Dpto. de Teoría de la Señal y Comunicaciones
Universidad Carlos III de Madrid

TESIS DOCTORAL

SIMILARITY MEASURES FOR CLUSTERING SEQUENCES AND SETS OF DATA

Autor: Darío García García
Directores: Dr. Fernando Díaz de María
Dr. Emilio Parrado Hernández

Leganés, 2011

Tesis Doctoral:
SIMILARITY MEASURES FOR CLUSTERING SEQUENCES AND SETS OF DATA

Autor:
DARÍO GARCÍA GARCÍA

Directores:
DR. FERNANDO DÍAZ DE MARÍA
DR. EMILIO PARRADO HERNÁNDEZ

El tribunal nombrado para juzgar la tesis doctoral arriba citada, compuesto por los doctores

Presidente:

Vocales:

Secretario:

acuerda otorgarle la calificación de

Leganés, a

RESUMEN EXTENDIDO

En este resumen se pretende dar una visión de conjunto del trabajo realizado durante la elaboración de la presente Tesis Doctoral. Tras introducir el objetivo general de la misma, describimos la organización y las aportaciones originales del trabajo de investigación para por último presentar las conclusiones más relevantes.

Motivación y metodología

El objetivo de esta Tesis Doctoral es la definición de nuevas medidas de similitud para secuencias y conjuntos de datos, con la finalidad de servir de entrada a un algoritmo de agrupamiento o *clustering* [Xu and Wunsch-II, 2005]. El agrupamiento es una de las tareas más habituales dentro del ámbito del aprendizaje máquina (*machine learning*) [Mitchell, 1997]. Dicha tarea consiste en la partición de un conjunto de datos en subconjuntos aislados (*clusters*), de tal forma que los datos asignados a un mismo subconjunto sean parecidos entre sí, y distintos a los datos pertenecientes a otros subconjuntos. Una de sus principales particularidades es que se trata de una tarea *no supervisada*, lo cual implica que no requiere de un conjunto de ejemplos etiquetados. De esta forma se reduce la interacción humana necesaria para el aprendizaje, haciendo del agrupamiento una herramienta ideal para el análisis exploratorio de datos complejos. Por otro lado, es precisamente esta falta de supervisión la que hace fundamental el disponer de una medida adecuada de similitud entre elementos, ya que es la única guía durante el proceso de aprendizaje.

El agrupamiento de secuencias es una tarea cada día más importante debido al reciente auge de este tipo de datos. Cabe destacar el ámbito multimedia, en el que muchos contenidos presentan características secuenciales: señales de voz, audio, vídeo, etc. No es un ejemplo aislado, ya que en muchos otros ámbitos se producen casuísticas similares: desde los datos de bolsa y mercados financieros diversos al problema del análisis de movimiento. En la mayoría de estos casos la complejidad de los datos de entrada se une a la dificultad y elevado coste del etiquetado manual de dichos datos. Es precisamente en este tipo de escenarios en los que el agrupamiento es especialmente útil, debido a que no requiere de un etiquetado previo.

En muchos casos es posible prescindir de la dinámica de las secuencias sin perjudicar el proceso de aprendizaje. Son aquellos casos en los que las características estáticas de los datos de entrada son suficientemente discriminativas. Al obviar la dinámica, las secuencias se transforman en *conjuntos de datos*, que se interpretan como muestras (no necesariamente independientes) de unas determinadas distribuciones de probabilidad subyacentes. Ejemplos prácticos de ámbitos en los que se trabaja con conjuntos de datos incluyen el agrupamiento de locutores [Campbell, 1997], los modelos de bolsa de palabras (*bag of words*) para texto/imagen [Dance et al., 2004], etc.

En la presente Tesis propondremos métodos y, sobre todo, puntos de vista innovadores para la definición de similitudes entre secuencias o conjuntos de datos. Todos los métodos propuestos han sido analizados desde un punto de vista tanto teórico como empírico. Desde la perspectiva experimental se ha tratado de trabajar con la mayor cantidad de datos reales posibles, haciendo especial hincapié en las tareas de agrupamiento de locutores y reconocimiento de género músical.

Aportaciones originales de la Tesis

La primera parte de la Tesis se centra en el desarrollo de medidas de similitud basadas en *modelos dinámicos*, mediante los cuales se pueden capturar las relaciones entre los elementos de las secuencias. Bajo esta idea, se trabaja en dos líneas principales:

- **Medidas basadas en verosimilitudes:** Partiendo de un marco de trabajo estándar, como es el de medidas de similitud entre secuencias basadas en una matriz de verosimilitudes [Smyth, 1997], introducimos un nuevo método basado en una re-interpretación de dicha matriz. Dicha interpretación consiste en asumir la existencia de un *espacio latente de modelos*, y considerar los modelos empleados para la obtención de la matriz de verosimilitud como muestras de dicho espacio. De esta forma, es posible definir similitudes entre secuencias trabajando sobre las probabilidades definidas por las columnas de la matriz de verosimilitud (debidamente normalizadas). Por tanto, la medida de similtudes entre secuencias se transforma en el problema habitual de medi-

das de distancia entre distribuciones. El método es extremadamente flexible, ya que permite el uso de cualquier modelo probabilístico para representar las secuencias individuales.

- **Medidas basadas en trayectorias en el espacio de estados:** Con el objetivo de aliviar los problemas más notorios de los métodos basados en verosimilitudes, se introduce una nueva vía para definir medidas de similitud entre secuencias. Al trabajar con *modelos de espacio de estados* es posible identificar las secuencias con las trayectorias que inducen en tal espacio de estados. De esta forma, la comparación entre secuencias se traduce en la comparación entre las trayectorias correspondientes. El uso de un modelo oculto de Markov [Rabiner, 1989] común para todas las secuencias permite además que dicha comparación sea muy sencilla, ya que toda la información acerca de una trayectoria queda resumida en la matriz de transiciones que induce en el modelo. Estas ideas conducen a la distancia SSD (*space-state dynamics*) entre secuencias. Esta distancia permite reducir la carga computacional cuando el número de secuencias en el conjunto de datos es elevado, sorteando la necesidad de calcular la matriz de verosimilitudes. Asimismo, ofrece unas mejores prestaciones en secuencias cortas, debido a que las probabilidades de emisión son estimadas de forma global para todo el conjunto de datos. Como contraprestación, el tamaño del modelo global depende de la complejidad total del conjunto de datos. Por tanto, este método es especialmente interesante en escenarios en los que hay que agrupar un gran número de secuencias pertenecientes a un pequeño número de clases.

La segunda parte de la Tesis aborda el caso en el que se descarta la dinámica temporal de las secuencias, que pasan a ser un conjunto de puntos o vectores. En un buen número de escenarios es posible dar este paso sin perjudicar el aprendizaje, ya que las características estáticas (densidades de probabilidad) de los distintos conjuntos son suficientemente informativas de cara a realizar la tarea correspondiente. El trabajo realizado en esta parte se divide a su vez en dos líneas:

- **Agrupamiento de conjuntos de vectores basado en el soporte de las**

distribuciones en un espacio de características: Se propone agrupar los conjuntos tomando como noción de similitud una medida de la intersección de sus soportes en un espacio de Hilbert. La estima del soporte es un problema inherentemente más simple que la estima de densidades de probabilidad, ruta habitual hacia la definición de similitudes entre distribuciones. El trabajar en un espacio de características definido por un *kernel* permite obtener representaciones muy flexibles mediante modelos conceptualmente simples. Más concretamente, la estimación del soporte se basa en *hiperesferas* en espacios de dimensión potencialmente infinita [Shawe-Taylor and Cristianini, 2004]. El agrupamiento en sí se puede realizar de manera eficiente en forma jerárquica mediante un algoritmo de fusión de esferas basado en argumentos geométricos. Dicho algoritmo es una aproximación "*greedy*" al problema de encontrar las hiperesferas que cubren el conjunto de datos con la mínima suma de radios, y puede ser aplicado en espacios de características de dimensión potencialmente infinita.

- **Medidas de afinidad y divergencia basadas en clasificadores:** Partiendo de una interpretación de la similitud entre conjuntos de datos relativa a la *separabilidad* de dichos conjuntos, se propone cuantificar esta separabilidad empleando *clasificadores*. Esta idea se formaliza empleando el concepto de **riesgo** del problema de clasificación binaria entre pares de conjuntos. Como ejemplo práctico de este paradigma, demostramos que la tasa de error de un clasificador *nearest-neighbor* (NN) presenta varias características muy deseables como medida de similitud: existen algoritmos eficientes y con sólidas garantías teóricas para su estima, es un kernel definido positivo sobre distribuciones de probabilidad, presenta invariancia de escala, etc. La evolución natural de las medidas basadas en riesgos de clasificación pasa por su conexión con el concepto de *divergencias* entre distribuciones de probabilidad. Para ello se definen y analizan generalizaciones de la familia de f-divergencias [Ali and Silvey, 1966], que contiene muchas de las divergencias más habituales en los campos de la estadística y el aprendizaje máquina. Concretamente, proponemos dos generalizaciones: *class-restricted f-divergences* (CRFDs) y

loss-induced divergences o (f, l)-divergencias. Estas generalizaciones trasladan a la medida de divergencia las principales características de una tarea práctica de clasificación: la definición de un conjunto permisible de funciones de clasificación y la elección del coste o pérdida a optimizar.

- **CRFDs:** La idea detrás de las CRFDs es la sustitución de los riesgos de Bayes por riesgos óptimos dentro de familias restringidas de funciones de clasificación. De esta forma se generan divergencias que están intimamente relacionadas con clasificadores que trabajan sobre una determinada clase de funciones (por ejemplo, clasificadores lineales). Presentamos resultados teóricos mostrando las propiedades más importantes de esta familia generalizada de divergencias, y cómo dichas propiedades se relacionan con la familia de funciones de clasificación elegida. Uno de los resultados principales es que el conjunto de funciones lineales de clasificación define una familia de divergencias propias (en el sentido de que cumplen el principio de identidad de los indiscernibles), que a su vez son cotas inferiores de las divergencias f equivalentes.
- (f, l)**-divergencias:** Las (f, l)-divergencias nacen de la sustitución del coste 0-1 (o error de clasificación) por costes alternativos, denominados generalmente *surrogate losses* en la literatura. La conexión entre (f, l) y f-divergencias es muy estrecha. Bajo condiciones simples y naturales en las funciones de coste empleadas es posible demostrar que se mantienen las propiedades más importantes de las f-divergencias. Por otra parte, también demostramos que es posible obtener expresiones alternativas de muchas f-divergencias estándar en forma de (f, l)-divergencias con funciones de coste distintas al error de clasificación. Estas re-expresiones proporcionan nuevas visiones de divergencias bien conocidas, y permiten el desarrollo de nuevos métodos de estima. Como ejemplo, tras demostrar un resultado relacionando el error asimptótico de un clasificador NN [Devroye et al., 1996] con el riesgo de Bayes bajo la pérdida cuadrática, obtenemos nuevos estimadores y cotas de la divergencia de Kullback-Leibler [Kullback and Leibler, 1951]. Dichos estimadores están basados

únicamente en el orden de proximidad de los vecinos de cada punto en el conjunto de datos, y resultan competitivos con el estado del arte, presentando además la gran ventaja de su independencia respecto a la dimensionalidad del espacio de entrada.

A pesar de que en cada capítulo se realiza un trabajo experimental con datos tanto sintéticos como reales, en el Capítulo 6 se presenta una aplicación más elaborada de los métodos desarrollados. Se trata de una tarea de reconocimiento automático no supevisado de género musical. El uso de la divergencia KL estimada mediante errores NN presenta un rendimiento magnífico en este complejo escenario.

Conclusiones

A lo largo de la Tesis hemos propuesto una variedad de métodos destinados a avanzar el estado del arte en agrupamiento de secuencias y conjuntos de datos. Hemos trabajado en diversos frentes: tanto métodos basados en modelos dinámicos para explotar las relaciones entre elementos de las secuencias como métodos no paramétricos de gran capacidad expresiva para discriminar entre conjuntos de datos.

En cuanto a los métodos basados en modelos, hemos propuesto dos alternativas, denominadas KL-LL y SSD. El método KL-LL presenta la ventaja de su gran flexibilidad, permitiendo el empleo de cualquier modelo generativo probabilístico para representar las secuencias. Como contrapartida, requiere la evaluación de un número de verosimilitudes que es cuadrático en el número de secuencias en el conjunto de datos. Además, el ajustar modelos a secuencias individuales puede presentar problemas de sobreajuste cuando la longitud de las secuencias es baja. La distancia SSD alivia estos problemas, pero en principio su aplicación está limitada a modelos ocultos de Markov. Los resultados empíricos en multitud de bases de datos reales y sintéticas muestran que ambas propuestas ocupan un puesto de honor entre el estado del arte.

Por otra parte, cuando se descartan las características dinámicas de las secuencias, de forma que se transforman en conjuntos de datos, es posible trabajar con representaciones muy flexibles del espacio de entrada. De esta forma se evitan las

fuertes asunciones que generalmente hacen los modelos dinámicos sobre las distribuciones de probabilidad en el espacio de entrada para permitir que la inferencia sea viable. Un ejemplo de esta flexibilidad es el poder emplear espacios de características inducidos por *kernels*, como mostramos en el Capítulo 4. Tanto la combinación de la distancia MMD [Gretton et al., 2007] con *clustering* espectral como nuestra propuesta de fusión jerárquica de hiperesferas permiten el trabajo en dichos espacios.

Por último, hemos presentado un paradigma para la definición de afinidades entre conjuntos de datos basado en riesgos de clasificación. Esta conexión intuitiva ha sido generalizada desde el punto de vista de las divergencias entre distribuciones de probabilidad, dando lugar a generalizaciones de la familia de f-divergencias. El estudio de estas generalizaciones ha resultado muy fructífero desde el punto de vista teórico, ya que los resultados obtenidos han permitido estrechar el vínculo entre medidas de divergencia y clasificación. De esta forma se ha avanzado hacia la unificación de conceptos que a simple vista pueden parecer distantes. También se han obtenido resultados relevantes en el terreno práctico, como el nuevo estimador para la divergencia KL. Los resultados experimentales demuestran que tanto el uso de divergencias para definir afinidades de cara a un agrupamiento como el estimador concreto propuesto son herramientas muy útiles que constituyen una aportación relevante.

Cabe resaltar que, aunque las medidas propuestas han sido inicialmente empleadas para la tarea de agrupamiento, todas ellas son útiles en otras tareas. Como ejemplo, en el Apéndice C mostramos como una pequeña variación sobre el algoritmo de *clustering* espectral permite abordar la tarea de segmentación de secuencias.

Líneas futuras

A continuación enunciamos algunas de las líneas de investigación más prometedoras que se derivan de los contenidos de la presente Tesis. Desde el punto de vista de las aplicaciones de los métodos desarrollados las posibilidades son prácticamente ilimitadas, por lo cual nos centramos en extensiones teóricas y algorítmicas.

Existen varias cuestiones abiertas en el área de las medidas de afinidad basadas en matrices de verosimilitud. Por ejemplo, la posibilidad de entrenar modelos en

subconjuntos de secuencias (en lugar de secuencias individuales) como forma de sortear las principales limitaciones de este tipo de métodos. También resulta de interés el estudio del comportamiento de las afinidades basadas en verosimilitudes cuando los modelos son muestreados de forma aleatoria, en vez de aprendidos para representar secuencias/conjuntos de secuencias.

El trabajo del Capítulo 3 puede continuarse de forma natural extendiendo la idea de la distancia SSD a otro tipo de modelos de espacio de estados. Es de especial interés el caso de modelos cuyo espacio de estados sea contínuo en vez de discreto.

El algoritmo de agrupamiento basado en fusión de esferas está intrínsecamente conectado con el problema de *set covering*, esto es, encontrar una cubierta óptima de un conjunto. Se trata de un problema topológico, que en los últimos años se ha estudiado dentro del aprendizaje máquina para obtener nuevos métodos de clasificación [Marchand and Taylor, 2003]. Conectar el trabajo presentado en el Capítulo 4 con la literatura relativa al *set-covering* ayudaría a extraer nuevas conclusiones e inspiración para trabajos futuros.

Hablemos por último de las generalizacions de la familia de f-divergencias que se proponen en el Capítulo 5. Una de las líneas de investigación más obvias consiste en lograr obtener estimadores prácticos de las CRFDs. Para ello habría que encontrar maneras eficientes de estimar el riesgo restringido a una familia de funciones de clasificación para todo el rango de probabilidades a priori. Esto supone un problema de gran interés desde el punto de vista teórico, y cuya aplicación práctica es inmediata. En cuanto a la familia de (f, l)-divergencias, su tremenda flexibilidad abre un amplio abanico de posibilidades. Por ejemplo, resulta sencillo definir divergencias sensibles al coste, utilizando para ello funciones de pérdidas asimétricas. Para finalizar, destacar el interés de la combinación de CRFDs y (f, l)-divergencias presentada en la Sección 5.5.6. Dicha combinación define de forma natural divergencias basadas en clasificadores, y su estudio es muy prometedor tando desde el punto de vista teórico como práctico.

ABSTRACT

The main object of this PhD. Thesis is the definition of new similarity measures for data sequences, with the final purpose of *clustering* those sequences. Clustering consists in the partitioning of a dataset into isolated subsets or clusters. Data within a given cluster should be similar, and at the same different from data in other clusters. The relevance of data sequences clustering is ever-increasing, due to the abundance of this kind of data (multimedia sequences, movement analysis, stock market evolution, etc.) and the usefulness of clustering as an unsupervised exploratory analysis method. It is this lack of supervision that makes similarity measures extremely important for clustering, since it is the only guide of the learning process.

The first part of the Thesis focuses on the development of similarity measures leveraging *dynamical models*, which can capture relationships between the elements of a given sequence. Following this idea, two lines are explored:

- **Likelihood-based measures:** Based on the popular framework of likelihood-matrix-based similarity measures, we present a novel method based on a re-interpretation of such a matrix. That interpretations stems from the assumption of a *latent model space*, so models used to build the likelihood matrix are seen as samples from that space. The method is extremely flexible since it allows for the use of any probabilistic model for representing the individual sequences.

- **State-space trajectories based measures:** We introduce a new way of defining affinities between sequences, addressing the main drawbacks of the likelihood-based methods. Working with *state-space models* makes it possible to identify sequences with the trajectories that they induce in the state-space. This way, comparisons between sequences amounts to comparisons between the corresponding trajectories. Using a common hidden Markov model for all the sequences in the dataset makes those comparisons extremely simple, since trajectories can be identified with transition matrices. This new paradigm improves the scalability of the affinity measures with respect to the dataset size, as well as the performance of those measures when the sequences are

short.

The second part of the Thesis deals with the case where the dynamics of the sequences are discarded, so the sequences become *sets of vectors* or points. This step to be taken, without harming the learning process, when the statical features (probability densities) of the different sets are informative enough for the task at hand, which is true for many real scenarios. Work along this line can be further subdivided in two areas:

- **Sets-of-vectors clustering based on the support of the distributions in a feature space:** We propose clustering the sets using a notion of similarity related to the intersection of the supports of their underlying distributions in a Hilbert space. Such a clustering can be efficiently carried out in a hierarchical fashion, in spite of the potentially infinite dimensionality of the feature space. To this end, we propose an algorithm based on simple geometrical arguments. Support estimation is inherently a simpler problem than density estimation, which is the usual starting step for obtaining similarities between probability distributions.

- **Classifer-based affinity and divergence measures:** It is quite natural to link the notion of similarity between sets with the *separability* between those sets. That separability can be quantified using *binary classifiers*. This intuitive idea is then extended via generalizations of the family of f-divergences, which originally contains many of the best-known divergences in statistics and machine learning. The proposed generalizations present interesting theoretical properties, and at the same time they have promising practical applications, such as the development of new estimators for standard divergences.

AGRADECIMIENTOS

Vaya, parece que ahora sí que es de verdad. Se acaban estos cuatro años inolvidables, y qué mejor manera de poner punto y final que echar la vista atrás y dar las gracias.

Empezaré por lo más obvio: a mis directores Fernando y Emilio por haber confiado en mi peculiar forma de hacer las cosas, lo cual muchas veces debe resultar complicado. Fernando siempre me lo ha puesto todo fácil desde que llegué a la UC3M, y ha escuchado con interés todo lo que tenía que decir. Emilio ha difuminado la barrera entre ser mi director y ser uno más de mis amigos, a base de horas de conversación acerca de lo humano y lo divino (¡incluso la reina de Inglaterra!). En lo estrictamente profesional, siempre me ha apuntado en la dirección correcta y me ha transmitido la obsesión por la investigación de calidad (las grandes ligas).

Es un placer agradecerle también a Ule von Luxburg el haberme permitido pasar unos meses en Hamburgo (no dejéis de pasaros por allí si podéis) que resultaron profesionalmente fructíferos y personalmente inolvidables. También por su apoyo, fundamental de cara a mi próxima aventura transcontinental.

A toda esa gente que me recibió con los brazos abiertos en Madrid, y que increíblemente todavía me siguen aguantando, pese a que escaparon de la universidad hace tiempo: Javi, murciano orgulloso y siempre con alguna paradoja en mente. Manu, con su abrumadora amplitud de intereses y contagiosa energía. Jesús de Vicente, siempre poniéndole a la vida el toque de pausa que a otros nos falta. Esperemos que la nueva incorporación a la familia herede esa tranquilidad y no le mantenga muchas noches en vela. Miguel, paisano brillante como he conocido pocos y gran esperanza del TSC. Jaisiel y su conciencia social. Y David, con quien he pasado tantas tardes de press y tribulus, y noches de Vendetta y mandrágora.

Qué decir de la gente del TSC. He de agradecer a todos los que se han interesado por mi trabajo durante estos años, principalmente a Jesús Cid, Fernando Pérez y Jero, que siempre tienen algo interesante que aportar. A Rocío y Sara, siempre dispuestas a ayudar y a dar un poco de conversación en las aburridas tardes departamentales. A Iván y su paciencia infinita, a Edu y su cabeza, a Rubén, Sergio, Manolo. Mención especial para Raúl, con quien he recorrido ya medio mundo, desde Bristol a Canberra pasando por más sitios de los que pensaba visitar en mi vida.

Tanto viaje ha dado para hablar del trabajo, de intereconomía y, sobre todo, para darme cuenta de que es un amigo de verdad.

Le debo mucho a la gente de Santander, que consigue que volver a casa sea una experiencia distinta pero idéntica cada vez. A Manny, que siempre me recibe en su hogar, me obliga a estar en forma y me trata mejor de lo que merezco, así como a las mujeres de la casa: Patri y Erin!. A Jony, compañero fiel de aventuras de lo más diversas y mi mejor apoyo en Madrid (¡esta es para vos!). A Carlos y Sara por llevar tantos años siempre ahí, bien sea en el local de ensayo, en el difunto Woodstock o en sitios aún más absurdos. Al resto de Mandanga Bros, Jaime y Fonso, cracks de la música y amigos inmejorables. A todos los clásicos de la carrera (Chus, Nacho, Vali, Rubio, Moreno, Pablo y Lucía, Ramírez, Salas, Moi,...), porque con ellos me lo pasé tan bien que no me pareció mala idea seguir más años metido en una universidad. Os debo unas setas. Y por supuesto, a Marina, que aguanta todas mis ideas geniales y mi incapacidad patológica para saber qué voy a hacer con más de dos días de antelación.

Por último, a mi familia. A los que están y a los que ya no están, porque no soy más que un revoltijo de lo que me han enseñado (bueno, las cosas malas son de mi cosecha). A mis tíos y mi primo Fabián, que siempre animan el cotarro. A mi abuela, que prácticamente me crió y sigue siendo el eje de la familia. Y llego a mis padres. Es demasiado típico para mi gusto, pero hay que decir alto y claro que se lo debo todo. Cariño y comprensión absoluta para con mis rarezas, y abnegación ante la vida nómada que me ha tocado vivir. ¡Sois la leche!

Bueno, me vais a disculpar, pero se me hace tarde y me esperan en la imprenta. Me habré dejado algo en el tintero, no me lo tengáis a mal. A todos, gracias y nos vemos pronto. ¡Ha sido un placer!

Darío

Consistency is the last refuge of the unimaginative

Oscar Wilde

Contents

List of Figures

List of Tables

Chapter 1

Introduction and goals of the Thesis

1.1 General aspects

1.1.1 Clustering and similarity functions

Clustering [Xu and Wunsch-II, 2005] is a core task in machine learning and statistical data analysis. Its main goal is to find a natural partition of a given dataset into a certain number of disjoint groups or *clusters*. Data within a given cluster must be similar, and at the same time different from data on other clusters. In contrast with classification, which is arguably the best known machine learning task, clustering is an *unsupervised* learning technique. By unsupervised we mean that there is no training set, that is to say, a set of examples with labels associated to them. Instead, a clustering algorithm receives just unlabeled samples. This minimizes human interaction and the influence of domain knowledge, making clustering a very useful technique for exploratory data analysis or, in general, to explore a pool of data in search of interesting relationships. At the same time, the lack of supervisions makes clustering appear as a subjective task, even an art [Guyon et al., 2009]. Having no labels available, all information is extracted solely from the metric structure imposed over the data. Generally speaking, a good clustering is one that generates a parti-

tion that is *smooth* with respect to the underlying metric. Broadly speaking, this can be intuitively interpreted as not assigning close points to different clusters. In a classification setting, the dependence on the metric is not so dramatic, since labels are the main reference used to guide the learning process towards the desired solution. However, the impact of the metric in clustering is totally crucial. It is usually defined in terms of an *affinity or similarity function* $w : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ that assigns a certain similarity to each pair of points in a set $\mathcal{X}$. Equivalently, a dissimilarity function could be used. Note that, for many algorithms, the affinity functions used do not need to satisfy all the properties of a strict metric. This mainly applies to the subadditivity or *triangular inequality*.

It is thus extremely important to define adequate affinity functions in order to achieve good clustering results. In fact, the choice of the similarity measure is often more important than the choice of the clustering algorithm itself. In the standard case of data points living in a common vector space $\mathbb{R}^D$ [1], the choice of a similarity/dissimilarity function is usually restricted to a small list of well-known alternatives. Some of the most obvious choices are:

- **Euclidean distance:** $d_E(\mathbf{x}, \mathbf{y}) = \sqrt{(\mathbf{x} - \mathbf{y})^T (\mathbf{x} - \mathbf{y})} = \sqrt{\sum_{i=1}^{D} (x_i - y_y)^2}$

  Arguably, the most widely used metric for vectors, and the natural metric in $\mathbb{R}^D$.

- **Mahalanobis distance:** $d_M(\mathbf{x}, \mathbf{y}) = \sqrt{(\mathbf{x} - \mathbf{y})^T \mathbf{S}^{-1} (\mathbf{x} - \mathbf{y})}$

  Proposed in [Mahalanobis, 1936], it can be seen as a generalization of the Euclidean distance when a covariance matrix $\mathbf{S}$ is available. In fact, the standard Euclidean distance can be recovered by setting $\mathbf{S} = \mathbf{I}$, where $\mathbf{I}$ is an identity matrix of appropriate dimensions. If $\mathbf{S}$ is diagonal, it amounts to a weighted euclidean distance. In general, Mahalanobis distance is specially useful in cases where the amount of information provided by different dimensions is very different, or when there are differences in the scales.

- **Gaussian affinity:** $w_G(\mathbf{x}, \mathbf{y}) = \exp \frac{-||\mathbf{x}-\mathbf{y}||^2}{\sigma^2}$

[1] In fact, every finite-dimensional vector space (with dimension D)is isomorphic to $\mathbb{R}^D$.

> The Gaussian affinity is a positive definite kernel function [Berg et al., 1984], so it can be interpreted as an inner product in some Hilbert space. It can thus be used via the *kernel trick* to turn a linear algorithm into its nonlinear version, or to capture high-order correlations between data in a simple way [Schoelkopf and Smola, 2001]. We will elaborate on this further down the road, mainly on Chapter 4.

Things become wilder when more complex (structured) kinds of data are involved. In these cases, there are rich relations and redundancies in the input data that need to be accounted for. In this Thesis we will focus on defining similarity/dissimilarity functions for sequences and sets of data. These scenarios will be clearly defined in Section 1.1.2.

The interest on having meaningful similarity functions is not restricted to the clustering problem. In fact, an adequate characterization of the input space is key to obtain good-performing classifiers. This is done via an appropriate *regularization* of the classification problem [Hastie et al., 2003]. This implies that the goal is not just to get a function that correctly discriminates between classes in the training dataset but also a function which is *smooth* enough. Then, good generalization properties of the classification function when facing unseen data can be expected. Smoothness can be enforced in several ways. A modern approach to regularization consists in the introduction of a penalty term dependent on the Hilbertian norm of the candidate function [Schoelkopf and Smola, 2001]. This can be efficiently done if a positive definite kernel function on the input space is defined.

Smoothness (and, thus, similarity functions) is also specially important for semi-supervised learning [Chapelle et al., 2006]. This learning task can be considered as a middle-ground between clustering (unsupervised) and classification (supervised). In semi-supervised learning we are given both labeled and unlabeled datasets, and the goal is to find a classification function for unseen data (so it is an inductive learning task). The typical approach is to use unlabeled data to get an idea of the marginal distribution $P(x)$ of the data and use that information to enforce smoothness. A remarkable example of this idea is manifold regularization [Belkin et al., 2006].

So, even though our main focus will be clustering, the work in this Thesis will

also help to leverage all these powerful methods and ideas for general learning with sequential data. For example, in Appendix C we show how to use the proposed affinities for *segmentation* purposes.

1.1.2 Sequences of data

We are interested in sequential scenarios where the smallest meaningful unit is no longer a single data vector, but a sequence $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ of vectors. For example, in a classification setting there would be $(\mathbf{X}, y) = (\{\mathbf{x}_1, \ldots, \mathbf{x}_n\}, y)$ pairs of sequences and labels, instead of $(\mathbf{x}, y)$ pairs of vectors and labels. Analogously, in a sequential data clustering task the goal is not to group individual vectors, but whole sequences.

Obviously, the information that a sequence conveys may not be encoded only in the data vectors themselves, but also in the way they evolve along a certain dimension (usually time). Standard machine learning techniques assume individual data points to be independent and identically distributed (i.i.d.), so accounting for the temporal dependencies is a very important particularity for sequence-based algorithms. Moreover, sequences in a given dataset can (and most surely will!) present different lengths. This implies that different sequences can not be directly seen as points in a common vector space, although the individual vectors forming the sequences actually are. It is then obvious that somewhat more involved techniques need to be used to evaluate the relations between sequences, compared with the standard case.

The application of machine learning techniques to sequential and general structured data is lately receiving growing attention [Dietterich, 2009], and clustering is no exception to this trend [Liao, 2005]. Sequential data arise in many interesting and complex scenarios where machine learning can be applied. Here we briefly present some of them:

- **Audio:** Sequentiality and dynamics are utterly important for many audio-related tasks. As a simple example, consider *speech recognition* [Rabiner and Juang, 1993]. The human auditory system is much more sophisticated than the best artificial system so far. However, if you take a speech signal and randomly scrambles the time dimension, the resulting signal is com-

pletely impossible to understand, even though it shares the exact same "statical" probability distribution as the original signal. Apart from the classical speech recognition task, there are loads of audio-related applications. In fact, *music analysis* is one of the hot topics of the last years, given the ubiquity and size of music repositories. Efficient machine learning techniques are essential to fully explode the potential of such repositories. Examples of this kind of applications include similarity-based search [West et al., 2006] and musical genre recognition. We will further explore this last example in Chapter 6.

- **Video/Multimedia:** Multimedia material is complex and highly structured, and also inherently sequential. *Event recognition* in multimedia databases [Zelnik-Manor and Irani, 2001, Hongeng et al., 2004] is a flourishing machine learning task, with applications in sport-related material [Xu et al., 2003] or surveillance video [Cristani et al., 2007].

- **Stock markets:** On the recent years, machine learning has entered the investment community, answering to an ever-growing interest in automatic trading algorithms with solid statistical foundations [Huang et al., 2005, Hassan and Nath, 2006]. The evolution of stocks and derivatives prices exhibits highly complex dependencies, but also a strong sequential behavior. It is thus necessary to use specific machinery for sequential data in order to obtain good-performing algorithms (and not loose too much money!).

- **Gesture recognition:** Gestural communication is really important for human interaction, and it has to be cast in a sequential framework. The order in which movements are performed is essential to grasp the conveyed concept. Nowadays, there is a big interesting in leveraging this way of communication for a wide range of applications. For example, *sign-language gesture recognition* is an exciting example of machine learning with a social edge. It can be performed using special sensors [Liang and Ouhyoung, 2002] or standard cameras [Wu and Huang, 1999], and is a key accessibility technology. On the opposite side of the spectrum, electronic entertainment systems (mostly video games) are bound to take advantage of this new way of interaction with the user. A

groundbreaking example is Microsoft's Kinect [Microsoft, 2011].

- **Biomedical applications:** Lots of biomedical problems deal with the analysis of time-varying signal such as electroencephalography (EEG), electrocardiography (ECG), etc. In those cases, the temporal dimension is essential to find interesting patterns. Applications of machine learning to this kind of data are almost endless. Some relevant examples range from the purely medical applications, such as epileptic seizure detection [Shoeb and Guttag, 2010], to emerging topics such as EEG-based authentication [Marcel and Millan, 2007] or brain-computer interfacing [Dornhege et al., 2007].

1.1.3 Dropping the dynamics

As previously stated, sequential data carry part of their information in the evolution of the individual vectors. In fact, sometimes all the relevant information is encoded in the dynamics. Consider two sources θ_1, θ_2 of binary data, so $\mathcal{X} = \{0, 1\}$. One of them, θ_1, emits sequences of the form $101010101\ldots$, while θ_2 produces sequences $11001100\ldots$. If we are given a dataset comprised of sequences from θ_1 and θ_2, they will be indistinguishable by looking just at their probability distributions, since they will be exactly identical: $P(X = 1|\theta_1) = P(X = 1|\theta_2) = \frac{1}{2}$.

However, in many practical scenarios the dynamical characteristics of the sequences can be discarded without severe degradation in the performance of the learning task at hand. A classical example of this is speaker recognition [Campbell, 1997]. In this field, sequences are typically modeled as mixtures of Gaussians (MoGs) [Bishop, 2006]. Information about the dynamics is thus lost, considerably simplifying the learning process while keeping a good performance. This is possible because the "statical" probability distributions of the feature vectors of different speakers are separable enough.

When the dynamics can be safely ignored, sequences can be seen as sets of vectors coming from some underlying distributions. It is then natural to assume that a desirable clustering solution would consist on grouping sequences according to the similarity of their generating probability distributions. This way, the problem of defining affinity functions for the original sequences reduces to the well-known prob-

lem of measuring similarities or divergences between distributions. In those cases, we talk about *sets of vectors* or *samples* instead of sequences. The classical statistical approach to quantify the dissimilarity between distributions is the use of *divergence functionals*. There are lots of widely used divergences, many of which are member of the Csiszar's or f-divergence family [Ali and Silvey, 1966, Csiszár, 1967]. Members of this family share many interesting properties, and are closely related to classification risks. On the other hand, some modern approaches rely on embeddings of probability distributions into reproducing kernel Hilbert spaces (RKHS) [Smola et al., 2007], where high-order moments can be dealt with in a simple manner.

As a last note, it is obvious that in practice we do not have access to the actual distributions, but just to sets of samples from them. It is thus of utmost importance to define affinities that can be efficiently estimated in an empirical setting. There is always a trade-off between expressivity/flexibility and complexity involved.

We will discuss in depth all these aspects in Section 1.2.4 and Chapters 4 and 5.

1.2 State of the art in clustering sequential data

In this section we will briefly review the best-known techniques involved in the clustering of sequences or sets of data. The process is typically separated in two different stages: obtaining an adequate affinity or distance matrix and then feeding that matrix to a similarity-based clustering algorithm. Consequently, in the following we will present both standard clustering algorithms and state-of-the-art proposals for measuring affinity between sequences or sets of vectors.

1.2.1 Clustering algorithms

There exists a wide body of work on clustering algorithms [Xu and Wunsch-II, 2005], since it is one of the core machine learning tasks. Here we are specifically interested in *affinity based* algorithms. By this we mean those algorithms which take as an input an affinity matrix. This is not much of a constraint, since the best-known clustering methods fall into this category. The complexity of those algorithms range from the

simple ideas of the standard k-means algorithm [Hastie et al., 2003] to margin-based algorithms which require costly optimization processes [Xu et al., 2004]. Amongst all those methods, the family of algorithms collectively known as spectral clustering [Wu and Leahy, 1993, Shi and Malik, 2000, von Luxburg, 2007] has recently stood out in the crowd and received a lot of attention due to its good practical performance and solid theoretical foundation. These methods share graph-theoretic roots that results in non-parametric partitions of a dataset, in the sense that they do not impose any parametric structure on the cluster structure of the input data. They are based on the *Laplacian matrix* [Chung, 1997] induced by a given affinity function. Analogously to the Laplacian operator in calculus, this Laplacian matrix can be used to measure the smoothness of functions over the nodes of an undirected graph $G = (V, E)$. The vertices V correspond with the data points, while the weights of the edges E denote the similarity between those points. Those weights are given by the elements of the *affinity or similarity matrix*. The desired partition function is then found by eigendecomposition of the Laplacian matrix. Another possible interpretation of spectral clustering algorithms arise when seen as a relaxation of a graph-cut [Chung, 1997] minimization.

There exist many flavors of spectral clustering, differing mainly on the choice of the Laplacian: graph Laplacian, normalized Laplacian, etc. Each one of them presents some particularities, although the underlying concept is very similar in all cases. Since spectral clustering (specifically, normalized-cut [Shi and Malik, 2000]) will be our clustering method of choice, we devote Appendix A to a little more in-depth explanation of this algorithm.

