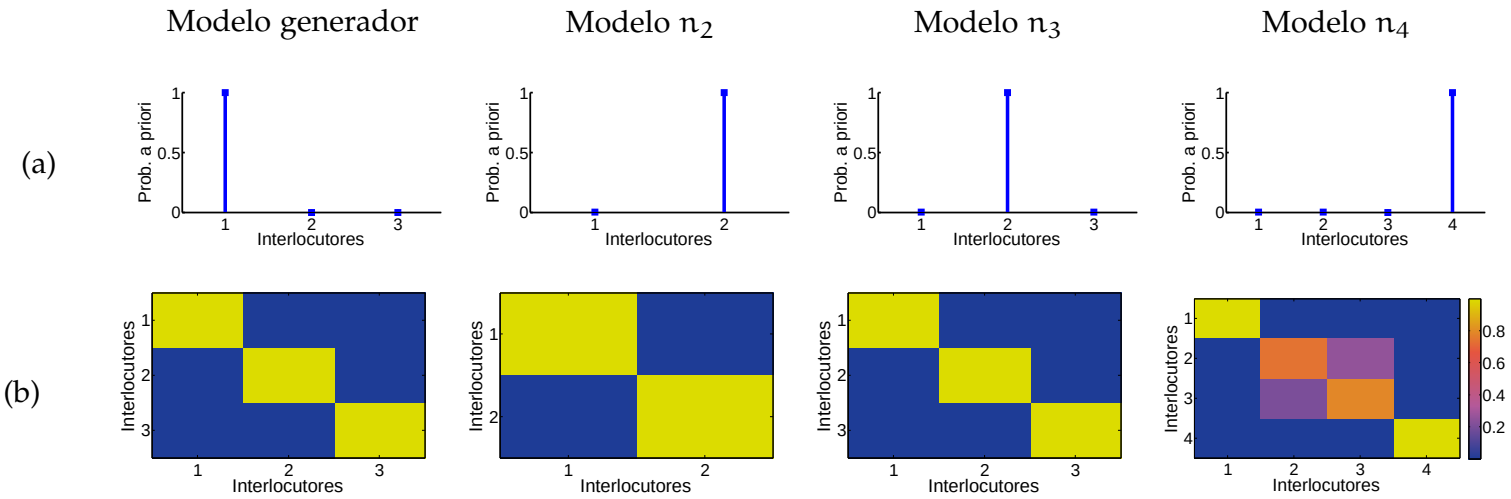


Figura 6.16: *Por columnas, se muestran los parámetros obtenidos para varios modelos propuestos. La primer columna corresponde a los parámetros verdaderos. En la primer fila* (a) *se muestran las probabilidades a priori de que cada uno de los interlocutores empiecen la conversación. En la segunda fila* (b) *se muestra en falso color la matriz de transición entre interlocutores para cada uno de los modelos mostrados.*

la probabilidad de cambio entre las personas que participan en la conversación.

De la misma manera, la matriz de transición que se recupera tiene estructura diagonal, como en la primer prueba.

En la Figura 6.17 se muestran las probabilidades de emisión para cada uno de los participantes, de acuerdo a las palabras en las que se ha discretizado la conversación. Idealmente, cada interlocutor tiene asignadas con mayor probabilidad un cierto conjunto de palabras del diccionario, lo que hace que la identificación de las personas sea mucho más fácil de realizar.

Para la selección del modelo y como se explicó en el Capítulo 5, primero se realiza una exploración del conjunto de posibles soluciones, por medio de BIC regularizado para encontrar cuál o cuáles son los modelos más probables.

En la Figura 6.18b se muestra la superficie generada al variar el valor de λ para diferentes curvas de selección BIC. Haciendo un análisis de sensibilidad se puede determinar cuál es el parámetro λ adecuado que penaliza de buena forma la log-verosimilitud.

En la Figura 6.18a se muestran las curvas de nivel de la figura anterior, además de la dirección del gradiente de la misma. Así

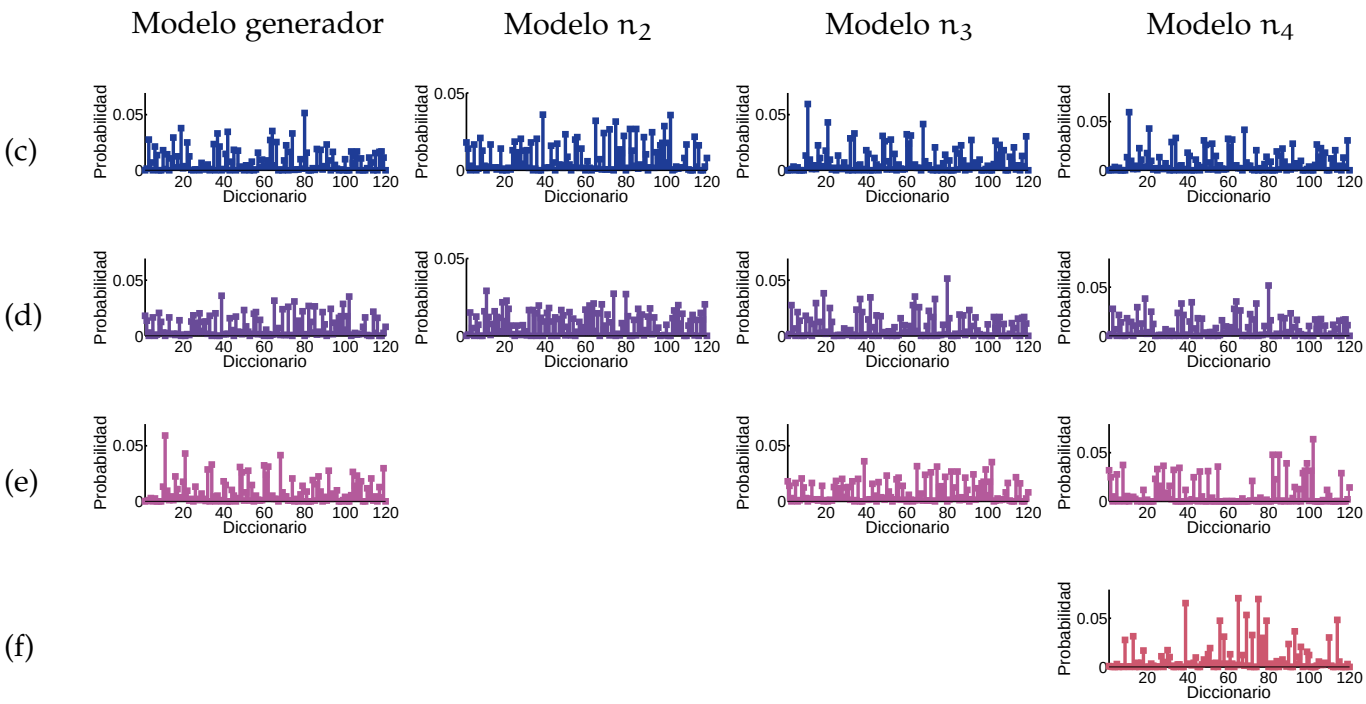


Figura 6.17: *Por columnas, se muestran las probabilidades de emisión para cada uno de los interlocutores de acuerdo al modelo. Cada fila representa las probabilidad de emisión de las palabras en el diccionario. La fila* (c) *para la primer persona, la fila* (d) *para la segunda persona y así de forma consecutiva, de acuerdo al número de personas que contempla cada modelo propuesto.*

mismo, en falso color se resaltan las zonas en las que el gradiente es menor.

Por último, se muestra en Figura 6.18c la curva BIC con el valor de λ encontrado a partir del análisis de sensibilidad realizado en la Figura 6.18a. El o los modelos que tengan un mayor valor BIC serán los que se seleccionarán como posibles modelos ganadores.

En caso de que después de utilizar BIC se presente ambigüedad para determinar un modelo ganador, o ya sea para realizar un análisis más exhaustivo, se puede propone hacer una prueba de hipótesis para determinar cuál modelo se ajusta mejor a los datos.