1.2.2 Dynamical models

Arguably, the best-known way to extract dynamical information is via *dynamical models*. These are probabilistic models that do not make the usual assumption of independence between samples. This way, the usual factorization of the probability of a set of samples does not hold

$$P(\mathbf{x}_1, \ldots, \mathbf{x}_n) \neq P(\mathbf{x_1}) \ldots P(\mathbf{x}_n)$$

Instead, dynamical models make different assumptions about the relationships between points in a sequence/stochastic process. A usual assumption is Markov assumption. We state it here in the discrete case:

$$P(\mathbf{x}_t|\mathbf{x}_1, \ldots, \mathbf{x}_{t-1}) = P(\mathbf{x}_t|\mathbf{x}_{t-1}),$$

that is to say, the conditional (on both past and present values) probability of future states of the process depends only on the present state. This directly implies the following factorization of the marginal probability of the sequence:

$$P(\mathbf{x}_1, \ldots, \mathbf{x}_n) = P(\mathbf{x}_1)P(\mathbf{x}_2|\mathbf{x}_1) \ldots P(\mathbf{x}_n|\mathbf{x}_{n-1}),$$

There exist many different dynamical models, differing mainly on the kind of relationships they account for. Most well-known dynamical models falls into the *state-space* category. These are methods that assume that, at each time instant, there is an underlying hidden (non-observable) state of the world that generates the observations. This hidden state evolves along time, possibly depending on the inputs. A very general family of state-space models is known as dynamic Bayesian network (DBNs) [Murphy, 2002]. DBNs extend the well-known Bayesian network [Bishop, 2006] formalism to handle time-based relationships. Viewed from a graphical-model perspective, this allows to easily specify the (conditional) independences that are assumed. Out of this very general family, one of the simplest but more effective models is the hidden Markov model (HMM) [Rabiner, 1989]. As it name conveys, it is based on a Markov assumption. Namely, that the time evolution of the hidden state q follows a first-order Markov chain. Moreover, the observation $\mathbf{x}_t$ at an instant t depends only on the hidden state q_t at that same instant. This reduced set of dependencies allows for full model specification using a small number of parameters that can be estimated in fast and convenient ways. HMMs have been widely used in signal processing and pattern recognition because they offer a good trade-off between complexity and expressive power, and a natural model for many phenomena. We will make extensive use of hidden Markov models in this Thesis, so we have included all the necessary information about this well-known models in Appendix B.

1.2.3 Model-based clustering of sequences

There are two classic approaches to perform clustering of sequences with dynamical models: fully-parametric and semi-parametric. Fully parametric methods [Alon et al., 2003, Li and Biswas, 2000] assume that a sequence is generated by a mixture of HMMs (or any other model for sequential data). This way, the likelihood function given a dataset $\mathcal{S} = \mathbf{X}_1, \ldots, \mathbf{X}_N$ of N sequences can be written as

$$\mathcal{L} = \prod_{n=1}^{N} \sum_{m=1}^{M} z_{nm} p(\mathbf{X}_n | \theta_m),$$

where M is the number of mixture components, z_{nm} is the probability of sequence n being generated by the m^{th} element of the mixture, and $p(\mathbf{X}_n|\theta_m)$ is the likelihood of the m^{th} model given the n^{th} sequence. If each membership variable z_{nm} is assumed to be binary, the problem takes the form of k-means [Hastie et al., 2003] clustering. It implies a hard assignment of sequences to clusters at each iteration, so only the sequences classified as being generated by a certain model affect the re-estimation of its parameters [Li and Biswas, 2000]. Another alternative is the use of soft assignments by means of an EM method [Dempster et al., 1977]. This way, each sequence has a certain probability of being generated by each model of the mixture, so each $\mathbf{z}_n = \{z_{n1}, \ldots, z_{nM}\}$ is a vector living in the M-simplex. Each of these vectors is interpreted as a collection of missing variables that are estimated by the EM algorithm at each iteration [Alon et al., 2003]. A mixture model is a reasonable assumption in many scenarios, but it imposes severe restrictions in the cluster structure which limit the flexibility in the general case.

On the other hand, semi-parametric methods [Yin and Yang, 2005, García-García et al., 2009c] assume some parametric model of the individual sequences, define an affinity measure based on that parametric representations and then feed the resulting similarity matrix into a non-parametric clustering algorithm. These semi-parametric methods have been shown [Yin and Yang, 2005] to outperform both fully parametric methods like mixture of HMMs [Alon et al., 2003] or combinations of HMMs and dynamic time warping [Oates et al., 2001]. The work in [Smyth, 1997] proposes a framework for defining model-based distances between sequences. Specifically, it takes a likelihood-based approach: the main

idea is to fit individual probabilistic models to each sequence, and then obtain a likelihood matrix that represents the probability of each sequence being generated by each model. Following this work, many researchers [García-García et al., 2009c, Panuccio et al., 2002, Porikli, 2004, Yin and Yang, 2005] have proposed different distance measures based on such a likelihood matrix. All these works share the need to train a model on each single sequence. Besides, [Jebara et al., 2007] defines the similarity between two sequences as the probability product kernel (PPK) between HMMs trained on each sequence. Probability product kernels [Jebara et al., 2004] are a kind of affinity between probability distributions that can be efficiently calculated for many probabilistic models.

1.2.4 Affinity measures for sets of vectors

As previously stated, when the dynamics of a sequence of data is discarded, it can be interpreted as a sample from some underlying distribution. For example, if a sequence is generated by a hidden Markov model with Gaussian emissions, its static probability distribution will be a mixture of Gaussians. This way, the similarity between two sequences whose dynamics are discarded can be defined as the similarity between their corresponding probability distributions. In this section we will briefly present two important approaches for measuring similarity between probability distributions: feature space embeddings and the family of f-divergences.

RKHS embeddings of distributions

In recent years, the methodology of kernel methods [Schoelkopf and Smola, 2001] has been extended to deal with analysis of probability distributions [Smola et al., 2007]. Applications include the two sample problem [Gretton et al., 2007], independence measurement, etc. A key point behind these methods is that the reproducing kernel Hilbert space (RKHS) $\mathcal{H}$ induced by some kernel functions $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ are dense in the space of continuous bounded functions $C_O(\mathcal{X})$. Kernels that satisfy this property are called *universal*. Examples of universal kernels include the Gaussian and Laplacian RBF kernels. Under this condition there is a natural injective embedding between probability distributions $P(x)$ on $\mathcal{X}$ and their mean in the RKHS $\mu_P =$

$\mathbb{E}_{x\ P}[k(x,\cdot)]$. This injectivity implies that, via a universal kernel, any probability distribution P is uniquely represented by $\mu[P]$ and $\mu_P = \mu_Q$ iff $P = Q$. It is then natural to define the distance between two distributions P and Q as

$$D_{\mathcal{H}}(P,Q) = ||\mu_P - \mu_Q||_{\mathcal{H}}\,, \tag{1.1}$$

where $||\cdot||_{\mathcal{H}}$ stands for the kernel-induced norm in $\mathcal{H}$. Such a distance can also be motivated from a different point of view. Given the reproducing property of an RKHS, it is easy to see that $D_{\mathcal{H}}(P,Q) = \sup_{||f||_{\mathcal{H}}\leq 1} \mathbb{E}_P[f(x)] - \mathbb{E}_Q[f(x)]$. That is to say, it coincides with the maximum mean discrepancy (MMD) [Gretton et al., 2007] of P and Q over the class of functions given by the unit ball in $\mathcal{H}$. MMD has received widespread attention in recent years, but to the best of our knowledge it has not been used as an affinity measure for clustering sets of vectors. We will give more details about this measure in Chapter 4.

f-divergences

There is a wide body of work regarding divergences for probability distributions. Most of the proposed divergences an be seen as particular instances of some generic family, such as Bregman divergences [Bregman, 1967], integral probability metrics (IPMs) [Sriperumbudur et al., 2009] or f-divergences (also known as Csiszar's divergences) [Ali and Silvey, 1966]. In particular, f-divergences encompass many very well-known divergences such as the Kullback-Leibler (KL) divergence [Kullback and Leibler, 1951], Pearson's χ^2 divergence [Pearson, 1900] or the variational divergence [Devroye et al., 1996], amongst others. This is a very relevant subset of divergences, specially when considering that the intersection between different families of divergences is usually extremely small. For example, the intersection between Bregman and f-divergences comprises only the KL divergence [Reid and Williamson, 2009], while the variational divergence is the only f-divergence which is an IMP [Sriperumbudur et al., 2009].

All the instances of f-divergence admit an integral representation [Österreicher and Vajda, 1993] in terms of statistical informations [DeGroot, 1970]. These magnitudes are closely related to Bayes classification risks, showing the deep

connections between divergence measurement and discrimination. We will deal with f-divergences and their integral representations in Chapter 5.

1.3 Goals, contributions and organization of the Thesis

The main goal of the present Thesis is to develop a body of principled methods for obtaining similarity/dissimilarity measures for sequences, for the purpose of clustering. On the one hand, we will use a model-based approach for those scenarios where the dynamics are important. On the other hand, we will tackle the "sets-of-vectors" scenario (that is to say, sequences where the dynamics are discarded). We will focus on both theoretical and practical issues, paying special attention to real-world scenarios.

Here we briefly summarize the main contributions of each chapter:

On Chapter 2 we present the framework proposed in [Smyth, 1997] for clustering of sequences of data, and how it has evolved into a successful semi-parametric approach. After reviewing the most relevant works on likelihood-matrix-based distances between sequences, we present an alternative proposal based on a latent model space-based interpretation of the likelihood matrix.

Chapter 3 aims at solving a main weakness of the kind of semi-parametric models explored in Chapter 2. The fact that each model is trained using just one sequence can lead to severe overfitting or non-representative models for short or noisy sequences. In addition, the learning of a likelihood matrix as required by such methods involves the calculation of a number of likelihoods or probability product kernels that is quadratic in the number of sequences, which hinders the scalability of the method. To overcome these disadvantages, we propose to train a single HMM using all the sequences in the dataset, and then cluster the sequences attending to the transition matrices that they induce in the state-space of the common HMM. The approach taken in this chapter is radically different from the aforementioned methods in the sense that it is not based on likelihoods, but on divergences between the transition probabilities that each sequence induces under the common model. In other words, we no longer evaluate the likelihoods of the sequences on some models and then define the distance accordingly. Instead, the focus is now shifted towards

the parameter space. Moreover, the identification of each sequence with a transition matrix opens up new possibilities since the metric can be based on the short term transitions, the long term stationary state distribution or on some middle ground.

The following chapters deal with the case where dynamics are dropped, and thus we look at the sequences as (not strictly independent) samples from some underlying probability distribution. In Chapter 4 we define a clustering procedure based on the overlap of the distributions in a feature space. The main assumption is that the underlying distributions do not overlap too much. This is usually a very strong assumption in the input space, but RKHS embedding arguments [Gretton et al., 2007, Sriperumbudur et al., 2009] show that it is reasonable in a universal kernel (such as the Gaussian kernel) induced feature space. We leverage ideas from support estimation in RKHS which have previously been applied to novelty-detection [Shawe-Taylor and Cristianini, 2004]. In particular, we obtain an approximation empirical support of each set consisting on hyperspheres in Hilbert spaces induced by Gaussian kernels. Assuming that we are looking for K clusters, the final goal is to obtain K hyperspheres with the smallest total radius that encompass all the points sets. Sets within the same sphere will lay on the same cluster. To this end, we propose a greedy algorithm based on a geometric sphere-merging procedure.

In Chapter 5 we introduce a new framework for defining affinity measures between sets of vectors using classification risks. The main idea is that, for clustering purposes, it is natural to relate the similarity between sets to how hard to separate they are, for a given family of classification functions. In many scenarios, practitioners know what kind of classifier (e.g. linear classifiers, SVMs with a certain kernel, etc.) works well with the kind of data they are trying to cluster. Then, it is natural to use affinity measures derived from such classifiers. We choose the nearest neighbor (NN) classifier and show how its asymptotic error rate exhibits some very interesting properties as an affinity function. This idea may seem intuitive, but simplistic at the same time. We address this by linking it with f-divergences, presenting a couple of generalized families of divergences consistent with the idea of classifier-based measures. We do this by exploiting the integral representation in [Österreicher and Vajda, 1993], which relates f-divergences and binary classification

error rates. On the one hand, we propose an extension of f-divergences which is related to restrictions in the set C of allowed classification functions. Controlling this set is equivalent to selecting the features of the distributions that we consider relevant for each application. We show what conditions need to be imposed on C so that the resulting divergences present key properties. On the other hand, the second generalization deals with substitutions of the 0-1 loss (error rates) for more general *surrogate losses*. Many interesting results arise, including relationships between standard divergences and surrogate losses (under what circumstances can we express a standard f-divergence in terms of a surrogate loss?) and between the properties of surrogate losses and their induced divergences. We also contribute another result linking the asymptotic error of a NN classifier and the Bayes risk under the squared loss. Coupled with the aforementioned results, this leads to a new way of empirical estimation or bounding of the KL divergence.

Though we present experimental results using both synthetic and real-world data on every chapter, we present a more elaborate application of clustering of sequences on Chapter 6, dealing with songs. Effective measures of similarity between music pieces are very valuable for the development of tools that help users organize and listen to their music, and can also serve to increase the effectiveness of current recommender systems, thus improving users' experience when using these services. In this chapter, we specifically address the problem of musical genre recognition. This is a complex task and a useful testbed for the methods developed in this Thesis.

Finally, in Chapter 7 we look back at the main results of the Thesis and contextualize them.

As a last note, we will make a slight digression in Appendix C, where we show how to adapt the spectral clustering methodology in order to use it for segmentation purposes. This implies a very simple change in the algorithm, allowing the previously presented methods to be used in segmentation scenarios. This way, a sequence can be broken down into segments which are coherent attending to their dynamical features (if the model-based affinities are used). The resulting segmentation can be used as-is, or employed as a initialization point for a further generative-model-based analysis.

Chapter 2

Clustering sequences using a likelihood matrix: A new approach

We review the existing alternatives for defining likelihood matrix-based distances for clustering sequences and propose a new one based on a latent model space view. This distance is shown to be especially useful in combination with spectral clustering. For improved performance in real-world scenarios, a model selection scheme is also proposed.

2.1 Introduction and chapter structure

As commented on the previous chapter, an intuitive framework for defining model-based distances for clustering sequential data consists on, first, learning adequate models for the individual sequences in the dataset, and then use these models to obtain a likelihood matrix, from which many different distances between sequences can be derived. This was originally proposed in [Smyth, 1997], and has since proved itself a very popular framework that has spanned many related works [Panuccio et al., 2002, Porikli, 2004, Yin and Yang, 2005], all of them being very similar in their philosophy.

In this chapter, we will first study the existing proposals under this framework and then explore a different approach to define a distance measure between sequences

by looking at the likelihood matrix from a probabilistic perspective. We regard the patterns created by the likelihoods of each of the sequences under the trained models as samples from the conditional likelihoods of the models given the data. This point of view differs largely from the existing distances. One of its differentiating properties is that it embeds information from the whole dataset or a given subset of it into each pairwise distance between sequences. This gives rise to highly structured distance matrices, which can be exploited by spectral methods to give a very high performance in comparison with previous proposals. Moreover, we also tackle the issue of selecting an adequate representative subset of models, proposing a simple method for that purpose when using spectral clustering. This greatly increase the quality of the clustering in those scenarios where the underlying dynamics of the sequences do not adhere well to the employed models.

This chapter is organized as follows: In Section 2.2 we review the general framework for clustering sequential data, together with the most employed tools within that framework, namely HMMs as generative models and hierarchical and spectral clustering, whose main characteristics are briefly outlined. The existing algorithms under this framework are also reviewed. Section 2.3 introduces our proposal of a new distance measure between sequences . Performance comparisons are carried out in Section 2.4, using both synthetic and real-world data. Finally, Section 2.5 collects the main conclusions of this work and sketches some promising lines for future research.

2.1.1 Related publications

This chapter is mainly based on [García-García et al., 2009c].

2.2 A Framework for Likelihood-Based Clustering Sequential Data

The seminal work of Smyth [Smyth, 1997] introduces a probabilistic model-based framework for sequence clustering. Given a dataset $\mathcal{S} = \{S_1, \ldots, S_N\}$ comprised of N sequences, it assumes that each one of them is generated by a single probabilistic

model from a discrete pool and the final aim is to cluster the sequencesa according to those underlying models.

The main idea behind this framework is to learn a generative model θ_i for every individual sequence S_i and then use the resulting models $\theta_1, \ldots, \theta_N$ to obtain a length-normalized log-likelihood matrix $\mathbf{L}$. The ij^{th} element l_{ij} of such a matrix is defined as

$$l_{ij} = \log p_{ij} = \frac{1}{\text{length}(S_j)} \log f_{\mathbf{S}}(S_j; \theta_i), \quad 1 \leq i, j \leq N, \tag{2.1}$$

where $f_{\mathbf{S}}(\cdot; \theta_i)$ is the probability density function (pdf) over sequences according to model θ_i. Based on this likelihood matrix, a new distance matrix $\mathbf{D}$ can be obtained so that the original variable-length sequence clustering problem is reduced to a typical similarity-based one. One of the strongest point of this approach is that it is very flexible, in the sense that any probabilistic generative model can be seamlessly integrated. This allows for the application of this methodology to a wide range of problems.

The following subsections will briefly describe the most usual tools under this framework: HMMs for the individual sequence models and hierarchical and spectral clustering for the actual partitioning of the dataset. Then,we briefly address the existing algorithms in the literature under this framework.

2.2.1 Hidden Markov Models

Hidden Markov models (HMMs)[Rabiner, 1989] are a type of parametric discrete state-space model, widely employed in signal processing and pattern recognition. Their success comes mainly from their relative low complexity compared to their expressive power and their ability to model naturally occurring phenomena. Its main field of application has traditionally been speech recognition [Rabiner, 1989], but they have also found success in a wide range of areas, from bioinformatics [Baldi et al., 1998] to video analysis [Jin et al., 2004].

In an HMM, the (possibly multidimensional) observation $\mathbf{y}_t$ at a given time instant t (living in a space $\mathbf{Y}$) is generated according to a conditional pdf $f_{\mathbf{Y}}(\mathbf{y}_t|q_t)$, with q_t being the hidden state at time t. These states follow a time-homogeneous first-order Markov chain, so that $P(q_t|q_{t-1}, q_{t-2}, \ldots, q_0) = P(q_t|q_{t-1})$. Bearing this

in mind, an HMM θ can be completely defined by the following parameters:

- The discrete and finite set of K possible states $\mathcal{X} = \{x_1, x_2, \ldots, x_K\}$
- The state transition matrix $\mathbf{A} = \{a_{ij}\}$, where each a_{ij} represents the probability of a transition between two states: $a_{ij} = P\left(q_{t+1} = x_j | q_t = x_i\right), 1 \leq i, j \leq K$
- The emission pdf $f_{\mathbf{Y}}(\mathbf{y}_t | q_t)$
- The initial probabilities vector $\pi = \{\pi_i\}$, where $1 \leq i \leq K$ and $\pi_i = P\left(q_0 = x_i\right)$

The parameters of an HMM are traditionally learnt using the Baum-Welch algorithm [Rabiner, 1989], which represents a particularization of the well-known Expectation-Maximization (EM) algorithm [Dempster et al., 1977]. Its complexity is $O(K^2T)$ per iteration, with T being the length of the training sequence. A hidden Markov model can be seen as a simple Dynamic Bayesian Network (DBN) [Murphy, 2002], an interpretation which provides an alternative way of training this kind of models by applying the standard algorithms for DBNs. This allows for a unified way of inference in HMMs and their generalizations. A more thorough explanation of HMMs can be found at Appendix B.

2.2.2 Hierarchical clustering

Hierarchical clustering (HC) [Xu and Wunsch-II, 2005] algorithms organize the data into a hierarchical (tree) structure. The clustering proceeds in an iterative fashion in the following two ways. Agglomerative methods start by assigning each datum to a different cluster and then merging similar clusters up to arriving at a single cluster that includes all data. Divisive methods initially consider the whole data set as a unique cluster that is recursively partitioned in a way such that the resulting clusters are maximally distant. In both cases, the resulting binary tree can be stopped at a certain depth to yield the desired number of clusters.

2.2.3 Spectral clustering

Spectral clustering (SC) [Wu and Leahy, 1993] casts the clustering problem into a graph partitioning one. Data instances form the nodes of a weighted graph whose

edges represent the adjacency between data. The clusters are the partitions of the graph that optimize certain criteria. These criteria include the normalized cut, that takes into account the ratio between the cut of a partition and the total connection of the generated clusters. To find these optimal partitions is an NP-hard problem, that can be relaxed to a generalized eigenvalue problem on the Laplacian matrix of the graph. The spectral techniques have the additional advantage of providing a clear and well-founded way of determining the optimal number of clusters for a dataset, based on the eigengap of the similarity matrix [Ng et al., 2002]. A deeper explanation of Spectral Clustering is provided in Appendix A.

2.2.4 Existing algorithms for likelihood-based clustering of sequences

The initial proposal for model-based sequential data clustering of [Smyth, 1997] aims at fitting a single generative model to the entire set $\mathcal{S}$ of sequences. The clustering itself is part of the initialization procedure of the model. In the initialization step, each sequence S_i is modeled with a HMM θ_i. Then, the distance between two sequences S_i and S_j is defined based on the log-likelihood of each sequence given the model generated for the other sequence:

$$d_{SYM}^{ij} = \frac{1}{2}\left(l_{ij} + l_{ji}\right), \tag{2.2}$$

where l_{ij} represents the (length-normalized) log-likelihood of sequence S_j under model θ_i. In fact, this is the symmetrized distance previously proposed in [Juang and Rabiner, 1985]. Given these distances, the data is partitioned using agglomerative hierarchical clustering with the "furthest-neighbor" merging heuristic.

The work in [Panuccio et al., 2002] inherits this framework for sequence clustering but introduces a new dissimilarity measure, called the BP metric:

$$d_{BP}^{ij} = \frac{1}{2}\left\{\frac{l_{ij} - l_{ii}}{l_{ii}} + \frac{l_{ji} - l_{jj}}{l_{jj}}\right\}. \tag{2.3}$$

The BP metric takes into account how well a model represents the sequence it has been trained on, so it is expected to perform better than the symmetrized distance in cases where the quality of the models may vary along different sequences.

Another alternative distance within this framework is proposed in [Porikli, 2004], namely

$$d^{ij}_{POR} = |p_{ij} + p_{ji} - p_{ii} - p_{jj}|\,, \tag{2.4}$$

with p_{ij} as defined in eq. (2.1).

Recently, the popularity of spectral clustering has motivated work in which these kinds of techniques are applied to the clustering of sequences. In [Yin and Yang, 2005] the authors propose a distance measure resembling the BP metric

$$d^{ij}_{YY} = |l_{ii} + l_{jj} - l_{ij} - l_{ji}|\,, \tag{2.5}$$

and then apply spectral clustering on a similarity matrix derived from the distance matrix by means of a Gaussian kernel. They reported good results in comparison to traditional parametric methods using initializations such as those proposed in [Smyth, 1997] and [Oates et al., 2001], called Dynamic Time Warping (DTW).

2.3 KL-LL disimilarity

Our proposal is based on the observation that the aforementioned methods define the distance between two sequences S_i and S_j using solely the models trained on them (θ_i and θ_j). We expect a better performance if we add into the distance some global characteristics of the dataset. Moreover, since distances under this framework are obtained from a likelihood matrix, it seems natural to take the probabilistic nature of this matrix into account when selecting adequate distance measures.

Bearing this in mind, we propose a novel sequence distance measure based on the Kullback-Leibler (KL) divergence [Kullback and Leibler, 1951], which is a standard measure for the similarity between probability density functions.

The first step of our algorithm involves obtaining the likelihood matrix $\mathbf{L}$ as in eq. (2.1) (we will assume at first that an HMM is trained for each sequence). The i^{th} column of $\mathbf{L}$ represents the likelihood of the sequence S_i under each of the trained models. These models can be regarded as a set of "intelligently" sampled points from the model space, in the sense that they have been obtained according to the sequences in the dataset. This way, they are expected to lie in the area of the model

space $\boldsymbol{\theta}$ surrounding the HMMs that actually span the data space. Therefore, these trained models become a good discrete approximation $\tilde{\boldsymbol{\theta}} = \{\theta_1, \ldots, \theta_N\}$ to the model subspace of interest. If we normalize the likelihood matrix so that each column adds up to 1, we get a new matrix $\mathbf{L}_N$ whose columns can be seen as the probability density functions over the approximated model space conditioned on each of the individual sequences:

$$\mathbf{L}_N = \left[f_{\tilde{\boldsymbol{\theta}}}^{S_1}(\theta), \ldots, f_{\tilde{\boldsymbol{\theta}}}^{S_N}(\theta) \right].$$

This interpretation leads to the familiar notion of dissimilarity measurement between probability density functions, the KL divergence being a natural choice for this purpose. Its formulation for the discrete case is as follows:

$$D_{KL}(f_P || f_Q) = \sum_i f_P(i) \log \frac{f_P(i)}{f_Q(i)}, \tag{2.6}$$

where f_P and f_Q are two discrete pdfs. Since the KL divergence is not a proper distance because of its asymmetry, a symmetrized version is used

$$D_{KL_{\mathrm{SYM}}}(f_P || f_Q) = \frac{1}{2} \left[D_{KL}\left(f_P || f_Q\right) + D_{KL}\left(f_Q || f_P\right) \right]. \tag{2.7}$$

This way, the distance between the sequences S_i and S_j can be defined simply as

$$d^{ij} = D_{KL_{\mathrm{SYM}}}\left(f_{\tilde{\boldsymbol{\theta}}}^{S_i} || f_{\tilde{\boldsymbol{\theta}}}^{S_j} \right). \tag{2.8}$$

We denote this disimilarity measure as KL-LL. This definition implies a change of focus from the probability of the sequences under the models to the likelihood of the models given the sequences. Distances defined this way are obtained according to the patterns created by each sequence in the probability space spanned by the different models. With this approach, the distance measure between two sequences S_i and S_j involves information related to the rest of the data sequences, represented by their corresponding models.

This redundancy can be used to define a representative subset $\mathcal{Q} \subseteq \mathcal{S}$ of the sequences, so that $\tilde{\boldsymbol{\theta}} = \{\theta_{Q_1}, \ldots, \theta_{Q_P}\}, P \leq N$. In this way, instead of using the whole dataset for the calculation of the distances, only the models trained with sequences belonging to $\mathcal{Q}$ will be taken into account for that purpose. The advantage of defining such a subset is twofold: on the one hand, computational load can be reduced

since the number of models to train is reduced to P and the posterior probability calculations drop from NxN to PxN. On the other hand, if the dataset is prone to outliers or the models suffer from overfitting, the stability of the distance measures and the clustering performance can be improved if $\mathcal{Q}$ is carefully chosen. Examples of both of these approaches are shown in the experiments included in Section 2.4. Obtaining this measure involves the calculation of $N(N-1)$ KL divergences, with a complexity linear in the number of elements in the representative subset. Therefore, its time complexity is $O(P \cdot N(N-1))$. Nevertheless, it is remarkable that the processing time devoted to the distance calculation is minimal in comparison to those involved in training the models and evaluating the likelihoods.

Finally, before applying a spectral clustering, the distance matrix $\mathbf{D} = \{d_{ij}\}$ must be transformed into a similarity matrix $\mathbf{A}$. A commonly used procedure is to apply a Gaussian kernel so that $a_{ij} = \exp\left(\frac{-d_{ij}^2}{2\sigma^2}\right)$, with σ being a free parameter representing the kernel width. Next, a standard normalized-cut algorithm is applied to matrix $\mathbf{A}$, resulting in the actual clustering of the sequences in the dataset. In the sequel, we will refer to this combination of our proposed KL-based distance and spectral clustering as KL+SC.

2.3.1 Model Selection

Since real-world data are inherently noisy and the sequences do not perfectly fit a markovian generative model, the property of embedding information about the entire set of sequences in each pairwise distance can become performance-degrading. Thus, it becomes interesting to select only an adequate subset C of the models for obtaining the distance matrix. This way, we will be performing the clustering in a reduced subspace spanned just by the chosen models.

For this purpose, we propose a simple method to determine which models to include in the KL+SC method. Since exhaustive search of all the possible subsets is intractable, we devise a growing procedure which sequentially adds sequences to the pool of representative models, and propose a simple heuristic to select the optimal pool.

- **Pool building:** First, we need to choose a initial sequence to represent

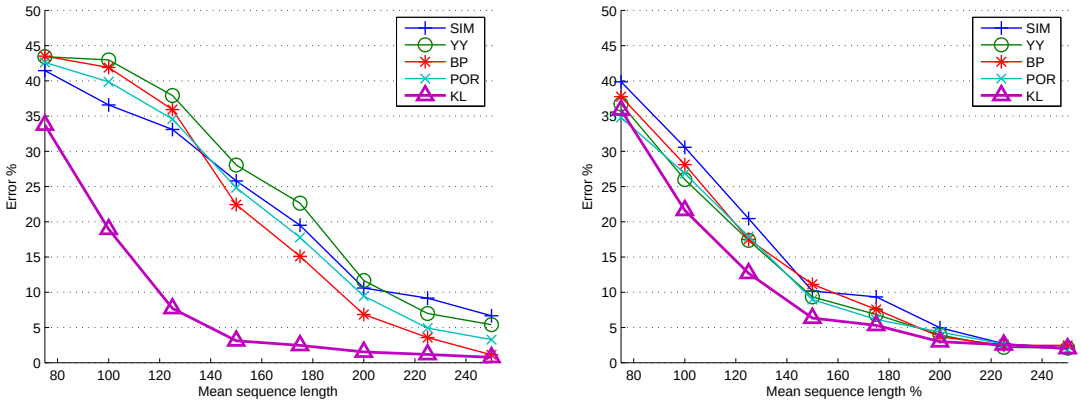
via its corresponding model, yielding the initial pool C_0. This can be done randomly or using some simple criterion like the length of the sequences, since models coming from lengthier sequences are expected to be less influenced by outliers and to provide more information about the underlying processes. The initial likelihood matrix $\mathbf{L}_0$ is then obtained by evaluating log-likelihoods of all sequences under the model in C_0. The pool is built up from there by adding at each step t models corresponding to the sequences which are poorly represented by current models. That is to say, sequences with low mass of probability as given by $\sum_{\theta \in C_{t-1}} f_{\mathbf{S}}(S;\theta)$, where C_{t-1} is the pool of models at step $t-1$. This is proportional to the row-sum of the likelihood matrix $\mathbf{L}_{t-1}$.

- **Pool selection:** For each candidate likelihood matrix, the KL-based distance is evaluated and a tentative clustering is carried out. We choose as the optimal clustering the one with the largest eigengap. Depending on the computational/time constraints, it is possible to try every candidate pool or to use an early stopping procedure, halting the process when the eigengap stops decreasing.

As previously stated, this is a simple method with no aspirations of being optimum but developed just for illustrating that an adequate selection of models can be advantageous, or even necessary, for attaining good clustering results in noisy scenarios. We refer to the KL+SC method coupled with this model selection scheme as KL+SC+MS. In the experiments below, no early stopping is used, so all the candidate pools returned by the pool building procedure are tried out.

2.4 Experimental Results

This section presents some experimental results concerning several synthetic and real-world sequence clustering problems. Synthetic data experiments aim at illustrating the performance of the different sequence clustering algorithms under tough separability conditions but fulfilling the assumption that the sequences are generated by hidden Markov models. This way, we focus the analysis on the impact of the distance measures as we isolate the adequateness of the modeling (except in



(a) Spectral clustering

(b) Hierarchical clustering

Figure 2.1: Clustering error percentage achieved by the compared methods against different mean sequence lengths ($V = 40\%$,$N = 80$)

overfitting). Besides, we also use two real-word scenarios, namely speech data and electroencephalogram (EEG) data, to show a sample application of sequence clustering in two fields where HMMs have typically been used as rough approximate generative models.

Apart from our proposal (KL-LL), we also introduce into the comparison all the likelihood-based measures that we have mentioned in Section 2.2.4. Namely:

SYM Symmetrized distance (eq. (2.2))

BP BP distance (eq. (2.3))

POR Porikli distance (eq. (2.4))

YY Yin - Yang distance (eq. (2.5))

KL Proposed KL-LL distance (eq. (2.8))

All of them will be paired with both an agglomerative hierarchical clustering using the furthest-neighbor merging heuristic, as in [Smyth, 1997], and a normalized-cut spectral clustering. For the spectral clustering algorithm, the value of parameter σ of the Gaussian kernel is selected empirically in a completely unsupervised fashion

as the one that maximizes the eigengap for each distance measure in each case (as proposed in [Ng et al., 2002]). It is also remarkable that the k-means part of the spectral clustering algorithm, due to its strong dependence on the initialization, is run 10 times at each iteration, and we choose as the most adequate partition the one with the minimal intra-cluster distortion, defined as:

$$D_{cluster} = \sum_{k=1}^{K} \sum_{i \in \mathcal{C}_k} ||\mathbf{x}_i - \mathbf{c}_k||^2,$$

where K is the number of clusters, $\mathcal{C}_k$ is the set of the indices of points belonging to the k^{th} cluster, $\mathbf{c}_k$ is the centroid of that cluster and $\mathbf{x}_i$ is the i^{th} data point. This distortion can be seen as the "tightness" of the resulting clusters, and it is also well known that this minimum distortion criterion implies a maximum separation amongst centroids [Shawe-Taylor and Cristianini, 2004].

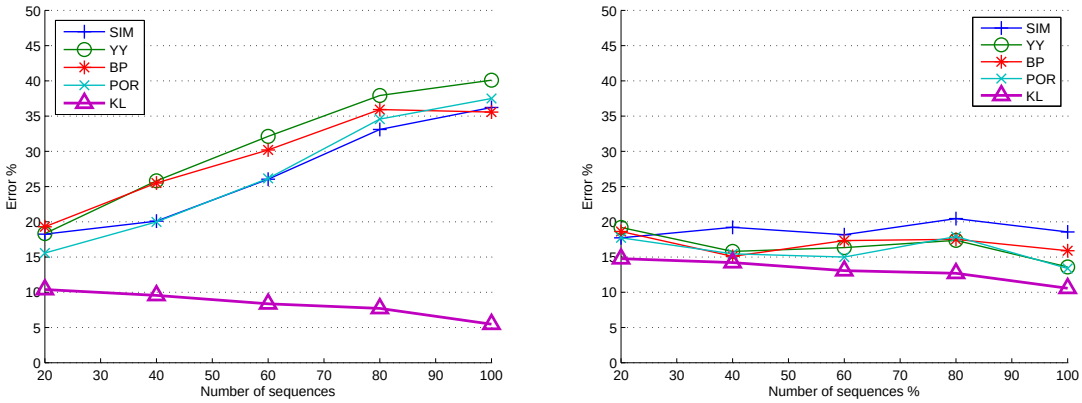
2.4.1 Synthetic data

The first scenario under which the comparison is carried out is the original example from [Smyth, 1997]: each sequence in the dataset is generated with equal probability by one of two possible HMMs θ_1 and θ_2, each one of them having two hidden states ($m = 2$). Transition matrices for the generating HMMs are given by

$$\mathbf{A}_1 = \begin{pmatrix} 0.6 & 0.4 \\ 0.4 & 0.6 \end{pmatrix} \qquad \mathbf{A}_2 = \begin{pmatrix} 0.4 & 0.6 \\ 0.6 & 0.4 \end{pmatrix}.$$

Initial states are equiprobable and emission probabilities are the same in both models, specifically $N(0, 1)$ in the first state and $N(3, 1)$ in the second. This scenario represents a very appropiate testbed for sequence clustering techniques, since the only way to differentiate sequences generated by each model is to attend to their dynamical characteristics. These, in turn, are very similar, making this a hard clustering task. The length of each individual sequence is obtained by sampling a uniform pdf in the range $[\mu_L(1 - V/100) \quad \mu_L(1 + V/100)]$, where μ_L is the sequence's mean length and V is a parameter which we will refer to as the percentage of variation in the length. All the given results are averaged over 50 randomly generated datasets.

Figure 2.1 shows the results of the performance comparison of the different distance measures and clustering methods against variations of the mean length μ_L of

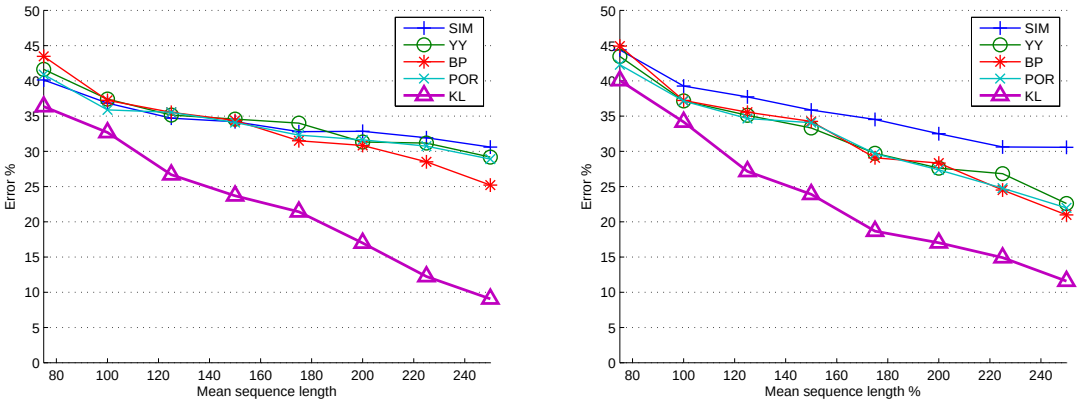


(a) Spectral clustering

(b) Hierarchical clustering

Figure 2.2: Clustering error against number of sequences in the synthetic dataset

the data sequences for a fixed length variation V of 40% in a dataset comprised of $N = 80$ sequences. Performance is measured in the form of clustering error, understood as the percentage of incorrectly classified samples under an optimal permutation of the cluster labels. It can be seen that, as expected, the longer the sequences the more accurate the clustering. It is also clear that our proposed distance measure outperforms the previous proposals under both hierarchical and spectral clustering, attaining specially good results using the latter technique. Specifically, the proposed KL+SC method yields the best performance for every mean sequence length, showing consistent improvements which are more dramatic for short mean sequence lengths ($\mu_L < 200$). Models trained with such short sequences suffer from severe overfitting, not being able to adequately capture the underlying dynamics and thus giving unrealistic results when evaluated using the sequences in the dataset. This results into incoherent distance matrices using the typical methods which renders the use of spectral clustering algorithms unproductive. Nonetheless, our proposal is more resilient against this issue since it takes a global view on the dataset that allows for the correct clustering of sequences even if the models are rather poor. Evaluating the sequences on a large enough number of individually inadequate models can generate patterns that our distance measure can capture, which translates into a consistent distance matrix very suitable for applying spectral methods. This shows that our



(a) Spectral clustering

(b) Hierarchical clustering

Figure 2.3: Performance in a multiclass ($K = 3$) clustering task against different mean sequence lengths (V=40%,N=100)

approach is efficient even when the models are poor so they cannot be expected to correctly sample the model space. In these scenarios, the probabilistic interpretation of the proposed distance is not clear and it takes more of a pattern matching role.