Por último, ya con el modelo seleccionado, se compara la segmentación recuperada con la secuencia o la segmentación originalde la conversación:

Más a detalle, en la Figura 6.20 se observa en azul el orden en el que participan los interlocutores de acuerdo a la secuencia recuperada. En rojo se marcan tanto los falsos positivos como los falsos negativos, de acuerdo al ground truth. Se observa también que la

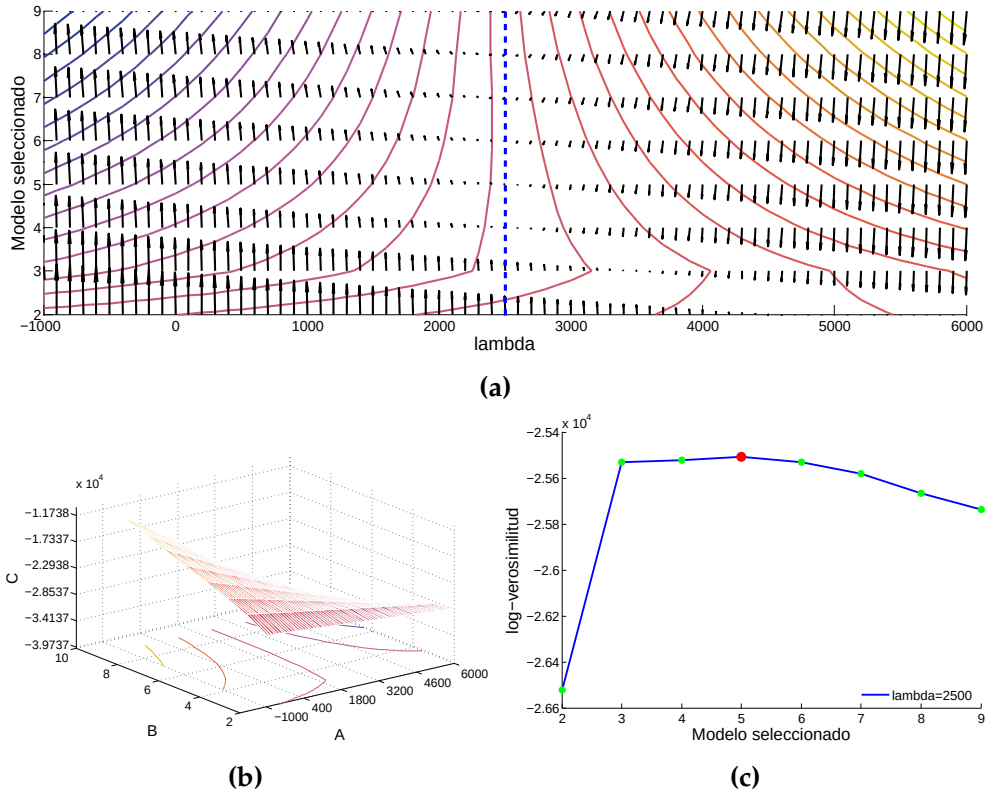


Figura 6.18: *En la Figura 6.18a, se muestra las curvas de nivel de la superficie al variar el valor de lambda para evaluar BIC, así como la dirección del gradiente en la misma. En la Figura 6.18b se muestra una perspectiva general de superficie en Figura 6.18a. En la Figura 6.18c se muestra el valor seleccionado para lambda de acuerdo al análisis de sensibilidad realizado en la superficie anterior.*

mayoría de las veces, en la secuencia recuperada se encuentran algunos brincos entre personas, pero en esencia la estructura y el orden en que hablan los interlocutores es el correcto.

6.1.5 *Secuencia 5: Calderón de la Barca*

En esta quinta secuencia se trabajó con varios fragmentos del poema 'La vida es sueño' del escritor Pedro Calderón de la Barca, usándose 3 diferentes voces en español. La secuencia de audio original es de 11:18 min.

Para la etapa de agrupación de los vectores MFCC con k-means++ se usaron 160 centros iniciales, mientras que el banco de filtros fue el usado en las secuencias anteriores.

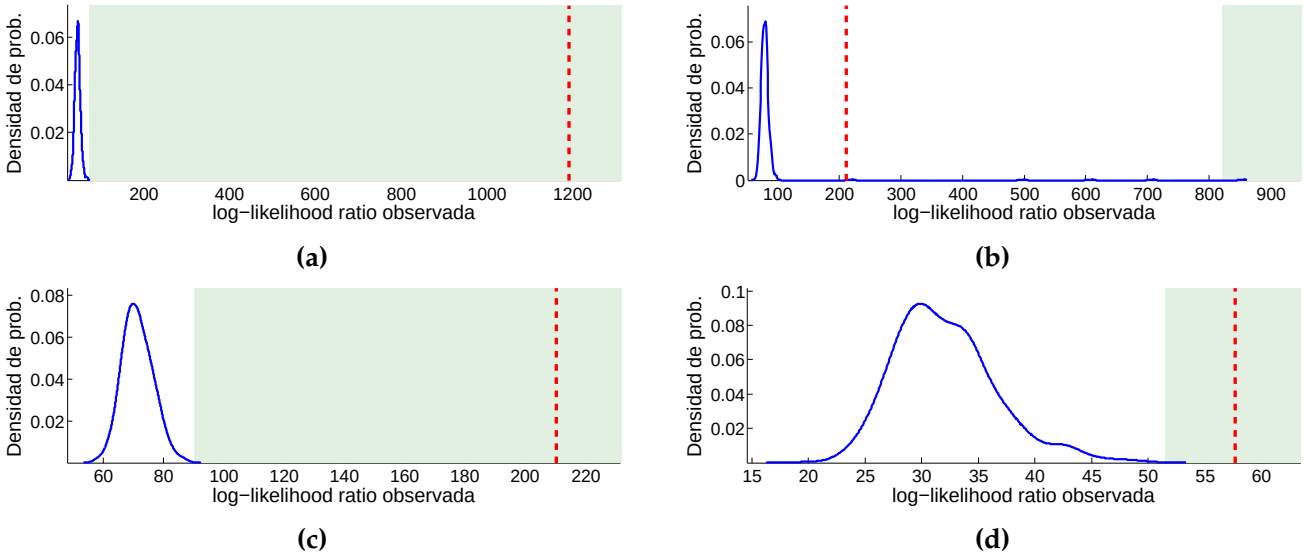


Figura 6.19: *En la Figura 6.19a se muestra la prueba de hipótesis realizada para comparar el modelo n_2 contra n_3. Como se observa, se rechaza la hipótesis de que el modelo correcto sea n_2. En la Figura 6.19b se hace la prueba del modelo n_3 contra n_4, y de la misma manera, se rechaza la hipótesis de que el modelo correcto sea n_3. Se sigue con la prueba de hipótesis del modelo n_4 contra n_5 en la Figura 6.19c, y en este caso el valor observado no cae dentro de la región de rechazo, por lo que no podemos rechazar que el modelo n_4 sea el correcto. Por último en la Figura 6.19d, se hace la prueba del modelo n_5 contra el modelo n_6, y se vuelve a rechazar la hipótesis nula.*

En las Figura 6.21 y Figura 6.22 se muestran los parámetros estimados para los diferentes modelos que se propusieron. La primer columna corresponde a los parámetros verdaderos, mientras que las demás columnas son los parámetros obtenidos para diferentes modelos estimados.

En la primer fila de la Figura 6.21 se muestran las probabilidades a priori de que cada uno de los interlocutores empiecen el diálogo. En la segunda fila se muestra en falso color la matriz de transición para cada uno de los modelos. Esta matriz representa la probabilidad de cambio entre las personas que participan en la conversación.

En este caso, la matriz de transición que se recupera también tiene estructura diagonal, como en las otras prueba.

En la Figura 6.22 se muestran las probabilidades de emisión para cada uno de los participantes, de acuerdo a las palabras en las que se ha discretizado la conversación. Idealmente, cada interlocutor tiene asignadas con mayor probabilidad un cierto conjunto de palabras del diccionario, lo que hace que la identificación de las personas sea mucho más fácil de realizar.

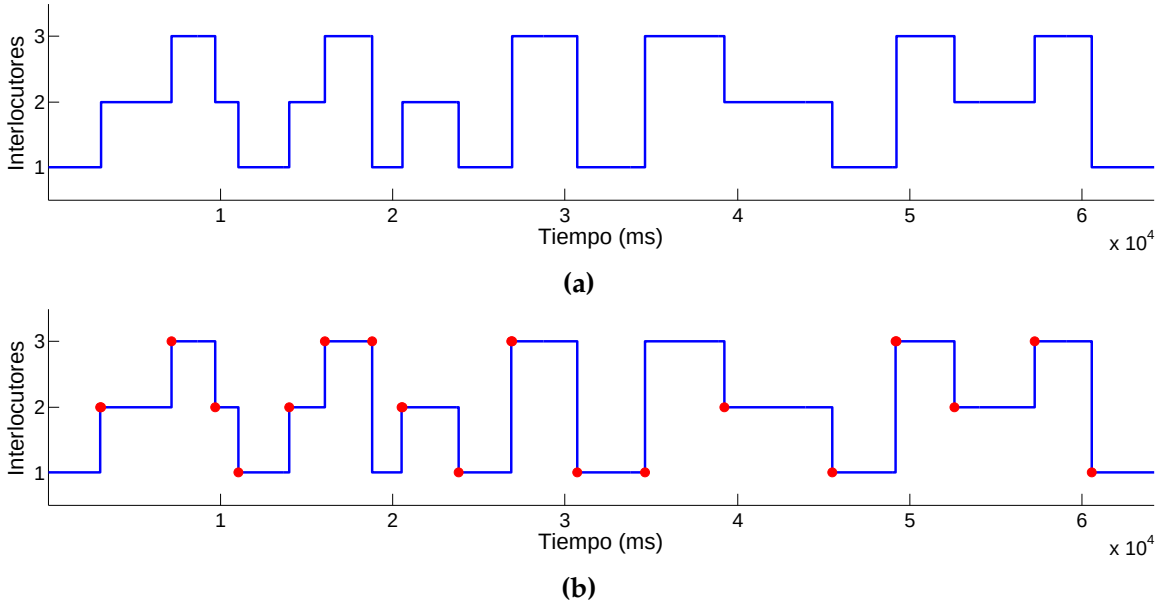


Figura 6.20: *En Figura 6.20a se muestra la secuencia original de la Subsección 6.1.4. En comparación, en Figura 6.20b se muestra la secuencia recuperada del modelo ganador y se marcan en rojo los errores cometidos respecto a la primera secuencia.*

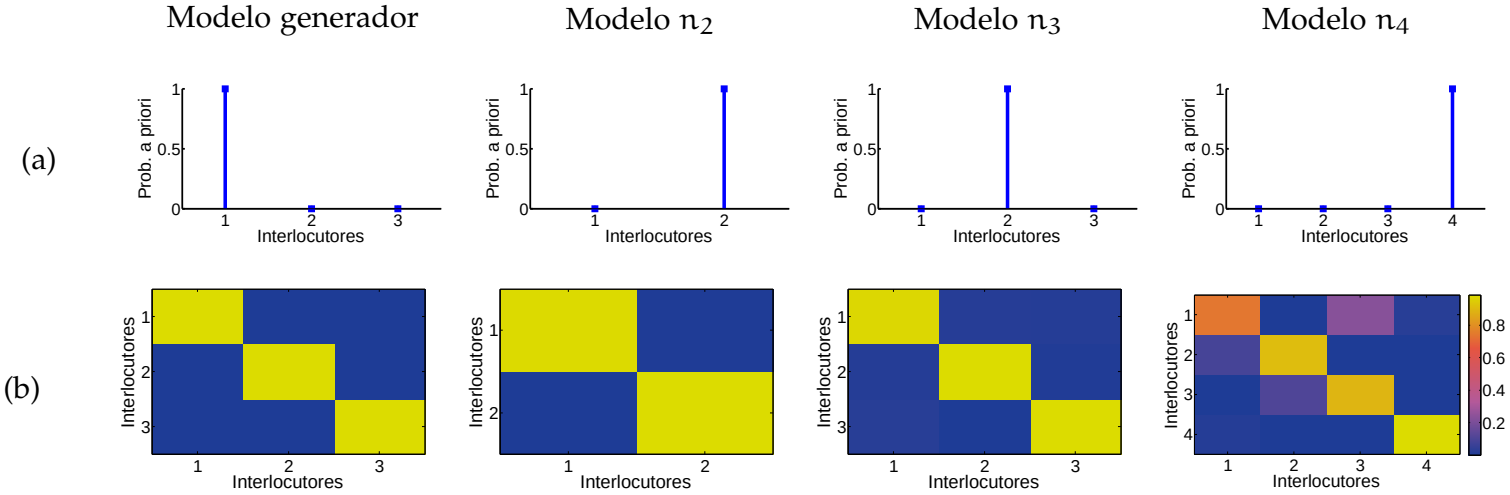


Figura 6.21: *Por columnas, se muestran los parámetros obtenidos para varios modelos propuestos. La primer columna corresponde a los parámetros verdaderos. En la primer fila* (a) *se muestran las probabilidades a priori de que cada uno de los interlocutores empiecen la conversación. En la segunda fila* (b) *se muestra en falso color la matriz de transición entre interlocutores para cada uno de los modelos mostrados.*

Para la selección del modelo y siguiendo la metodología propuesta, primero se efectúa una inspección por medio de BIC regu-

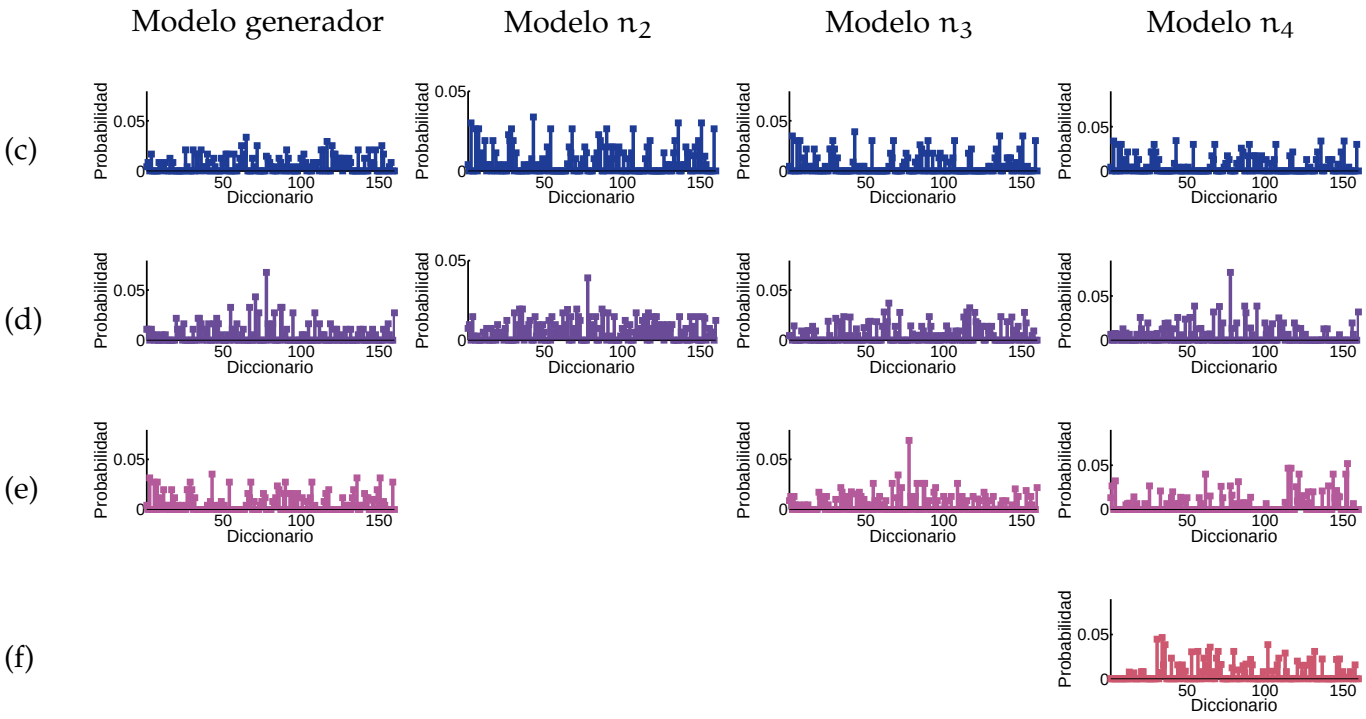


Figura 6.22: *Por columnas, se muestran las probabilidades de emisión para cada uno de los interlocutores de acuerdo al modelo. Cada fila representa las probabilidad de emisión de las palabras en el diccionario. La fila* (c) *para la primer persona, la fila* (d) *para la segunda persona y así de forma consecutiva, de acuerdo al número de personas que contempla cada modelo propuesto.*

larizado para encontrar cuál o cuáles son los modelos más probables.

En la Figura 6.23b se muestra la superficie generada al variar el valor de λ para diferentes curvas de selección BIC. Haciendo un análisis de sensibilidad se puede determinar cuál es el parámetro λ adecuado que penaliza de buena forma la log-verosimilitud.

En la Figura 6.23a se muestran las curvas de nivel de la figura anterior, además de la dirección del gradiente de la misma. Así mismo, en falso color se resaltan las zonas en las que el gradiente es menor.

Por último, se muestra en Figura 6.23c la curva BIC con el valor de λ encontrado a partir del análisis de sensibilidad realizado en la Figura 6.23a. El o los modelos que tengan un mayor valor BIC serán los que se seleccionarán como posibles soluciones.