Agglomerative hierarchical clustering is more forgiving of loosely structured distance matrix, since it merges clusters based on pairwise distance comparisons instead of taking a global view. Therefore, it seems more suitable than spectral clustering methods for its use with the previously proposed model-based sequence distances. On the other hand, it also implies that it cannot benefit from the use of our proposed distance as much as spectral techniques can.

Figure 2.2 displays the evolution of the error along the number of sequences in the dataset. As more sequences are present in the dataset, the aforementioned problems of the previous proposals in combination with spectral clustering become clearer, while our method manages to improve its performance. Using hierarchical clustering, all the distances achieve stable results irrespective of the number of sequences, but once again this comes at the expense of an inferior performance compared to the KL+SC combination.

Figure 2.3 shows the results for a multi-class clustering with $K = 3$ classes. The sequences being clustered were generated using the two previously employed HMMs

(θ_1 and θ_2) and a third one θ_3 that only differs from them in the transition matrix. Specifically,

$$\mathbf{A}_3 = \begin{pmatrix} 0.7 & 0.3 \\ 0.3 & 0.7 \end{pmatrix}.$$

The additional class makes this a harder problem than the two-class scenario, so it is logical to assume that lengthier sequences are required to achieve comparable results. Nonetheless, the use of our proposed distance still shows significant improvements over the rest of the distances, all of which give almost identical results.

2.4.2 Real-world data

In this section, different sequential-data clustering algorithms will be evaluated on real-world scenarios. The first scenario is speaker clustering: we are given a set of audio files, each one of them containing speech from a single speaker, and the task is to group together files coming from the same speaker (two speakers per experiment). We simulate spaker clustering tasks using two different datasets: the GPM PDA and Japanese Vowels (JV) datasets (see Appendix D. The coefficient sequences were directly fed into the different clustering algorithms without any further processing. We use the JV dataset to simulate a 9-class speaker clustering task, while the GPM dataset is used for a pairwise 2-class task.

The other scenario used for testing purposes is clustering of electroencephalogram (EEG) signals. Given a subject, the purpose is to find clusters of sequences representing the same activity. Concretely, we perform seven clusterings (one for each subject) of 50 sequences (10 per mental activity, randomly chosen) into five groups.

Table 2.1 shows the results (averaged over 15 iterations) of the compared methods in the different datasets using spectral clustering. Hierarchical clustering results are not shown because of space restrictions, but they were clearly inferior to those attained via spectral clustering. Agglomerative methods fail in these scenarios because the relationships among the data that must be exploited in order to obtain an adequate partition are impossible to capture in a pairwise fashion. Results are given under varying number of hidden states, in the range where the different methods

perform best for each dataset.

All in all, the KL+SC+MS combination noticeably outperforms the alternatives, specially in the speech datasets. The use of KL+SC without model selection does not work as well as in the synthetic experiments because sequences belonging to the same cluster are not actually drawn from a unique HMM. The clustering just relies on the assumption that these sequences lead to similar models. In this scenario there are no such things as "true" HMMs generating the dataset, so the interpretation of the likelihood matrix that gives birth to our proposal loses part of its strength. However, it can be seen that if the KL+SC combination is coupled with the proposed model selection method it produces convincing results even in such adverse conditions.

A remarkable fact is that the previously proposed distances suffer from a severe performance loss in the speaker clustering tasks as the number of hidden states increases. This is caused by the models overfitting the sequences in these datasets because of the high dimensionality of the data and the short mean length of the sequences. The evaluation of likelihoods under these models produces results that does not reflect the underlying structure of the data. This distortion severely undermines the performance of previously proposed distances, yielding poorly structured distance matrices that seriously hinders the spectral clustering. The use of our proposed KL distance, specially in combination with model selection, has a smoothing effect on the distance matrices. This effect makes it less sensitive to overfit models, resulting in an improved performance relative to the other distances as overfitting becomes more noticeable. This robustness is a very useful property of our proposal since, in practice, it is usually hard to determine the optimum model structure and overfitting is likely to occur. It is also worth noting that the advantage of using our method is clearer in the GPM-UC3M dataset, because the number of sequences considered in each clustering task is larger. This agrees with the conclusions drawn from the experiments with synthetic data.

The dimensionality of the data in the EEG dataset is lower than in the speaker verification ones. This allows for an increase in the number of hidden states without suffering from overfitting. The KL+SC+MS method performs best also in this dataset, followed closely by the YY distance. It is remarkable that the improvement

Table 2.1: Mean and standard deviation of clustering error (%) on the real datasets using HMMs with different number of hidden states and spectral clustering: Japanese Vowels (JV), GPM-UC3M and EEG.

Dataset	# of hidden states	SYM	BP	YY	POR	KL	KL+MS
JV	m=2	33.33% (±2.8)	14.70% (±1.0)	14.89% (±2.1)	58.99% (±1.1)	12.72 (±1.3)	**9.85**% (±1.7)
	m=3	32.82% (±3.0)	24.66% (±3.7)	20.99% (±4.1)	54.88% (±2.3)	19.68 (±3.8)	**18.86%** (±4.0)
	m=4	32.04% (±3.4)	21.45% (±4.8)	26.59% (±6.2)	61.49% (±1.0)	16.33 (±4.3)	**14.45%** (±5.4)
	m=5	29.56% (±3.6)	20.23% (±4.4)	17.19% (±5.2)	58.60% (±1.3)	18.05 (±5.1)	**17.70%** (±5.5)
GPM	m=2	15.46 (±3.67)	12.62 (±3.97)	15.02 (±3.95)	49.91 (±0.19)	15.72 (±4.30)	**9.82** (±3.45)
	m=3	26.95 (±4.34)	24.07 (±5.25)	25.11 (±4.83)	49.82 (±0.29)	23.87 (±4.52)	**9.65** (±2.93)
	m=4	38.32 (±2.73)	38.18 (±3.44)	35.12 (±4.33)	49.74 (±0.26)	29.75 (±3.76)	**10.04** (±2.50)
EEG	m=5	60.57 (±0.49)	37.01 (±0.88)	**27.47** (±0.60)	51.36 (±0.70)	33.47 (±0.82)	29.89 (±0.85)
	m=6	60.56 (±0.40)	34.08 (±0.90)	30.06 (±0.76)	52.75 (±0.82)	33.63 (±0.85)	**28.86** (±0.89)
	m=7	60.56 (±0.44)	31.62 (±0.94)	28.50 (±0.87)	54.38 (±0.71)	29.53 (±0.84)	**16.79** (±0.76)

Table 2.2: Optimal percentage of models chosen by the model selection algorithm

Dataset	m=2	m=3	m=4
JAPANESE VOWELS	66.63%	72.28%	75.3%
GPM-UC3M	43.1%	58.61%	63.20%
Dataset	m=5	m=6	m=7
EEG	57.14%	60.12%	64.46%

in performance due to the use of model selection is less dramatic in this scenario because of both the absence of overfitting and the equal length of all sequences.

In Table 2.2 we show the number of models chosen for consideration by the model selection algorithm in each of the clustering tasks. Notice how in the three cases as the complexity of the models (in terms of number of hidden states) increases, the model selection scheme picks a larger number of them. In general, more complex models lead to more varying probabilities when evaluated on the different sequences. This way, the effective dimension of the model-induced subspace where the sequences lie grows with the complexity of the models, which agrees with the aforementioned behavior of the model selection scheme.

2.5 Summary

We have proposed a new distance measure for sequential data clustering, based on the Kullback-Leibler divergence. It embeds information of the whole dataset into each element of the distance matrix, introducing a structure that makes it specially suitable for its use in combination with spectral clustering techniques. This measure also allows for the use of a reduced representative subset of models, which, if chosen properly, can give an increase in performance in real-world scenarios potentially containing outliers and misleading data.

The reported results have been obtained using HMMs as generative models for the individual sequences, although the method is independent of this selection. In fact, exploring more expressive models is a straightforward and promising future line of research in order to successfully apply this clustering technique to a wider range

of problems, such as video event detection, text mining,etc.

Chapter 3

State Space Dynamics for Clustering Sequences of Data

We propose a novel similarity measure for clustering sequential data. We first construct a common state-space by training a single probabilistic model with all the sequences in order to get a unified representation for the dataset. Then, distances are obtained attending to the transition matrices induced by each sequence in that state-space. This approach solves some of the usual overfitting and scalability issues of the existing semi-parametric techniques, that rely on training a model for each sequence.

3.1 Introduction and chapter structure

In the previous chapter we presented the general framework for likelihood-based clustering of sequences. In [Jebara et al., 2007] another method for constructing the similarity matrix in a model-based approach is proposed which avoids the calculation of the likelihood matrix $\mathbf{L}$. It is based on the definition of probability product kernels (PPK) [Jebara et al., 2004] between two densities $p(x), q(x)$ over $\mathcal{X}$:

$$\kappa_{\text{PPK}}(p,q) = \int_{\mathcal{X}} p(x)^{\rho} q(x)^{\rho} dx = \langle p^{\rho}, q^{\rho} \rangle_{L_2}, \tag{3.1}$$

where L_2 is the space of square-integrable functions and ρ is a free parameter. Again an HMM is trained on each individual sequence, but then similarities between sequences are computed directly through a PPK. Specifically, the PPK with $\rho = 2$ of the probability densities spanned by two HMMs in the space of the sequences of a fixed length T_{PPK}, which is a design parameter. This way, $\mathbf{L}$ is no longer necessary because the similarities are obtained directly from parameters of the models. The calculation of the PPK between two HMMs can be carried out in $O(K^2 T_{\text{PPK}})$ time.

All the aforementioned semi-parametric methods share the need to train an individual HMM on each sequence in the dataset (or in a subset of it, as we did in Chapter 2). We see this as a disadvantage for several reasons. Short sequences are likely to lead to overfitted models, providing unrealistic results when used to evaluate likelihoods or PPKs. Moreover, training individual models in an isolated manner prevents the use of similar sequences to achieve more accurate representations of the states. As for the computational complexity, these methods do not scale well with the dataset size N. Specifically, the number of likelihoods that need to be obtained is N^2 (or PN using the KL method). In the case of PPKs, $N^2/2$ evaluations are required, since the kernel is symmetric. This quadratic number of likelihood evaluations hinders the scalability of the methods to large datasets.

This chapter aims at overcoming these weaknesses. Specifically, we propose to train a single HMM using all the sequences in the dataset, and then cluster the sequences attending to the transition matrices that they induce in the state-space of the common HMM. This approach is radically different from the aforementioned methods in the sense that it is not based on likelihoods, but on divergences between the transition probabilities that each sequence induces under the common model. In other words, we no longer evaluate the likelihoods of the sequences on some models and then define the distance accordingly. Instead, the focus is now shifted towards parameter space. Moreover, the identification of each sequence with a transition matrix opens up new possibilities since the metric can be based on the short term transitions, the long term stationary state distribution or on some middle ground.

The rest of this chapter is organized as follows: Section 3.2 describes how to cluster sequences using information about their dynamics in a common state-space.

This new proposal is empirically evaluated in comparison with other methods in Section 3.3. Finally, Section 3.4 draws the main conclusions of this work and sketches some promising lines for future research.

3.1.1 Related publications

This chapter is mainly based on [García-García et al., 2011a].

3.2 State Space Dynamics (SSD) Distance

In this chapter, we propose to take a novel approach in order to overcome the need of fitting an HMM to each sequence. To this end, we propose to train a single, large HMM Θ of K hidden states using all the sequences in the dataset. This will allow for a better estimation of the emission probabilities of the hidden states, compared to the case where an HMM is trained on each sequence. Then, we use the state-space of Θ as a common representation for the sequences. Each sequence $\mathbf{S}_n$ is linked to the common state-space through the transition matrix that it induces when is fed into the model. This matrix is denoted as $\tilde{\mathbf{A}}^n = \left\{\tilde{a}_{ij}^n\right\}_{i,j=1}^K$, where

$$\tilde{a}_{ij}^n = p(q_{t+1}^n = s_j | q_t^n = s_i, \mathbf{S}_n, \Theta). \tag{3.2}$$

In order to obtain each $\tilde{\mathbf{A}}^n$, we run the forward-backward algorithm for the sequence $\mathbf{S}_n$ under the parameters Θ (including the learned transition matrix $\mathbf{A} = \{a_{ij}\}$) and then obtain the sequence-specific transition probabilities by using equation (B.5):

$$\tilde{a}_{ij}^n \propto \sum_{t'=1}^{T} \alpha_i^n(t') a_{ij} p(\mathbf{x}_{t'+1} | q_{t'+1} = s_j) \beta_j^n(t'+1), \tag{3.3}$$

where $\alpha_i^n(t)$ and $\beta_j^n(t'+1)$ are the forward and backward variables for $\mathbf{S}_n$, respectively. This process can be seen as a projection of the dynamical characteristics of $\mathbf{S}_n$ onto the state-space defined by the common model Θ. Therefore, the overall transition matrix $\mathbf{A}$ of the large model Θ acts as a common, data-dependent "prior" for the estimation of these individual transition matrices.

When using dynamical models for sequence clustering, the use of the proposed common set of emission distributions can be motivated as follows: if this kind of algorithms are used, the most likely situation is that there is a high degree of state-sharing between the models for the different classes. If that were not the case, the "static" probability densities of the different classes would hardly overlap and the dynamical information would not be required, so simpler models such as mixtures of Gaussians could be used.

This procedure is somewhat equivalent to obtaining individual HMMs with emission distributions that are shared or "clamped" amongst the different models. Clamping is a usual and useful tool when one wants to reduce the number of free parameters of a model in order to either obtain a better estimate or reduce the computational load. In our case, the effects of clamping the emission distributions are two-fold: we get the usual benefit of better estimated parameters and, at the same time, it allows for simple distance measures between hidden Markov models using the transition distributions. This happens because the transition processes of the different models now share a common support, namely the fixed set of emission distributions.

As previously mentioned, running the forward-backward algorithm implies a time complexity of $O(K^2T)$ for a sequence of length T, which is the same complexity required for obtaining the likelihood of an HMM. Our proposal only requires N of these calculations, instead of N^2 likelihood evaluations or $N^2/2$ PPKs as the methods mentioned in the previous section do. This makes the SSD algorithm a valuable method for working with large datasets.

At this point, we have each sequence $\mathbf{S}_n$ represented by its induced transition matrix $\tilde{\mathbf{A}}^n$. In order to define a meaningful distance measure between these matrices, we can think of each $\tilde{\mathbf{A}}^n = [\mathbf{a}_{n1}, \ldots, \mathbf{a}_{nK}]^T$ as a collection of K discrete probability functions $\mathbf{a}_{n1}, \ldots, \mathbf{a}_{nK}$, one per row, corresponding with the transition probabilities from each state to every other state. In this manner, the problem of determining the affinity between sequences can finally be transformed into the well-studied problem of measuring similarity between distributions. In this work, we employ the

Bhattacharyya affinity [Bhattacharyya, 1943], defined as:

$$D_B(p_1, p_2) = \sum_x \sqrt{p_1(x)p_2(x)}, \tag{3.4}$$

where p_1 and p_2 are discrete probability distributions. We consider the affinity between two transition matrices to be the mean affinity between their rows. The distance between two sequences $\mathbf{S}_i$ and $\mathbf{S}_j$ can then be written as:

$$d_{ij}^{\text{BHAT}} = -\log \frac{1}{K} \sum_{k=1}^{K} D_B(p_{ik}, p_{jk}). \tag{3.5}$$

Once the distances between all the sequences are obtained, the actual clustering can be carried out using spectral clustering (or any other typical technique). We refer to this algorithm as state-space dynamics (SSD) clustering. It is summarized in Alg. 1.

It is worth noting that our proposal does not include any special initialization of the large model representing the dataset, such as imposing a block-diagonal structure on the transition matrix to encourage the clustering [Smyth, 1997]. We do not aim to obtain a single generative model of the complete dataset, but an adequate common representation that allows for a subsequent successful non-parametric clustering.

An important free parameter of our method is K, the number of hidden states of the common model. It should be chosen accordingly to the richness and complexity of the dataset. In the worst case (that is to say, assuming that there is no state sharing amongst different groups), it should grow linearly with the number of groups. In the experiments included in this work, we have fixed this size a priori, but it could be estimated using well-known criteria such as BIC or AIC [Bishop, 2006].

Nonetheless, we do not expect this parameter K to be critical for the success of the method as long as it is within a sensible range , and we prove this point in the experiments by trying a wide range of hidden space cardinalities. Further discussion about this topic is provided in Sec. 3.3.3.

Recall that the forward-backward algorithm for HMMs is $O(K^2T)$, where K is the number of states and T the sequence length. This indicates that our proposal is specially suitable for datasets consisting of a large number of sequences coming from a small number of clusters, which is a usual case. In such a scenario, the number

of hidden states required for a successful clustering is low, so the time penalty in the FW-BW algorithm will be more than compensated by the significant computational load reduction coming from the linear complexity in the number of sequences. If sequences coming from different clusters share some emission distributions, the improvements will be even more notorious, because the algorithm will exploit that sharing in a natural way.

As previously explained, dynamic model-based distance measures are specially useful when there is a significant overlap between the emission distributions of different classes. Our proposal is thus very suitable for working in this interesting scenario. If there were no such sharing, the model would have to be large enough to accommodate the individual models for all the classes. If the emission models for the different classes were distinct enough, a given sequence would only transition between the states representing its class and thus the distances between sequences belonging to different clusters would be infinite.

3.2.1 Relationships with similar methods

On some sense, the SSD distance can be seen to be similar in spirit to the Fisher kernel [Jaakkola and Haussler, 1998]. Both methods define the affinity between sequences by means of some kind of "projection" of the different sequences onto a common generative model. In the Fisher kernel, the gradients of the log-likelihoods $\nabla \log P(\mathbf{X}_i|\boldsymbol{\theta})$ around the parameters $\boldsymbol{\theta}$ (called *Fisher scores*) are used as the feature vector for sequence $\mathbf{X}_i$. This way, there is a close link between SSD and a Fisher kernel where only the scores for transition parameters are taking into account. One of the main problems of the Fisher kernel is its excessive dependence on the global model $\boldsymbol{\theta}$. Since Fisher scores are gradients around the parameters of the global model, the meaningfulness of those scores depends heavily on the shape of the log-likelihood at that point. This may be not too bad in a supervised scenarios, where the global model is learned to represent one of the classes or to maximize its discriminative power. However, in an unsupervised setting there is no simple a-priori way to know if a given θ will result in meaningful scores. This problem is alleviated in SSD, since the similarity is based on the transition parameters, which are re-estimated for each

sequence. This way, the local properties of the log-likelihood are not so relevant.

Finally, we would like to comment on the relationship between our proposal and that of [Ramoni et al., 2002]. There, the authors propose a bayesian clustering method based on transition matrices of Markov chains. They assume that the sequences are discrete, so a Markov chain can be directly estimated via transition counts. Our proposal, on the other hand, uses Markov chains on the latent variables (states), what makes it far more general. Moreover, our focus is on defining a general model-based distance between sequences, so that the SSD distance can be directly coupled with a wide range of clustering algorithms depending on the task at hand.

3.2.2 Extensions of SSD: Diffusion and Time Warping

Other approaches could be used in order to define distances between the different transition matrices. For example, instead of using $\tilde{\mathbf{A}}^n$ directly, an idea similar to diffusion distances [Szlam et al., 2008] could be applied by using different powers of the transition matrices $\left(\tilde{\mathbf{A}}^n\right)^t$, where t is a time index. This is equivalent to iterating the random walk defined by the transition matrices for t time steps. The j^{th} row of such an iterated transition matrix encodes the probabilities of transitioning from state j to each other state in t time steps. However, this approach would introduce the extra parameter t, which must be set very carefully. For example, many transition matrices converge very quickly to the stationary distribution even for low t (specially if the number of states is small). This could be a problem in cases where the stationary distributions for sequences in different clusters are the same. An example of such a scenario is presented in Section 3.3.

Moreover, the SSD distance measure is very flexible. Measuring distances between sequences is highly subjective and application dependent. For example, in a certain scenario we may not be interested in the rest time for each state, but only in the transitions (similar to Dynamic Time Warping [Sakoe and Chiba, 1978]). To this end, a good alternative would be to obtain the transition matrices $\tilde{\mathbf{A}}^n$ for every sequence, but ignore the self transitions in the distance measurement. That can be easily done by setting all the self-transitions to 0 and then re-normalizing the rows of the resulting transition matrices.

Algorithm 1 SSD distance for clustering sequential data

Inputs:

Dataset $\mathcal{S} = \{\mathbf{S}_1, \ldots, \mathbf{S}_N\}$, N sequences

K: Number of hidden states

Algorithm:

Step 1: Learning the global model (Baum-Welch)

$\Theta = \arg\max_{\theta'} P(\mathbf{S}_1, \ldots, \mathbf{S}_N | \Theta')$

Step 2: Estimating $\tilde{\mathbf{A}}^n = \left\{\tilde{a}_{ij}^n\right\}$ (Forward/Backward)

for all $\mathbf{S}_n$ **do**

$\alpha_k(t) = P(\mathbf{S}_n(1), \ldots, \mathbf{S}_n(t), q_t = k|\Theta)$

$\beta_k(t) = P(\mathbf{S}_n(t+1), \ldots, \mathbf{S}_n(T_n), q_t = k|\Theta)$

$\tilde{a}_{ij}^n \propto \sum_{t=1}^{T_n} \alpha_i(t) a_{ij} p(\mathbf{S}_n(t+1)|q_{t+1} = i)\beta_j(t+1)$

end for

Step 3: Obtaining the distance matrix $\mathbf{D} = \{d_{ij}\}$

for all i, j **do**

$\mathbf{p}_{ik} \equiv k^{th}$ row of $\tilde{\mathbf{A}}^i$

$d_{ij} = -\log \frac{1}{K} \sum_{k,k'=1}^{K} \sqrt{p_{ik}(k') p_{jk}(k')}$

end for

Step 4: Clustering using $\mathbf{D}$

3.3 Experimental Results

In this section we present a thorough experimental comparison between SSD and state of the art algorithms using both synthetic and real-world data. Synthetic data include an ideal scenario where the sequences in the dataset are actually generated using HMMs, as well as a control chart clustering task. Real data experiments include different scenarios (character, gesture and speaker clustering) selected from the UCI-ML [Frank and Asuncion, 2010] and UCI-KDD [Hettich and Bay,] repositories. We use the implementation of PPK available at the author's website `http://www1.cs.columbia.edu/~jebara/code.html`[1].

The compared methods for obtaining the distance matrix are: **SSD**, state-space dynamics clustering with Bhattacharyya distance; **PPK**, Probability Product Kernels [Jebara et al., 2007]; **KL**, KL-LL distance (see Chapter 2) [García-García et al., 2009c]; **BP**, BP metric [Panuccio et al., 2002]; **YY**, Yin-Yang distance [Yin and Yang, 2005] and **SYM**, Symmetrized distance [Smyth, 1997].

We denote the number of hidden states of the global model used by SSD as K, and the number of states per model of the methods that rely on training a HMM on each single sequence as K_m.

Once a distance matrix is available, we perform the actual clustering using the spectral algorithm described in [Ng et al., 2002]. The different distance matrices are turned into similarity matrices by means of a Gaussian kernel whose width is automatically selected in each case attending to the eigengap. Though more elaborated methods such as [Zelnik-Manor and Perona, 2004] can be used to select the kernel width, in our experiments it is automatically selected in each case attending to the eigengap since the experimental results are good enough. We assume that the number of clusters is known a priori. If this is not the case, automatic determination of the number of clusters can be carried out by methods such as those in [Sanguinetti et al., 2005, Zelnik-Manor and Perona, 2004].The PPK method directly returns a similarity matrix, that is first converted into a distance matrix by taking the negative logarithm of each element. Then, it is fed into the clustering algorithm

[1] Accessed on 26/07/2010

with automatic kernel width selection. The final k-means step of the spectral clustering algorithm is run 10 times, choosing as the final partition the one with the minimum intra-cluster distortion. The free parameter T_{PPK} of the PPK method is fixed to 10 following [Jebara et al., 2007].

The results shown in the sequel are averaged over a number of iterations in order to account for the variability coming from the EM-based training of the HMM.

3.3.1 Synthetic data

In this subsection we test the algorithms using two kinds of synthetically generated data: a mixture-of-HMMs (MoHMM) scenario as in Chapter 2, following [Smyth, 1997, García-García et al., 2009c], and a UCI-ML dataset representing control charts.

Mixture of HMMs

Each sequence in this dataset is generated by a mixture of two equiprobable HMMs θ_1 and θ_2. Each of these models has two hidden states, with an uniform initial distribution, and their corresponding transition matrices are

$$\mathbf{A}_1 = \begin{pmatrix} 0.6 & 0.4 \\ 0.4 & 0.6 \end{pmatrix} \qquad \mathbf{A}_2 = \begin{pmatrix} 0.4 & 0.6 \\ 0.6 & 0.4 \end{pmatrix}.$$

Emission probabilities are the same in both models, specifically $N(0,1)$ in the first state and $N(3,1)$ in the second. This is a deceptively simple scenario. Since both the emission probabilities and the equilibrium distributions are identical for both models, the only way to differentiate sequences generated by each of them is to attend to their dynamical characteristics. These, in turn, are very similar, making this a hard clustering task. The length of each individual sequence is uniformly distributed in the range $[0.6\mu_L, 1.4\mu_L]$, where μ_L is the mean length.

Figure 3.1 shows the clustering error achieved by the compared algorithms in a dataset of $N = 100$ sequences, averaged over 50 runs. All the algorithms use a correct model structure ($K_m = 2$ hidden states per class) to fit each sequence. For SSD, this implies using 4 hidden states for the common model ($K = 4$). Note that even better

results can be obtained by selecting $K = 2$, but that can be considered cheating since it implies knowing that the emission distributions are shared by the two generating models. As expected, when the sequences follow an HMM generative model and the representative model structure is chosen accordingly, SSD achieves impressive performance improvements for short sequence lengths. In contrast, algorithms that rely on training an HMM for each sequence suffer from poor model estimation when the mean sequence length is very low ($\leq$ 100), which in turn produces bad clustering results. Our proposal overcomes this difficulty by using information from all the sequences in order to generate the common representative HMM. Consequently, the emission probabilities are estimated much more accurately and the distances obtained are more meaningful, leading to a correct clustering. Nonetheless, when the sequences are long ($\geq$ 200) very accurate models can be obtained from each single sequence and the different methods tend to converge in performance.

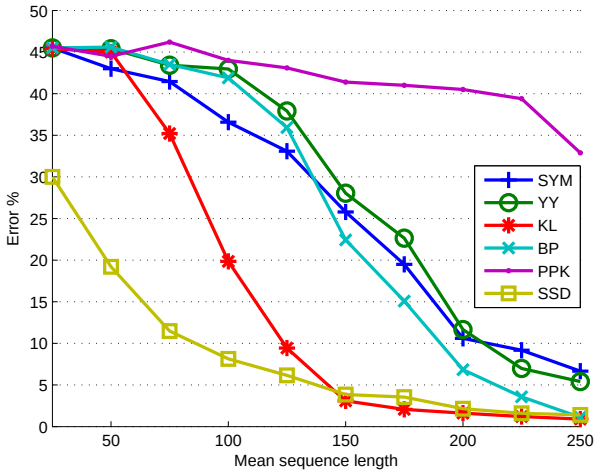


Figure 3.1: Clustering error for the MoHMM case

Synthetic Control Chart

We carry out a multi-class clustering task on this dataset (see Appendix D), partitioning the corpus into 6 groups which we expect to correspond with the different classesof control charts. As explained in Sec. 3.2, the size of the state-space for the

HMM in SSD clustering should be chosen accordingly to the number of classes, so we employ a number of hidden states in the range 12-40. It also allows us to show that our proposal is robust enough to produce good performance in such an extense range. Results, averaged over 10 runs, are shown in Table 3.1.

In general, all methods obtain good results, specially taking into account the multi-class nature of the dataset (a trivial clustering assigning all sequences to the same cluster would attain a 16.66% accuracy / 83.33% error). However, SSD clearly outperforms all the other compared algorithms. It is specially remarkable the very high performance achieved in the "large-dataset" of 100 sequences per class. Such good results are obtained because, in contrast to previous proposals, the modeling employed by SSD distance improves as the dataset size increases. It is worth noting how SSD provides better results than any other compared method even for the very modest number of 12 hidden states. This implies that there is a great amount of state-sharing between the different classes. Also of interest is the fact that the performance gets even better as that number of states is increased. We tried to push the limits of the method by using the largest number of hidden states that could be handled in a reasonable time (40 states), and even in that case we obtained very good results. This supports the idea that the size of the common model space is not a critical parameter of the algorithm. The confusion matrix when $N = 30$ and $K = 20$ (averaged over the 10 runs) is shown in Fig. 3.2 in the form of a Hinton diagram.

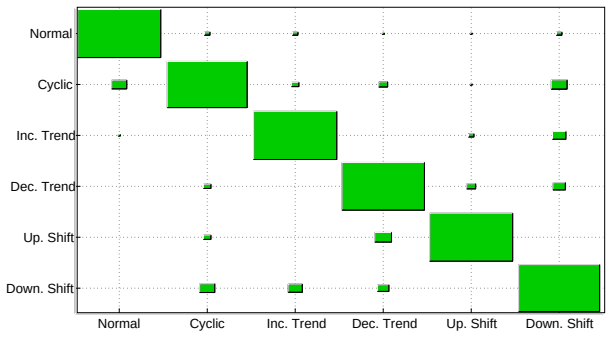


Figure 3.2: Confusion matrix for SSD clustering with 30 sequences per class and 20 hidden states

3.3.2 Real-world data clustering experiments

We use the following datasets from the UCI ML and KDD archives:

- Character Trajectories
- AUSLAN
- Japanese Vowels

Please refer to Appendix D for a description of the different datasets.

Table 3.2 shows the numerical results, averaged over 10 runs. In the Character Trajectories dataset, the individual sequences are fairly long and the classes are mostly well separated, so this is an easy task. Single sequences are informative enough to produce good representative models and, consequently, most methods achieve very low error rates. Nonetheless, using the SSD distance outperforms the competitors.

For the AUSLAN dataset, following [Jebara et al., 2007], we used HMMs with $K_m = 2$ hidden states for the methods that train a single model per sequence. The sequences were fed directly to the different algorithms without any preprocessing. We reproduce the results for the PPK method from [Jebara et al., 2007]. The common model for SSD employs $K = 4$ hidden states ($2K_m$), since the 2-way clustering tasks are fairly simple in this case. It is worth noting that the bad performance in the 'Yes' vs 'No' case is due to the fact that the algorithms try to cluster the sequences attending to the recording session instead of to the actual sign they represent. Our proposal produces great results in this dataset, surpassing the rest of the methods in every pairing except for the pathological 'Yes' vs 'No' case.

Finally, we carry out a 9-class speaker clustering task using the full Japanese Vowels dataset. There are 30 sequences per class, and the trivial clustering error baseline is 11.11%. The large number of classes and their variability demands a large number of hidden states in the common HMM of SSD. This, in turn, means a time penalty as the HMM training time is quadratic in the number of hidden states. Nonetheless, the performance obtained in this dataset by our proposal is very competitive in terms of clustering accuracy, only being surpassed by the KL

method. It is also remarkable how the SSD-based clustering exhibits a very stable performance in a huge range of state-space cardinalities, what once again coincides with our intuition that an accurate determination of that parameter is not crucial to the algorithm.

As the results confirm, using the SSD distance is very adequate in this scenario because the number of subsequences is quite large and, at the same time, all of them are very short. This way, we exploit both the reduced time complexity in the number of sequences and the better estimation of the emission distributions.

3.3.3 On the number of hidden states for SSD distance

A general conclusion that can be drawn from the experimental results is that the proposed distance measure generally increases its performance as the common state-space grows larger. This way, the size of the initial HMM can be chosen according to the available computational power. A performance loss is predictable at some point, although we have not witnessed it in our experiments, since there the limiting factor have always been the computational load. A possible way to test if the number of hidden states is getting too large for the problem at hand is to look at the mean occupancy (or stationary probability) of those states. If there are some of them which are occupied only for a very small fraction of the time, this can be seen as a sign that the common model is starting to overfit to particular characteristics of some individual sequences. In this case, the state space is too large and should be reduced.

Table 3.1: Mean clustering error (standard deviation in brackets) in the Control Chart dataset. Using (a) 30 and (b) 100 sequences per class.

K_m	SYM	YY	KL	BP	PPK	K	SSD
$K_m = 2$	23.89% (±0.1)	25.89% (±5.3)	25.33% (±5.3)	21.67% (±0.2)	53.11% (±3.3)	$K = 12$	28.29% (±6.0)
K_m=3	26.00% (±0.6)	25.33% (±0.5)	21.27% (±3.9)	24.11% (±3.6)	44.78% (±0.6)	$K = 16$	13.72% (±6.3)
K_m=4	25.22% (±0.6)	23.56% (±1.6)	20.67% (±3.1)	24.00% (±4.4)	48.78% (±4.8)	$K = 20$	12.67% (±4.1)
K_m=5	23.00% (±1.8)	20.89% (±3.2)	22.56% (±4.9)	20.22% (±1.6)	58.22% (±3.1)	$K = 28$	**11.81**% (±4.7)
K_m=6	25.33% (±0.3)	23.11% (±3.1)	23.78% (±2.9)	25.56% (±2.0)	59.89% (±3.5)	$K = 40$	14.17% (±2.7)

(a)

30 sequences per class

K_m	SYM	YY	KL	BP	PPK	K	SSD
K_m=2	32.34% (±4.35)	26.70% (±7.1)	29.81% (±0.08)	27.09% (±4.9)	54.21% (±1.8)	$K = 12$	8.43% (±1.5)
K_m=3	20.77% (±0.26)	14.47% (±3.2)	29.80% (±0.07)	22.67% (±0.6)	48.77% (±1.9)	$K = 16$	7.29% (±1.8)
K_m=4	20.72% (±0.7)	12.85% (±3.6)	27.77% (±3.0)	15.08% (±4.3)	46.85% (±3.7)	$K = 20$	6.77% (±1.4)
K_m=5	18.88% (±3.4)	15.34% (±4.9)	29.10% (±0.5)	17.65% (±2.5)	59.25% (±5.8)	$K = 28$	**5.93%** (±1.1)
K_m=6	21.37% (±5.5)	17.22% (±4.1)	16.05% (±4.5)	19.27% (±5.2)	60.85% (±1.7)	$K = 40$	6.40% (±1.0)

(b)

100 sequences per class

Table 3.2: Clustering error on the Character Trajectories (top), AUSLAN (middle) and JV (bottom, 9-class clustering task) datasets. Standard deviation of the results for the AUSLAN dataset is 0 except for 'SPEND' vs 'COST' using YY distance, with a value of 0.8. The number of hidden states is $K = 4$ for SSD and $K_m = 2$ for the rest of methods (best case)

# hidden stat.	SYM	YY	KL	BP	PPK	# hidden stat.	SSD
$K_m = 2$	3.90% (±0.4)	2.98% (±0.2)	3.56% (±0.1)	3.43% (±0.2)	23.28% (±0.8)	K=14	2.83% (±0.3)
$K_m = 3$	3.70% (±0.1)	3.10% (±0.2)	3.55% (±0.0)	4.77% (±0.3)	22.34% (±0.8)	K=16	2.42% (±0.2)
$K_m = 4$	4.69% (±0.2)	4.47% (±0.0)	4.31% (±0.1)	16.08% (±0.8)	37.75% (±0.2)	K=20	1.77% (±0.2)
$K_m = 5$	3.72% (±0.3)	3.60% (±0.1)	3.42% (±0.1)	15.30% (±0.1)	38.61% (±0.2)	K=22	**1.65%** (±0.2)

SIGNS	SYM	YY	KL	BP	PPK	SSD
'HOT' vs 'COLD'	**0%**	**0%**	**0%**	**0%**	**0%**	**0%**
'EAT' vs 'DRINK'	48.15%	7.41%	7.41%	7.41%	7%	**4.63**%
'HAPPY' vs 'SAD'	40.74%	1.85%	**0%**	1.85%	13%	**0%**
'SPEND' vs 'COST'	45.93%	0.37%	**0%**	**0%**	20%	**0%**
'YES' vs 'NO'	**39.64%**	45.45%	45.45%	45.45%	41%	45.45%

# hidden stat.	SYM	YY	KL	BP	PPK	# hidden stat.	SSD
$K_m = 2$	32.33% (±2.8)	14.89% (±2.1)	**9.85%** (±1.7)	14.70 % (±1.0)	24.52% (±3.7)	K=20	17.70% (±3.7)
$K_m = 3$	32.72% (±3.0)	20.99% (±4.1)	18.86% (±4.0)	24.56% (±3.7)	17.41% (±5.1)	K=30	14.52% (±4.5)
$K_m = 4$	32.04% (±3.4)	26.59% (±6.2)	14.45% (±5.4)	21.45% (±4.8)	20.22% (±3.7)	K=40	12.07% (±4.4)
$K_m = 5$	29.40% (±3.6)	17.19% (±5.2)	17.70% (±5.5)	20.23% (±4.4)	21.85% (±3.9)	K=50	23.63% (±6.9)

3.4 Summary

In this chapter we have presented a new distance for model-based sequence clustering using state-space models. We learn a single model representing the whole dataset and then obtain distances between sequences attending to their dynamics in the common state-space that this model provides. It has been empirically shown that the proposed approach outperforms the previous semi-parametric methods, specially when the mean sequence length is short. Furthermore, the proposed method scales much better with the dataset size (linearly vs quadratically). As drawback of this method it should be mentioned that, as the number of classes grow, the common model may need a large number of hidden states to correctly represent the dataset (although the method is empirically shown not to be too sensitive to the accurate determination of the model size). In the case of hidden Markov models, the time complexity of the training procedure is quadratic in this number of states, so total running time can be high in these cases. Consequently, we find our proposal specially appealing for scenarios with a large number of sequences coming from a few different classes, which is a very usual case.