En caso de que después de utilizar BIC se presente ambigüedad para determinar un modelo ganador, o ya sea para realizar un

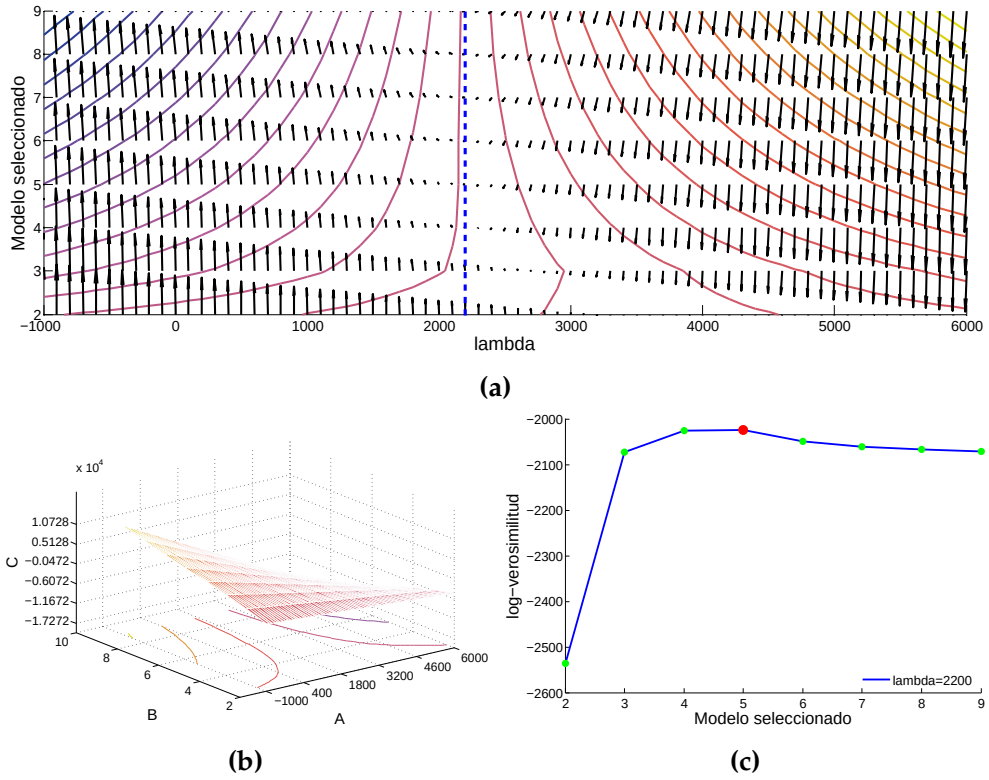


Figura 6.23: *En la Figura 6.23a, se muestra las curvas de nivel de la superficie al variar el valor de lambda para evaluar BIC, así como la dirección del gradiente en la misma. En la Figura 6.23b se muestra una perspectiva general de superficie en Figura 6.23a. En la Figura 6.23c se muestra el valor seleccionado para lambda de acuerdo al análisis de sensibilidad realizado en la superficie anterior.*

análisis más exhaustivo, se puede propone hacer una prueba de hipótesis para determinar cuál modelo se ajusta mejor a los datos.

A diferencia de la primer etapa, en la que se usa BIC como criterio para seleccionar el mejor modelo de un conjunto no definido de modelos con diferentes parámetros, la intención de hacer pruebas de hipótesis es determinar en un pequeño conjunto de probables modelos, cuál es mejor, y qué tan bueno es un modelo respecto a otro.

Al plantear la prueba de hipótesis se harán una gran cantidad de simulaciones para ver qué tan bien se ajusta cada modelo a los datos originales, por lo que este proceso es computacionalmente intensivo y sólo se recomienda hacerlo para evitar la ambigüedad entre un par de modelos.

Por último, ya con el modelo seleccionado, se compara la segmentación recuperada con la secuencia o la segmentación originalde la conversación:

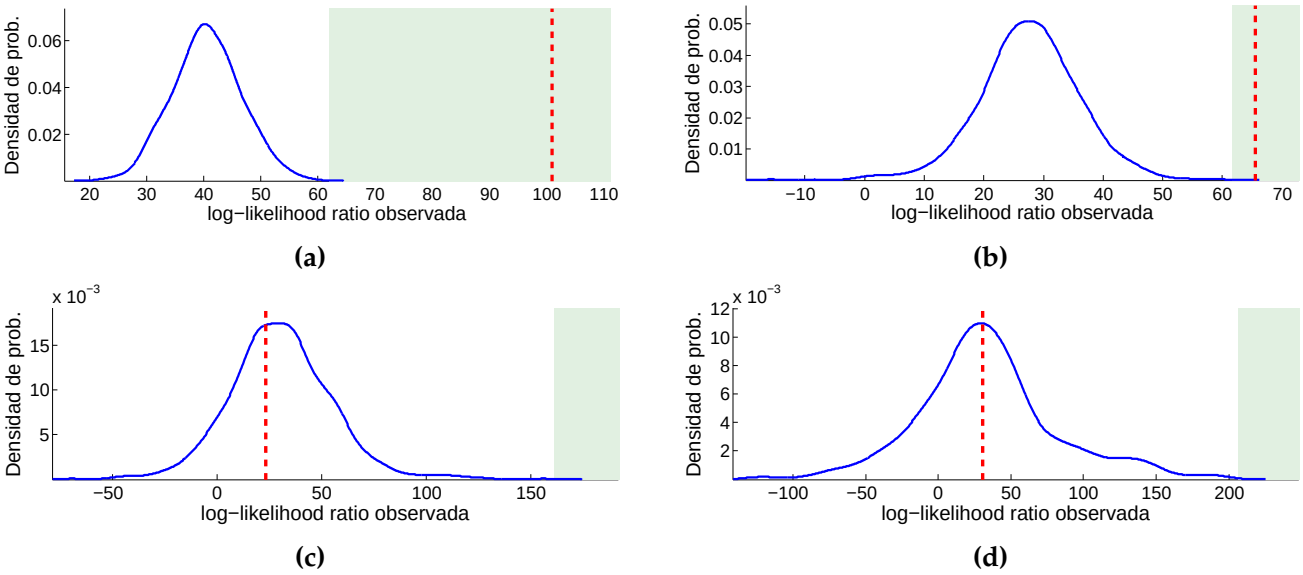


Figura 6.24: *En la Figura 6.24a se muestra la prueba de hipótesis realizada para comparar el modelo n_2 contra n_3. Como se observa, se rechaza la hipótesis de que el modelo correcto sea n_2. En la Figura 6.24b se hace la prueba del modelo n_3 contra n_4, y se vuelve a rechazar la hipótesis nula, aunque queda muy cerca del umbral de decisión. Realizando la prueba de n_4 contra n_5 en la Figura 6.24c en este caso el valor observado no cae dentro de la región de rechazo, por lo que no podemos rechazar que el modelo n_4 sea el correcto. Por último en la Figura 6.24d, se hace la prueba del modelo n_5 contra el modelo n_6, y vuelve a suceder lo mismo que en el caso anterior. Como para las últimas dos pruebas no se pudo rechazar la hipótesis nula, se preferirá siempre la que involucre al modelo más sencillo. Cabe recalcar también que si se hicieran las pruebas de hipótesis con una significancia menor, entonces desde la segunda prueba se tendría al modelo ganador,*

Más a detalle, en la Figura 6.25 se observa en azul el orden en el que participan los interlocutores de acuerdo a la secuencia recuperada. En rojo se marcan tanto los falsos positivos como los falsos negativos, de acuerdo al ground truth. Hay que notar que cuando el número de estados para un modelo no es el correcto, entonces inminentemente el número de errores en la secuencia obtenida será mayor, pues al menos todas las intervenciones de un hablante no podrán ser emparejadas o serán asignadas a alguien más.

Se observa también que la mayoría de las veces, en la secuencia recuperada se encuentran algunos brincos entre personas, pero en esencia la estructura y el orden en que hablan los interlocutores es el correcto.