Promising lines for future work include the application of this methodology to other state-space models, both discrete and continuous, and to semi-supervised scenarios. We are also investigating alternative definitions of distance measures between transition matrices in order to take into account the potential redundancy of the state-space.

Chapter 4

Sphere packing for clustering sets of vectors in feature space

We propose a simple method for clustering sets of vectors by packing spheres learnt to represent the support of the different sets. This can be done efficiently in a kernel-induced feature space by using the kernel trick. Our main assumption is that the supports of the distribution in that feature space present modest overlap.

4.1 Introduction and chapter structure

On previous chapters of this Thesis we have focused on using probabilistic models to elicit the dynamical characteristics of sequences of data. This allowed us to define model-based disimilarity measures that capture those dynamics. However, there are many scenarios where the sequences can be accurately classified or clustered without attending to their dynamical features. Examples include bag-of-words models for image analysis [Dance et al., 2004], speech-independent speaker verification [Reynolds, 2002],etc. In those scenarios, the statical probability distributions of the elements belonging to the different sequences (or, in this case, *sets of vectors*) are distinct enough, so a learning task can succeed without considering the dynamics.

In these cases the sequences can be viewed simply as (not strictly independent) samples from some underlying distributions, and can thus be characterized in terms

of the probability density functions (PDFs) of those distributions. The PDFs can be estimated in many ways depending on the application. For example, many works in bag-of-words modeling for topic analysis or image classification employ a simple histogram of the visual words [Dance et al., 2004], while for general continuous distributions the most extended model is arguably the Gaussian mixture model (GMM) [Bishop, 2006]. Probabilistic models need a fine-tuning of their parameters for them to be effective, and in many cases that can be a hard problem in itself. Moreover, it is well known that when the input space is high dimensional and/or the number of samples per sequence is low, the estimation of probabilistic models is likely to be an ill-posed problem.

In many cases the input space does not provide an adequate representation of the data, so it is beneficial to work in an alternative feature space. Inspired by this, some works define affinity measures between sequences in a feature space as a combination of individual vector kernels. This idea was first presented in the field of speaker verification in [Campbell et al., 2006], where an explicit representation of the feature space vectors was required. In contrast, [Louradour et al., 2007] propose a modification where the affinity can be obtained only in terms of inner products in feature space. This allows for the use of the kernel trick [Schoelkopf and Smola, 2001] in order to perform implicit expansions into a high (possibly infinite) dimensional feature space via a kernel function. The drawback of this method is that it requires a set of labeled sequences in order to learn the covariance matrices defining each class, so it is not directly applicable in an unsupervised scenario.

In this chapter we aim at providing a direct non-parametric feature-space clustering of different sets of vectors, in contrast with the usual two-stage (obtain a distance between sets and then apply a standard clustering algorithm) approach. To this end, we employ recent ideas in the field of support estimation for distributions in a reproducing kernel Hilbert space (RKHS) [Shawe-Taylor and Cristianini, 2004]. Specifically, we model the support of each set of vectors via its minimum-enclosing sphere in feature space. In essence, we substitute the usual density estimation for the much simpler support estimation procedure. The goal of the clustering step is two find a certain number of hyperspheres (correspoding with the desired number of

clusters) encompassing the whole dataset with the smallest total radius. Since this is computationally a hard problem, we use a greedy approximation carried out by iteratively finding the smallest sphere encompassing two of the existing ones. As we show in the paper, this procedure is very efficient due to the sparsity of the solution to the support estimation and can work seamlessly in kernel-induced feature spaces of arbitrary dimensions. The method can be interpreted as a hierarchical-clustering scheme, and can naturally handle multi-class clustering.

This chapter is structured as follows: First, Section 4.2 provides a brief overview of existing distance/affinity measures for sets of vectors. Then, Section 4.3 contains the theoretical background of the support estimation algorithms. A detailed description of our proposed clustering algorithm can be found in Section 4.4. Empirical performance evaluation is shown in Section 4.5, and finally Section 4.6 summarizes the contributions presented in the chapter and sketches some lines for further research.

4.1.1 Related publications

This chapter is mainly based on [García-García and Santos-Rodríguez, 2011].

4.2 Distance measures for sets of vectors

Assume we are given a dataset $\mathcal{S} = \{\mathbf{X}_1, \ldots, \mathbf{X}_N\}$ of bags-of-vectors, with $\mathbf{X}_i = \left[\mathbf{x}_1^{(i)}, \ldots, \mathbf{x}_{N_i}^{(i)}\right]$. Different sequences may present different number of elements N_i. All the individual vectors $\mathbf{x}_j^{(i)}$ of every set live in the same input space $\mathcal{X}$.

4.2.1 General model-based distances

Firstly, let us make clear that the model-based approaches shown in Chapter 2 can be directly translated to a set-of-vectors scenario by substituting the dynamical model (HMM) for a "statical" counterpart, such as MoG models. This is also the case with the probability product kernel (PPK) (see Chapter 3, Sec. 3.1), which is just a generalized inner product of probability densities. Specifically, the PPK between two Gaussian mixture models with $\rho = 1$ corresponds with the expected likelihood (EL) kernel, and can be obtained in closed form [Lyu, 2005]. Assuming we have a

pair of k-component GMMs with parameter sets θ_a and θ_b, their EL kernel can be written as:

$$\kappa_{\mathrm{EL}}(\theta_a, \theta_b) = \int_{\mathbb{R}^d} p(\mathbf{z}|\theta_a) p(\mathbf{z}|\theta_b) d\mathbf{z} = (2\pi)^{-d/2} \mathbf{w}_a^T \Gamma \mathbf{w}_b,$$

where $\mathbf{w}_a$ is a column vector containing the weights of the different components of θ_a and Γ is a $k \times k$ symmetric matrix whose ij^{th} element is given by integration of the product of the i^{th} Gaussian component of θ_a and the j^{th} component of θ_b. For a pair of Gaussian distributions (μ_1, C_1) and (μ_2, C_2), that integration can be written in closed form as

$$g(\mu_1, C_1, \mu_2, C_2) = \frac{|C|^{\frac{1}{2}} \exp\left(\frac{1}{2}\mu^T C^{-1} \mu\right)}{\prod_{k=1}^{2} |C_k|^{\frac{1}{2}} \exp\left(\frac{1}{2}\mu_k^T C_k^{-1} \mu_k\right)},$$

where $\mu = C_1^{-1}\mu_1 + C_2^{-1}\mu_2$ and $C = (C_1^{-1} + C_2^{-1})^{-1}$.

4.2.2 Feature space methods

In many cases, the input space does not provide an adequate representation of the data for learning purposes. In those cases, it may be beneficial to project the input vectors into a higher dimensional space in order to increase the separability of the classes. This can be done via an embedding function $\Phi : \mathbf{x} \to \Phi(\mathbf{x})$, as shown in the Generalized Linear Discriminant Sequence kernel (GLDS) of [Campbell et al., 2006]. This method defines a kernel between sets of vectors as a rescaled dot product between average polynomial expansions. The scaling is given by the second-moment matrix of the polynomial expansions, estimated on some background population.

Working with explicit feature vectors can be intractable as the size of the feature space grows. In order to work in very high (possibly infinite) dimensional feature spaces, the *kernel trick* [Schoelkopf and Smola, 2001] is usually employed. If an algorithm can be expressed in terms of inner products of feature space vectors, the kernel trick consists in substituting these inner products for evaluations of a kernel function $\kappa(\mathbf{x}, \mathbf{y})$ on the corresponding input vectors. If this function satisfies the Mercer conditions, it represents an inner product in an induced feature space. In [Louradour et al., 2007], the authors modify the idea of the GLDS in a way that it can be applied to any feature space defined in terms of a kernel function. For

a comparative evaluation of these and other specific kernels in the field of speaker verification, please refer to [Daoudi and Louradour, 2009].

Probabilistic distances can also be defined in feature space. In [Zhou and Chellappa, 2006], the authors show how some of the most usual distances between probability distributions can be obtained in the space induced by a given kernel function. They assume that the data vectors are Gaussian-distributed in the RKHS, and that they follow a factor analysis model. This is key in order to reduce the effective dimension and avoid singular covariance matrices. Thus, this kind of methods relies heavily on the right choice of the latent space dimensionality, which is a hard problem in itself.

Some recent works have analyzed the embedding of distributions in a reproducing kernel Hilbert space (RKHS). Specifically, these embedding have been used to define tests for checking independence [Gretton et al., 2005] or for addressing the two-sample problem [Gretton et al., 2007]. Many of these tests use a statistic called the maximum mean discrepancy (MMD), which is defined as follows:

Definition (Maximum Mean Discrepancy) Let $\mathcal{F}$ be a class of functions $f : \mathcal{X} \to \mathbb{R}$ and let P and Q be two Borel probability measures defined on the domain $\mathcal{X}$. The MMD is defined as

$$\text{MMD}\left[\mathcal{F}, P, Q\right] = \sup_{f \in \mathcal{F}} \left(\mathbb{E}_P[f(x)] - \mathbb{E}_Q[f(y)]\right)$$

When $\mathcal{F}$ is the unit ball in a reproducing kernel Hilbert space (RKHS) $\mathcal{H}$, the MMD can be computed very efficiently, as stated in the following lemma from [Gretton et al., 2007]:

Lemma 4.2.1. *Denote* $\mu_P = \mathbb{E}_P\left[\Phi(\mathbf{x})\right]$. *Then* $MMD[\mathcal{F}, P, Q] = ||\mu_P - \mu_Q||_{\mathcal{H}}$,

Moreover, if the RKHS $\mathcal{H}$ is universal then MMD is a metric, allowing us to identify $P = Q$ uniquely [Gretton et al., 2007]:

Theorem 4.2.2. *Let* $\mathcal{F}$ *be a unit ball in a universal RKHS* $\mathcal{H}$. *Then* $MMD[\mathcal{F}, P, Q] = 0$ *if and only if* $P = Q$

To the best of our knowledge, the MMD has not been used as a distance measure for clustering sets of vectors.

4.3 Support estimation via enclosing hyperspheres

A recent algorithm for novelty detection [Shawe-Taylor and Cristianini, 2004] consists in enclosing the training data in a hypersphere and then for each new point checking whether that point is inside or outside the hypersphere. This way, the sphere acts as an estimation of the *support* of the underlying distribution.

Support of a distribution Let P be a probability measure over a space $\mathcal{X}$, and $S_{x,\epsilon}$ the closed ball of radius ϵ centered at x. Its support is defined as the collection of all $x \in \mathcal{X}$ with $P(S_{x,\epsilon}) > 0$ for all $\epsilon > 0$.

Assuming that points in the training set are i.i.d., the most obvious choice for such an hypersphere would be one centered at the mean of their distribution. However, this is not the smallest possible one. Moreover, if we are working with sequences whose dynamics have been discarded, independence is too much of an assumption, which further discourages the use of the sample mean.

Assume we are given a set of points $\mathbf{X} = \{\mathbf{x}_i, \ldots, \mathbf{x}_j\}$ with an associated embedding $\boldsymbol{\phi}$ into a feature space with associated positive semi-definite kernel $\kappa(\mathbf{x}, \mathbf{z}) = \langle \boldsymbol{\phi}(\mathbf{x}), \boldsymbol{\phi}(\mathbf{z}) \rangle_{\mathcal{H}}$, where $\mathcal{H}$ is the corresponding Hilbert space. The problem of finding the minimum sphere that contains all the embedded points can be written as:

$$\min r^2 \quad \text{s.t.} \quad ||\boldsymbol{\phi}(\mathbf{x}_i) - \mathbf{c}||^2 \leq r^2, \quad i = 1, \ldots, N_i, \tag{4.1}$$

where $\mathbf{c}$ represents the center of the optimal hypersphere, and r its radius. This is a constrained optimization problem, which can be solved by introducing a Lagrangian involving one Lagrange multiplier α_i for each constraint. Then, it is easy to solve for those multipliers by setting the derivatives of the Lagrangian w.r.t. $\mathbf{c}$ and r equal to 0, yielding

$$\sum_{i=1}^{n} \alpha_i = 1 \Rightarrow \mathbf{c} = \sum_{i=1}^{n} \alpha_i \boldsymbol{\phi}(\mathbf{x}), \tag{4.2}$$

which shows that the center of the optimal sphere always lie on the span of the input data points, so it can be expressed in the dual representation. Substituting back in

Eq. (4.1) we get

$$L(\mathbf{c}, r, \boldsymbol{\alpha}) = \sum_{i=1}^{n} \alpha_i \kappa(\mathbf{x}_i, \mathbf{x}_i) - \sum_{i,j=1}^{n} \alpha_i \alpha_j \kappa(\mathbf{x}_i, \mathbf{x}_j), \tag{4.3}$$

which is a convex problem (since the matrix with entries $\kappa(\mathbf{x}_i, \mathbf{x}_j)$ is positive semi-definite for any dataset) where the optimization is subject to the non-negativity and sum to one conditions of the Lagrangian coefficients $\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_{N_i}]^T$.

By virtue of the Karush-Kuhn-Tucker (KKT) conditions [Boyd and Vandenberghe, 2004], the only non-zero Lagrange multipliers $\boldsymbol{\alpha}$ will be those corresponding to points that lie on the surface of the hypersphere. This implies the very desirable property of sparsity of the solution, in the same vein as support vector machines (SVMs) [Shawe-Taylor and Cristianini, 2004]. Due to this analogy, we will refer to the input points with non-zero coefficients as support vectors (SVs).

This algorithm is very sensitive to outliers, since we are forcing the hypersphere to enclose all the points in the dataset. To alleviate this, "soft" versions of the support estimation procedure can be defined by using slack variables that penalizes points outside the hypersphere. The resulting optimization problem is practically equivalent to Eq. (4.3), but substituting the non-negativity constraint of the Lagrange multipliers for the more restrictive one $0 \leq \alpha_i \leq C$, where C is a penalty parameter. Since the sum-to-one constraint still holds, this means that penalties $C > 1$ are meaningless.

Moreover, this restriction has a natural interpretation in terms of sparsity. Given a penalty parameter C, the center of the optimal sphere must be be represented by at least $\lceil 1/C \rceil$ points with strictly positive Lagrange multipliers. This way, the trade-off between the robustness to outliers and the sparsity of the solution is made explicit.

4.4 Clustering sets of data by sphere packing

Suppose we have a dataset $\mathcal{S} = [\mathbf{X}_1, \ldots, \mathbf{X}_m]$ of m sets of data (samples). We want to divide that dataset into K disjoint groups (clusters), with samples belonging to a given cluster being similar to each other and different from samples in other

clusters. Obviously, this is a very subjective definition, totally dependent on the choice of metric with which the similarity between the sequences is measured. We propose to view each individual set $\mathbf{X}_i$ as an i.i.d. sample from some underlying probability distribution P_i, and then consider the overlap between the support of the different distributions as a measure of similarity between the corresponding sets. To that end, we will employ the support estimation methods presented in the previous section. This way, on the first step of the algorithm each sample $\mathbf{X}_i$ is associated with an hypersphere $S^{(i)}$ in feature space, defined by its radius $r^{(i)}$ and its center $\mathbf{c}^{(i)}$. Those individual spheres can be learned using the hard or soft support estimation procedures discussed in the previous section, depending on the scenario. If our data is likely to present outliers, the soft formulation will yield better results because of its robustness.

The goal of the algorithm is to obtain a set of K spheres $S^{(1)}, \ldots, S^{(K)}$(one per final cluster), each one of them formed by "packing" together the original spheres. To this end, we propose a hierarchical recursive sphere-merging mechanism. At each step the algorithm checks what pair of spheres can be fused using a minimum-radius sphere. The assumption behind this algorithm is that the support of different classes do not overlap much. This may be seen as a too restrictive assumption in input space, but is quite sensible in a rich feature space such as those induced by Gaussian kernels. Moreover, by restricting our attention to the support of the distribution we can get much better estimates for small sample size. Estimating a full distribution requires a large number of samples to be accurate (specially in high dimensional settings), while the support estimation is a much simpler problem and can thus be accurately solved given small sample sizes.

The crucial part of the algorithm is the sphere-packing procedure. We will illustrate this step in the first iteration, when we have the original spheres $S^{(1)}, \ldots, S^{(m)}$. The goal of this step is to find the smallest (in terms of radius) sphere $S^* = S^{(i^*,j^*)}$ from all the $S^{(i,j)}$, where $S^{(i,j)}$ is the smallest sphere encompassing $S^{(i)}$ and $S^{(j)}$. So the problem now remains how to obtain each $S^{(i,j)}$. Obviously, if some $S^{(i)}$ is contained within another $S^{(j)}$, then $S^{(i,j)} = S^{(j)}$. If that is not the case, $S^{(i,j)}$ can be obtained in a simple geometrical way, as depicted in Fig. 4.1. The center $\mathbf{c}^{(i,j)}$ of

the sphere will be the middle point of the segment going through the two existing centers and joining the outer edges of the spheres, and its radius $r^{(i,j)}$ can be simply written as

$$r^{(i,j)} = \frac{1}{2}\left(d_{i,j} + r^{(i)} + r^{(j)}\right). \tag{4.4}$$

The most important aspect of this construction is that it can be carried out in a kernel-induced feature space. To this end we first need to find the distance $d_{i,j}$ between each pair of centers $\mathbf{c}^{(i)}$ and $\mathbf{c}^{(j)}$. Since each center is a point in a possibly infinite feature space, we represent them, by virtue of Eq. (4.2), by the corresponding dual coefficients $\boldsymbol{\alpha}^{(i)}$ and $\boldsymbol{\alpha}^{(j)}$. This way, we can write:

$$\begin{aligned} d_{i,j} &= ||\mathbf{c}^{(i)} - \mathbf{c}^{(j)}||_{\mathcal{H}} = \sqrt{||\mathbf{c}^{(i)}||^2_{\mathcal{H}} + ||\mathbf{c}^{(j)}||^2_{\mathcal{H}} - 2\left\langle \mathbf{c}^{(i)}, \mathbf{c}^{(j)} \right\rangle_{\mathcal{H}}} \\ &= \sqrt{\boldsymbol{\alpha}^{(i)T}\mathbf{K}^{(i)}\boldsymbol{\alpha}^{(i)} + \boldsymbol{\alpha}^{(j)T}\mathbf{K}^{(j)}\boldsymbol{\alpha}^{(j)} - 2\boldsymbol{\alpha}^{(i)T}\mathbf{K}^{(i,j)}\boldsymbol{\alpha}^{(j)}}, \end{aligned} \tag{4.5}$$

where $\mathbf{K}^{(i)}$ and $\mathbf{K}^{(j)}$ are the kernel matrices of the support vectors defining $\mathbf{c}^{(i)}$ and $\mathbf{c}^{(j)}$, respectively, and $\mathbf{K}^{(i,j)}$ is the corresponding crossed kernel matrix between the two sets of support vectors. Since the support estimation solution is sparse, the number of support vectors is expected to remain low in comparison with the total number of samples, and thus these kernel calculations are computationally very cheap. Moreover, the support set does not change in further steps, so there is no need to re-calculate the kernel values. The radius of the corresponding encompassing sphere can be obtained using Eq. (4.4), and thus the optimal pair (i^*, j^*) of spheres to merge can be selected accordingly.

Now we need a way to represent the center of the new sphere for further iterations. Noting that

$$\mathbf{c}^* = \mathbf{c}^{(i^*)} - \left(r^* - r^{(i^*)}\right)\frac{\mathbf{c}^{(i^*)} - \mathbf{c}^{(j^*)}}{||\mathbf{c}^{(i^*)} - \mathbf{c}^{(j^*)}||} \tag{4.6}$$

and recalling Eq. (4.2), it is easy to see that $\mathbf{c}^*$ lies in the span of $SV^* = SV(i^*) \cup SV(j^*)$, where $SV(i)$ is the set of support vectors of the sphere learned on the i^{th} sample. The corresponding coefficients for that support vector expansion are given by:

$$\boldsymbol{\alpha}^* = \begin{bmatrix} \boldsymbol{\alpha}^{(i^*)} \\ \mathbf{0} \end{bmatrix} - \frac{\left(r^* - r^{(i^*)}\right)}{d_{i^*,j^*}} \begin{bmatrix} \boldsymbol{\alpha}^{(i^*)} \\ -\boldsymbol{\alpha}^{(j^*)} \end{bmatrix}. \tag{4.7}$$

Then, $S^{(i^*)}$ and $S^{(j^*)}$ are eliminated from the pool of spheres to merge, while S^* is added to that pool. This step will be repeated until there are only K remaining spheres. Sets whose support have been embedded into the same sphere will belong to the same cluster. The process, using a linear kernel, is shown in Fig. 4.2, starting with the original samples, then performing support estimation on each individual sample and finally packing the individual spheres until there are only as many left as clusters we are looking for.

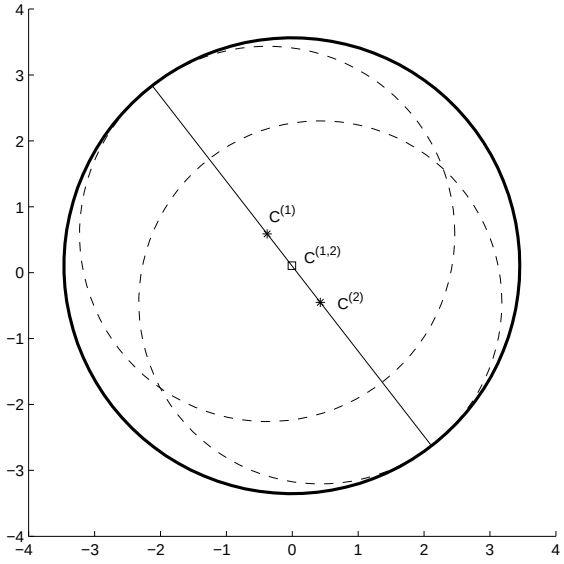
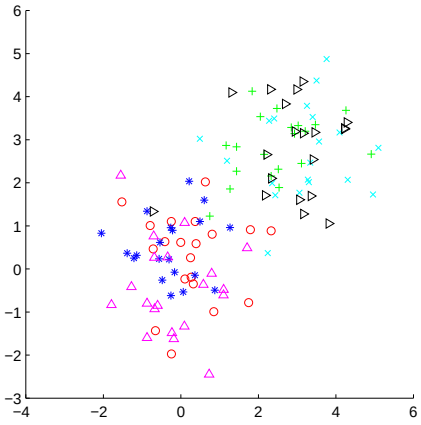


Figure 4.1: $S^{(1,2)}$ is the smallest encompassing sphere of $(S^{(1)}, S^{(2)})$

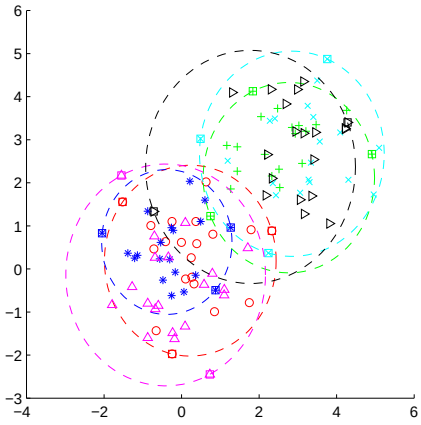
Note that the smallest sphere encompassing $S^{(i)}$ and $S^{(j)}$ is not necessarily the smallest sphere encompassing samples $\mathbf{X}_i$ and $\mathbf{X}_j$. We build all the packing procedure upon the support estimates instead of on the samples because this way the process becomes computationally much lighter, since the optimal packings can be obtained in close form instead of having to solve a QP problem for each pair of samples at each step. Moreover, the slack introduced by this approximation has been

empirically shown to be quite small.

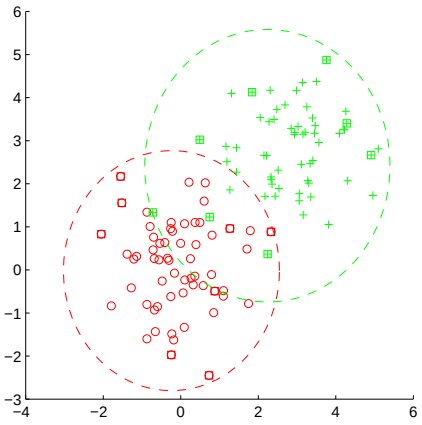
The clustering procedure is summarized in Alg. 2. Note that the sphere-packing algorithm can be easily adapted to a semi-supervised learning setting. In that scenario, we are given a (usually small) set of labeled points, together with a (generally larger) set of unlabeled points. Sphere-packing can work seamlessly on such a setting by grouping all the points with each given label as separate sets (and then learn the corresponding spheres), and viewing the unlabeled points as 1-element sets, so their corresponding spheres will have a null radius and a center given by the point itself. If this reasoning is taken to the extreme, we arrive at the standard clustering problem of individual data points. Sphere-packing in that case reduces to an agglomerative variant of hierarchical clustering.



(a) Original data, 6 sets



(b) Individual support estimates



(c) Final spheres, 2 clusters

Figure 4.2: Sphere packing procedure: Each original set is represented by its smallest-encompassing sphere, and the resulting spheres are iteratively packed until there are only K spheres remaining.

Algorithm 2 Hierarchical sphere-packing for clustering sets of vectors

Inputs:

Dataset $\{\mathbf{X}_1, \ldots, \mathbf{X}_N\}$ of N sets of vectors

Desired number of clusters K, kernel parameters

Algorithm:

Step 1: Support estimation

$\mathbf{S} = []$

for all $\mathbf{X}_i$ **do**

$S_i = (r^{(i)}, \boldsymbol{\alpha}^{(i)})$: Smallest sphere containing $\mathbf{X}_i$, learnt using Eq. (4.3) or similar.

$\mathbf{S} \leftarrow \mathbf{S} \cup S_i$

end for

Step 2: Sphere-packing

while $|\mathbf{S}| > K$ **do**

$i^*, j^* = \arg\min_{i,j} \frac{1}{2}\left(d_{i,j} + r^{(i)} + r^{(j)}\right)$

$r^* = \min \frac{1}{2}\left(d_{i,j} + r^{(i)} + r^{(j)}\right)$

$$\boldsymbol{\alpha}^* = \begin{bmatrix} \boldsymbol{\alpha}^{(i^*)} \\ \mathbf{0} \end{bmatrix} - \frac{\left(r^* - r^{(i^*)}\right)}{d_{i^*,j^*}} \begin{bmatrix} \boldsymbol{\alpha}^{(i^*)} \\ -\boldsymbol{\alpha}^{(j^*)} \end{bmatrix}.$$

$S^* = (r^*, \boldsymbol{\alpha}^*)$

$\mathbf{S} \leftarrow \mathbf{S} \cup S^*$

$\mathbf{S} \leftarrow \mathbf{S} \backslash (S^{i^*}, S^{j^*})$

end while

4.5 Experimental results

In this section we evaluate the performance of the sphere-packing clustering algorithm (SPH) using both synthetic and real-world datasets. We compare our method with a two-step approach combining Maximum Mean Discrepancy (MMD) [Gretton et al., 2007] as a distance measure between sequences/samples and a normalized-cut spectral clustering algorithm [von Luxburg, 2007], both methods being state-of-the-art in their respective fields. Moreover, we also compare with the KL-LL model-based approach described in Chapter 2. Taking advantage of the flexibility of the method, we will use both a single Gaussian (KL-Gauss) or a two-components Gaussian mixture (KL-GMM) as the generative model, apart from the HMM case (KL-HMM). We also introduce into the comparison the Expected Likelihood (EL) kernel described in Section 4.2.1, using two-components GMMs. Results are shown in the form of clustering error, understood as the corresponding sequence-wise classification error under an optimal permutation of the labels. All the experiments use Gaussian kernels with a width parameter σ automatically selected as the median distance between points in the dataset, which is a poular heuristic [Schoelkopf and Smola, 2001]. For support estimation, we use the soft version of the algorithm, with a penalty parameter $C = 0.05$.

In the synthetic case, we want to cluster samples from two different distributions. Both distributions are 2-D zero-mean Gaussians, with covariance matrices $C_1 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$ and $C_2 = \begin{pmatrix} 1.3 & 0 \\ 0 & 1 \end{pmatrix}$, respectively. The dataset is comprised of 50 samples, each one of them of a size randomly selected between 80 and 120. That conforms scenario (a). In scenario (b), the samples are contaminated with a 10% of outliers coming from a Gaussian distribution with mean $[55]^T$ and identity covariance matrix. Table 4.1 shows the results, averaged over 50 simulations. In absence of outliers, the Gaussian distribution approach obtains the better results. That is sensible, since data actually follows a Gaussian distribution. However, the non-parametric methods are much more robust in the presence of outliers, even with the inclusion of a second mixture component in the GMM model to account for the outlying points. The SPH algorithm is even able to outperform the gold-standard

Table 4.1: Clustering error for the synthetic scenarios

	SPH	MMD	KL-Gauss	KL-GMM (m=2)	EL
(a)	20.05%	24.10%	16.50%	17.10%	24.20%
(b)	21.90%	25.30%	42.30%	40.00%	40.50%

Table 4.2: Clustering error for the speaker clustering tasks

Dataset	SPH	MMD	KL-Gauss	KL-GMM	KL-HMM	SSD	EL
JV	10.00%	20.37%	16.37%	15.93%	9.85%	12.07%	18.33%
GPM	10.56%	9.57%	15.22%	10.84%	9.65%	10.71%	15.10%

combination of MMD+SC.

As an example of a real-world application, we tackle two speaker clustering tasks. It is quite a natural choice, since it is a task where the dynamics of the sequences are usually discarded. We have tried both the Japanese Vowels (JV) and GPM datasets (see Appendix D). We simulate a 9-class speech-independent speaker clustering task with the JV dataset, and a 2-class task using the GPM dataset. Results are shown in Table 4.2, showing that SPH achieves a much better performance than the other methods which use only "static" information in the JV dataset. It is only surpassed by the best-performing method from previous comparisons: the KL-LL distance with HMM as generative model. The SPH algorithm also performs at a good level in the GPM dataset. The performance of MMD is excellent in the two-class clustering task, but very poor in the 9-class task. A more challenging scenario will be presented in Chapter 6.

4.6 Summary

We have presented a new approach for clustering sets of vectors, based on support estimation in a kernel-induced feature space. Starting with the hyperspheres approximating the support of each individual set, we cluster them by iteratively packing together the pair of spheres that results in the smallest encompassing sphere. Empirical results are promising and show that the method is competitive with the state of the art.

Chapter 5

Risk-based affinities for clustering sets of vectors

We investigate the definition of similarity between sets of data based on their separability, as measured by the risk of some classifier. This idea is further developed and linked with the framework of f-divergences. Two generalizations of this family of divergences are proposed and studied: CRFDs are based on restricting the class of allowable of classification functions and Loss-Induced divergences are based on the use of surrogate losses instead of the 0-1 loss. As a first example of this theory, we use loss-induced divergences to define both a new estimator and bound for the well-known Kullback-Leibler divergence

5.1 Introduction and chapter structure

In this chapter we are once again concerned with the sets-of-vectors scenario, that is to say, the case where the dynamics of the sequences are discarded. While in Chapter 4 we developed a support-based approach to clustering, here we will present more general and expressive methods. The main intuition behind our results is the identification of the affinity of a pair of sets of vectors with how hard to separate those sets are. Separation can be quantified with the help of binary classifiers, opening a wide range of possibilities for defining classifier-based affinity functions.

Intuitively, the similarity between two sets $\mathbf{X}$ and $\mathbf{Y}$ can be defined as a function of how *separated* the sets are. It is quite natural to measure the amount of separation between two sets with the help of a classifier. In the simplest case, this idea reduces to training a a classifier that is supposed to separate the points $\mathbf{X}$ from $\mathbf{Y}$ and use its error rate as a similarity score. Intuitively, if the sets $\mathbf{X}$ and $\mathbf{Y}$ "overlap a lot", then the classifier will have a high error rate, which we interpret as a high similarity. If, on the other hand, $\mathbf{X}$ and $\mathbf{Y}$ are well separated from each other, the classifier will achieve a low error, leading to a low similarity score. This can be understood as a *discriminative* approach to measuring affinity, in contrast to the standard methods. A particularity of error-based affinity measures is that they make no distinctions between distributions that can be perfectly separated. Consider, for example, a pair of uniform distributions $P = U[0, 1]$ and $Q = U[2, 3]$. Any sensible classifier will be able to separate samples from those two distributions perfectly, since their support is disjoint, and thus their affinity will be 0. The exact same will happen if $Q = U[10^3, 10^4]$ or $Q = U[1.01, 2]$, although one may intuitively assume that the latter can be somewhat "closer" to P than the former. By restricting our affinity measure to be risk-based, we are effectively considering that two distributions that can be perfectly separated are totally dissimilar. This has the desirable effect of bounding the affinity measure.

Thus, by using classifiers it is possible to obtain useful information about the affinity of the sets of vectors without having to explicitly fit probabilistic models to the individual sequences. Learning this kind of models in high dimensional spaces or when a sequence is very short is an ill-posed problem, requiring very careful regularization procedures. This may hinder their practical application in scenarios such as kernel-induced high dimensional feature spaces. In contrast, learning classifiers in such spaces is usually a straightforward procedure.

The choice of the classifier to use as a basis for defining affinities can be motivated by domain knowledge. For example, a practitioner may know that, in his field, linear classifiers (or SVMs with a given kernel) work well. It is then natural to use those classifiers to induce an affinity function for clustering purposes, providing a simple and intuitive way to leverage this knowledge. However, such an approach is clearly

suboptimal if there is no such prior knowledge. In those cases it would benefitial to use very flexible class of functions in order to capture all the possible information. For example, the Nearest Neighbor (NN) rule provides a simple classifier (or, strictly, classification rule) that exhibits many interesting properties for our purposes. The resulting classification functions are inherently non-linear, capturing arbitrary shapes of the distributinons, its error rate can be estimated both easily and efficiently and it is amenable to theoretical analysis. For these reasons, we will use it as a basis for many of the results in this chapter.

The idea of risk-based affinities can be cast naturally in the framework of f-divergences [Ali and Silvey, 1966, Csiszár, 1967]. Many well-known divergence measures can be seen to be members of this family, which is closely related to Bayes classification errors. There are two main paths to further generalize the concept of f-divergences for our purposes: on the one hand, it is possible to define divergences based on restricted sets of classification functions. On the other hand, losses othar than the 0-1 loss associated to Bayes errors can be considered. Moreover, these two paths can be combined, yielding a very general and appealing notion of classifier-induced divergences.

This chapter is structured as follows: we start in Section 5.2 by introducing the intuitions and some formalities regarding the notion of classifier-based affinities. Section 5.2.1 is devoted to the analysis of the NN error as an affinity function. We then present the foundations for more involved risk-based measures in Section 5.3, introducing the well-known family of f-divergences. In Section 5.4 we present a first generalization of f-divergences, based on the idea of restricting the class of permissible classification functions, and explore the most interesting theoretical properties of these generalized divergences. Section 5.5 deals with another possible generalization, this time related to surrogate Bayes risks. We present some theoretical results about this loss-induced divergences, showing their most appealing properties as well as the deep connections with standard f-divergences. Together with a result linking NN error and a certain surrogate Bayes risk, we use the developed theory to obtain new estimators for the Kullback-Leibler divergence.

5.1.1 Related publications

Parts of this chapter appear in [García-García et al., 2011b].