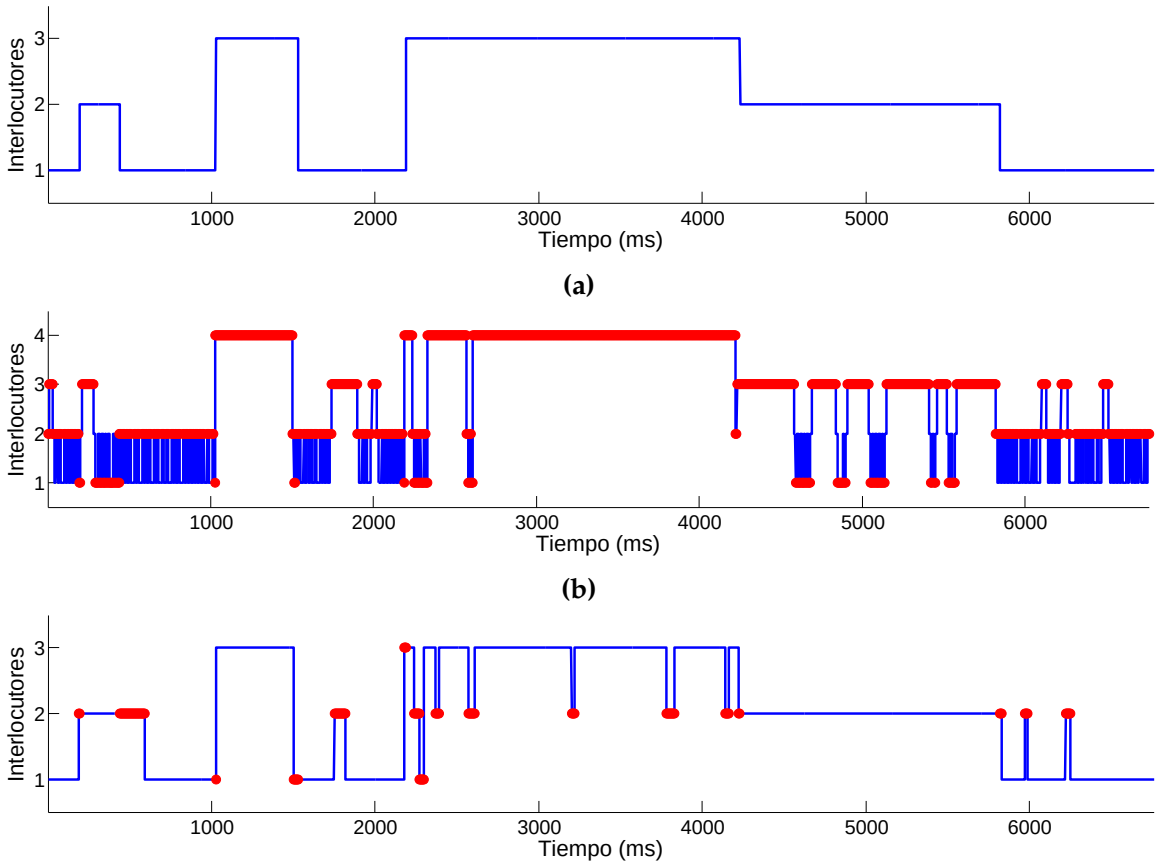


Figura 6.25: *En Figura 6.25a se muestra la secuencia original de la Subsección 6.1.5. En comparación, en Figura 6.25b se muestra la secuencia recuperada del modelo ganador y se marcan en rojo los errores cometidos respecto a la primera secuencia. Por último, se muestra en Figura 6.25c la el modelo que debió haber ganado en caso de que se usara una significacia más baja para las pruebas.*

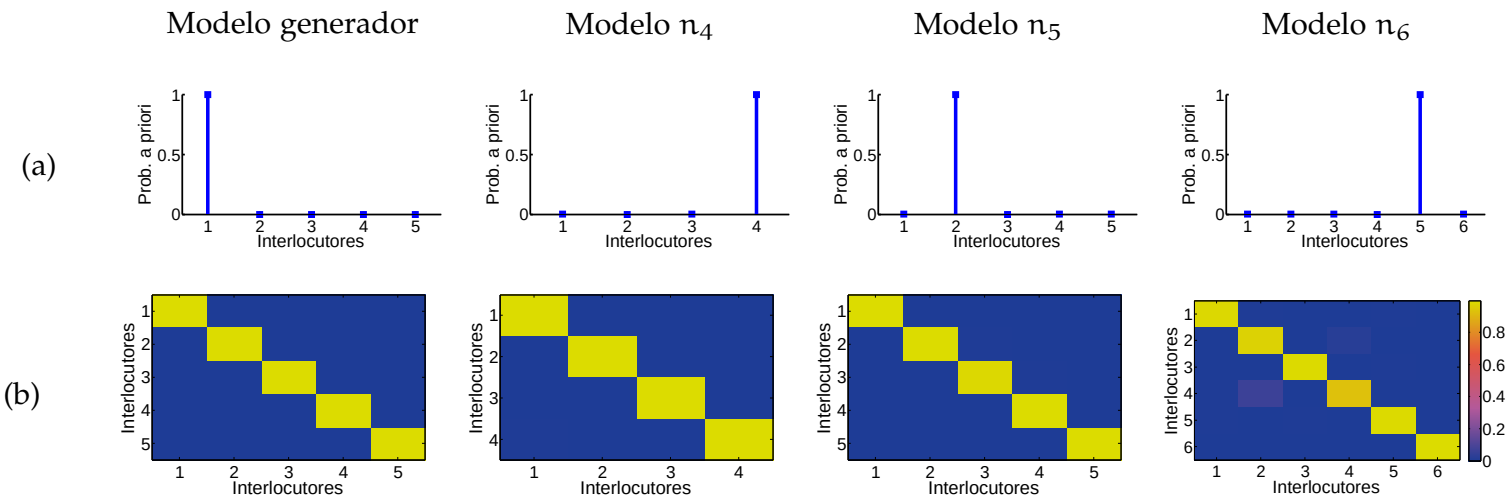


Figura 6.26: *Por columnas, se muestran los parámetros obtenidos para varios modelos propuestos. La primer columna corresponde a los parámetros verdaderos. En la primer fila* (a) *se muestran las probabilidades a priori de que cada uno de los interlocutores empiecen la conversación. En la segunda fila* (b) *se muestra en falso color la matriz de transición entre interlocutores para cada uno de los modelos mostrados.*

6.1.6 *Secuencia 6: Andrew Lloyd Webber*

Para la última secuencia de prueba se utilizaron fragmentos de la obra musical 'Cats', escrita por Andrew Lloyd Webber, usando 5 voces distintas en inglés. La secuencia de audio original tiene una duración de 6:48 min.

Para la etapa de agrupación de los vectores MFCC con k-means++ se usaron 90 centros iniciales, usando el mismo banco de filtros.

En las Figura 6.26 y Figura 6.27 se muestran los parámetros estimados para los diferentes modelos que se propusieron. La primer columna corresponde a los parámetros verdaderos, mientras que las demás columnas son los parámetros obtenidos para diferentes modelos estimados.

En la primer fila de la Figura 6.26 se muestran las probabilidades a priori de que cada uno de los interlocutores empiecen el diálogo. En la segunda fila se muestra en falso color la matriz de transición para cada uno de los modelos. Esta matriz representa la probabilidad de cambio entre las personas que participan en la conversación.

De la misma manera, la matriz de transición que se recupera tiene estructura diagonal, como en las demás pruebas.