5.2 Classifier-based affinity measures

Let us now conceptualize the classifier-based framework more abstractly. Consider two probability distributions P, Q on the same space $\mathcal{X}$ and their convex combination $M := \pi P + (1-\pi)Q$ for some weight parameter $\pi \in [0,1]$. Assign labels $+1$ to all points that have been drawn from P, and labels -1 to all points drawn from Q. The classification task (π, P, Q, l) consists in finding the optimal classification function $\hat{y} : \mathcal{X} \to \mathcal{V}$, $\mathcal{V} \subseteq \mathbb{R}$, for this setting under a given loss function $l : \{0,1\} \times \mathcal{V} \to \mathbb{R}$. That is to say, find the function $\hat{y}$ minimizing the *risk* under the loss l

$$\mathbb{L}_l(\hat{y}, \pi, P, Q) = \mathbb{E}_M[\eta(x) l(1, \hat{y}(x)) + (1-\eta(x)) l(0, \hat{y}(x))], \tag{5.1}$$

where

$$\eta = P(Y = 1 | X = x) = \pi \frac{dP}{dM} \tag{5.2}$$

is the *posterior class probability function*. In case densities $p(x), q(x)$ exist, $\eta(x) = \frac{\pi p(x)}{\pi p(x) + (1-\pi) q(x)}$. We will define notions of affinity based on optimal risks

$$\mathbb{L}_l^C(\pi, P, Q) = \min_{\hat{y} \in C} \mathbb{L}_l(\hat{y}, \pi, P, Q) \tag{5.3}$$

where C is a given set of functions. In case we allow any possible classification function and choose the 0-1 loss

$$l_{0-1}(Y, \hat{y}) = I_{Y \neq \hat{y}}, \tag{5.4}$$

where I stands for the indicator function, this similarity score becomes the *Bayes error* of the classification task.

$$\underline{\mathbb{L}}_{0-1}(\pi, P, Q) = \mathbb{E}[\min(\eta(x), 1-\eta(x))], \tag{5.5}$$

We use the underline to denote an optimal risk. This approach immediately rises a couple of questions: what is the value of π we should use, and what is the best

loss function l in order to obtain a meaningful similarity score? Moreover, from a practical perspective, using the Bayes error as a similarity measure is problematic, since it is really hard to estimate. It requires a consistent classifier or posterior class probability estimator, and the convergence speed is usually quite slow (in fact, it can be arbitrary slow [Devroye et al., 1996]).

Risk estimation becomes easier if we restrict the classification function to a simple (e.g. parametric) family, like linear classifiers. However, this effectively imposes severe limitations on the features of the distributions that are being taken into account by the similarity measures. This can be really beneficial if there is some domain knowledge substantiating that limitations, since it allows the similarity measure to focus on the relevant "features" of the distributions. However, if we remain agnostic about our scenario and would like the data to speak for itself, such a rigid approach can be detrimental. Moreover, optimizing the 0-1 loss is still a complex problem as it cannot be handled analytically. Instead, *surrogate losses* [Bartlett et al., 2006] are usually employed. They are functions that share some features of the 0-1 loss while being well-behaved. This mainly implies being smoothness and differentiability.

A first pragmatic approach for defining risk-based affinities is to use the nearest neighbor (NN) rule (Sec. 5.2.1). Being a non-parametric method, it can capture arbitrary "shapes" of the distributions, making it flexible enough to define similarities between sets of points. From a practical point of view, there exist several efficient alternatives for obtaining training-set based error estimates with good distribution-free performance guarantees. In this chapter we will show how the asymptotic NN error presents a nice property for an affinity measure: it is a definite-positive kernel over probability distributions.

5.2.1 The Nearest Neighbor Rule

The k-nearest neighbor (k-NN) rule [Devroye et al., 1996] has enjoyed great popularity since its conception in the early fifties. This popularity is arguably the product of the following factors, amongst others:

- Intuitive interpretation

- Good performance in real-world problems
- Lends itself well to theoretical analysis

The goal of this section is to briefly describe the aspects of the k-NN rule that will serve as the basis for further developments in this chapter. Given a training dataset comprised of n pairs $(X_i, Y_i) \in \mathbb{R}^d \times \{0, 1\}$, we can formally define the k-NN rule as the mapping $g_n : \mathbb{R}^d \to \{0, 1\}$ such that:

$$g_n(\mathbf{x}) = \begin{cases} 1 & \text{if } \sum_{i=1}^{n} w_{ni} I_{\{Y_i=1\}} > \sum_{i=1}^{n} w_{ni} I_{\{Y_i=0\}} \\ 0 & \text{otherwise,} \end{cases} \tag{5.6}$$

where $w_{ni} = 1/k$ if X_i is among the k nearest neighbors of x and $w_{ni} = 0$ otherwise. A training sample X_i is said to be the k-th nearest neighbor of x (which we denote by $X_{(k)}(x)$) if the distance $||X_i - x||$ is the k-th smallest among $||X_1 - x||, \ldots, ||X_n - x||$. If k is odd (thus avoiding voting ties), the rule can be interpreted as classifying a sample x according to the majority vote of its k nearest neighbors in the training set. For $k = 1$, this reduces to the standard *NN rule*, where a test point is assigned the label of its closest point in the training set. Such a simple rule has been widely studied in the literature for decades, and will be the one we will base our further analyses upon.

One of the most interesting properties of the NN rule (and one that we will extensively explote) is that there is a convenient closed-form expression for its asymptotic error. Denoting by L_n^{NN} the error rate of the NN rule on a training set of size n we have the following theorem:

Theorem 5.2.1. *For the nearest neighbor rule and any pair P, Q of distributions:*

$$\lim_{n\to\infty} \mathbb{E}_M[L_n^{NN}] = \mathbb{E}_M[2\eta(x)(1 - \eta(x))] = \mathbb{L}_{0-1}^{NN}, \tag{5.7}$$

where $\eta = \frac{\pi dP}{dM}$ is the posterior class probability, and $M = \pi dP + (1 - \pi)dQ$ is the reference measure.

This theorem, under various continuity conditions, appears in [Cover and Hart, 1967]. In its most general form, it is due to [Stone, 1977]. It can be intuitively understood by noting that, asymptotically, the nearest neighbor

of a point x will converge to that exact same point. This way, the probability of error at a given point will be $\eta(x)(1-\eta(x)) + (1-\eta(x))\eta(x) = 2\eta(x)(1-\eta(x))$, since $\eta(x)$ is the probability of the point x belonging to class 1, $1-\eta(x)$ is the probability of point x belonging to class 0 and, as aforementioned, those exact same values also hold for the nearest neighbor of x.

Another interesting aspect of the NN rule is the good behaviour of its training-set set based error estimates. Arguably the simplest possible option is the *deleted estimate* or leave-one-out cross validation. For general k-NN, a simple distribution-free finite sample performance bound of this estimate in terms of squared error is given in [Rogers and Wagner, 1978]. Particularized for the NN case, it leads to the following L1 bound [Devroye et al., 1996]:

$$\mathbb{E}[|\hat{\mathbb{L}}_n^{(D)} - \mathbb{L}_n|] \leq \sqrt{\frac{7}{n}},$$

where $\hat{\mathbb{L}}_n^{(D)}$ denotes the deleted estimate on a sample size n, and $\mathbb{L}_n$ is the actual NN error on that sample. The most serious disadvantage of the deleted estimate is its large variance. This can be alleviated by using more complex methods, like multiple-fold cross validation (CV). In fact, the NN rule lends itself very well to CV, making it possible to get closed-form expressions for complete cross-validation [Mullin and Sukthankar, 2000]. Standard CV relies on resampling in order to get train/test divisions of the original sample, obtaining the test error on those division and then averaging to get the final estimate. On the other hand, the main idea behind complete cross-validation is to directly obtain the expectation over all the possible partitions of the dataset (for a fixed training set size). For the NN rule, this can be done in closed form. Specifically, the number of train/test partitions for which a given x in the original sample is correctly satisfied is given by:

$$A(x) = \sum_{i=1}^{n-1} I_{Y_x = Y_{X_{(k)}(x)}} \binom{n-i-1}{\alpha-1},$$

where $X_{(k)}(x)$ represents the k-th nearest neighbor of x, Y_x the label associated with x and α is the chosen size for the training set. Obviously, the number of possible partitions with such a training set size (and forcing x to be in the test set) is simply

$A_\alpha = \binom{n-1}{\alpha}$. This way, the expected error for a given point x is given by $1 - \frac{A(x)}{A_\alpha}$. Averaging over all the points in the sample gives the final estimate.

5.2.2 Properties of NN error as an affinity measure

Using NN errors $\mathbb{L}_{0-1}^{NN}(P,Q)$ as affinity measures between distributions P, Q is a simple approach with good theoretical properties. In the following we present and prove our results regarding some of the most important of those properties.

- **NN risk is a non-negative, bounded similarity measure with the supremum attained when $P = Q$**

  Obviously, these conditions are highly desirable for an affinity measure. We make them explicit in the following theorem:

  Theorem 5.2.2. *$0 \leq \mathbb{L}_{0-1}^{NN}(P,Q) \leq \frac{1}{2}$, with $\mathbb{L}_{0-1}^{NN}(P,Q) = \frac{1}{2}$ iff $P = Q$*

  Proof. Non-negativity is obvious, since $\mathbb{L}_{0-1}^{NN}$ (as any other error) is bounded below by the Bayes error. The equality condition can be shown in many ways. For the sake of interest, we will prove it based on the relation between NN and Bayes errors. The expression for $\mathbb{L}_{0-1}^{NN}$ can be bounded from above terms of Bayes error $\mathbb{L}^*$:

  $$\begin{aligned}
  \mathbb{L}_{0-1}^{NN}(P,Q) &= \mathbb{E}_M[2\eta(x)(1-\eta(x))] \\
  &= 2\mathbb{E}_M[\min(\eta(x), 1-\eta(x)) \cdot (1 - \min(\eta(x), 1-\eta(x)))] \\
  &\leq 2\mathbb{E}_M[\min(\eta(x), 1-\eta(x))]\mathbb{E}_M[1 - \min(\eta, 1-\eta(x))] \\
  &= 2\mathbb{L}^*(P,Q)(1 - \underline{\mathbb{L}}_{0-1}(P,Q)) \leq 2\underline{\mathbb{L}}_{0-1}(P,Q),
  \end{aligned}$$

  where the first inequality comes from the well-known association inequality $\mathbb{E}[f(x)g(x)] \leq \mathbb{E}[f(x)]\mathbb{E}[g(x)]$ for f monotone increasing and g monotone decreasing ([Devroye et al., 1996], Theorem A.19). The next equality comes from the definition of Bayes error as in Eq. (5.5). Since the Bayes risk has a maximum of $\frac{1}{2}$ when $\eta(x) = \frac{1}{2}$ for all x (and thus $P = Q$) it follows that $\mathbb{L}_{0-1}^{NN}(P,Q) \leq \frac{1}{2}$ with equality iff $P = Q$. □

- **NN risk as a positive-definite kernel over probability distributions** Many machine learning algorithms require affinity functions or *kernels* which are positive definite (p.d.). A symmetric function $\kappa : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a positive definite (p.d.) kernel on $\mathcal{X}$ if

$$\sum_{i,j=1}^{n} \kappa(x_i, x_j) c_i c_j \geq 0 \tag{5.8}$$

holds for any $n \in \mathbb{N}$, $x_1, \ldots, x_n \in \mathcal{X}$ and $c_1, \ldots, c_n \in \mathbb{R}$.

The reason behind the interest of p.d. similarity functions is mainly two-fold: On the one hand, positive definiteness of affinity matrices is fundamental for ensuring the convergence of convex optimization [Boyd and Vandenberghe, 2004] procedures, which underlies many learning algorithms. On the other hand, and arguably more interestingly, a positive definite kernel κ can be interpreted as defining dot-products between feature representations in a space of sequences l^2 [Berlinet and Thomas-Agnan, 2003]. This way, any linear algorithm which can be expressed in terms of inner products can be inmediately made non-linear by substituting the euclidean inner products for evaluations of a kernel function. This amounts to working on the induced feature space, without needing to perform any explicit expansion. This is known as the *kernel trick*.

We now proceed to establish and proof that $\mathbb{L}_{0-1}^{NN}$ is a positive-definite kernel for probability distributions:

Theorem 5.2.3. *The asymptotic error $\mathbb{L}_{0-1}^{NN}$ of the nearest neighbor rule is a positive definite kernel on the space of probability distributions $\mathcal{P} = \mathcal{M}_+^1(\mathcal{X})$*

Before starting with the proof, we present and prove the following intermediate result

Lemma 5.2.4. *The function $f(x, y) = \frac{1}{x+y}$ is a positive definite kernel in $\mathbb{R}^+/\{0\}$*

Proof. Let $g_x(t) = e^{-xt} I_{t>0}$ and $g_y(t) = e^{-yt} I_{t>0}$. For $x, y \in \mathbb{R}^+/\{0\}$ we have that $g_x, g_y \in L^2$, that is to say, the Hilbert space of square-integrable functions.

The standard inner product of g_x and g_y then yields:

$$< g_x, g_y >= \int_0^\infty g_x(t)g_y(t)dx = -\frac{1}{x+y}\, e^{-(x+y)t}\Big|_{t=0}^{\infty} = \frac{1}{x+y}.$$

So the function $f(x,y) = \frac{1}{x+y}$ for $x, y \in \mathbb{R}^+/\{0\}$ is the inner product of a certain L^2 embedding of x and y, and is thus p.d. □

We are now in condition of proving the main theorem:

Proof. For notational simplicity, we proof the theorem for the case where $\pi = \frac{1}{2}$ and assuming that densities $p(x), q(x)$ exist. Under these conditions, we can write the asymptotical NN error as

$$\begin{aligned}
\mathbb{L}_{0-1}^{NN}(P,Q) &= \mathbb{E}_M[2\eta(x)(1-\eta(x))] \\
&= 2\int_{\mathcal{X}} \eta(x)(1-\eta(x)) \cdot \frac{1}{2}(p(x)+q(x))dx \\
&= 2\int_{\mathcal{X}} \frac{p(x)}{p(x)+q(x)} \cdot \frac{q(x)}{p(x)+q(x)} \cdot \frac{1}{2}(p(x)+q(x))dx \\
&= \int_{\mathcal{X}} \frac{p(x)q(x)}{p(x)+q(x)}dx = \int_{\mathcal{X}} L(p(x),q(x))dx, \qquad (5.9)
\end{aligned}$$

where $L(a,b) = \frac{ab}{a+b}$. Note that if $x \notin \mathcal{R}_{P,Q} = \text{Supp}(P) \cap \text{Supp}(Q)$, where Supp denotes the support, then $L(p(x), q(x)) = 0$. So we can write

$$\mathbb{L}_{0-1}^{NN}(P,Q) = \int_{\mathcal{R}_{P,Q}} L(p(x),q(x))dx.$$

Let us analize the integrand $L(a,b)$. Since $p(x), q(x) \geq 0$ and the integral is restricted to $\mathcal{R}_{P,Q}$, we only need to worry about $a, b \in \mathbb{R}^+/\{0\}$. Trivially, ab is a p.d. kernel on $\mathbb{R}$, since it is the standard euclidean inner product in that space. Since, by Lemma 5.2.4, $\frac{1}{a+b}$ is p.d. in $\mathbb{R}^+/\{0\}$ then we can apply an elementary property of p.d. functions (see, e.g., [Berg et al., 1984] Chap. 3) to claim that $L(a,b) = \frac{ab}{a+b}$ is also p.d. Finally, it is easy to show that the integral is a p.d. function by proving that it satisfies the condition in Eq. (5.8).

$$\begin{aligned}
\sum_{i,j=1}^{n} c_j c_k \mathbb{L}_{0-1}^{NN}(P,Q) &= \sum_{i,j=1}^{n} c_j c_k \int_{\mathcal{R}_{P,Q}} L(p(x),q(x))dx \\
&= \int_{\mathcal{R}_{P,Q}} dx \sum_{i,j=1}^{n} c_j c_k L(p(x),q(x)) \geq 0,
\end{aligned}$$

the second equality is due to uniform convergence, and the inequality comes from $\sum_{i,j=1}^{n} c_j c_k L(a,b) \geq 0$ for all $a, b \in \mathbb{R}^+/\{0\}$, since L is p.d. □

- **NN risk is a scale-invariant affinity measure**

  A measure between probability distributions over $\mathcal{X}$ is scale-invariant if it is insensitive to scalings of $\mathcal{X}$. For clustering purposes it is positive to have such a coherent similarity measure, so that if we have scaled versions of the same distribution in our dataset we obtain a similar scatter within the corresponding clusters. We now proceed to state and proof that the NN risk satisfies such a property:

  Theorem 5.2.5. *The NN risk $\mathbb{L}_{0-1}^{NN}$ is a scale-invariant measure of affinity between probability distributions*

  The result can be intuitively understood by considering the way the NN classifier works. We can also write down a very simple explicit proof:

  Proof. For notational simplicity, we will once again assume that densities exist and that $\mathcal{X} \subseteq \mathbb{R}$. The scaled densities can thus be written as $p^*(x) = \frac{1}{a}p(\frac{x}{a})$, $q^*(x) = \frac{1}{a}q(\frac{x}{a})$. The NN error between these scaled densities is simply:

$$\begin{aligned} \mathbb{L}_{\not<-\not<}^{NN}(p^*, q^*) &= \int_{\mathcal{X}} \frac{\frac{1}{a^2}p(\frac{x}{a})q(\frac{x}{a})}{\frac{1}{a}\left(p(\frac{x}{a}) + q(\frac{x}{a})\right)} dx \\ &= \frac{1}{a}\int_{\mathcal{X}} \frac{p(x')q(x')}{p(x') + q(x')} a dx' \\ &= \int_{\mathcal{X}} \frac{p(x')q(x')}{p(x') + q(x')} dx' = \mathbb{L}_{0-1}^{NN}(p, q), \end{aligned}$$

  where the second equality comes from the change of variable $x' = \frac{x}{a}$, so $dx = adx'$. □

5.3 Generalizing risk-based affinities

Using the risk of a certain classifier as a measure of affinity between sequences for a further clustering is a quite intuitive idea, but maybe lacking flexibility and being too simplistic. The main goal of this section is to look at risk-based measures

from a broader perspective, trying to generalize the basic idea and, at the same time, connecting it with the broader literature of divergences between probability distributions. Specifically, the focus will be on the connection of classifier-induced affinities with the well-known family of f-divergences [Ali and Silvey, 1966, Csiszár, 1967].

5.3.1 f-divergences

Given a convex function $f : (0, \infty) \to \mathbb{R}$, with $f(1) = 0$, we can define the corresponding f-divergence between two distributions $P, Q \in \mathcal{P} = \mathcal{M}_+^1(\mathcal{X})$ over an input space $\mathcal{X}$ as:

$$\mathbb{I}_f(P, Q) = \mathbb{E}_Q \left[f\left(\frac{dP}{dQ}\right)\right] = \int_{\mathcal{X}} dQ f\left(\frac{dP}{dQ}\right),$$

if P is absolutely continuous with respect to Q, and ∞ otherwise. Many well-known divergences can be cast into this framework by adequately choosing the generating function f. Some important examples include

- **Variational Divergence**

  The Variational Divergence is deeply connected with the Bayes risk of a binary classification problem (see, e.g. [Devroye et al., 1996] Chap. 3). Its name arises from the following variational interpretation of the divergence:

  $$V(P, Q) = 2||P - Q||_\infty = 2 \sup_{A \subseteq \mathcal{X}} |P(A) - Q(A)|.$$

  It can be written in the canonical f-divergence form as follows:

  $$V(P, Q) = \int_{\mathcal{X}} |dP - dQ| \int_{\mathcal{X}} dQ \frac{|dP - dQ|}{dQ} = \int_{\mathcal{X}} dQ f_V\left(\frac{dP}{dQ}\right),$$

  with $f_V(t) = |t - 1|$.

- **Kullback-Leibler (KL) Divergence**

  The KL divergence is arguably one of the best known f-divergences. It is a very important magnitude in information theory, since it is closely related to Shannon entropy, mutual information, cross entropy,etc. [Cover and Thomas, 1991]. Its standard representation is given by

  $$KL(P, Q) = \int_{\mathcal{X}} dP \log \frac{dP}{dQ}$$

which is obviously equivalent to the following expression

$$KL(P,Q) = \int_{\mathcal{X}} \frac{dP}{dQ} dQ \log \frac{dP}{dQ} = \int_{\mathcal{X}} dQ f_{KL}\left(\frac{dP}{dQ}\right),$$

with $f_{KL}(t) = t \log t$.

Table 5.1, extracted from [Reid and Williamson, 2009], summarizes the generating functions f and associated weights of these and several other important divergences.

Our discussion will be mainly based on a nice classic result (see for example [Österreicher and Vajda, 1993]) that shows how f-divergences can be represented by a weighted integral of *statistical informations*, which are closely related to Bayes risks for 0-1 loss. Specifically:

$$\mathbb{I}_f(P,Q) = \int_0^1 \Delta\underline{\mathbb{L}}_{0-1}(\pi, P, Q)\gamma_f(\pi)d\pi, \tag{5.10}$$

where $\Delta\underline{\mathbb{L}}_{0-1}(\pi, P, Q)$ is the *statistical information* defined in [DeGroot, 1970] as:

$$\Delta\underline{\mathbb{L}}_{0-1}(\pi, P, Q) = \underline{\mathbb{L}}(\pi) - \underline{\mathbb{L}}(\pi, P, Q) = \min(\pi, 1-\pi) - \underline{\mathbb{L}}(\pi, P, Q), \tag{5.11}$$

where the underline denotes a minimization and $\mathbb{L}_l$ represents the expected risk under loss l (when no loss is explicitly indicated, 0-1 loss is assumed). Specifically, $\mathbb{L}(\pi)$ stands for the *prior* expected risk, that is to say, the expected risk when only the prior probability of the binary classes is known. On the other hand, $\mathbb{L}(\pi, P, Q)$ represents the *posterior* expected risk, where both the prior probability and the class densities (and thus, the posterior probability) are known. This way, the statistical information can be intuitively interpreted as the risk reduction provided by the knowledge of the exact posterior probability instead of just the prior probability π.

The weights $\gamma_f(\pi)$ of the integral representation are related to the curvature of the function f defining the divergence, showing that f-divergences are invariant to affine transformations of the generating function f. Specifically, the weights for a given f can be written down as follows:

$$\gamma_f(\pi) = \frac{1}{\pi^3} f''\left(\frac{1-\pi}{\pi}\right), \tag{5.12}$$

where the derivatives are interpreted in a distributional way. Since f is a convex function, the weights $\gamma_f(\pi)$ are non-negative.

This representation makes explicit the relationship between binary risks and f-divergences, already hinted by the presence of the likelihood ratio in the original definition of the divergences. Coupling our interest on defining classifier-based affinity/divergence measures for sets of vectors with this convenient representation of f-divergences we will devote this chapter to proposing interesting generalizations of this kind of divergences. Our aim will be finding divergences that are analogous to standard f-divergences (such as KL, Jensen-Shannon, ...) but with the particularities of classifier-based measures. We will do this from two different perspectives: restriction of the class of admissible classification functions that we want to consider (yielding class-restricted f-divergences) and substitution of the 0-1 loss for other kind of loss functions (obtaining loss-induced f-divergences).

Symbol	$\gamma(\pi)$	$f(t)$	Name
$V(P,Q)$	$4\delta(\pi - \frac{1}{2})$	$\lvert t-1\rvert$	Variational Divergence
$\Delta(P,Q)$	8	$(t-1)^2/(t+1)$	Triangular Discrimination
$\mathrm{KL}(P,Q)$	$\frac{1}{\pi^2(1-\pi)}$	$t \ln t$	Kullback-Leibler Divergence
$I(P,Q)$	$\frac{1}{2\pi(1-\pi)}$	$\frac{t}{2}\ln t - \frac{t+1}{2}\ln(t+1) + \ln 2$	Jensen-Shannon Divergence
$J(P,Q)$	$\frac{1}{\pi^2(1-\pi)^2}$	$(t-1)\ln t$	Jeffreys Divergence
$\chi^2(P,Q)$	$\frac{2}{\pi^3}$	$(t-1)^2$	Pearson Chi Squared Divergence
$h^2(P,Q)$	$\frac{1}{2[\pi(1-\pi)]^{\frac{3}{2}}}$	$(\sqrt{t}-1)^2$	Hellinger Divergence

Table 5.1: Some well-known f-divergences with their associated weights. Extracted from [Reid and Williamson, 2009].

5.4 Class-restricted divergences (CRFDs)

The weighted integral representation in Eq. (5.10) suggests a simple way of generalizing the idea of using the risk of certain classifiers as affinity measures between sets of vectors. Specifically, given a class C of classification functions $C \subseteq \{0,1\}^{\mathcal{X}}$, we can use the optimal risks over C to define a "restricted" version of the statistical information as follows:

$$\Delta\underline{\mathbb{L}}^{C}_{0-1}(\pi, P, Q) = \underline{\mathbb{L}}^{C}(\pi) - \underline{\mathbb{L}}^{C}(\pi, P, Q), \tag{5.13}$$

so the risk minimization is performed over the functions in C instead of over the whole set of classification functions. This way, we directly obtain the class-restricted f-divergences (CRFDs):

$$\mathbb{I}^{C}_{f}(P, Q) = \int_{0}^{1} \Delta\underline{\mathbb{L}}^{C}_{0-1}(\pi, P, Q)\gamma_{f}(\pi)d\pi, \tag{5.14}$$

The class-restricted statistical information $\Delta\underline{\mathbb{L}}^{C}_{0-1}(\pi, P, Q)$ can be defined in the following alternate form:

$$\begin{aligned} \Delta\underline{\mathbb{L}}^{C}_{0-1}(\pi, P, Q) &= \underline{\mathbb{L}}^{C}(\pi) - \underline{\mathbb{L}}^{C}(\pi, P, Q) \\ &= \max_{P', Q' \in \mathcal{P}} \underline{\mathbb{L}}^{C}(\pi, P', Q') - \underline{\mathbb{L}}^{C}(\pi, P, Q), \end{aligned} \tag{5.15}$$

where the restricted risk $\underline{\mathbb{L}}^{C}(\pi, P, Q)$ can be written in terms of the posterior probability function $\eta = \frac{\pi dP}{dM}$, with $M = \pi P + (1-\pi)Q$, as follows

$$\underline{\mathbb{L}}^{C}(\pi, P, Q) = \min_{y \in C} \mathbb{E}_{M}\left[\eta(x) I_{y(x)=0} + (1 - \eta(x)) I_{y(x)=1}\right]. \tag{5.16}$$

This definition explicitly shows that $\Delta\underline{\mathbb{L}}^{C}_{0-1}(\pi, P, Q)$ is always a non-negative quantity.

Equation (5.14) provides a way to define f-like divergences using optimal risks within a predefined set of classification functions, instead of full Bayes risks. The main points behind this idea are the following:

- Use the allowed classification functions C to select what aspects of the distributions we are interested in from the divergence measurement perspective.

For example, in a set-of-vectors clustering scenario, practitioners may know a priori that the kind of sets that they want to cluster can be well separated by a linear classifier (or a SVM with some predefined kernel). This way, using standard divergences will capture more information than the user is interested on.

- CRFDs depend on optimal risks within a family of classification functions, which are "easier" to estimate than full Bayes risks. This is usually known as the trade-off between approximation and estimation errors: the smaller the class of functions, the easier it is to get close to the optimal classification function within the class, but the further away that function will be (in general) from the best possible (or Bayes) classification function.

- Restricting C has a "regularizing" effect on the divergence. As we will see in the next section, reducing the set of classification functions C compress the range of the divergences. In practical applications, this amounts to disencouraging wide divergence fluctuations due to peculiarities of the samples (such as outliers).

Obviously, $\mathbb{I}_f^C(P,Q) = \mathbb{I}_f(P,Q)$ whenever $C = \{0,1\}^{\mathcal{X}}$, so *the standard f-divergences can be seen as a special case of class-restricted f-divergences when the class includes every possible classification function.*

5.4.1 Properties of class-restricted divergences

In this section we will explicit the conditions that a class of functions C must satisfy in order for some desirable properties of the resulting $\mathbb{I}_f^C$ divergences to hold. We will show how many very desirable properties can be obtained using quite simple choices for C.

1. **Lower bounds of original f-divergences:** $\mathbb{I}_f^C(P,Q) \leq \mathbb{I}_f(P,Q)$
   This is an interesting property to have if class-restriced divergences are to be used as surrogates for standard f-divergences. Moreover, it has a natural interpretation in terms of C defining the "features" of the probability distributions that we are interested in. Taking into account more features by making C

larger should increase the divergence, the limiting case being $C = \{0,1\}^{\mathcal{X}}$ and, as previously stated, $\mathbb{I}_f^C(P,Q) = \mathbb{I}_f(P,Q)$. This property can be enforced by a very simple condition on C.

Theorem 5.4.1. *If the constant functions* $\mathbf{0}, \mathbf{1} \in C$*, then* $\mathbb{I}_f^C(P,Q) \leq \mathbb{I}_f(P,Q)$

Proof. When $\mathbf{0}, \mathbf{1} \in C$, the whole input space $\mathcal{X}$ can be assigned to any of the two classes and thus $\underline{\mathbb{L}}^C(\pi) = \min(\pi, 1-\pi) = \underline{\mathbb{L}}(\pi)$. Obviously, $\underline{\mathbb{L}}^C(\pi, P, Q) \geq \underline{\mathbb{L}}(\pi, P, Q)$ since the minimization is carried out over a subset of classification functions, so $\Delta\underline{\mathbb{L}}_{0-1}^C(\pi, P, Q) \geq \Delta\underline{\mathbb{L}}_{0-1}(\pi, P, Q)$. The result then follows by the analogy between Eqs. 5.10 and 5.14. □

Moreover, we can trivially get the following corollary:

Corollary 5.4.2. *If* $\mathbf{0}, \mathbf{1} \in C$ *and* $C \subseteq C'$*, then* $\mathbb{I}_f^C(P,Q) \geq \mathbb{I}_f^{C'}(P,Q)$

So, if the condition holds, the CRFDs are non-increasing in C (in the order given by inclusion).

2. **Non-negativity:** $\mathbb{I}_f^C(P,Q) \geq 0$ This is a basic property for a measure of divergence. Here we show how it holds for any subset C of classification functions defining the restricted divergence:

   Theorem 5.4.3. *For any convex* f *and* $C \subseteq \{0,1\}^{\mathcal{X}}$*,* $\mathbb{I}_f^C(P,Q) \leq 0 \ \forall P, Q \in \mathcal{P}$

   Proof. Convexity of f implies non-negativity of the weights $\gamma_f(\pi)$. From previous discussion, $\Delta\underline{\mathbb{L}}_{0-1}^C(\pi, P, Q) \geq 0$. From the definition of class-restricted divergences in Eq. (5.14), non-negativity of $\mathbb{I}_f^C(P,Q)$ is assured. □

3. **Identity of indiscernibles:** $\mathbb{I}_f^C(P,Q) = 0$ iff $P = Q$.
   A divergence satisfying this property can be used for checking the equality of two distributions, and can be thus used e.g. for solving the two-sample problem [Gretton et al., 2007]. Together with non-negativity, this property is essential for proper measures of difference. It is then very interesting to find the conditions that C must satisfy in order for it to hold. In the following theorem, we link this property with the discriminative capacity of C:

Theorem 5.4.4. *For any strictly convex f and any class of classification functions $C \subseteq \{0,1\}^{\mathcal{X}}$ with $\mathbf{0}, \mathbf{1} \in C$ such that, for some $\pi \in [0,1]$, $\underline{\mathbb{L}}^C(\pi, P, Q) = \pi$ iff $\underline{\mathbb{L}}(\pi, P, Q) = \pi$, then $\mathbb{I}_f^C(P, Q) = 0$ iff $P = Q$*

Proof. If f is strictly convex then, by Eq. (5.12), $\gamma_f(\pi) > 0$ for all $\pi \in [0,1]$. From this and Eq. (5.14), the identity of indiscernibles condition translates into $\mathbb{I}_f^C(P, Q) = 0$ iff $\Delta\underline{\mathbb{L}}_{0-1}^C(\pi, P, Q) = 0 \ \forall \pi \in [0,1]$. Given the conditions imposed in the theorem, there is some π such that $\Delta\underline{\mathbb{L}}_{0-1}^C(\pi, P, Q) = 0$ iff $\underline{\mathbb{L}}(\pi, P, Q) = \pi$. Since the Bayes error rate $\underline{\mathbb{L}}(\pi, P, Q) = \pi$ iff $P = Q$, this completes the proof. □

This theorem shows that if the class of classification functions is one that can always improve (at least marginally) over the trivial classifier, unless that trivial classifier is already optimal, then the resulting class-restricted f-divergence satisfies the identity-of-indiscernibles property. The ability of improving over the trivial classifier can be verified for many "sensible" classes of classification functions. We cite the following lemma from [Devroye et al., 1996], Chap. 4.

Lemma 5.4.5. *The optimal expected risk $\underline{\mathbb{L}}^l$ of a linear classifier satisfies $\underline{\mathbb{L}}^l < \frac{1}{2}$ with equality iff the Bayes error rate $\underline{\mathbb{L}} = \frac{1}{2}$.*

This shows that $\mathbb{L}^l(\frac{1}{2}, P, Q) = \frac{1}{2}$ iff $\mathbb{L}(\frac{1}{2}, P, Q) = \frac{1}{2}$, so the conditions of Theorem 5.4.4 apply and we obtain the following corollary:

Corollary 5.4.6. *If $C_l = \{y : y(x) =< w, x > +b, w \in \mathbb{R}^d, b \in \mathbb{R}\}$ is the set of all linear classification functions, then $\mathbb{I}_f^{C_l}(P, Q) = 0$ iff $P = Q$*

4. **Symmetry:** $\mathbb{I}_f^C(P, Q) = \mathbb{I}_f^C(Q, P)$ if $f(t) = f(t)^* + c(t-1)$, $c \in \mathbb{R}$

Symmetry of an affinity / divergence function is a intuitively desirable property. In fact, many learning algorithms require symmetric affinity functions to work correctly. For general f-divergences it is well known (see, for example, [Liese and Vajda, 2006]) that

$$\mathbb{I}_f(P, Q) = \mathbb{I}_f(Q, P) \text{iff} f(t) = f^*(t) + c(t-1), \tag{5.17}$$

where f^* is the Csiszar dual of f, defined as follows:

$$f^*(t) = tf\left(\frac{1}{t}\right) \tag{5.18}$$

Our intention here is to propose an equivalent theorem for class-restricted divergences, imposing a very simple condition on C:

Theorem 5.4.7. *If C is such that $f \in C \Rightarrow \overline{f} \in C$, then the symmetry property holds.*

To prove this theorem we will use a result from [Reid and Williamson, 2009], relating the condition on f for the symmetry of an f-divergence with a condition on the corresponding weights $\gamma_f(\pi)$:

Lemma 5.4.8. *Suppose $\mathbb{I}_f$ is an f-divergence with weight function γ_f. Then $\mathbb{I}_f$ is symmetric iff $\gamma_f(\pi) = \gamma_f(1-\pi)$*

Now we can proceed with the proof:

Proof. Assume that f is such that $\mathbb{I}_f$ is symmetric, so $\gamma_f(\pi) = \gamma_f(1-\pi)$. Let us start by writing $\mathbb{I}_f^C(Q,P)$:

$$\begin{aligned}
\mathbb{I}_f^C(Q,P) &= \int_0^1 \Delta\underline{\mathbb{L}}_{0-1}^C(\pi,Q,P)\gamma_f(\pi)d\pi \\
&= -\int_1^0 \Delta\underline{\mathbb{L}}_{0-1}^C(1-\pi',Q,P)\gamma_f(1-\pi')d\pi' \\
&= \int_0^1 \Delta\underline{\mathbb{L}}_{0-1}^C(1-\pi',Q,P)\gamma_f(\pi')d\pi',
\end{aligned}$$

where the first equality comes from the change of variable $\pi = 1-\pi'$ and the second one from the symmetry of γ_f. By comparison with Eq. (5.14), this implies that $\mathbb{I}_f^C(P,Q) = \mathbb{I}_f^C(Q,P)\ \forall P,Q \in \mathcal{P}$ if $\Delta\underline{\mathbb{L}}_{0-1}^C(\pi,P,Q) = \Delta\underline{\mathbb{L}}_{0-1}^C(1-\pi,Q,P)$. Changing the binary classification task from (π,P,Q) to $(1-\pi,Q,P)$ transforms the posterior from $\eta(x)$ to $1-\eta(x)$ while keeping the same base measure $M = \pi P + (1-\pi)Q$. Recalling the definition of restricted posterior risk in Eq. (5.16) , it is easy to see that if $y*$ is the minimizer within C of $\underline{\mathbb{L}}^C(\pi,P,Q)$ and the condition in the theorem holds, then $\overline{y^*}$ is the minimizer of

$\underline{\mathbb{L}}^C(1-\pi, Q, P)$ in C, yielding the same minimum. This way, $\Delta\underline{\mathbb{L}}^C_{0-1}(\pi, P, Q) = \Delta\underline{\mathbb{L}}^C_{0-1}(1-\pi', Q, P)$, completing the proof. □

In contrast, there are some properties of f-divergences that are intrinsically lost when the class of functions is restricted. The best example of this is the *information processing* property [Csiszár, 1967, Österreicher, 2002], which basically states that no transformation applied to the space over which the distributions are defined can increase the divergence. This can be easily explained in terms of equivalent results for the Bayes risk ([Devroye et al., 1996], Chap. 2) and exploiting the integral representation of f-divergences. Obviously, such a result can not hold in general for restricted risks. One of the motivations for using classification functions for defining divergence measures is that the chosen family C of functions can reflect some prior knowledge on the kind of "features" of the distributions that we are interested on for discrimination purposes. Thus, we are only looking at the information preserved by C, so an information processing-like property, while very important in general, is not interesting for our purpose.

5.4.2 Conclusions

We have introduced the concept of class-restricted f-divergences, based on the idea of minimizing the 0-1 loss under a limited set C of classification functions. This goes on the line of using the family of classification functions to define what features of the distributions we are interested on. We have seen how many interesting properties can be obtained by imposing simple conditions on C. For example, by defining C to be the set of linear classification functions we get a divergence measure with most interesting properties of f-divergences, but attending only to linear patterns. The problem remains how to efficiently estimate these divergences. The formulation requires finding the optimal error rate within C for every $\pi \in [0, 1]$. Sampling π yields approximation of the divergences, but still requires finding a lot of optimal classifiers. Moreover, the 0-1 loss is usually not easy to minimize, so classifiers usually optimize some related measures which are better behaved (mostly, continuous and differentiable). They are called surrogate losses. With those ideas in mind, in the

next section we study a different generalization of f-divergences from the perspective of surrogate loss functions. Nonetheless, we consider the theoretical framework behind CRFDs to be interesting and potentially applicable to real problems when efficient estimation procedures appear. This constitutes a interesting line for our future research.

5.5 Loss-induced divergences

5.5.1 Some motivation: NN error and surrogate Bayes risks

The previous section dealt with generalizations of f-divergences when the defining 0-1 risks are optimized over a subset $C \subseteq \{0,1\}^{\mathcal{X}}$ of binary classification functions. However, many classification rules do not directly involve a risk minimization over a predefined family of functions. Specifically, we are motivated here by the case of the nearest-neighbor rule. As previously stated, this rule has many nice properties which make it very appealing for defining risk-based measures. The problem is that it is quite unnatural to interpret it in the framework of CRFDs. Given a training set, the NN rule directly gives a classification function $g_{NN}(x)$ without any explicit optimization over a function class. In this sense, one could consider that $C = \{g_{NN}\}$, which is such a restricted class that it is not possible to deduce its properties using the Theorems in Sec. 5.4.1. However, here we propose another interpretation of the NN rule risk which directly suggests yet another way of generalizing f-divergences. We first must introduce the *square loss* $l_{SQ}(y, \hat{y})$:

$$l_{SQ}(y,\hat{y}) = \begin{cases} (1-\hat{y})^2 & , y = 1 \\ \hat{y}^2 & , y = 0 \end{cases} \tag{5.19}$$

This loss is well-known in the field of probability estimation. It naturally induces the following point-wise risk

$$L_{SQ}(\eta(x), \hat{\eta}(x)) = \eta(x)(1-\hat{\eta}(x))^2 + (1-\eta(x))\hat{\eta}(x)^2, \tag{5.20}$$

where η is the posterior class probability. It is very easy to obtain the minimum on $\hat{\eta}(x)$ of such a equation. Differentiation with respect to $\hat{\eta}(x)$ yields

$$\begin{aligned}\frac{\partial L_{SQ}(\eta(x),\hat{\eta}(x))}{\partial \hat{\eta}(x)} &= -2\eta(x)(1-\hat{\eta}(x)) + 2(1-\eta(x))\hat{\eta}(x) \\ &= 2(\hat{\eta}(x)-\eta(x)),\end{aligned}$$

so that, for a given $\eta(x)$, the minimum is achieved whenever $\hat{\eta}(x) = \eta(x)$. Losses that induce a point-wise risk satisfying this intuitive property are known as *Fisher consistent* or *proper losses* [Buja et al., 2005]. The corresponding optimal point-wise risk is obtained directly by substitution.

$$\begin{aligned}\underline{L}_{SQ}(\eta(x)) &= \eta(x)(1-\eta(x))^2 + (1-\eta(x))\eta(x)^2 \\ &= \eta(x)(1-\eta(x)). \end{aligned} \tag{5.21}$$

Now recall the expression for the asymptotic NN risk under the 0-1 loss

$$\mathbb{L}_{0-1}^{NN} = \mathbb{E}[2\eta(x)(1-\eta(x))].$$

We can directly re-write that expression as follows.

$$\begin{aligned}\mathbb{L}_{0-1}^{NN}(\eta, M) &= 2\mathbb{E}_{x\sim M}[\underline{L}_{SQ}(x)] \\ &= 2\underline{\mathbb{L}}_{SQ}(\eta(x), M) = 2\underline{\mathbb{L}}_{SQ}(\pi, P, Q), \end{aligned} \tag{5.22}$$

where the last equality is just an application of generative/discriminative duality. So it turns out that the expected error probability for the NN rule gives us a way to estimate the Bayes risk for the square loss. It is worth mentioning that the NN rule is not minimizing the risk under the square loss, since there is a factor of 2 in the formula, but nonetheless provides a univocal estimate of the optimal risk. Figure 5.1 provides a visual representation of the point-wise Bayes risks induced by both the square and 0-1 losses, as well as the asymptotic point-wise nearest neighbor error rate.

5.5.2 (f, l)-divergences

In this section we we define another generalization of f-divergences, called (f, l)-divergences. This generalization provides an additional degree of freedom by allowing

the substitution of the 0-1 loss in the original definition (Eq. (5.10)) for any loss. This way, we can express this new generalization as follows.

$$\mathbb{I}_{f,l} = \int_0^1 \Delta\underline{\mathbb{L}}_l(\pi, P, Q)\gamma_f(\pi)d\pi, \tag{5.23}$$

where

$$\Delta\underline{\mathbb{L}}_l(\pi, P, Q) = \underline{\mathbb{L}}_l(\pi) - \underline{\mathbb{L}}_l(\pi, P, Q). \tag{5.24}$$

We denote this generalized family as (f, l)-divergences. Once again, the original f-divergences can be obtained as a particular case of (f, l)-divergences by setting $l = l_{0-1}$.

Note that the idea of substituting 0-1 for more general losses is at the core of almost every classifier. This is the idea of *surrogate losses* [Bartlett et al., 2006]. Since the 0-1 loss is not very well behaved and thus hard to handle, most learning algorithms use, explicitly or implicitly, other kind of losses that approximate the 0-1 loss while being much more amenable to theoretical analysis, numerical optimization, etc. Thus, if the goal is to define divergences that can be nicely estimated using classification risks it is very natural to work with surrogate losses, since they are what most practical classifiers optimize.

5.5.3 Some properties of (f, l)-divergences

Analogously to what we did with CRFDs in Sec. 5.4.1, we devote this section to study how we can get interesting properties for (f, l)-divergences by adequately choosing the loss l. We will implicitly assume all losses to be *proper*. As explained in the previous section, a loss $l(y, \hat{y})$ is proper or Fisher-consistent if it satisfies

$$\begin{aligned} \underline{L}_l(\eta(x)) &= \min_{\hat{\eta}(x)} L_l(\eta(x), \hat{\eta}(x)) = L_l(\eta(x), \eta(x)) \\ &= \eta(x)l(1, \eta(x)) + (1 - \eta(x))l(0, \eta(x)) \end{aligned} \tag{5.25}$$

that is to say, the minimum point-wise risk for a given η is attained when the estimate $\hat{\eta} = \eta$. This is thus a very natural condition that all sensible losses satisfy.

As we will show in Sec. 5.5.4, (f, l) and f-divergences are deeply connected, so it is natural to recover most properties of standard f-divergences with a sensible

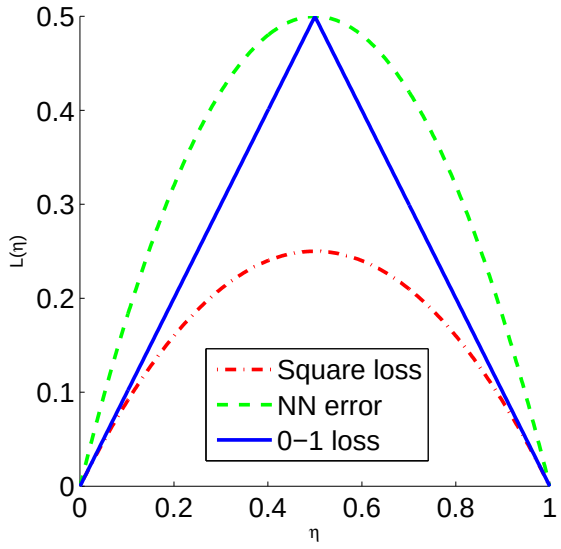


Figure 5.1: Point-wise Bayes risk $L(\eta)$ under the square and 0-1 losses, and thenearest neighbor asymptotic error.

election of the surrogate loss function. We now proceed to show a small representative selection of such properties, along with the conditions that the losses must satisfy in order for those properties to hold.

1. **Non-negativity and identity of indiscernibles:** $\mathbb{I}_{f,l}(P,Q) \leq 0$, with equality iff $P = Q$

   Theorem 5.5.1. *For any convex f and any proper loss l, $\mathbb{I}_{f,l}(P,Q) \geq 0$ for all P, Q. Moreover, if f is non-trivial ($\exists \pi \in (0,1) \mid \gamma_f(\pi) > 0$) and l is such that $\underline{L}_l$ is strictly concave, then equality holds iff $P = Q$.*

   Before stating the proof, we need the following lemma from [L.J.Savage, 1971].

   Lemma 5.5.2. *The point-wise Bayes risks $\underline{L}_l$ induced by a proper loss l is always a concave function.*

   We can now proceed to prove the theorem.

Proof. Based on the above lemma, we can write

$$\begin{aligned}\underline{\mathbb{L}}_l(\pi, P, Q) &= \mathbb{E}_{x\sim M}\left[\underline{L}_l\left(\frac{\pi dP(x)}{\pi dP(x) + (1-\pi)dQ(x)}\right)\right] \\ &\leq \underline{L}_l\left(\mathbb{E}_{x\sim M}\left[\frac{\pi dP(x)}{\pi dP(x) + (1-\pi)dQ(x)}\right]\right) \\ &= \underline{L}_l(\pi), \end{aligned} \tag{5.26}$$

where we have applied Jensen's inequality and then the fact that the expectation of the posterior probability coincides with the prior class probability π. This equation direclty shows that $\Delta\underline{\mathbb{L}}_l(\pi, P, Q) = \underline{\mathbb{L}}_l(\pi) - \underline{\mathbb{L}}_l(\pi, P, Q) \geq 0$, since $\underline{\mathbb{L}}_l(\pi) = \underline{L}(\pi)$. Given the non-negativity of the weight function γ_f, non-negativity of $\mathbb{I}_{f,l}$ arises. Moreover, if $\underline{L}_l(\eta)$ is strictly concave, then Jensen's inequality becomes a equality iff η is constant, which implies $P = Q$ and $\mathbb{I}_{f,l}(P, Q) = 0$ iff $P = Q$. □

Common surrogate losses induce strictly concave point-wise Bayes risks $\underline{L}_l$. Here we show it for two of the best-known ones:

- **Square loss**

  Recalling Eq. (5.21), $\underline{L}_{SQ}(\eta) = \eta(1-\eta)$. Directly,

  $$\begin{aligned}\underline{L}'_{SQ}(\eta) &= 1 - 2\eta \\ \underline{L}''_{SQ}(\eta) &= -2,\end{aligned}$$

  and the point-wise Bayes risk is strictly concave.

- **Log-loss**

  The log-loss is defined as follows: $l_{\log}(y, \hat{y}) = -y\log(\hat{y}) - (1-y)\log(1-\hat{y})$, for $y \in \{0, 1\}$ and $\hat{y} \in [0, 1]$. Its point-wise risk is thus given by

  $$L_{\log}(\eta(x), \hat{\eta}(x)) = \eta(x)(-\log(\hat{\eta}(x))) + (1-\eta(x))(-\log(1-\hat{\eta}(x)))$$

  It is easy to see that it is a proper loss

  $$\frac{\partial L_{\log}(\eta(x), \hat{\eta}(x))}{\partial\hat{\eta}(x)} = \frac{1-\eta(x)}{1-\hat{\eta}(x)} - \frac{\eta(x)}{\hat{\eta}(x)},$$

which equals 0 when $\hat{\eta}(x) = \eta(x)$. Then, the point-wise Bayes risk can be written as

$$\underline{L}_{\log}(\eta(x)) = \eta(x)(-\log(\eta(x))) + (1-\eta(x))(-\log(1-\eta(x))). \quad (5.27)$$

We can now obtain the derivatives

$$\begin{aligned} \underline{L}'_{\log}(\eta) &= \log(1-\eta) - \log(\eta) \\ \underline{L}''_{\log}(\eta) &= -\frac{1}{\eta(1-\eta)}. \end{aligned}$$

Since $\underline{L}''_{\log}(\eta) < 0$ for $\eta \in [0,1]$, the point-wise Bayes risk is strictly concave.

2. **Symmetry:** $\mathbb{I}_{f,l}(P,Q) = \mathbb{I}_{f,l}(Q,P)$ if $f(t) = f(t)^* + c(t-1)$, $c \in \mathbb{R}$
The following theorem shows how we can get a symmetry property analogous to the standard f-divergence case by imposing a really simple condition on the loss used to define the (f,l)-divergence.

Theorem 5.5.3. *If l is a proper loss such that $l(0,\hat{\eta}) = l(1,1-\hat{\eta})$, then the symmetry property for $\mathbb{I}_{f,l}$ holds*

Proof. Recalling the proof of Theorem 5.4.7, given the conditions on f (or, equivalently, γ_f), symmetry holds if $\Delta\mathbb{L}_l(\pi,P,Q) = \Delta\mathbb{L}_l(1-\pi,Q,P)$ for $\pi \in [0,1]$. Recalling the definition of statistical information in Eq. (5.24), this is equivalent to the condition $\underline{\mathbb{L}}_l(\pi,P,Q) = \underline{\mathbb{L}}_l(1-\pi,Q,P)$ or, equivalently, $\underline{\mathbb{L}}_l(\eta,M) = \underline{\mathbb{L}}_l(1-\eta,M)$. This holds if $\underline{L}_l(\eta) = \underline{L}_l(1-\eta)$. Finally, writing $\underline{L}_l$ in terms of l we get

$$\begin{aligned} \underline{L}_l(\eta) &= \eta l(1,\eta) + (1-\eta) l(0,\eta) \\ \underline{L}_l(1-\eta) &= (1-\eta) l(1,1-\eta) + \eta l(0,1-\eta). \end{aligned}$$

If $l(0,\eta) = l(1,1-\eta)$ both expressions coincide, and thus symmetry holds. □

Once again, standard surrogate losses satisfy this simple and natural condition. Note that the condition can be expressed in the alternative form $l(y,\hat{\eta}) = l(|y-\hat{\eta}|)$.

3. **Information processing:** $\mathbb{I}_{f,l}(P,Q) \geq \mathbb{I}_{f,l}(\Phi(P),\Phi(Q))$, where Φ is any transformation.

   In contrast with the CRFD case, (f,l)-divergences always preserve the information processing property of f-divergences. By simple inspection of the definition of (f,l)-divergences in Eq. (5.23) and non-negativity of the weights γ_f, the information processing property holds if $\Delta\mathbb{L}_l(\pi,P,Q) \geq \Delta\mathbb{L}_l(\pi,\Phi(P),\Phi(Q))$. This implies $\underline{\mathbb{L}}(\pi,P,Q) \leq \underline{\mathbb{L}}(\pi,\Phi(P),\Phi(Q))$, which is an intrinsic property of Bayes risks.

4. **Generalized Blackwell's Theorem:**

   Blackwell's theorem [Blackwell, 1951] is a well-known result from statistics, which in machine learning terms basically says that if a pair (P,Q) of distributions presents a larger f-divergence than another pair (T,Z), then there exists some prior probabilities for the class labels such that the classification error is smaller for the former pair [X.L. Nguyen and Jordan, 2009]. Although this is usually proved using complex arguments, we argue that the integral representation of f-divergences in Eq. (5.10) makes the proof trivial. Analogously, the definition of (f,l)-divergences in Eq. (5.23) implies the generalized version of the theorem that we provide.

   Theorem 5.5.4. *Let (P,Q) and (T,Z) be two pairs of probability distributions. For any (f,l)-divergence, if $\mathbb{I}_{f,l}(P,Q) > \mathbb{I}_{f,l}(T,Z)$ then $\exists\pi \in [0,1]$ such that $\underline{\mathbb{L}}_l(\pi,P,Q) < \underline{\mathbb{L}}_l(\pi,T,Z)$*

   Proof. Recalling Eq. (5.23), since $\gamma_f(\pi) \geq 0$ for all π, $\mathbb{I}_{f,l}(P,Q) > \mathbb{I}_{f,l}(T,Z)$ implies that there is at least some π such that $\Delta\underline{\mathbb{L}}_l(\pi,P,Q) > \Delta\underline{\mathbb{L}}_l(\pi,T,Z)$, which from the the definition of statistical information (Eq. (5.24)) in turn implies $\underline{\mathbb{L}}_l(\pi,P,Q) < \underline{\mathbb{L}}_l(\pi,T,Z)$. □

5.5.4 Connecting f and (f,l)-divergences

In this section we will take a look at the connections between (f,l)-divergences and standard f-divergences. Specifically, we will study if we can represent a given

(f, l)-divergence as an equivalent f-divergence. This will provide great insight into the effect of using a surrogate loss for divergence definition, as well as motivating surprising ways of estimating some well-known divergences.

As shown in [Österreicher and Vajda, 1993], statistical informations and f-divergences are in a one-to-one relationship. This is formalized in a constructive way in the following theorem:

Theorem 5.5.5 (Österreicher and Vajda 1993, Thm. 2)**.**
Given an arbitrary loss l, then defining

$$f_l^\pi(t) = \underline{L}(\pi) - (\pi t + 1 - \pi)\underline{L}\left(\frac{\pi t}{\pi t + 1 - \pi}\right). \tag{5.28}$$

for $\pi \in [0, 1]$ implies f_l^π is convex and $f_l^\pi(1) = 0$, and

$$\Delta\underline{\mathbb{L}}_l(\pi, P, Q) = \mathbb{I}_{f_l^\pi}(P, Q) \tag{5.29}$$

for all distributions P and Q.

Exploiting this representation of statistical information for arbitrary losses, Eq. (5.23) can be rewritten as follows:

$$\mathbb{I}_{f,l} = \int_0^1 \mathbb{I}_{f_l^\pi}(P, Q)\gamma_f(\pi)d\pi \tag{5.30}$$

Now we can leverage the weighted integral representation of $\mathbb{I}_{f_l^\pi}$ (Eq. (5.10)), yielding:

$$\begin{aligned}
\mathbb{I}_{f,l} &= \int_0^1 \left(\int_0^1 \Delta\underline{\mathbb{L}}_{0-1}(\pi', P, Q)\varphi_{l,\pi}(\pi')d\pi'\right)\gamma_f(\pi)d\pi \\
&= \int_0^1 \Delta\underline{\mathbb{L}}_{0-1}(\pi', P, Q)\left(\int_0^1 \varphi_{l,\pi}(\pi')\gamma_f(\pi)d\pi\right)d\pi' \\
&= \int_0^1 \Delta\underline{\mathbb{L}}_{0-1}(\pi, P, Q)\gamma_{f,l}(\pi)d\pi,
\end{aligned} \tag{5.31}$$

where $\varphi_{l,\pi}(\pi')$ is the weight function corresponding to f_l^π, as given by Eq. (5.12)

$$\varphi_{l,\pi}(\pi') = \frac{1}{\pi'^3} f_l^{\pi\prime\prime}\left(\frac{1 - \pi'}{\pi'}\right) \tag{5.32}$$

This way, the (f,l)-divergence has now the aspect of a standard f-divergence in integral form, with a new weight function $\gamma_{f,l}(\pi)$. This function is obtained by a linear integral operator T_l acting on the original weights $\gamma_f(\pi)$:

$$\gamma_{f,l}(\pi) = \int_0^1 \varphi_l(\pi,\pi')\gamma_f(\pi')d\pi' = \left(T_l\gamma_f\right)(\pi), \tag{5.33}$$

where $\varphi_l(\pi,\pi')$ acts as the kernel of the transformation. We have written it as an explicit function of two variables for consistency with the usual nomenclature for integral operators, and will keep this convention in the following.

Whenever the integral in Eq. (5.33) converges, we obtain an equivalent f-divergence representation of the (f,l)-divergence, so both divergences are intrinsically the same one, but expressed on different bases. This allows estimation of some standard f-divergences by using statistical informations under adequate surrogate losses. These surrogate statistical informations can be then obtained using classifiers. In the next section we delve deeper into this aspect, showing how the Nearest Neighbor classifier can be used to estimate KL divergences.

Note that [X.L. Nguyen and Jordan, 2009] and [Reid and Williamson, 2009] connect losses and f-divergences by associating a loss l with a divergence with $f = f_l^{\frac{1}{2}}$. That can be seen to be a particular case of (f,l)-divergences when f is chosen to represent the *variational divergence* V, since $\gamma_V \propto \delta(\pi - \frac{1}{2})$.

5.5.5 Leveraging the NN rule for divergence estimation

Given the nice properties of the NN rule, in this section we study its applicability to divergence estimation. Recall from above the close relationship between the NN error rate and the Bayes risk under the square loss. This way, we can study the viability of NN-based divergence estimation by analizing the (f,l)-divergences induced by the square loss.

Using Eq. (5.28) we can get the f function associated to the statistical information under the square loss:

$$
\begin{aligned}
f^{\pi}_{SQ}(t) &= \underline{L}_{SQ}(\pi) - (\pi t + 1 - \pi)\underline{L}_{SQ}\left(\frac{\pi t}{\pi t + 1 - \pi}\right) \\
&= \pi(1-\pi) - (\pi t + 1 - \pi)\frac{\pi t}{\pi t + 1 - \pi}\frac{1-\pi}{\pi t + 1 - \pi} \qquad (5.34) \\
&= \pi(1-\pi) - \frac{\pi(1-\pi)t}{\pi t + 1 - \pi} \qquad (5.35)
\end{aligned}
$$

The weight function of the integral representation of $\mathbb{I}_{f^{\pi}_{SQ}}$ can be obtained by plugging in the above result into Eq. (5.32):

$$
\begin{aligned}
\varphi_{SQ}(\pi, \pi') &= \frac{1}{\pi'^3} {f^{\pi}_{SQ}}''\left(\frac{1-\pi'}{\pi'}\right) \\
&= 2(1-\pi')^2\pi'^2\frac{1}{(\pi'(1-2\pi)+\pi)^3} \qquad (5.36)
\end{aligned}
$$

Let us now apply this operator to find the f-divergence equivalent of some square loss-induced divergences. With some hindsight, we start with Jeffreys (J) divergence, which is a symmetrized version of the KL divergence. According to Table 5.1, the weight function corresponding to the J divergence is given by

$$
\gamma_J(\pi) = \frac{1}{\pi^2(1-\pi)^2}. \qquad (5.37)
$$

We then get this very simple and interesting expression for the final weights

$$
\begin{aligned}
\gamma_{J,SQ}(\pi) &= (T_{SQ}\gamma_J)(\pi) = \int_0^1 \varphi_{SQ}(\pi, \pi')\gamma_J(\pi')d\pi' \\
&= 2\int_0^1 \frac{1}{(\pi'(1-2\pi)+\pi)^3}d\pi' \\
&= 2\frac{1}{2(2\pi-1)}\left.\frac{1}{(\pi'+\pi-2\pi\pi')^2}\right|_{\pi'=0}^{1} \\
&= \frac{1}{\pi^2(1-\pi)^2} = \gamma_J(\pi), \qquad (5.38)
\end{aligned}
$$

that is to say, the weight function associated with the f-divergence equivalent of the (J, SQ)-divergence is exactly the same weight function of the standard Jeffrey's divergence:

$$
\mathbb{I}_J(P,Q) = \int_0^1 \Delta\underline{\mathbb{L}}_{0-1}(\pi, P, Q)\gamma_J(\pi)d\pi = \int_0^1 \Delta\underline{\mathbb{L}}_{SQ}(\pi, P, Q)\gamma_J(\pi)d\pi = \mathbb{I}_{J,SQ}(P,Q)
$$

An analogous result holds for the KL divergence, whose weights are given by

$$\gamma_{KL}(\pi) = \frac{1}{\pi^2(1-\pi)}, \tag{5.39}$$

and the corresponding weights for the (KL, SQ)-divergence are

$$\begin{aligned} \gamma_{KL,SQ}(\pi) &= (T_{SQ}\gamma_{KL})(\pi) = \int_0^1 \varphi_{SQ}(\pi,\pi')\gamma_{KL}(\pi')d\pi' \\ &= 2\int_0^1 \frac{1-\pi}{(\pi'(1-2\pi)+\pi)^3}d\pi' = \frac{1}{\pi^2(1-\pi)} = \gamma_{KL}(\pi), \end{aligned} \tag{5.40}$$

These results imply that the weight functions for both KL and Jeffrey's divergences are eigenfunctions of the integral operator T_{SQ} with eigenvalue 1. We say that both KL and Jeffrey's are *eigendivergences* of the square loss. The result may seem quite counterintuitive, which adds to its interest. The square loss has been previously linked to triangular discrimination [Reid and Williamson, 2009], since it coincides (up to scaling) with the divergence $\mathbb{I}_{f^{\frac{1}{2}}_{SQ}}$ associated to $\Delta\underline{\mathbb{L}}_{SQ}(\frac{1}{2}, P, Q)$. We can recover the link between triangular discrimination and square loss as a particular instance of our general loss-induced divergences framework by using the variational divergence under the squared loss $\mathbb{I}_{V,SQ}$. Applying the integral operator associated to the square loss to the weight function of the variational divergence $\gamma_v = 4\delta(\pi - \frac{1}{2})$ we get:

$$\begin{aligned} \gamma_{V,SQ}(\pi) &= (T_{SQ}\gamma_V)(\pi) = 4\left(T_{SQ}\delta(\pi - \frac{1}{2})\right)(\pi) \\ &= 4\varphi_{SQ}(\pi, \frac{1}{2}). \end{aligned}$$

Using the value of φ_{SQ} from Eq. (5.36) yields:

$$\gamma_{V,SQ}(\pi) = 4 \cdot 1 = 4,$$

which is a scaled version of the integral weight function for the triangular discrimination divergence $\gamma_\Delta(\pi) = 8$ (see Table 5.1).

On the contrary, many (f, SQ)-divergences can not be realized as standard f-divergences. Consider for example the (χ^2, SQ)-divergence. Since $\gamma_{\chi^2}(\pi) = \frac{1}{\pi^3}$, applying the integral operator to try to express it as an f-divergence yields

$$\begin{aligned}\gamma_{\chi^2,SQ}(\pi) &= (T_{SQ}\gamma_{\chi^2})(\pi) = \int_0^1 \varphi_{SQ}(\pi,\pi')\frac{1}{\pi'^3}d\pi' \\ &= 2\int_0^1 \frac{(1-\pi')^2}{\pi'}\frac{1}{(\pi'(1-2\pi)+\pi)^3}d\pi',\end{aligned}$$

which diverges, showing that the (χ^2, SQ)-divergence is not an f-divergence. This also shows that the (f,l)-divergence family is strictly larger than the f-divergence family.

The above results relating the (KL, SQ)-divergence with the standard KL-divergence is actually telling us that it is possible to estimate KL divergences using nearest-neighbor risks. Without this result, the obvious way of using the integral representation to estimate an f-divergence would be to plug-in a consistent classifier (such as k-NN with an adequate election of k) to estimate full 0-1 Bayes risks. Most recent proposals for KL divergence estimation [Nguyen et al., 2008, Wang et al., 2009, Suzuki et al., 2009] rely on direct estimation of the likelihood ratio [Nguyen et al., 2008, Wang et al., 2009, Suzuki et al., 2009], and thus of the posterior class probabilities, avoiding individual density estimations. Our proposal avoids any explicit density and likelihood ratio or posterior estimation. Instead, we have shown that it is possible to use the risk of a simple, non-consistent classifier such as NN to obtain an error-rate based exact expression for the KL divergence (and, even funnier, using the exact same weight function as we would use with the full Bayes risks).

Estimating the NN error

Now that we have a expression relating the Bayes risk under the square loss (and thus, the NN error rate) with the KL and Jeffreys divergences we are close to proposing a novel way of estimating those divergences in a non-parametric way. From an empirical estimation point of view, the problem remains to obtain good estimates of the NN error rate for the whole range of prior probabilities $\pi \in [0,1]$. If the sample size for each class is really large, a *hold-out* set could be used to evaluate the risks, with the precaution of maintaining the correct proportions for each π. Since data is usually scarce, in practice it is usually required to perform *training*

set error estimation, that is to say, estimate the error without explicitly separating the samples into training and hold-out sets. A standard way to do so would be to perform empirical resampling of the different sequences as in the standard cross-validation (CV) procedure, but with additional constraints. These constraints must force the ratio of points from each sequence in the training and test sets to comply with the desired π. This is called *stratified cross-validation* [Kohavi, 1995].

However, the particularities of the NN rule can be exploited to obtain closed-form estimates of the error rate. We will now show a couple of alternatives to carry out that estimation in an efficient manner. The first alternative is to perform *complete stratified cross validation* [Mullin and Sukthankar, 2000]. As mentioned in Section 5.2.1, the idea behind complete cross-validation is to average over all the possible test/train partitions of data. That is to say, instead of resorting to empirical resampling, and approximating the expectation over the partitions by the average over the actual sampled partitions, obtain a closed-form expression for the expectation. This can be done because of the nature of the NN rule, which makes it relatively easy to know how many of the potential partitions will result in a correct or incorrect classification of a given point. Nonetheless, running this process for a dense enough set of π's is a very time consuming process.

To speed things up we have devised a simple "closed-form sampling scheme", specially tailored for the task of estimating risks over the whole range of prior probabilities, which we now sketch. The main idea is to subsample just one of the sets, depending on π. Assume we are given two sets $\mathbf{X}$ and $\mathbf{Y}$, with n_X and n_Y elements, so the estimated prior probability is just $\pi_0 = \frac{n_X}{n_X+n_Y}$. The error for $\pi = \pi_0$ can be estimated using standard methods such as deleted estimate ([Devroye et al., 1996] Chap. 24), yielding error estimates in $\{0, 1\}$ for each point $z \in (\mathbf{X} \cup \mathbf{Y})$. In order to obtain error estimates for $\pi \neq \pi_0$, our proposal is to calculate the expectation of the probability of error at each point z given that we are subsampling $\mathbf{X}$ if $\pi < \pi_0$ or $\mathbf{Y}$ if $\pi > \pi_0$. We can obtain such expectation just by knowing the order of the closest point to z in both $\mathbf{X}$ and $\mathbf{Y}$ and calculating the ratio of partitions that result in the point changing its label with respect to case $\pi = \pi_0$. For example, consider the case of a point $z \in \mathbf{X}$ which is correctly classified for $\pi = \pi_0$, and whose closest

point in $\mathbf{Y}$ occupies the $k_Y(z)$ position in the ordered list of neighbors. Let $n_s(\pi)$ be the number of points from $\mathbf{X}$ that must be taken away for the desired π to hold. Point z will become incorrectly classified whenever its nearest neighbor after subsampling belongs to $\mathbf{Y}$. That is to say, whenever the $k_Y(z) - 1$ first neighbors of z are taken away from $\mathbf{X}$. This is a typical sampling-without-replacement scenario, and the probability of such an event is given by the hypergeometric distribution, yielding

$$P_e(z;\pi) = \frac{\binom{n_x-(k_Y(z)-1)}{n_s(\pi)-(k_Y(z)-1)}}{\binom{n_x}{n_s(\pi)}}. \tag{5.41}$$

The reasoning is similar for points which are originally incorrectly classified: they can become correctly classified if we take away from the adversarial class its closest neighbors. Note that this method is based solely on the order of the neighbors of each point.

Figure 5.2 displays some different error estimates for $P = \mathcal{N}(0,1)$ and $Q = \mathcal{N}(1,1)$ and a sample size of 1000 instances per class. Specifically, we compare our closed-form sampling (CFS) with two different empirical sampling schemes: Scheme A randomly selects subsets of a fixed size maintaining the proportions imposed by each π. Scheme B is the empirical analogue of our closed-form estimator: it enforces the desired proportion between the two classes by subsampling from the minority class. We also include the theoretical NN error rate for that sample. The graphic summarizes well the usual behaviour: all the estimates behave very similarly and are quite close to the theoretical error for reasonable sample sizes, so the election is a matter of computational efficiency. In this aspect, our proposed closed-form sampling scheme has been empirically shown to perform much faster than both stratified CV or empirical resampling.

Risk-based bounds of KL and Jeffreys divergences

Finally, upper bounds on the NN error rate can be used to obtain lower bounds on the estimated divergences. For example, consider the following result from [Devijver and Kittler, 1982]

Lemma 5.5.6 (Devijver and Kittler (1982, p.166))**.** *For all distributions P, Q over*

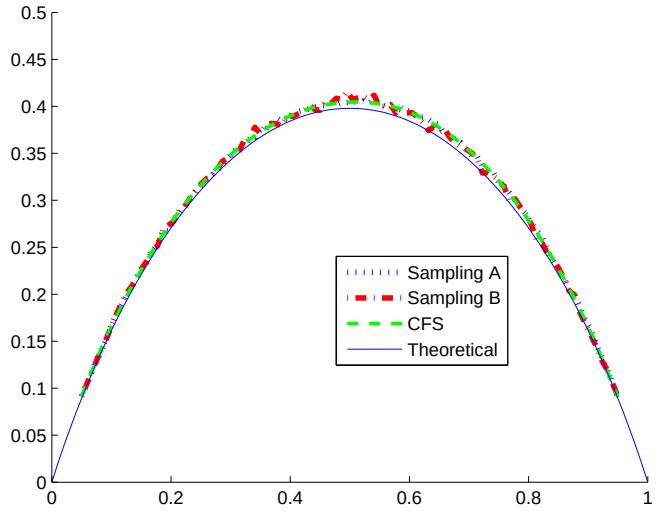


Figure 5.2: $\mathbb{L}_{0-1}^{NN}(P,Q)$ estimates as a function of $\pi \in [0.05, 0.95]$, $P = \mathcal{N}(0,1)$ and $Q = \mathcal{N}(1,1)$.

$\mathcal{X}$ with finite second moment we have

$$\mathbb{L}_{0-1}^{NN}(\pi, P, Q) \leq \frac{2\pi(1-\pi)}{1+\pi(1-\pi)\Delta^2(\pi, P, Q)},$$

where Δ stands for the Mahalanobis distance between P and Q with mixture parameter π

$$\Delta(\pi, P, Q) = \sqrt{(\mu_p - \mu_q)^T \Sigma^{-1} (\mu_p - \mu_q)},$$

with $\Sigma(\pi, P, Q) = \pi\Sigma_p + (1-\pi)\Sigma_q$, $\Sigma_p = \mathbb{E}[(x-\mu_p)(x-\mu_p)^T]$ and analogously for Σ_q.

Recalling the connection between NN error rate and square loss we can trivially get the following corollary:

Corollary 5.5.7. *For all distributions P, Q over $\mathcal{X}$ with finite second moment we have*

$$\underline{\mathbb{L}}_{SQ}(\pi, P, Q) \leq \frac{\underline{\mathbb{L}}_{SQ}(\pi)}{1+\underline{\mathbb{L}}_{SQ}(\pi)\Delta^2(\pi, P, Q)},$$

and

$$\Delta\underline{\mathbb{L}}_{SQ}(\pi, P, Q) \geq \underline{\mathbb{L}}_{SQ}(\pi)\left[1 - \frac{1}{1+\underline{\mathbb{L}}_{SQ}(\pi)\Delta^2(\pi, P, Q)}\right],$$

where $\underline{\mathbb{L}}_{SQ}(\pi) = \pi(1-\pi)$ *is the prior Bayes risk under the square loss.*

We can plug this result into Eq. (5.23) to get a lower bound for the KL divergence in terms of the Mahalanobis distance:

$$\begin{aligned}
\mathbb{I}_{KL}(P,Q) &= \mathbb{I}_{KL,SQ}(P,Q) = \int_0^1 \Delta\underline{\mathbb{L}}_{SQ}(\pi,P,Q)\gamma_{KL}(\pi)d\pi \\
&\geq \int_0^1 (\pi(1-\pi))\left[1 - \frac{1}{1+\pi(1-\pi)\Delta^2(\pi,P,Q)}\right]\frac{1}{\pi^2(1-\pi)}d\pi \\
&= \int_0^1 \frac{(1-\pi)\Delta^2(\pi,P,Q)}{1+\pi(1-\pi)\Delta^2(\pi,P,Q)}d\pi, \qquad (5.42)
\end{aligned}$$

and an analogue lower bound for the Jeffreys divergence:

$$\begin{aligned}
\mathbb{I}_{J}(P,Q) &= \mathbb{I}_{J,SQ}(P,Q) = \int_0^1 \Delta\underline{\mathbb{L}}_{SQ}(\pi,P,Q)\gamma_{J}(\pi)d\pi \\
&\geq \int_0^1 (\pi(1-\pi))\left[1 - \frac{1}{1+\pi(1-\pi)\Delta^2(\pi,P,Q)}\right]\frac{1}{\pi^2(1-\pi)^2}d\pi \\
&= \int_0^1 \frac{\Delta^2(\pi,P,Q)}{1+\pi(1-\pi)\Delta^2(\pi,P,Q)}d\pi. \qquad (5.43)
\end{aligned}$$

5.5.6 Further generalization: Classifier-induced divergences

Putting together the class-restricted and loss-induced divergences, we get to a very general expression which can encompass divergences induced by classifiers. Most classifiers can be interpreted as minimizing a surrogate loss [Bartlett et al., 2006] over a family of classification functions, so the natural f-like divergence induced by the risks of such classifiers can be defined as follows:

$$\mathbb{I}^C_{f,l} = \int_0^1 \Delta\underline{\mathbb{L}}^C_l(\pi,P,Q)\gamma_f(\pi)d\pi, \qquad (5.44)$$

where l stands for the surrogate loss minimized by the chosen classifier, and C represents the subset of functions over which the minimization is carried out. For example, a Support Vector Machine (SVM) classifier would induce such a divergence with C being the set of linear classifiers in some kernel-induced feature space, and l being the hinge loss. The study of such expressions is quite complex, and stands as a very promising work in progress.

5.5.7 Experimental results

To bridge the gap between theory and practice, in this section we will study the square loss-based divergence measures in both synthetic and real-world applications. Given the enormous flexibility of the (f, l)-divergence framework, we have to restrict ourselves to some particular case. Specifically, we will focus on using the Nearest Neighbor classifier to estimate KL divergences, since that is one of the most straightforward application of the theoretical results. Moreover, given a KL estimation procedure it is straightforward to extend it to mutual information, cross entropy and other information-theoretic magnitudes estimation.