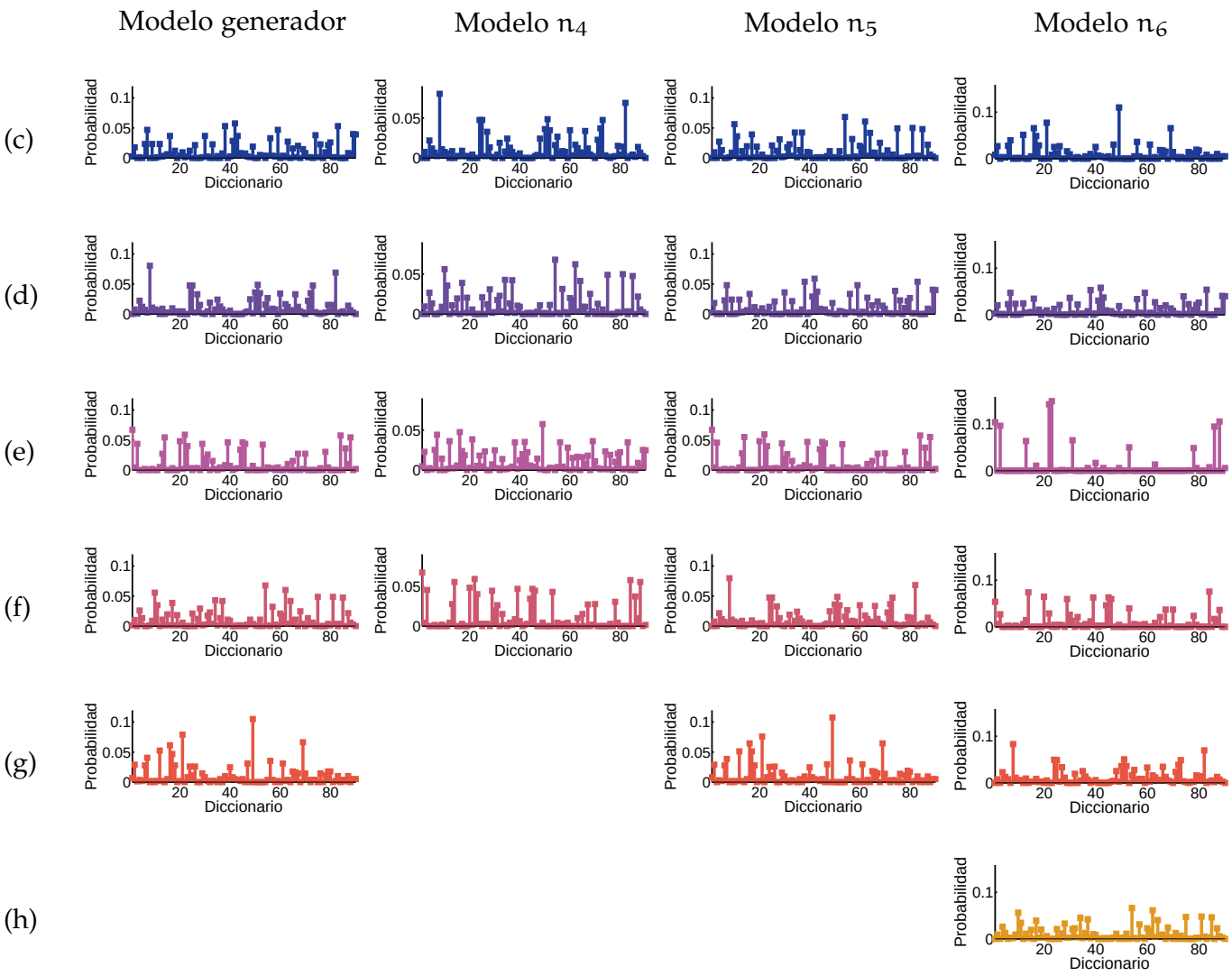


Figura 6.27: *Por columnas, se muestran las probabilidades de emisión para cada uno de los interlocutores de acuerdo al modelo. Cada fila representa las probabilidad de emisión de las palabras en el diccionario. La fila* (c) *para la primer persona, la fila* (d) *para la segunda persona y así de forma consecutiva, de acuerdo al número de personas que contempla cada modelo propuesto.*

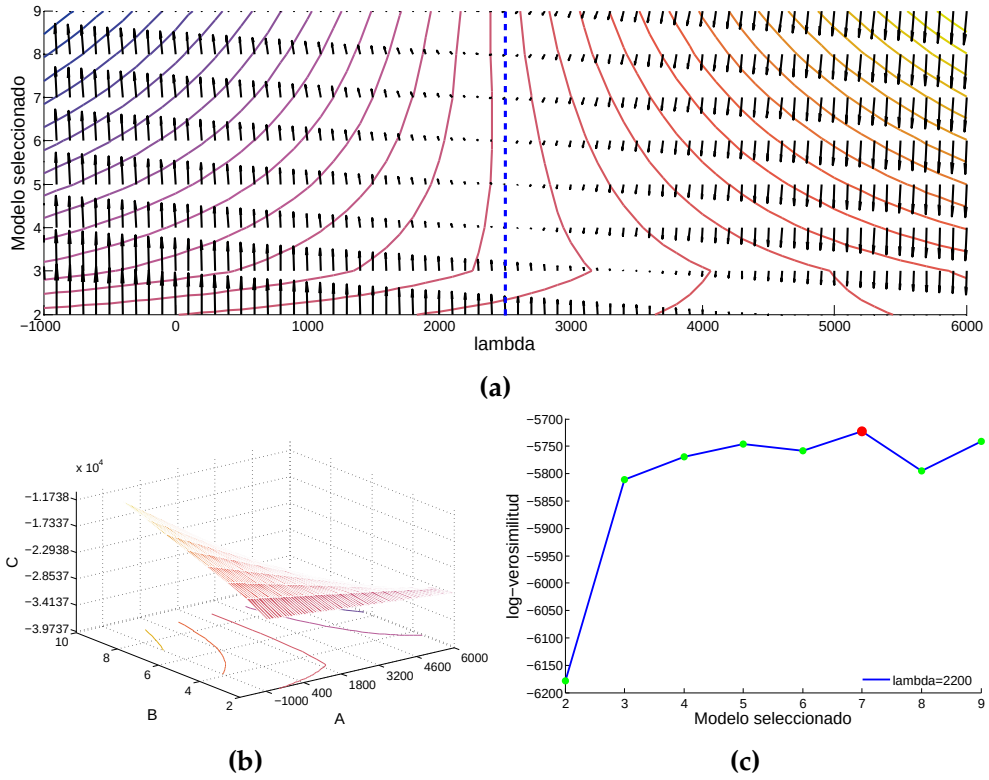


Figura 6.28: *En la Figura 6.28a, se muestra las curvas de nivel de la superficie al variar el valor de lambda para evaluar BIC, así como la dirección del gradiente en la misma. En la Figura 6.28b se muestra una perspectiva general de superficie en Figura 6.28a. En la Figura 6.28c se muestra el valor seleccionado para lambda de acuerdo al análisis de sensibilidad realizado en la superficie anterior.*

En la Figura 6.27 se muestran las probabilidades de emisión para cada uno de los participantes, de acuerdo a las palabras en las que se ha discretizado la conversación. Idealmente, cada interlocutor tiene asignadas con mayor probabilidad un cierto conjunto de palabras del diccionario, lo que hace que la identificación de las personas sea mucho más fácil de realizar.

Para la selección del modelo y como se explicó en el Capítulo 5, primero se realiza una exploración del conjunto de posibles soluciones, por medio de BIC regularizado para encontrar cuál o cuáles son los modelos más probables.

En la Figura 6.28b se muestra la superficie generada al variar el valor de λ para diferentes curvas de selección BIC. Haciendo un análisis de sensibilidad se puede determinar cuál es el parámetro λ adecuado que penaliza de buena forma la log-verosimilitud.

En la Figura 6.28a se muestran las curvas de nivel de la figura anterior, además de la dirección del gradiente de la misma. Así

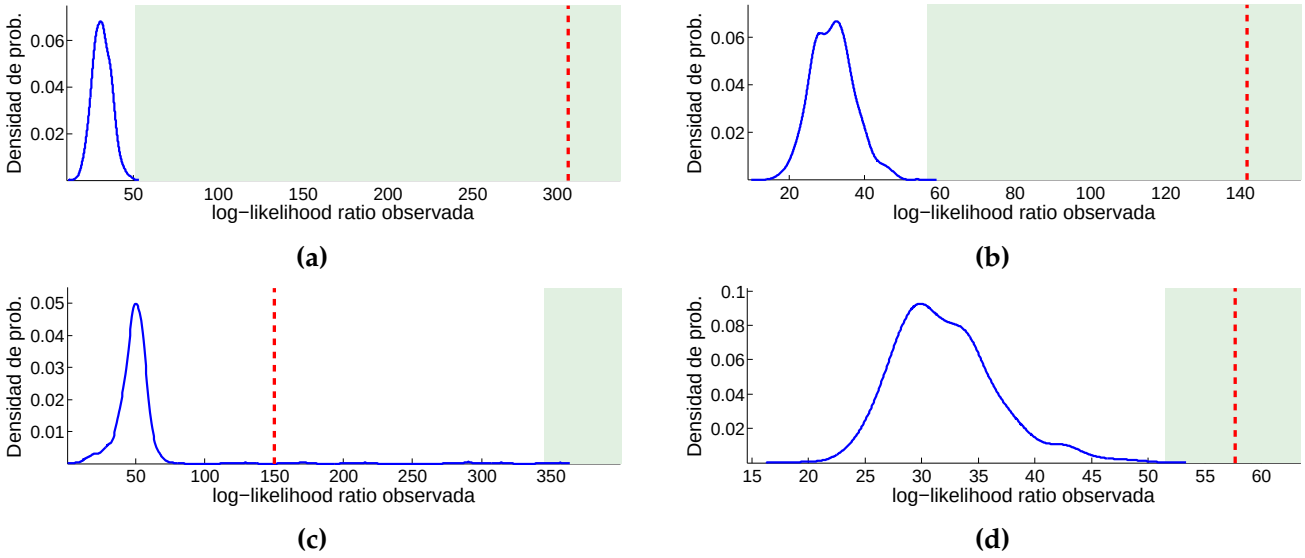


Figura 6.29: *En la Figura 6.29a se muestra la prueba de hipótesis realizada para comparar el modelo n_3 contra n_4. Como se observa, se rechaza la hipótesis de que el modelo correcto sea n_3. En la Figura 6.29b se hace la prueba del modelo n_4 contra n_5, y de la misma manera, se rechaza la hipótesis de que el modelo correcto sea n_4. Se sigue con la prueba de hipótesis del modelo n_5 contra n_6 en la Figura 6.29c, y en este caso el valor observado no cae dentro de la región de rechazo, por lo que no podemos rechazar que el modelo n_5 sea el correcto. Por último en la Figura 6.29d, se hace la prueba del modelo n_6 contra el modelo n_7, y se vuelve a rechazar la hipótesis nula.*

mismo, en falso color se resaltan las zonas en las que el gradiente es menor.

Por último, se muestra en Figura 6.28c la curva BIC con el valor de λ encontrado a partir del análisis de sensibilidad realizado en la Figura 6.28a. El o los modelos que tengan un mayor valor BIC serán los que se seleccionarán como posibles modelos ganadores.

En caso de que después de utilizar BIC se presente ambigüedad para determinar un modelo ganador, o ya sea para realizar un análisis más exhaustivo, se puede propone hacer una prueba de hipótesis para determinar cuál modelo se ajusta mejor a los datos, como ya se explicó en el Capítulo 5.