Starting from the expression of the KL divergence in terms of NN risk:

$$\begin{aligned}\mathbb{I}_{KL}(P,Q) &= \int_0^1 \Delta\underline{\mathbb{L}}_{SQ}(\pi,P,Q)\gamma_{KL}(\pi)d\pi \\ &= \frac{1}{2}\int_0^1 \Delta\underline{\mathbb{L}}_{0-1}^{NN}(\pi,P,Q)\gamma_{KL}(\pi)d\pi \\ &= \frac{1}{2}\int_0^1 \left(2\pi(1-\pi)-\mathbb{L}_{0-1}^{NN}(\pi,P,Q)\right)\gamma_{KL}(\pi)d\pi, \end{aligned} \tag{5.45}$$

we have devised a naïve estimation procedure, consisting on quadrature integration with uniform sampling of $\pi \in [\pi_{min}, \pi_{max}]$. A more sophisticated approach could be taken by using some kind of importance sampling depending on the weight function γ_{KL}. The error rates $\mathbb{L}_{0-1}^{NN}$ at each π are estimated using our procedure described in Section 5.5.5. The thresholds on π can be used in a way akin to the usual assumption in divergence estimation that the likelihood ratio is bounded and do not fall below a given threshold. Statistical informations outside these thresholds are assumed to be 0, effectively regularizing the divergence estimate. In our experiments we fix $\pi_{min} = 10^{-3}, \pi_{max} = 1 - 10^{-3}$. We denote this non-parametric estimator NN-KL. The same approach has been used for obtaining an estimator of the bound in Eq. (5.42), yielding algorithm NNbound-KL.

KL divergence estimation

Our benchmark for divergence estimation is the proposal in [Wang et al., 2009], which is arguably the state-of-the-art in non-parametric estimators for KL divergences. It is mainly based on direct estimation of the likelihood ratio $\frac{dP}{dQ}$ at each

point using nearest-neighbor distances. Assume we have a sample $X = \{x_1, \ldots, x_{n_X}\}$ coming from the distibution P and another sample $Y = \{y_1, \ldots, y_{n_Y}\}$ coming from Q. The estimator can be written as

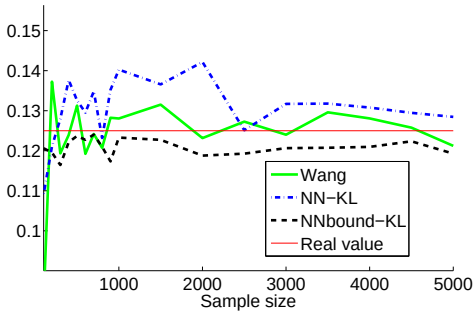
$$\hat{\mathbb{I}}_{KL}(P, Q) = \frac{d}{n} \sum_{i=1}^{n_X} \log \frac{\nu_k(x_i)}{\rho_k(x_i)} + \log \frac{n_Y}{n_X - 1},$$

where $\nu_k(x)$ and $\rho_k(x)$ are the distances from x to its k-th nearest neighbor in Y and X respectively, and d is the dimension of the data. This algorithm was shown to outperform previous proposals, like divergence estimation based on data-dependent partitions [Wang et al., 2005] or direct kernel plug-in estimates. In our experiments we haves used $k = 1$. For more details and convergence results please refer to [Wang et al., 2009] and [Perez-Cruz, 2008].

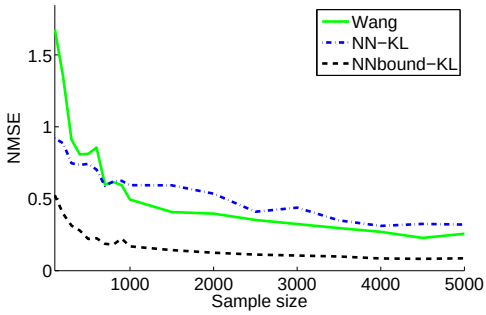
We have run the algorithms in synthetic datasets comprised of samples from Gaussian distributions of different dimensionalities and separations, with unit covariance matrices. Figures 5.5.7 and 5.5.7 show plots of mean divergence and normalized mean square error (NMSE) (both averaged over 100 runs) using a separation of $0.5\mathbf{e}_D$ (where $\mathbf{e}_D$ is the unit vector in $\mathbb{R}^D$) and $0.75\mathbf{e}_D$ for different dimensionalites $D = \{1, 5, 10\}$. The NN-KL estimator improves its performance in comparison with both the Wang estimator and the risk-based lower bound as the dimensionality increases. It coincides with the intuition that, in high-dimensional settings, it may be easier to estimate classification risks than likelihood ratios. The abrupt change in MSE slope of the NN-KL estimator in Fig.5.4f is due to the thresholds on π limiting the divergence estimate.

Figure 5.5 shows the results for KL divergence estimation between samples from Gaussian distribution $\mathcal{N}(\mathbf{0}, \mathbf{I}_3)$ and a uniform distribution Unif$[-3, 3]^3$. In this case, the Mahalanobis-based bound for the KL divergence is totally useless, since both distributions have the same mean. The Wang estimator achieves an impressive performance in this scenario. Nonetheless, the NN-error based estimator remains competitive, specially in terms of MSE.

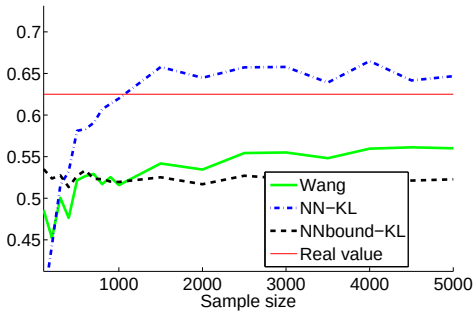
In general terms, our proposed estimator appears to be very competitive with the state of the art, showing that risk-based estimation of divergence measures is a promising line to explore in depth.



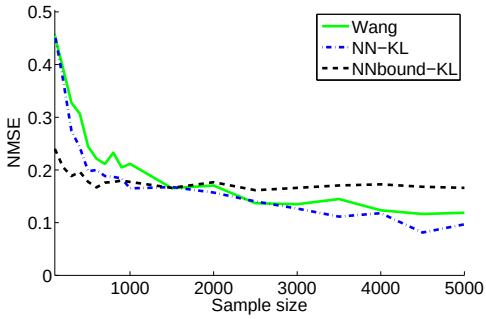
(a) Mean, D=1



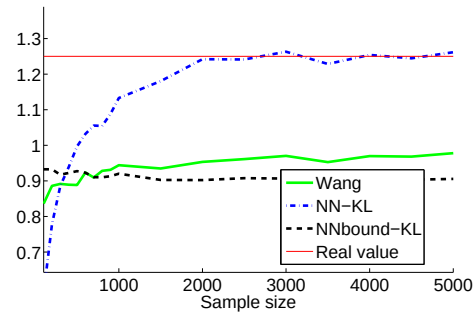
(b) NMSE, D=1



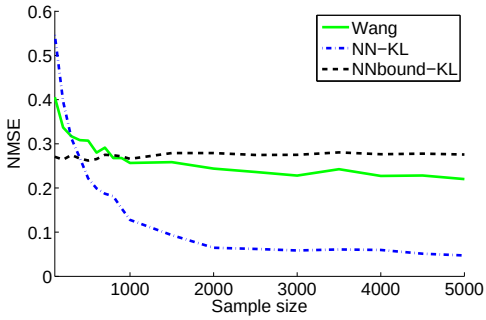
(c) Mean, D=5



(d) NMSE, D=5

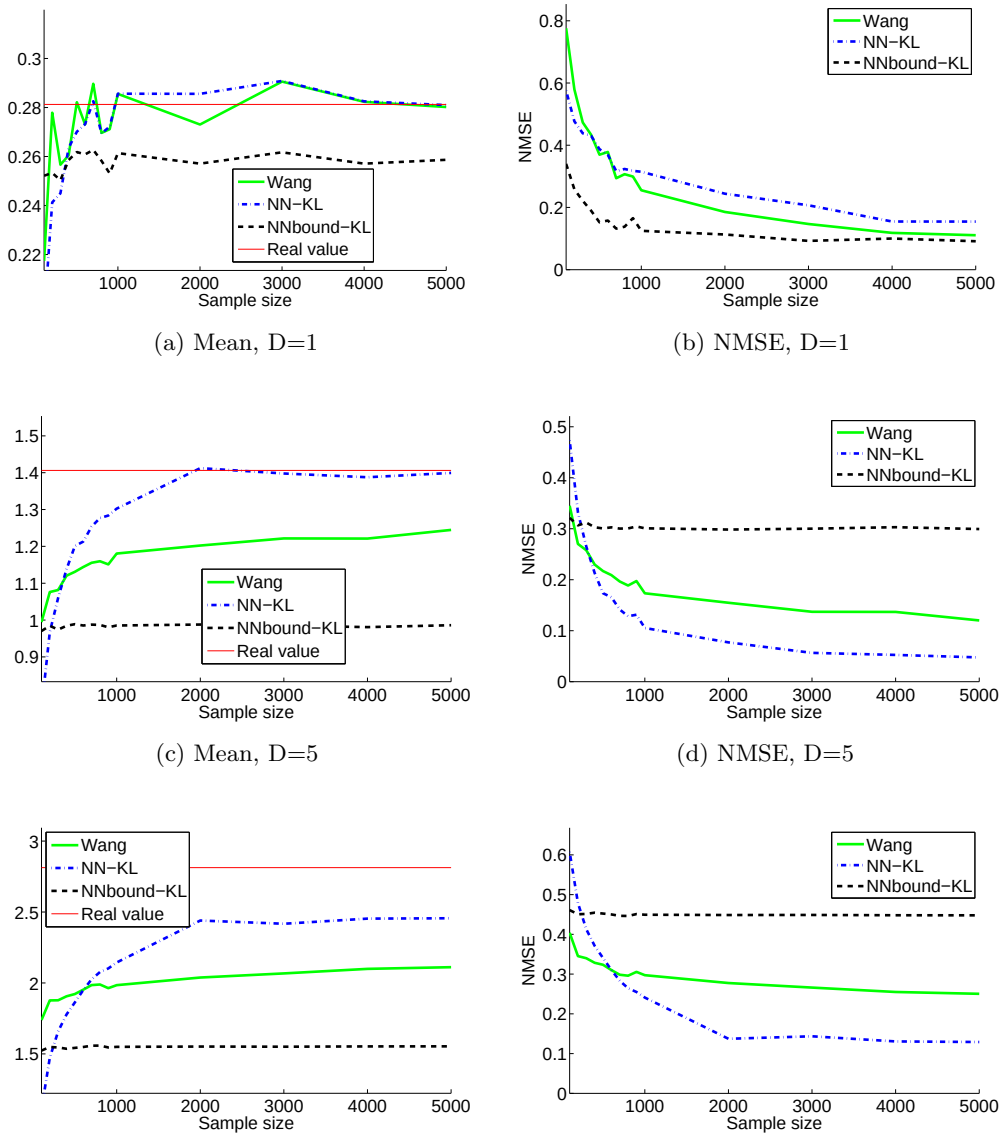


(e) Mean, D=10



(f) NMSE, D=10

Figure 5.3: NMSE and bias of the different estimators of $\mathrm{KL}(P, Q)$ divergence, $P = \mathcal{N}(\mathbf{0}, \mathbf{I}_D)$, $Q = \mathcal{N}(\frac{1}{2}\mathbf{e}_D, \mathbf{I}_D)$.



(a) Mean, D=1

(b) NMSE, D=1

(c) Mean, D=5

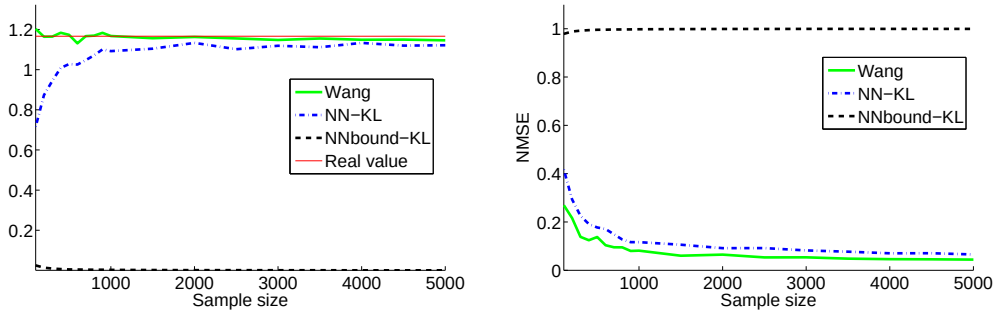
(d) NMSE, D=5

(e) Mean, D=10

(f) NMSE, D=10

Figure 5.4:]

NMSE and bias of the different estimators of KL(P,Q) divergence, $P = \mathcal{N}(\mathbf{0}, \mathbf{I}_D)$, $Q = \mathcal{N}(0.75\mathbf{e}_D, \mathbf{I}_D)$.



(a) Mean (b) NMSE

Figure 5.5: NMSE and bias of the different estimators of $\mathrm{KL}(P, Q)$ divergence, $P = \mathcal{N}(\mathbf{0}, \mathbf{I_3})$, $Q = \mathrm{Unif}[-3, 3]^3$.

Speaker clustering

Here we are interested in knowing whether the use of risk-based divergences can improve clustering performance by providing a more flexible representation of the input space, even though the dynamics are discarded. Since spectral clustering works with symmetric affinity matrices, a natural choice for the divergence is Jeffreys (or symmetric KL) divergence. Moreover, we also introduce into the comparison the maximum mean discrepancy (algorithm MMD) [Gretton et al., 2007], which was discussed in Section 4.2.2 and is a widely known measure of disimilarity based on RKHS embeddings of distributions. We use Gaussian kernels for those embeddings. As for the width parameter of the kernel, we sweep it within a sensible range and report the best performance. Finally, we also report clustering results using an affinity matrix based on simple nearest neighbor risks (algorithm NN). We obtain the risks running a complete cross-validation procedure [Mullin and Sukthankar, 2000] with a training set size parameter of 50%. The result from the sphere-packing (SPH) algorithm of Chapter 4 is also reproduced.

We simulate a 9-class speaker clustering scenario using the UCI Japanese Vowels dataset [Hettich and Bay,], and a 2-class task using the GPM PDA dataset. Table 5.2 holds the results of these tasks. Please refer to Appendix D for details on these datasets. Japanese Vowels is an easy dataset, and most algorithms perform quite well. It is remarkable how the three NN-error based algorithms give the best

	KL-LL	SSD	NN	NN-J	Wang-J	NN Bound-J	MMD	SPH
JV	9.85%	12.07%	8.15%	10.00%	16.30%	7.41%	20.37%	10.00%
GPM	9.65%	10.71%	16.67%	9.38%	18.22%	7.15%	9.57%	10.56%

Table 5.2: Clustering error for the **speaker clustering tasks** on the Japanese Vowels (JV, 9-class) and GPM (2-class) datasets

performance. In this particular case, the added expresiveness of KL divergence does not compensate the more complex (and noisier) estimation procedure, as can be seen by the Mahalanobis-based bound achieving the highest performance, followed by the simple NN risk. On the GPM dataset, the performance of the NN-based bound of the Jeffreys divergence is once again excellent, notably improving over the best-performing method from previous comparisons. The NN-based estimator also perfoms optimally, beating all previous approaches. All in all, our proposed estimator and bound perform impressively in the speaker clustering tasks.

We will evaluate our proposals in a more challenging scenario in Chapter 6.

5.5.8 Summary

The framework of (f, l)-divergences allows the generalization of standard f-divergences by substituting the 0-1 loss for any other loss l. If a sensible l is chosen, most properties of f-divergences are preserved. We have made explicit how some (f, l)-divergences are equivalent to f-divergences where the integral weights have undergone a transformation defined by the chosen l. Specifically, when the square loss is used, KL and Jeffreys divergences are equivalent to their standard counterparts, since their weight functions are eigenfunctions of the corresponding integral transform. Together with a result relating Nearest-Neighbor classification error and Bayes risk under the square loss, this allows for a new way of estimating or bounding these important divergences. Estimators based on this idea have proven to be useful in practice, but there is still a lot of room for improvement in terms of speed. On a more theoretical standpoint, analysis of the integral operators induced by several losses must be performed in order, for example, to determine what standard f-divergences can be estimated using a given surrogate loss. This is related to the inversion of the

integral operator.

Chapter 6

Musical genre recognition

We present an exciting real-world application of the affinity measures that we have developed during this Thesis: musical genre recognition. It is a quite complex scenario (high dimensions, scarce samples per sequence, etc.), making it a very hard and interesting testbed for the developed affinity measures.

6.1 Introduction and chapter structure

Nowadays, there exists an increasing interest on automatic methods for organization and navigation of music collections, which is mainly motivated by the extensive availability of music in digital format. Music distribution is no longer limited to physical media, and many users have become frequent clients of on-line services such as *Amazon* or *iTunes*, and download music titles using standard encoding formats such as MP3 or AAC. Furthermore, the capacity of current portable players allows users to store their personal music collections, and carry them everywhere. In this context, automatic methods that infer similarity between music pieces are very valuable for the development of tools that help users organize and share their music. They can also serve to increase the effectiveness of current recommender systems, thus improving users' experience when using these services.

It is obvious that music similarity can be defined in many different ways: For instance, we may be interested on extracting beat, instrumental or genre information,

just to name a few examples. We may expect that different similarities lie at different time scales. For instance, perceptual features such as the beat can be studied directly from short windows, while extraction of the genre information will typically require the examination of the whole music piece. In this paper, we are interested on the latter case, and will propose and analyze different metrics to be exploited in a genre classification task.

A typical approach to implement genre classifiers (see [Tzanetakis and Cook, 2002, Meng et al., 2007, Guaus, 2009], among others) is splitting the audio signal into several (typically overlapping) windows of short duration (10-50 miliseconds), from which the so-called *short-time audio features* are extracted. Some examples of such features are Mel Frequency Cepstral Coefficients (MFCCs) or the Zero Crossing Rate (ZCR) [Müller, 2007]. However, in spite of being related to perceptual characteristics of the audio signal, short-time features do not contain much valuable information about the genre. Therefore, in order to obtain a better classification accuracy, a temporal feature integration process is carried out [Meng et al., 2007, McKinney and Breebart, 2003], summarizing all short-time features extracted in longer windows of about one second. The new features obtained at this larger time scale, which are more correlated with the music genre information, are then feeded into a supervised classifier, and the genre is extracted via majority or weighted voting among all classifier outputs for the segments in one song. In spite of considering to some extent the temporal structure of the music, the previous procedure fails at capturing the whole dynamics characterizing each song, which include not just one or several seconds, but the whole piece of music. The idea behind this is that songs are characterized not only by the musical motifs present in them, but also by the way these motifs evolve along time.

In this Chapter we use this interesting problem to further evaluate the methods proposed in this Thesis. We work from a fully unsupervised perspective, so our methods do not require any intermediate segment classification step. Moreover, we consider songs as a whole, so there is no voting of the individual segments: the algorithms directly produce a single label for each sequence/song.

Specifically, we want to find out if high-level dynamics can be leveraged to improve the genre recognition procedure. To this end, we will compare the performance of the SSD distance (Chapter 3) with a statical counterpart. We are also interested in testing the non-parametric methods presented in the second half of the Thesis (Chapters 4 and 5). Those methods discard the dynamics, but are much more expressive in input space. Is that added expresiveness more important than the dynamics? Which method will finally get the best performance in this challenging scenario?

6.1.1 Related publications

Some of the results in this paper appeared originally in [García-García et al., 2010].

6.2 Song modelling

We use the song representation introduced in [Meng et al., 2007]. Previous to the classification phase, each song is pre-processed, extracting audio features at two different time levels:

- Short-time feature extraction: First, MFCCs are extracted in overlapped windows of short duration. These parameters were originally developed for automatic speech recognition tasks, but they have also been extensively applied in Music Information Retrieval (MIR) tasks [Sigurdsson et al., 2006] with generally good results. These features are inspired by the auditory perception of humans, and contain information about the variations in the spectral envelope of the audio signal.

  For this work, we have used the MFCC implementation of [Sigurdsson et al., 2006], using a bank with 30 filters, and keeping just the initial 6 coefficients (however, the first coefficient, which is associated to perceptual dimension of loudness, is discarded). The window size and hopsize have been fixed to 30 and 15 ms, respectively. Thus, a fragment of 60 seconds of music would be represented by a matrix of size 4000×6 after MFCC feature extraction.

- Temporal feature integration: It is well-known that the direct use of MFCCs does not provide an adequate song representation for music genre recognition tasks. Thus, a time integration process is needed in order to recover more relevant information. As an alternative to simpler procedures, such as using the mean and variance of the MFCCs, [Meng et al., 2007] proposed to adjust a Multivariate Autoregressive (MAR) model. To be more specific, for a set of consecutive MFCCs vectors, a MAR model of lag P is adjusted using the formula $\mathbf{z}_j = \sum_{p=1}^{P} \mathbf{B}_p \mathbf{z}_{j-p} + \mathbf{e}_j$, where $\mathbf{z}_j$ are the MFCCs extracted at the j^{th} window, $\mathbf{e}_j$ is the prediction error, and $\mathbf{B}_p$ are the model parameters. The values of matrices $\mathbf{B}_p$, $p = 1, \ldots, P$, together with the mean and covariance of the residuals $\mathbf{e}_j$ are concatenated into a single feature vector (MAR vector).

  In this paper, we have considered MAR models of order $P = 3$, resulting in MAR vectors of size 135. For this temporal integration phase, we have considered a window size and a hopsize of 2 and 1 seconds, respectively. Thus, an audio fragment of 60 seconds is represented by a matrix of size 60×135 after time integration.

Previous works have carried out classification at the time scale resulting from the temporal integration, using weighted or majority voting to obtain the final classification of each song. Following a different direction, in this paper we propose metrics which deal with the songs as a whole. Thus, no postprocessing is needed to obtain the final label for each song.

6.2.1 Dataset description

We use a subset of the *garageband* dataset described in [Arenas-García et al., 2007]. The data set consists of snippets of around 60s of 4412 songs downloaded from the online music site `http://www.garageband.com`[1]. The songs are in MP3 format, and belong to different music genres. For the experiments we consider a simplified problem where the goal is to discriminate between four different genres: "Punk", "Heavy Metal", "Classical", and "Reggae". In our opinion this selection is a good

[1] Downloaded in November, 2005.

representation of the dataset, since it includes both genres that are a priori hard to distinguish from one another (Punk and Heavy Metal) and others which are easily separated. Each genre is represented by 100 songs. MFCC and MAR extraction settings are as described in Subsection 6.2.

6.3 Influence of high-level dynamics: SSD vs SSD-ST

To check if paying attention to coarse time dynamics can be beneficial for genre classification, we will compare the standard SSD distance with its steady-state version, which we will denote by SSD-ST. The SSD-ST measure is obtained by letting the markov chains defined by the transition matrices induced by the different sequences reach their equilibrium. It can be interpreted as an extreme example of the diffusion process that we explain in Chapter 3, Section 3.2.2. Note that we do not introduce likelihood-based methods in the comparison since they are not useful for this task. This is due to the number of sequences being high and individual sequences being short and high-dimensional, which implies poor individual models.

The first experiment consists in performing 1vs1 clustering tasks between the selected subset of genres. Results are shown in Table 6.1, where performance is measured in the form of clustering error, understood as the percentage of incorrectly classified samples under an optimal permutation of the cluster labels. For the sake of comparison, the table also shows results obtained using the mean of all MAR vectors of each song as a representative vector. In all cases, the affinity matrices are generated using a gaussian kernel whose width is automatically selected attending to the eigengap [Ng et al., 2002]. The poor performance of the mean-vector approach shows the benefit of using mixture models to represent the individual songs. The stationary distance SSD-ST performs better for classes which are easily separated. The intuition is that for these classes, taking into account the dynamics do not improve the already high separability, while the higher number of parameters involved results in less stable distances and slightly poorer results. However, the hardest pairing ("Punk" vs "Heavy Metal") benefits from using the dynamics, due to the additional discriminative power.

Results of the 4-way clustering are displayed in Table 6.2, using $K = 30$ hidden

Table 6.1: 1 vs 1 clustering error for the chosen genres using K=20 states

	H.M.	Classical	Reggae
Punk	38.1%	9.0%	14.1%
H.M.	-	6.0%	12.7%
Classical	-	-	5.9%

a) SSD distance

	H.M.	Classical	Reggae
Punk	44.7%	8.1%	12.6%
H.M.	-	5.5%	11.7%
Classical	-	-	5.1%

b) SSD-ST distance

	H.M.	Classical	Reggae
Punk	45%	19%	35.5%
H.M.	-	7.5%	49%
Classical	-	-	8.5%

c) Mean vectors

Table 6.2: 4-way clustering error for the chosen genres, SSD vs SST

K	SSD	SSD-SST
25	38.25%	45.75%
30	36.75%	46.00%
40	33.25%	44.25%

states. A clear improvement is obtained in this case when taking into account the dynamics. The presence of classes with high overlap limits the performance of the multi-way clustering, and this overlap is bigger when using stationary distances. This comes to show that while for simpler problems it is enough to look at the probability distribution of the AR coefficients for the different sequences, the added discriminative power of dynamics-based distances comes in handy when handling complex scenarios.

6.4 Input space expressivity: Non-parametric methods

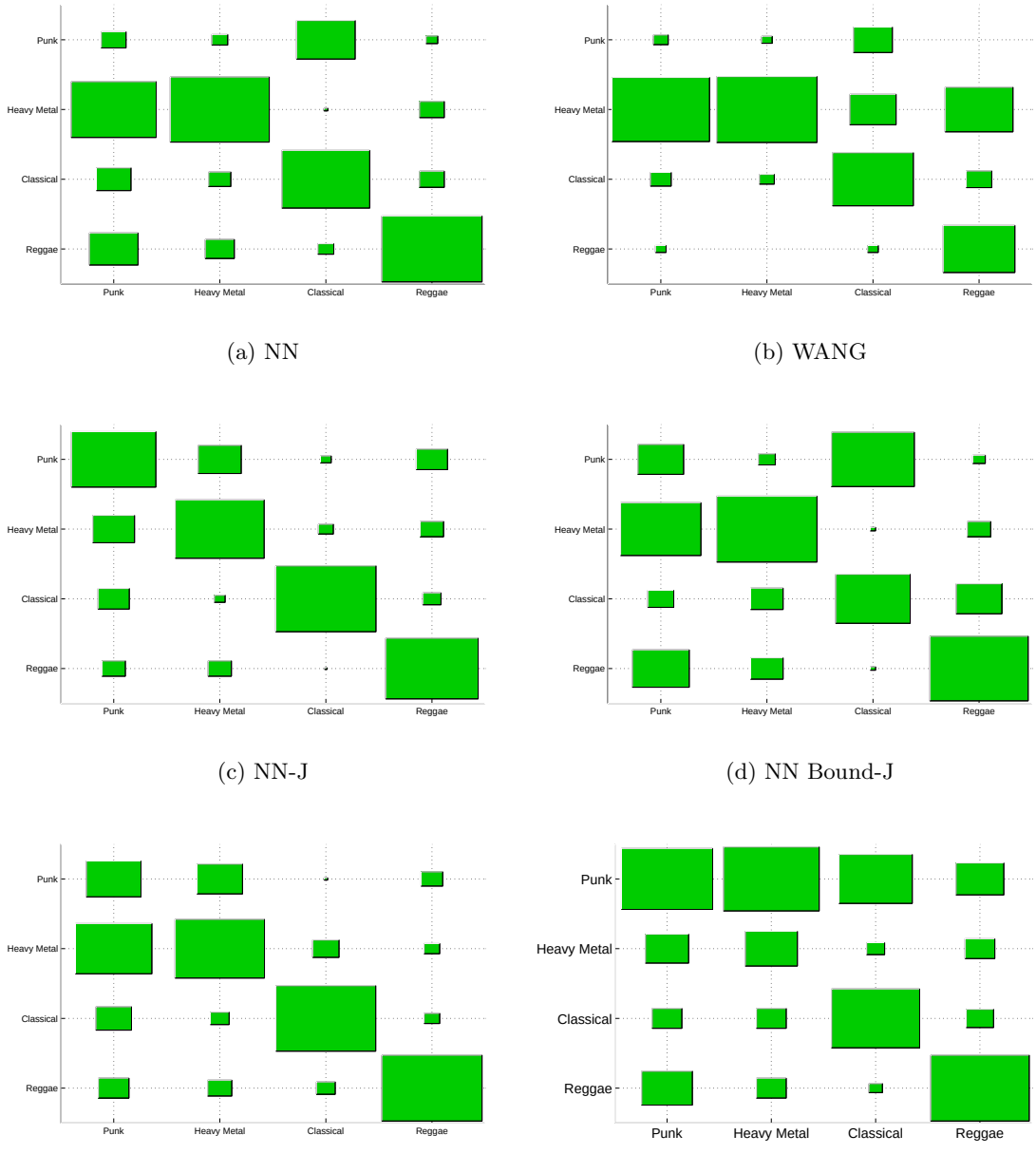
On the second part of the Thesis (Chapters 4 and 5) we have focused on developing non-parametric affinities for sets of vectors. This has the potential drawback of not taking into account the dynamics of the sequences, and the advantage of allowing for a much more flexible view of input space. In this section we apply these methods to the music genre recognition task.

We introduce into the comparison the sphere-packing algorithm (SPH) from Chapter 4, the NN-based estimator (NN-J) and bound (NN Bound-J) of the Jeffreys divergence from Chapter 5, as well as a Jeffreys-divergence estimator based on [Wang et al., 2009] (Wang-J), the MMD distance [Gretton et al., 2007] and the Nearest-Neighbor risk, estimated using complete CV with a training set size of 50% (NN). Both MMD and SPH use Gaussian kernels. The width of those kernels is determined using the same heuristic as in Chapter 4: it is chosen to be the median distance between points in the dataset.

Figure 6.4 provides a graphical view (in the form of Hinton diagrams) of the confusion matrices, and Table 6.3 holds the clustering error of the non-parametric methods, in comparison with the best-performing method of the previous section. There is a clear winner: the best performance is obtained by far when our proposed NN-risk based estimator of the Jeffreys divergence is used. It is remarkable how the other estimator of Jeffreys divergence (Wang-J) produces much worse results. This is likely due to the high dimensionality of the feature vectors, coupled with small sample size. This kind of data is likely to present a manifold structure, so the explicit dependance of the Wang estimator on the data dimensionality hinders its performance, since the actual intrinsic dimensionality is surely much lower than the ambient space dimension. The poor performance of the NN-based bound on the divergence can be explained by noting the complexity of the data, which renders the second-order moments not informative enough. The sphere-packing algorithm also performs poorly, specially in comparison with the MMD distance.

	SSD	NN	NN-J	Wang-J	NN Bound-J	MMD	SPH
Clustering error	33.25%	39.50%	**21.25%**	48.75%	46.00%	31.50%	48.50%

Table 6.3: Clustering error for the 4-way music genre clustering task



(a) NN

(b) WANG

(c) NN-J

(d) NN Bound-J

(e) MMD

(f) SPH

Figure 6.1: Confusion matrices for the different algorithms

6.5 Summary

This chapter focuses on the application of affinity measures for sequences of data to the musical genre recognition task. Standard methods work individually on segments of songs, and then make a final decission by voting. Alternatively, sequence-based affinities view the songs as a whole, and do not need such post-processing steps.

We evaluate whether the high-level dynamics of songs can be leveraged for the recognition procedure, by comparing the SSD distance with a statical counterpart. Experiments show that the dynamics can improve the performance in similar genres, where the instrumentation is similar so the low-level features of the genres overlap a lot.

Besides, we evaluate the non-parametric affinities we have proposed on the second half of the Thesis. The NN-based divergence estimator achieves an impressive performance, showing that it is very appropriate for complex scenarios with high-dimensional data.

Chapter 7

Conclusions

In this Thesis we have presented a set of ideas and methods for defining convenient and meaningful notions of similarity/dissimilarity between sequences of data. We have proposed *model-based measures* that use dynamical models for capturing the relationships between elements in a sequence, in order to introduce that structure in the similarity function. When the dynamics of the sequences are discarded, they become sets of vectors. We propose affinity measures for sets of vectors related to the overlap of their estimated supports in a high-dimensional feature space, and also a variety of classification-risk-based measures.

Regarding the model-based methods, we have presented two different alternatives, namely KL-LL and SSD (Chapters 2 and 3, respectively). KL-LL is a likelihood-based method, originating on a novel look at the likelihood matrix from a model-space perspective. One of the particularities of this method is that it includes information from the whole dataset in every pair-wise distance. We have also proposed a simple scheme to choose a good subset of the pool of models, for improved performance in practical and noisy scenarios. The benefits of KL-LL with this model-selection scheme in comparison with previous likelihood-based approaches are evident based on the empirical comparisons.

The SSD distance (Chapter 3) aims at alleviating the main problems of the likelihood-based methods: poor performance on short sequences due to overfitted models, and a number of required likelihood evaluations that is quadratic on the

dataset size. It works by assuming a common set of emission distributions in a global hidden Markov model for all the sequences in the dataset. Then, different sequences are compared according to the transition matrices that they induce in that model. This way, there are no likelihood evaluations involved. The computational complexity of learning a transition matrix is analogous to that of a likelihood evaluation for an HMM, and the number of matrices to learn is linear in the number of sequences.

We also present two different approaches for the sets-of-vectors scenario. The first of them, in Chapter 4, provides a way to work in a kernel-induced feature space. We provide a geometrical approach consisting on merging spheres learnt to represent the support of the distributions. This procedure can be carried out regardless of the (potentially infinite) dimensionality of the feature space. Estimating the supports is a much simpler problem than estimating the actual distributions. This makes the proposed recursive sphere-packing algorithm very useful in those settings where the overlap between the distributions in the feature space is modest.

Our other approach is based on the observation that the affinity between two sets of vectors can be linked with how hard to separate the elements of the two sets are. Moreover, that separation can be measured via *classification risks*. We present some results showing that the error rate of a simple nearest-neighbor classifier provides a convenient similarity measure. In fact, we show that it induces a scale-invariant, positive-definite kernel over probability distributions.

We then connect the idea of classification-risk based similarities with the well-known family of f-divergences. Specifically, we develop two natural generalizations of that family which are related to the two parameters that mainly defines a classifier: the choice of a *set of allowable classification functions* and a *loss function*. The first of those generalizations, the CRFD family, deals with the former aspect. It defines divergence measures which look at certain features of the distributions, implictly defined by the choice of the set C of classification functions. We prove some interesting properties of CRFDs, under mild conditions on C: they lower-bound standard divergences, they are monotonous non-decreasing on C and they can satisfy the identity of indiscernibles property.

In spite of their theoretical interest, the practical application of CRFDs is hindered by the complexity of estimating classification risks for the whole range of prior probabilities. In principle this approach is only practical for some specific classifiers, like the nearest neighbor rule. We have shown that the NN asymptotic error is in a 1-1 correspondence with the Bayes risk for the square loss. Inspired by this result, we have presented the *loss-induced divergences* or (f, l)-divergences, which are another generalization of f-divergences where the 0-1 loss is substituted by a surrogate loss. The loss-induced divergences share most of the properties of f-divergences, while being a more flexible family. They also provide alternative representations of standard divergences. Exploiting these alternative representations, we provide a new estimator of the KL divergence in terms of NN classification errors, as well as a lower-bound on that divergence. This estimator is not only theoretically interesting, but also of practical importance, as evidenced by the empirical results. One of its main assets is that it is independent of the input space dimensionality, unlike the current state-of-the-art methods. This makes it specially interested in high-dimensional settings, where the actual data is likely to lie on a lower-dimensional manifold.

Finally, in Appendix C we have shown how a simple modification of spectral clustering allows the use of any of the proposed affinity measures for tackling the *segmentation* problem.

Even though each chapter includes experimental results on both synthetic and real datasets, we have also tried the different methods out in a more complex task in Chapter 6. Specifically, we have addressed the unsupervised music genre recognition problem. The NN-based divergence estimator of the Jeffreys divergence clearly outperforms the rest of the methods, showing that the proposed estimator is really useful for defining disimilarities in complex data sets.

Future lines

Here we sketch some of the most promising future lines arising from the work in this Thesis. From the point of view of the applications of the different methods, the possibilities are endless, so we will focus on theoretical and algorithmical extensions.

There are many exciting open issues in the area of model-based similarities.

Let us consider first the likelihood-based methods. Training models on subsets of sequences is a feasible way of overcoming their main limitations. Moreover, the performance of such similarity measures under random sampling of models is a nice object of study from a theoretical point of view.

Extending the SSD idea to other kind of state-space models is quite a natural continuation for the work in Chapter 3. Specifically, working with continuous state space models is a promising path for exploration.

Regarding the sphere-packing clustering algorithm, studying alternative approaches for finding the optimal K-cover (where K is the desired number of clusters/spheres) of the dataset is a worthy research line. Further connections with the set-covering literature are likely to exist and yield interesting conclusions.

Regarding the generalizations of f-divergences, devising efficient methods for estimating CRFDs is a very interesting line of work. That could be done by, for example, finding clever ways to estimate the risks of linear classifiers for the whole range of prior probabilities. The (f, l) family of divergences can be used to define cost-sensitive divergences, by using adequate non-symmetric losses as the building blocks of the divergences. Finally, the proposed combination of CRFDs and (f, l)-divergences in Section 5.5.6 is a very promising line to explore. It naturally defines classifier-based affinities, and is an exciting research topic from both theoretical and practical viewpoints.

Part I

Appendices

Appendix A

Spectral clustering

Clustering [Xu and Wunsch-II, 2005] consists in partitioning a dataset $\mathcal{S}$ comprised of N elements into C disjoint groups called clusters. Data assigned to the same cluster must be similar and, at the same time, distinct from data assigned to the rest of clusters. It is an unsupervised learning problem, meaning that it does not require any prior labeling of the data, and thus it is very appropriate for exploratory data analysis or scenarios where obtaining such a labeling is costly.