Por último, ya con el modelo seleccionado, se compara la segmentación recuperada con la secuencia o la segmentación originalde la conversación:

Más a detalle, en la Figura 6.30 se observa en azul el orden en el que participan los interlocutores de acuerdo a la secuencia recuperada. En rojo se marcan tanto los falsos positivos como los falsos negativos, de acuerdo al ground truth.

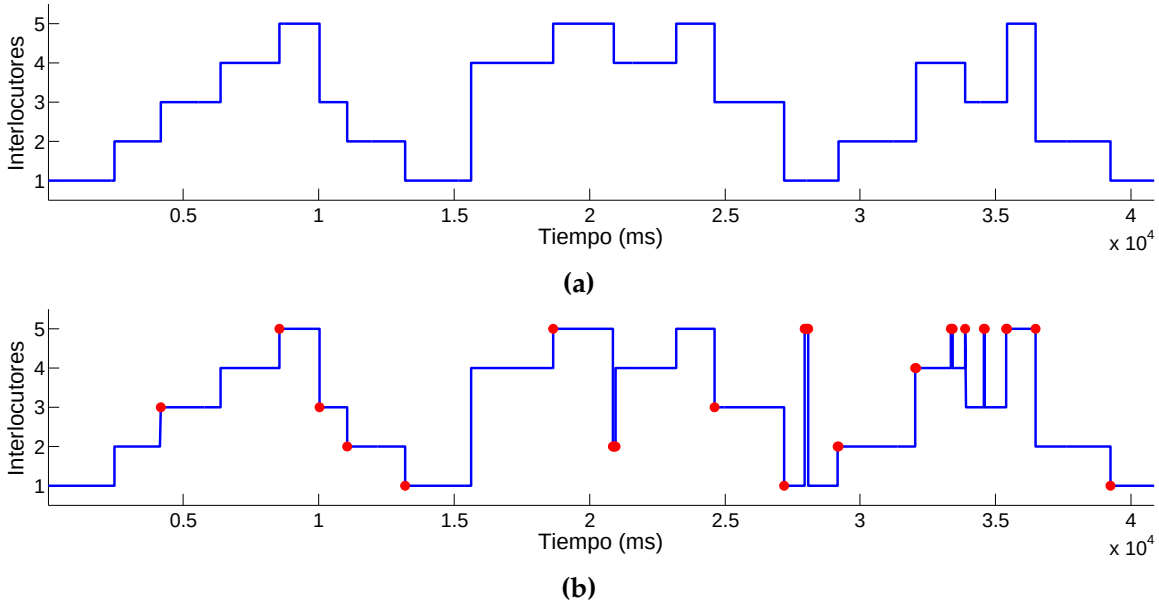


Figura 6.30: *En Figura 6.30a se muestra la secuencia original de la Subsección 6.1.6. En comparación, en Figura 6.30b se muestra la secuencia recuperada del modelo ganador y se marcan en rojo los errores cometidos respecto a la primera secuencia.*

Se observa también que la mayoría de las veces, en la secuencia recuperada se encuentran algunos brincos entre los interlocutores, pero en esencia la estructura y el orden en que hablan los interlocutores es el correcto.

6.2 RESULTADOS

A continuación se presentan la tabla Tabla 1 con las características de cada una de las secuencias utilizadas para las pruebas.

SEC. ID.	DURACIÓN	#SAMPLES	#WORDS	n_{TRUE}
ALLPOE	12:06 min	7218	140	6
GARMÁR	6:55 min	4154	90	4
WSHAKE	2:43 min	1636	160	4
MACUÑA	10:41 min	6414	120	3
CALDER	1:07 min	676	160	3
LLOYDW	6.48 min	4084	160	5

Tabla 1: *Descripción de características de secuencias utilizadas para las pruebas*

SEC. ID.	n_{TRUE}	n_{FOUND}	DER (%)
ALLPOE	6	6	03.324
GARMÁR	4	4	00.652
WSHAKE	4	4	12.353
MACUÑA	3	3	00.343
CALDER	4	3	*08.092
LLOYDW	5	5	01.170

Tabla 2: *Resultados de pruebas realizadas. Se observa en general que para la mayoría de las pruebas se encontró el modelo correcto, además de que se tiene un DER en general abajo del 10 %. En el caso de la secuencia `CALDER`, se muestra el DER correspondiente al modelo correcto.*

En la tabla Tabla 2 se muestran los resultados obtenidos de las pruebas realizadas. Para 5 de los 6 casos se acertó al modelo generador, siendo el caso presentado en la Subsección 6.1.5 el único en el que no se logró inferir de forma correcta. Como se observa en su curva BIC (Figura 6.23c), la mayoría de los modelos obtuvieron un puntaje similar, e incluso, el modelo con una mayor valoración fue otro ($n = 5$). De ahí la importancia de realizar la segunda etapa de refinamiento para mediante pruebas de hiótesis discernir al modelo correcto.

Como se muestra en la Figura 6.24, al momento de comparar el modelo verdadero ($n = 3$) contra otro modelo ($n = 4$), se rechaza la hipótesis nula de que el modelo generador sea el primero de estos, estando apenas el valor de LLR_{obs} por debajo del valor de significancia usado; por lo que en caso de haber usado un umbral mucho más pequeño se habría logrado identificar de forma correcta al modelo verdadero.

En la tabla Tabla 3 se muestra algunos detalles de cada una las pruebas que se realizaron.

Es importante resaltar también los tiempos de las pruebas que se realizaron, y que al ser técnicas computacionalmente demandantes, tardaron mucho en ejecutarse. Todas las pruebas se realizaron en una máquina de escritorio, con (...incluir especificaciones...).

Para cada prueba se realizaron el mismo número de simulaciones, aunque en cada una varía la duración de cada grabación de audio, así como la longitud del diccionario de palabras utilizado. También es importante la complejidad de los modelos con los que se hicieron las pruebas de hipótesis, pues mientras más complejo

SEC. ID.	DURACIÓN	$\#_{WORDS}$	MODELOS	EJECUCIÓN
ALLPOE	12:06 min	140	$\{4,5,6,7,8\}$	$\sim$ 402 hrs
GARMÁR	6:55 min	90	$\{2,3,4,5,6\}$	$\sim$ 140 hrs
WSHAKE	2:43 min	160	$\{2,3,4,5,6\}$	$\sim$ 66 hrs
MACUÑA	10:41 min	120	$\{2,3,4,5,6\}$	$\sim$ 176 hrs
CALDER	1:07 min	160	$\{2,3,4,5,6\}$	$\sim$ 23 hrs
LLOYDW	6.48 min	160	$\{3,4,5,6,7\}$	$\sim$ 193 hrs

Tabla 3: *Detalle de pruebas realizadas.*

un modelo, mayor número de parámetros se tienen que estimar a cada iteración del método.

7

DISCUSIÓN Y CONCLUSIONES

De acuerdo a los resultados descritos en el Capítulo 6, se observa que el método presentado muestra un buen desempeño en las pruebas realizadas. Es importante recalcar la naturaleza de estas pruebas, pues tienen ciertas características propias que facilitaron la realización de los experimentos.

Como se mencionó en el Capítulo 1, las pruebas se realizaron con secuencias de audio generadas artificialmente, utilizando un sintetizador de voz con motores en diferentes idiomas. Aunque esto nos permitió generar las secuencias de prueba de forma más fácil, esto implica que las voces puedan presentar algún patrón intrínseco por la misma naturaleza de los motores empleados. Además de esto, los valores que se utilizaron para construir el banco de filtros en la etapa de obtención de los MFCC puede que no funcionen de forma adecuada para voces humanas.

Otro de los problemas que se podría presentar en caso de utilizar otras secuencias de audio, seria que por la forma en que están generadas nuestras secuencias de audio, no se presentan empalmes entre los diferentes interlocutores; por lo que quizá sea mucho más fácil el poder obtener la secuencia correcta. Una forma de tratar de evitar esto, sería proponer en dado caso otra técnica para la selección de la secuencia óptima, en la que se puedan considerar estas anomalías.

Sin embargo, la aportación fuerte de este trabajo de tesis reside en la metodología propuesta para la selección de modelo correcto. Como ya se mencionó, se basa en técnicas sencillas, y que dado el poder de cómputo actual son fáciles de implementar.

Se presenta un método de selección de modelo en dos etapas:

EXPLORACIÓN: Mediante una versión de BIC regularizada; se escoge un subconjunto de soluciones más probables a partir del conjunto de modelos propuestos. Para esto, se usa una heurística establecida en un análisis de sensibilidad.

REFINACIÓN: A partir del subconjunto de soluciones obtenido en la primer etapa, se realiza un análisis estadístico mediante pruebas de hipótesis utilizando como estadístico a evaluar el LLR; lo que nos permite discriminar uno a uno los modelos finalistas.

A diferencia de otras técnicas, el método propuesto permite explorar diferentes posibles soluciones al mismo tiempo, y escoger de acuerdo a su verosimilitud cuál es la que representa una mejor solución. Además, se realiza un análisis estadístico sobre la pertenencia de los datos a modelo.

Sin embargo, y como se mostró en el Capítulo 6, se usan métodos computacionalmente demandantes, por lo que se debe de considerar qué tan importante es esta segunda etapa de refinación en algunos casos.

Como opción alternativa, también se implementó el algoritmo de Baum-Welch (B-W) en C++, buscando disminuir el tiempo de ejecución, y aunque se programó con OpenMP, por cuestiones de tiempo no se logró hacer un análisis completo sobre su desempeño.

7.1 TRABAJO FUTURO

Como trabajo futuro, hay varias etapas en las que se puede explorar mucho más el trabajo.

Principalmente, está el conseguir o armar de alguna manera un banco de pruebas con voces reales, que permitan analizar el comportamiento del método presentado en un entorno más real.

En caso de que se hagan pruebas con conversaciones reales, es importante también, mejorar la forma en que se eliminan los silencios y ruidos; pues en pruebas con un ambiente *normal*, hay muchas más fuentes de perturbación que las que hasta ahora se han considerado. Sería interesante también considerar la aplicación de algún filtro para atenuar ruidos propios de los micrófonos o del ambiente en que se graben las conversaciones.

También es importante explorar otras estrategias para selección de la segmentación más óptima, buscando realizar el cálculo de manera eficiente.

Evaluar la pertinencia de paralelizar el algoritmo principal de estimación de parámetros del HMM, para mejorar el rendimiento general del sistema.

[AMBE+12] Xavier Anguera Miró, Simon Bozonnet, Nicholas W. D. Evans, Corinne Fredouille, Gerald Friedland, and Oriol Vinyals. Speaker Diarization: A Review of Recent Research. *IEEE Transactions on Audio, Speech & Language Processing*, 20(2):356–370, 2012.

[AMWP06] Xavier Anguera Miró, Chuck Wooters, and José M. Pardo. Robust Speaker Diarization for Meetings: ICSI RT06S Meetings Evaluation System. In Steve Renals, Samy Bengio, and Jonathan G. Fiscus, editors, *MLMI*, volume 4299, pages 346–358, 2006.

[AV07] David Arthur and Sergei Vassilvitskii. k-means++: The advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, SODA '07, pages 1027–1035, Philadelphia, PA, USA, 2007. Society for Industrial and Applied Mathematics.

[BEF10] Simon Bozonnet, Nicholas W. D. Evans, and Corinne Fredouille. The LIA-EURECOM RT'09 Speaker Diarization System : enhancements in speaker modelling and cluster purification. In *ICASSP*, pages 4958–4961. IEEE, 2010.

[Bis06] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Information Science and Statistics. Springer, 2006.

[Bur13] Patrick Burns. The Statistical Bootstrap and Other Resampling Methods, 10 2013.

[CH10] Gerda Claeskens and Nils Lid Hjort. *Model Selection and Model Averaging*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, 2010.

[DE83] Persi Diaconis and Bradley Efron. Computer-Intensive Methods in Statistics. Technical Report 5, Stanford University, May 1983.

[DM80] Steven B. Davis and Paul Mermelstein. Comparison of Parametric Representations for Monosyllabic

Word Recognition in Continuously Spoken Sentences. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 28(4), 1980.

[Efr78] B. Efron. Bootstrap Methods: Another Look at the Jackknife. *The Annals of Statistics*, 7(1):1–26, 1978.

[FBE09] Corinne Fredouille, Simon Bozonnet, and Nicholas W. D. Evans. The LIA-EURECOM RT'09 Speaker Diarization System. In *RT 2009, NIST Rich Transcription Workshop, May 28-29, 2009, Melbourne, USA*, Melbourne, UNITED STATES, 05 2009.

[FE07a] Corinne Fredouille and Nicholas W. D. Evans. The LIA RT'07 Speaker Diarization System. In Rainer Stiefelhagen, Rachel Bowers, and Jonathan G. Fiscus, editors, *CLEAR*, volume 4625 of *Lecture Notes in Computer Science*, pages 520–532. Springer, 2007.

[FE07b] Corinne Fredouille and Nicholas W. D. Evans. The LIA RT'07 Speaker Diarization System. In Rainer Stiefelhagen, Rachel Bowers, and Jonathan G. Fiscus, editors, *CLEAR*, volume 4625 of *Lecture Notes in Computer Science*, pages 520–532. Springer, 2007.

[FSJW11] E.B. Fox, E.B. Sudderth, M.I. Jordan, and A.S. Willsky. A Sticky HDP-HMM with Application to Speaker Diarization. *Annals of Applied Statistics*, 5(2A):1020–1056, 2011.

[HMvL11] M. Huijbregts, M. McLaren, and D. van Leeuwen. Unsupervised acoustic sub-word unit detection for query-by-example spoken term detection. In *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, pages 4436–4439, 2011.

[HTF01] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer Series in Statistics. Springer New York Inc., New York, NY, USA, 2001.

[IFM+05] Dan Istrate, Corinne Fredouille, Sylvain Meignier, Laurent Besacier, and J.-F. Bonastre. NIST RT'05S Evaluation: Pre-processing Techniques and Speaker Diarization on Multiple Microphone Meetings. In Steve Renals and Samy Bengio, editors, *MLMI*, volume 3869 of *Lecture Notes in Computer Science*, pages 428–439. Springer, 2005.

[Jel98] Frederick Jelinek. *Statistical Methods for Speech Recognition*. Language, Speech, and Communication. MIT Press, 1998.

[JRP09] S. Jothilakshmi, V. Ramalingam, and S. Palanivel. Speaker diarization using autoassociative neural networks. *Engineering Applications of Artificial Intelligence*, 22(4–5):667–675, 2009.

[LG09] Howard Lei and Eduardo López Gonzalo. Mel, linear, and antimel frequency cepstral coefficients in broad phonetic regions for telephone speaker recognition. In *INTERSPEECH*, pages 2323–2326. ISCA, 2009.

[Mac07] James G. MacKinnon. Bootstrap Hypothesis Testing. Working Papers 1127, Queen's University, Department of Economics, June 2007.

[MBI01] S. Meignier, J.-F. Bonastre, and S. Igounet. E-HMM approach for learning and adapting sound models for speaker indexing. In *ISCA, A Speaker Odyssey, The Speaker Recognition Workshop*, Chiana (Crete), 18-22 Juin 2001 2001.

[MS91] David P. Morgan and Christopher L. Scofield. *Neural Networks and Speech Processing*. The Kluwer International Series in Engineering and Computer Science. Kluwer Academic Publishers, 1991.

[NLS09] Trung Hieu Nguyen, Haizhou Li, and Chng Eng Siong. Cluster criterion functions in spectral subspace and their applicationin speaker clustering. In *ICASSP*, pages 4085–4088, 2009.

[Rab89] Lawrence R Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77:257–286, 1989. errata at http://alumni.media.mit.edu/ rahimi/rabiner/rabiner-errata/rabiner-errata.html.

[RJ93] Lawrence R. Rabiner and Biing-Hwang Juang. *Fundamentals of speech recognition*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1993.

[RS07] Lawrence R. Rabiner and R.W. Schafer. Introduction to digital speech processing. *Foundations and trends in signal processing*, 1:1–194, 2007.

[Ryd08] Tobias Ryden. EM versus Markov chain Monte Carlo for Estimation of Hidden Markov Models: A Computational Perspective. *Bayesian Analysis*, 2008.

[SDDG13] Stephen Shum, Najim Dehak, Réda Dehak, and James R. Glass. Unsupervised methods for speaker diarization: An integrated and iterative approach. *IEEE Transactions on Audio, Speech & Language Processing*, 21(10):2015–2028, 2013.

[TR06] Sue Tranter and Douglas Reynolds. An Overview of Automatic Speaker Diarisation Systems. *IEEE Transactions on Speech, Audio & Language Processing, Special Issue on Rich Transcription*, pages 1557–1565, 2006.

[WH07] Chuck Wooters and Marijn Huijbregts. The ICSI RT07s Speaker Diarization System. In Rainer Stiefelhagen, Rachel Bowers, and Jonathan G. Fiscus, editors, *CLEAR*, volume 4625 of *Lecture Notes in Computer Science*, pages 509–519. Springer, 2007.

[ZGRD+11] Xinhui Zhou, Daniel Garcia-Romero, Ramani Duraiswami, Carol Y. Espy-Wilson, and Shihab A. Shamma. Linear versus mel frequency cepstral coefficients for speaker recognition. In David Nahamoo and Michael Picheny, editors, *ASRU*, pages 559–564. IEEE, 2011.