Algebraically, a clustering problem can be formulated in the following way. Given a dataset $\mathcal{S}$, one forms a $N \times N$ similarity matrix $\mathbf{W}$, whose ij^{th} element w_{ij} represents the similarity between the i^{th} and j^{th} instances. The clustering problem then consists in obtaining a $N \times C$ clustering matrix $\mathbf{Z}$, where $z_{ic} = 1$ if instance i belongs to cluster c and $z_{ic} = 0$ otherwise, which is optimal under some criteria.

Spectral clustering (SC) algorithms [von Luxburg, 2007] approach the clustering task from a graph-theoretic perspective. Data instances form the nodes V of a weighted graph $G = (V, E)$ whose edges E represent the similarity or adjacency between data, defined by the matrix $\mathbf{W}$. This way, the clustering problem is cast into a graph partitioning one. The clusters are given by the partition of G in C groups that optimize certain criteria such as the normalized cut [Shi and Malik, 2000]. Finding such an optimal partition is an NP-hard problem, but it can be relaxed into a (generalized) eigenvalue problem on the Laplacian matrix $\mathbf{L} = \mathbf{D} - \mathbf{W}$, where $\mathbf{D}$ is the diagonal matrix with elements $d_{ii} = \sum_{j=1}^{N} w_{ij}$, or one of its normalized versions, followed by k-means [Bishop, 2006] or any other clustering algorithm on the rows of the matrix of selected eigenvectors. Specifically, if the optimization criterion is the normalized cut, the corresponding eigenvalue problem is:

$$L\boldsymbol{\phi} = \mathbf{D}\boldsymbol{\phi}\boldsymbol{\lambda}, \tag{A.1}$$

where $\boldsymbol{\phi} = \{\phi_1, \ldots, \phi_N\}$ is the matrix whose columns corresponds with the generalized eigenvectors and $\boldsymbol{\lambda}$ is a diagonal matrix whose ii^{th} entry corresponds with λ_i, the i^{th} eigenvalue, which are assumed to be sorted by increasing magnitude. Then, the actual partitions are found by clustering the rows of $\tilde{\boldsymbol{\phi}} = \{\phi_2, \ldots, \phi_{K'}\}$[Shi and Malik, 2000]. The first eigenvector (associated with an eigenvalue 0) is not included because it is constant, assuming that the graph is connected [von Luxburg, 2007]. The number of selected eigenvectors is usually equal to the

number of clusters $K' = K$, but it can also be chosen more carefully, looking for the number of eigenvectors that provides a partition of the space in as many clusters as desired. More generally, if the number of clusters is unknown, it can be chosen attending to the eigengap, which is the difference in magnitude between two consecutive eigenvalues. From matrix perturbation theory, a large eigengap between the i^{th} and $(i+1)^{th}$ eigenvalue implies a stable principal subspace $\{\phi_1, \ldots, \phi_i\}$ [Stewart and Sun, 1990].

In recent years, many authors have analyzed spectral clustering algorithms from several new viewpoints. Specifically, in [Belkin and Niyogi, 2001] the authors show that the eigenvalue problem coming from the relaxation of the normalized-cut criterion clustering corresponds with a structure-preserving low-dimensional embedding of the original data, so the use of euclidean distances between the rows of $\tilde{\boldsymbol{\phi}}$ is justified. This embedding is called Laplacian Eigenmap. Attending to this interpretation, the number of eigenvectors K' selected according to the eigengap can be seen in some sense as the effective dimension of the manifold where the original data live.

The time complexity for the spectral clustering is dominated by the eigendecomposition of the normalized Laplacian, which in general is $O(N^3)$. However, if the affinity matrix is sparse (e.g. if only the affinities between the nearest neighbors of a given node are considered), there exist efficient iterative methods that notably reduce this complexity, such as the Lanczos method [Golub and Van Loan, 1989], which makes it feasible even for large datasets [von Luxburg, 2007].

Appendix B

Hidden Markov models (HMMs)

Hidden Markov models (HMMs) [Rabiner, 1989] are a type of parametric, discrete state-space model. They provide a convenient model for many real-life phenomena, while allowing for low-complexity algorithms for inference and learning. Their main assumptions are the independence of the observations given the hidden states and that these states follow a Markov chain.

Assume a sequence $\mathbf{S}$ of T observation vectors $\mathbf{S} = \{\mathbf{x}_1, \ldots, \mathbf{x}_T\}$. The HMM assumes that $\mathbf{x}_t$, the t^{th} observation of the sequence, is generated according to the conditional emission density $p(\mathbf{x}_t|q_t)$, with q_t being the hidden state at time t. The state q_t can take values from a discrete set $\{s_1, \ldots, s_K\}$ of size K. The hidden states evolve following a time-homogeneous first-order Markov chain, so that $p(q_t|q_{t-1}, q_{t-2}, \ldots, q_0) = p(q_t|q_{t-1})$.

In this manner, the parameter set θ that defines an HMM consists of the following distributions:

- The initial probabilities vector $\pi = \{\pi_i\}_{i=1}^K$, where $\pi_i = p(q_0 = s_i)$.
- The state transition probability, encoded in a matrix $\mathbf{A} = \{a_{ij}\}_{i,j=1}^K$ with $a_{ij} = p(q_{t+1} = s_j|q_t = s_i)$, $1 \leq i, j \leq K$.
- The emission pdf for each hidden state $p(\mathbf{x}_t|q_t = s_i), 1 \leq i \leq K$.

From these definitions, the likelihood of a sequence $\mathbf{S} = \{\mathbf{x}_1, \ldots, \mathbf{x}_T\}$ can be written in the following factorized way:

$$p(\mathbf{S}|\theta) = \sum_{q_0,\ldots,q_T} \pi_{q_0} p(\mathbf{x}_0|q_0) \prod_{t=1}^{T} p(\mathbf{x}_t|q_t) a_{q_{t-1},q_t}. \tag{B.1}$$

The training of this kind of models in a maximum likelihood setting is usually accomplished using the Baum-Welch method [Rabiner, 1989, Bilmes, 1997], which is a particularization of the well-known EM algorithm. The E-step finds the expected state occupancy and transition probabilities, which can be done efficiently using the forward-backward algorithm [Rabiner, 1989]. This algorithm implies the calculation of both the forward α and backward β variables that are defined as follows:

$$\alpha_k(t) = p(\mathbf{x}_1, \ldots, \mathbf{x}_t, q_t = s_k) \tag{B.2}$$

$$\beta_k(t) = p(\mathbf{x}_{t+1}, \ldots, \mathbf{x}_T|q_t = s_k). \tag{B.3}$$

These variables can be obtained in $O(K^2T)$ time through a recursive procedure and can be used to rewrite the likelihood from Eq. (B.1) in the following manner:

$$p(\mathbf{S}|\theta) = \sum_{k=1}^{K} \alpha_k(t)\beta_k(t), \tag{B.4}$$

which holds for all values of $t \in \{1, \ldots, T\}$.

Given a previously estimated $\mathbf{A}$, the state transition probabilities can be updated using the forward/backward variables and that previous estimation, yielding:

$$\tilde{a}_{ij} \propto \sum_{t'=1}^{T} \alpha_i(t')a_{ij}p(\mathbf{x}_{t'+1}|q_{t'+1} = s_j)\beta_j(t'+1). \tag{B.5}$$

Then, the M-step updates the parameters in order to maximize the likelihood given the expected hidden states sequence. These two steps are then iterated until convergence. It is worth noting that the likelihood function can have many local maxima, and this algorithm does not guarantee convergence to the global optimum. Due to this, it is common practice to repeat the training several times using different initializations and then select as the correct run the one providing a larger likelihood.

The extension of this training procedure to multiple input sequences is straightforward. The interested reader is referred to [Rabiner, 1989] for a complete description.

Appendix C

From spectral clustering to segmentation for sequences of data

Motivated by an interpretation of segmentation as clustering with additional constraints, we propose a new algorithm for sequence segmentation based on the methods developed in previous chapters. This approach implies the use of model-based distance measures between sequences, as well as a variant of spectral clustering specially tailored for segmentation.

C.1 Introduction

As we have previously emphasized, clustering is a very useful data exploration technique, consisting in finding a partition of a dataset resulting in meaningful disjoint groups. When the data we want to partition is actually a sequence, it is usually quite natural to look for temporally (or spatially) contiguous clusters. This way, the clustering task is turned into a *segmentation task´*. Arguably, the best-known approach to segmentation is the dynamic programming technique [Bellman, 1961]. In its most basic form, it looks for an optimal piecewise-constant approximation to the

original sequence, usually in terms of the L2 norm of the approximation error. Hidden Markov models are also usually employed for sequeece segmentation purposes. Recall that this family of models assumes a discrete latent variable z_t following a Markov chain with a certain transition matrix $A = \{a_{ij}\}, a_{ij} = P(z_t = j|z_{t-1} = i)$ and a observable variable $\mathbf{y}_t$. In a segmentation scenario, each block of contiguous data vectors assigned to a given latent variable value is considered a segment.

Note that these techniques treat data inside each segment as independent, identically-distributed samples. However, many times it is interesting to take into account the dynamical characteristics of the individual segments. This can be achieved by the use of Hierarchical Hidden Markov Models (HHMMs) [Fine et al., 1998], a multi-layered probabilistic model in which each layer form an HMM, or in general any kind of hierarchical dynamic bayesian network [Murphy, 2002]. The drawback of this approach is that the learning of such models is very computationally demanding and, even more importantly, usually requires very careful model selection and initialization procedures in order to produce adequate results.

In this brief chapter we propose to use the framework of semi-parametric model-based sequence clustering in order to achieve segmentations which take into account the dynamical characteristics of the individual segments. If a full generative characterization of the dataset is desired (e.g. an HHMM), the resulting segmentation can be used as an initialization for the learning procedure of the actual generative model.

The rest of this chapter is organized as follows: Section C.2 focus on defining the segmentation problem from a clustering perspective, and provide an

C.1.1 Related publications

This appendix is mainly based on publications [García-García et al., 2009b] and [García-García et al., 2009a].

C.2 Segmentation as a clustering problem

Data segmentation can, in many cases, be naturally interpreted as a clustering problem with additional connectivity constraints. In the case of sequential data, these constraints imply that the points belonging to a given cluster must be *compact* in the temporal dimension. This means that every node between the leftmost and the rightmost ones of a given cluster must belong to that cluster. With this in mind, we propose to extend the successful framework of semi-parametric sequence clustering to sequence segmentation. To this end, the original sequence $\mathcal{S}$ is divided into N sub-sequences $\mathcal{S} = S_1, \ldots, S_N$ which are then clustered while enforcing the aforementioned constraints. The length of the sub-sequences is called the *window size*, and it determines the temporal resolution of the segmentation. It is worth noting that if a high time resolution is required for the segment boundaries, it can be easily and progressively refined by focusing on the sub-sequences which define a boundary and analyzing them using a smaller window size.

If the KL-based method from Chap. 2 is used for defining affinities, then any probabilistic model can be used for the individual sub-sequences. If the state-space dynamics method from Chap. 3 is used, then HMMs will be the model of choice. If a dynamical model is used, the usual assumption of i.i.d. data can be avoided, thus taking into account the temporal evolution of the data *within each subsequence*. This is one of the main strengths of this proposal. For the actual clustering stage, we once again propose the use of spectral clustering algorithms, given their good performance, as reported in the model-based sequence clustering literature [García-García et al., 2009c, Yin and Yang, 2005] and the strength of its interpretation as a low-dimensional embedding, which we will use when defining our algorithm.

As previously stated, the problem of minimizing the normalized cut of a given affinity matrix is NP-hard. Nonetheless, the restriction of the clusters being compact in time allows for the development of a particular dynamic programming (DP) algorithm that find the optimal partition in K segments in polynomial time [Malioutov and Barzilay, 2006]. However, in this paper we propose to carry out the usual spectral decomposition of the affinity matrix and then obtain the actual seg-

mentation on the resulting eigenvectors using standard DP with L2 norm. This way, the implementation of a specific DP algorithm for solving the normalized cut is avoided, and we can resort to the standard spectral clustering literature in order to intelligently select the kernel width if the number of sources is known a priori [García-García et al., 2009c, Ng et al., 2002]. As previously commented, this parameter has dramatic effects on the clustering/segmentation performance and should be chosen carefully. Working on selected eigenvectors also has a denoising effect, coming from the dimensionality reduction. The drawback of this alternative is the need for a (partial) eigendecomposition of the $N \times N$ affinity matrix. The time complexity for this decomposition, assuming the matrix is full, is $O(N^3)$. However, if the affinity matrix is sparse (e.g. if only the affinities between the nearest neighbors of a given node are considered), there exist efficient iterative methods that reduce this complexity, such as the Lanczos method [Golub and Van Loan, 1989].

C.2.1 Segmenting the eigenvectors

As explained in Appendix A, the normalized-cut spectral clustering algorithm can be interpreted as an embedding of the original data into a low-dimensional space followed by a typical clustering applied on the embedded data. The low-dimensional embedding correspond with the rows of a subset of the eigenvectors of the normalized Laplacian of the affinity matrix. We can obtain a segmentation based on this paradigm by just substituting the clustering on the eigenvectors by a procedure which preserves the time-continuity of the clusters. If we know the number M_{SRC} of individual "sources" of the data (e.g. number of different speakers in a speaker segmentation task or number of mixture components in a mixture model segmentation scenario), that number should be the number of clusters that arise naturally from W. Thus, a large eigengap is expected between $\lambda_{M_{\text{SRC}}}$ and $\lambda_{M_{\text{SRC}}+1}$ and we can interpret $\tilde{\boldsymbol{\phi}} = \left\{\boldsymbol{\phi}_2, \ldots, \boldsymbol{\phi}_{M_{\text{SRC}}}\right\}$ as a good $M_{\text{SRC}} - 1$ dimensional embedding for clustering purposes. In fact, if M_{SRC} is unknown, we can simply choose as the dimension of the embedding the one that provides a larger eigengap.

As mentioned in Appendix A, euclidean distances between the rows of $\tilde{\boldsymbol{\phi}}$ are meaningful from the clustering point of view. Moreover, the resulting eigenvectors

approximate the desired piecewise-constant solution to the clustering problem. Thus, in order to obtain the actual segmentation, it is very natural to apply the well-known k-segmentation algorithm based on dynamic programming [Bellman, 1961] which, as previously stated, assume a piecewise-constant behavior of the sequence.

If we are also interested in simultaneous clustering of the segments according to their source, an inexpensive alternative would be to perform k-means clustering on the eigenspace to find as many clusters as sources are present, and then assign each segment to the same cluster of the majority of the subsequences within it.

C.3 Experimental Results

In this section we evaluate the performance of the proposed method for sequence-clustering-based segmentation. We focus on using dynamical models for the subsequences, since this is one of the main advantages of our proposal. To this end, we present results using both synthetic and real-world datasets, namely a speaker segmentation task. For the sake of simplicity we have assumed a known number of sources and segments in the experiments. If these parameters are unknown, they can be effectively estimated using well-known criteria. In order to determine the number of sources (or the effective dimension of the embedded space), as previously commented we can resort to eigengap-based heuristics [Ng et al., 2002]. Additionally, if the number of segments is unknown, the Bayes information criterion (BIC) [Bishop, 2006] could be used together with dynamic programming in order to obtain an optimal partition. The kernel width is automatically selected as the one that provides the largest eigengap between the K^{th} and $(K+1)^{th}$ eigenvalues. We report the results obtained applying KL-distance (see Chapter 2) based spectral clustering (KL-SC) as well as the proposed segmentation algorithm (KL-SS), and the same for the SSD distance from Chapter 3 (yielding SSD-SC and SSD-SS). Clustering error is defined as the percentage of incorrectly classified samples under an optimal permutation of the cluster labels. Note that when clustering the subsequences, our aim is to group them according to their source. On the other hand, segmentation error is naturally the percentage of subsequences assigned to the wrong segment. If we perform a further clustering of the estimated segments, such as the one mentioned

in the previous section, the resulting clustering error would be equal to the reported segmentation error if each segment is correctly clustered (as is always the case in the present experiments).

Since the segment size may not be an exact multiple of the window size, the ground-truth label of a given window for error calculation purposes is set as the label of the majority of the data points falling inside that window.

C.3.1 Synthetic Data: Segmenting a Mixture of HMMs

We adopt the scenario suggested in [Smyth, 1997] which we already used in Chapters 2 and 3, but adapted to the segmentation task. The data sequence is comprised of segments coming from a equiprobable mixture of two HMMs θ_1 and θ_2. Each of these models has two hidden states, with an uniform initial distribution, and their corresponding transition matrices are

$$\mathbf{A}_1 = \begin{pmatrix} 0.6 & 0.4 \\ 0.4 & 0.6 \end{pmatrix} \qquad \mathbf{A}_2 = \begin{pmatrix} 0.4 & 0.6 \\ 0.6 & 0.4 \end{pmatrix}.$$

Emission probabilities are the same in both models, specifically $\mathcal{N}(0,1)$ in the first state and $\mathcal{N}(3,1)$ in the second.

We generate random sequences consisting of 20 segments, 10 of them coming from each one of the aforementioned sources. The length of each segment is chosen accordingly to a uniform random variable in the range $[350, 2000]$. This experiment is repeated for different window lengths, and for each one of them 50 runs are executed and averaged. Since there are less sources than segments, we need to assign each segment to the corresponding source. This can be done by simply running a k-means algorithm on the selected eigenvectors to cluster individual nodes of the graph in as many groups as different sources exist (in this case, 2) and then assign each segment to the source corresponding to the majority of its nodes. We compare clustering and segmentation results are shown in Table C.1. As expected, enforcing the contiguity constraint turns the KL-SC clustering method into a powerful segmentation algorithm. This additional constraints greatly simplifies the original problem, since dynamic programming corrects isolated clustering errors inside a segment, resulting in very low segmentation error rates. However, when the window size is very short

Table C.1: Clustering and Segmentation error (mean and standard deviation) in the synthetic scenario (2 sources, 8 segments)

Window length	KL-SC	KL-SS
m=100	19.63% (±13.05) %	15.91% (±16.38)
m=125	7.25% (±6.30) %	4.23% (±7.63)
m=150	3.81% (±2.50) %	1.06% (±1.76)

(according to the complexity of the task), the affinity matrix does not reflect well the structure of the dataset, so the eigenvectors of the Laplacian matrix are not very informative from a clustering perspective and dynamic programming can not get a proper segmentation.

C.3.2 Speaker Segmentation

In order to show the real-world effectiveness of the proposed method, we carry out a speaker segmentation task on the UCI Japanese Vowels dataset (see Appendix D). Recall that it is comprised of utterances from 9 different speakers. We concatenate all the individual sequences to form the long sequence which we would like to divide into 9 segments. We use a windows of lengths ranging from 10 to 15 samples, which corresponds with time resolutions from 64 to 128ms. Each subsequence is modeled using a 2-state HMM.

Fig.C.1 shows the evolution of the eigenvectors along the subsequences. Their behavior is approximately piecewise-constant, as expected, so dynamic programming is able to find a very good segmentation of the original sequence. Note that, for the shake of clarity, there are just 5 eigenvectors in the plot and they clearly define the segment boundaries, showing that many times the dimensionality of the embedding can be lesser than the number of clusters while allowing for a clear partition. Table C.2 compares clustering and segmentation results for the likelihood-based methods, averaged over 50 runs. It is worth noting that the dynamic programming on the eigenvectors greatly alleviates the fluctuations amongst runs, giving extremely stable segmentation results in each execution as well as very good performance. On Table C.3 we compare the segmentation performance of KL and SSD measures.

Table C.2: Clustering and segmentation error (mean and standard deviation) in the Japanese Vowels dataset (9 sources and segments)

Window length	KL-SC	KL-SS
m=10	21.68% (±4.75)	1.0% (±0.22)
m=15	10.21% (±3.18)	3.5% (±0)
m=20	5.19% (±2.00)	1.4% (±0)

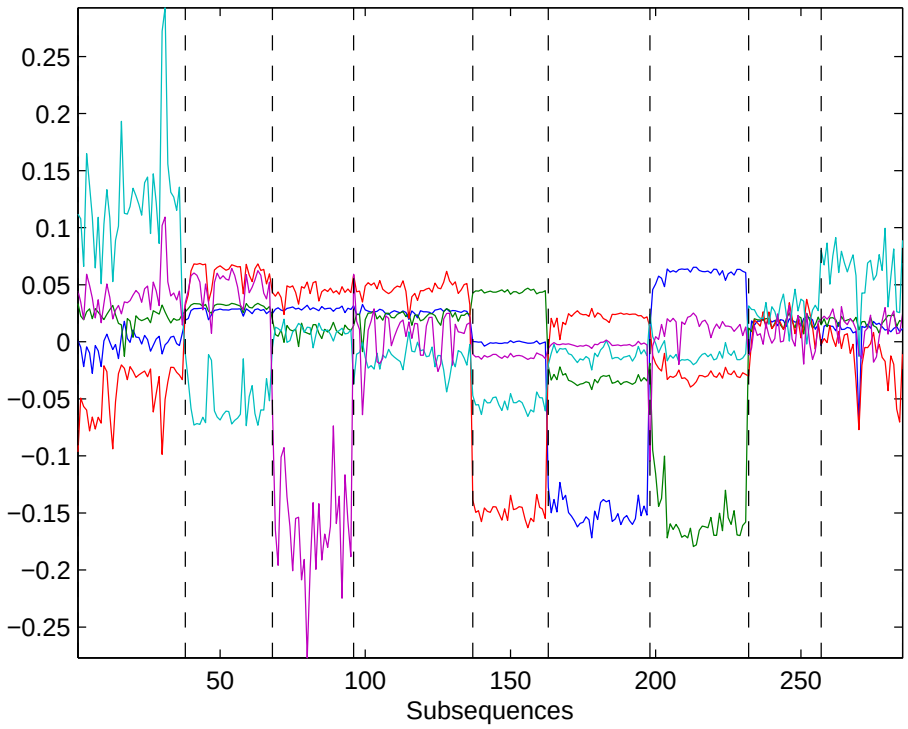


Figure C.1: Plot of the eigenmap and the segment boundaries found by DP for the speaker segmentation task

Table C.3: Segmentation error (mean and standard deviation) in the Japanese Vowels dataset (9 sources and segments) using KL and SSD distances

	KL-SS	SSD-SS			
Window length	K_m=2	K=18	K=24	K=32	K=40
W=10	1.0% (±0.22)	1.61% (±0)	1.0% (±0.29)	0.82% (±0.35)	0.72% (±0.33)
W=15	3.5% (±0)	2.39% (±0)	1.12% (±0.39)	0.95% (±0.29)	0.98% (±0.27)
W=20	1.4% (±0)	2.66% (±0)	1.26% (±0.44)	1.03% (±0.53)	0.75% (±0.45)

Appendix D

Dataset descriptions

D.1 EEG Data

This database was recorded by Zak Keirn at Purdue University, and can be found at `http://www.cs.colostate.edu/eeg/eegSoftware.html`. It consists of EEG recordings of seven subjects performing five different mental tasks, namely: baseline (rest), math calculations, composing a letter, rotating a geometrical figure and counting. Each recording comprises measures taken from seven channels at 250Hz for 10 seconds. We divide them into subsequences of l25 samples, and the only preprocessing applied to them is a first order derivative so they adjust better to a Markov model.

D.2 Japanese Vowels

To construct this dataset, nine male speakers uttered two Japanese vowels /ae/ consecutively. The actual data is comprised of the 12-dimensional time-series of LPC cepstrum coefficients for each utterance, captured at a sampling rate of 10KHz using a sliding window of 25.6ms with a 6.4ms shift. The number of samples per sequence varies in the range 7-29 and there are 30 sequences per user. This dataset can be downloaded from the UCI ML repository at `http://archive.ics.uci.edu/ml/datasets/Japanese+Vowels`.

D.3 GPM PDA speech data

This database was recorded at the Multimedia Processing Group of the University Carlos III of Madrid using a PDA. It is available at `http://www.tsc.uc3m.es/~dggarcia/code.html`. The dataset consists of speech coming from 30 differente speakers. Each speaker recorded 50 isolated words (each one of them being an individual sequence), yielding recordings with a mean length around 1.3 seconds. The audio files were processed using the freely available HTK software[1], using a standard parametrization consisting of 12 Mel-frequency cepstral coefficients (MFCCs) [Müller, 2007], an energy term and their respective

[1] http://htk.eng.cam.ac.uk

increments (δ), giving a total of 26 parameters. These parameters were obtained every 10ms with a 25ms analysis window. This dataset can be used to obtain 1176 2-speaker clustering tasks.

D.4 Synthetic Control Chart data

This dataset contains unidimensional time series representing six different classes of control charts: normal, cyclic, increasing trend, decreasing trend, upward shift and downward shift. The sequences are synthetically generated using the process in [Alcock and Manolopoulos, 1999]. There are 100 instances of each class, with a fixed length of 60 samples per instance. A sample of each class is plotted in Fig. D.1. The dataset is part of the UCI KDD repository, and can be found at `http://archive.ics.uci.edu/ml/datasets/Synthetic+Control+Chart+Time+Series`.

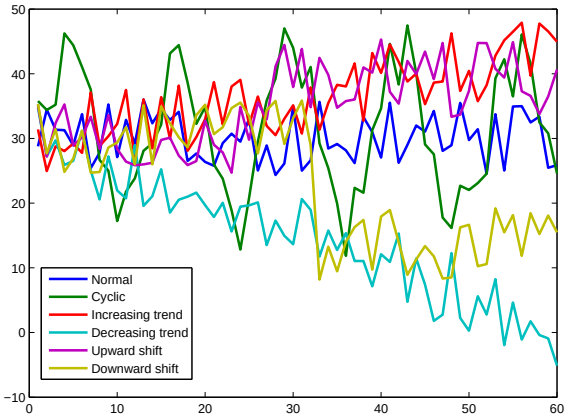


Figure D.1: Some samples from the Synthetic Control Chart dataset

D.5 Character Trajectories

This dataset consists of trajectories captured by a digitizing tablet when writing 20 different characters. Each sample is a 3-dimensional vector containing the x and y coordinates as well as the pen tip force. There are 2858 sequences in the dataset, which are already normalized, differentiated and smoothed using a Gaussian kernel. We use this dataset for carrying out two-class clusterings between all the possible combinations, giving a total of 190 experiments. The average length of the sequences in this dataset is around 170 samples. The dataset can be downloaded from the UCI ML repository at `http://archive.ics.uci.edu/ml/datasets/Character+Trajectories`.

D.6 AUSLAN

The Australian Sign Language dataset is comprised of 22-dimensional time series representing different sign-language gestures. The gestures belong to a single signer, and were collected in different sessions over a period of nine weeks. There are 27 instances per gesture, with an average length of 57 samples. Following [Jebara et al., 2007], we perform 2-class clustering tasks using semantically related concepts. These concepts are assumed to be represented by similar ges-

tures and thus provide a difficult scenario. The AUSLAN dataset can be found at `http://archive.ics.uci.edu/ml/datasets/Australian+Sign+Language+signs+(High+Quality)`.

Appendix E

Code

MATLAB implementations of the different methods mentioned in this Thesis can be found at `http://www.tsc.uc3m.es/~dggarcia/code.html`.

Bibliography

[Alcock and Manolopoulos, 1999] Alcock, R. and Manolopoulos, Y. (1999). Time-series similarity queries employing a feature-based approach. In *7th Hellenic Conference on Informatics*.

[Ali and Silvey, 1966] Ali, S. and Silvey, S. (1966). A general class of coefficients of divergence of one distribution from another. *Journal of the Royal Statistical Society. Series B (methodological)*, 28:131–142.

[Alon et al., 2003] Alon, J., Sclaroff, S., Kollios, G., and Pavlovic, V. (2003). Discovering Clusters in Motion Time-Series Data. In *Proc. of the 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'03)*.

[Arenas-García et al., 2007] Arenas-García, J., Parrado-Hernández, E., Meng, A., Hansen, L. K., and Larsen, J. (2007). Discovering music structure via similarity fusion. In *Music, Brain and Cognition Workshop, NIPS'07*.

[Baldi et al., 1998] Baldi, P., Brunak, S., and Stolovitzky, G. (1998). *Bioinformatics: The Machine Learning Approach*. MIT Press.

[Bartlett et al., 2006] Bartlett, P., Jordan, M., and McAuliffe, J. (2006). Convexity, classification and risk bounds. *Journal of the American Statistical Association*, 101:138–156.

[Belkin and Niyogi, 2001] Belkin, M. and Niyogi, P. (2001). Laplacian eigenmaps and spectral techniques for embedding and clustering. In *Advances in Neural Information Processing Systems 14*, pages 585–591. MIT Press.

[Belkin et al., 2006] Belkin, M., Niyogi, P., and Sindhwani, V. (2006). Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *J. Mach. Learn. Res.*, 7:2399–2434.

[Bellman, 1961] Bellman, R. (1961). On the approximation of curves by line segments using dynamic programming. *Communications of the ACM*, 4(6).

[Berg et al., 1984] Berg, C., Christensen, J. P. R., and Ressel, P. (1984). *Harmonic Analysis on Semigroups*. Springer-Verlag, New York.

[Berlinet and Thomas-Agnan, 2003] Berlinet, A. and Thomas-Agnan, C. (2003). *Reproducing Kernel Hilbert Spaces in Probability and Statistics*. Kluwer Academic Publishers.

[Bhattacharyya, 1943] Bhattacharyya, A. (1943). On a measure of divergence between two statistical populations defined by their probability distributions. *Bull. Calcutta Math Soc.*

[Bilmes, 1997] Bilmes, J. A. (1997). A gentle tutorial on the em algorithm and its application to parameter estimation for gaussian mixture and hidden markov models. Technical report.

[Bishop, 2006] Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer.

[Blackwell, 1951] Blackwell, D. (1951). Comparison of experiments. In *Proc. 2nd Berkeley Symp. Probab. Statist.*, volume 1, pages 93–102, Berkeley. Univ. California Press.

[Boyd and Vandenberghe, 2004] Boyd, S. and Vandenberghe, L. (2004). *Convex Optimization*. Cambridge University Press.

[Bregman, 1967] Bregman, L. M. (1967). The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming. *USSR Computational Mathematics and Mathematical Physics*, 7(3):200 – 217.

[Buja et al., 2005] Buja, A., Stuetzle, W., and Shen, Y. (2005). Loss functions for binary class probability estimation and classification: Structure and applications. Technical report, University of Pennsylvania.

[Campbell, 1997] Campbell, J. (1997). Speaker recognition: A tutorial. *Proceedings of the IEEE*, 85(9):1437–1462.

[Campbell et al., 2006] Campbell, W., Campbell, J., Reynolds, D., Singer, E., and Torres-Carrasquillo, P. (2006). Support vector machines for speaker and language recognition. *Computer Speech and Language*, 20:210–229.

[Chapelle et al., 2006] Chapelle, O., Schölkopf, B., and Zien, A., editors (2006). *Semi-Supervised Learning*. MIT Press, Cambridge, MA.

[Chung, 1997] Chung, F. R. K. (1997). *Spectral Graph Theory (CBMS Regional Conference Series in Mathematics, No. 92)*. American Mathematical Society.

[Cover and Hart, 1967] Cover, T. and Hart, P. (1967). Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13:21–27.

[Cover and Thomas, 1991] Cover, T. M. and Thomas, J. A. (1991). *Elements of Information Theory*. Wiley-Interscience.

[Cristani et al., 2007] Cristani, M., Bicego, M., and Murino, V. (2007). Audio-visual event recognition in surveillance video sequences. *IEEE Transactions on Multimedia*, 9(2):257–267.

[Csiszár, 1967] Csiszár, I. (1967). Information-type measures of difference of probability distributions and indirect observations. *Studia Scientiarum Mathematicarum Hungarica*, 2:29–318.

[Dance et al., 2004] Dance, C., Willamowski, J., Fan, L., Bray, C., and Csurka, G. (2004). Visual categorization with bags of keypoints. In *ECCV International Workshop on Statistical Learning in Computer Vision*.

[Daoudi and Louradour, 2009] Daoudi, K. and Louradour, J. (2009). A comparison between sequence kernels for svm speaker verification. In *ICASSP '09: Proceed-*

ings of the 2009 IEEE International Conference on Acoustics, Speech and Signal Processing, pages 4241–4244, Washington, DC, USA. IEEE Computer Society.

[DeGroot, 1970] DeGroot, M. (1970). *Optimal Statistical Decisions*. McGraw-Hill Book Company.

[Dempster et al., 1977] Dempster, A., Laird, N., and Rubin, D. (1977). Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society*, 39(1):1–38.

[Devijver and Kittler, 1982] Devijver, P. and Kittler, J. (1982). *Pattern Recognition: A Statistical Approach*. Prentice-Hall, Englewood Cliffs, NJ.

[Devroye et al., 1996] Devroye, L., Györfi, L., and Lugosi, G. (1996). *A Probabilistic Theory of Pattern Recognition*. Springer.

[Dietterich, 2009] Dietterich, T. (2009). Machine learning for sequential data: A review. *Structural, Syntactic, and Statistical Pattern Recognition*, pages 227–246.

[Dornhege et al., 2007] Dornhege, G., Krauledat, M., Müller, K.-R., and Blankertz, B. (2007). General signal processing and machine learning tools for BCI. In Dornhege, G., del R. Millán, J., Hinterberger, T., McFarland, D., and Müller, K.-R., editors, *Toward Brain-Computer Interfacing*, pages 207–233. MIT Press, Cambridge, MA.

[Fine et al., 1998] Fine, S., Singer, Y., and Tishby, N. (1998). The Hierarchical Hidden Markov Model: Analysis and Applications. *Machine Learning*, 32(1):41–62.

[Frank and Asuncion, 2010] Frank, A. and Asuncion, A. (2010). UCI machine learning repository.

[García-García et al., 2010] García-García, D., Parrado-Hernández, E., Arenas-García, J., and Díaz-de-María, F. (2010). Music genre classification using the temporal structure of songs. In *IEEE International Workshop on Machine Learning for Signal Processing*.

[García-García et al., 2009a] García-García, D., Parrado-Hernández, E., and Díaz-de-María, F. (2009a). Model-based clustering and segmentation of sequences. In *NIPS'09 Workshop on Temporal Segmentation*, Whistler, BC.

[García-García et al., 2009b] García-García, D., Parrado-Hernández, E., and de María, F. D. (2009b). Sequence segmentation via clustering of subsequences. In *International Conference on Machine Learning and Applications (ICMLA)*.

[García-García et al., 2011a] García-García, D., Parrado-Hernández, E., and de María, F. D. (2011a). State-space dynamics distance for clustering sequential data. *Pattern Recognition*, 44(5):1014–1022.

[García-García et al., 2009c] García-García, D., Parrado-Hernández, E., and F. Díaz-de-María (2009c). A New Distance Measure for Model-Based Sequence Clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 31(7):1325–1331.

[García-García and Santos-Rodríguez, 2011] García-García, D. and Santos-Rodríguez, R. (2011). Sphere packing in feature space for clustering sets of vectors. In *Proc. of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.

[García-García et al., 2011b] García-García, D., Santos-Rodríguez, R., and von Luxburg, U. (2011b). Risk-based generalizations of f-divergences. Submitted to ICML 2011.

[Golub and Van Loan, 1989] Golub, G. and Van Loan, C. (1989). *Matrix Computations*. John Hopkins University Press.

[Gretton et al., 2007] Gretton, A., Borgwardt, K., Rasch, M., Schoelkopf, B., and Smola, A. (2007). A kernel method for the two-sample-problem. In Schoelkopf, B., Platt, J., and Hoffman, T., editors, *Advances in Neural Information Processing Systems 19*, pages 513–520. MIT Press.

[Gretton et al., 2005] Gretton, A., Herbrich, R., Smola, A., Bousquet, O., and Schölkopf, B. (2005). Kernel methods for measuring independence. *Journal of Machine Learning Research (JMLR)*, 6:2075–2129.

[Guaus, 2009] Guaus, E. (2009). *Audio content processing for automatic music genre classification: descriptors, databases, and classifiers*. PhD thesis, Universitat Pompeu Fabra.

[Guyon et al., 2009] Guyon, I., von Luxburg, U., and Williamson, R. (2009). Clustering: Science or art? Technical report, NIPS'09 Workshop on Clustering: Science or Art? Towards Principled Approaches.

[Hassan and Nath, 2006] Hassan, M. and Nath, B. (2006). Stock market forecasting using hidden Markov model: a new approach. In *Intelligent Systems Design and Applications, 2005. ISDA'05. Proceedings. 5th International Conference on*, pages 192–196. IEEE.

[Hastie et al., 2003] Hastie, T., Tibshirani, R., and Friedman, J. H. (2003). *The Elements of Statistical Learning*. Springer, corrected edition.

[Hettich and Bay,] Hettich, S. and Bay, S. *The UCI KDD Archive*. University of California, Irvine, Dept. of Information and Computer Science, [http://kdd.ics.uci.edu].

[Hongeng et al., 2004] Hongeng, S., Nevatia, R., and Bremond, F. (2004). Video-based event recognition: activity representation and probabilistic recognition methods. *Computer Vision and Image Understanding*, 96(2):129 – 162. Special Issue on Event Detection in Video.

[Huang et al., 2005] Huang, W., Nakamori, Y., and Wang, S. (2005). Forecasting stock market movement direction with support vector machine. *Computers & Operations Research*, 32(10):2513–2522.

[Jaakkola and Haussler, 1998] Jaakkola, T. and Haussler, D. (1998). Exploiting generative models in discriminative classifiers. In *Advances in Neural Information Processing Systems 11 (NIPS 11)*, pages 487–493. MIT Press.

[Jebara et al., 2004] Jebara, T., Kondor, R., and Howard, A. (2004). Probability product kernels. *Journal of Machine Learning Research (JMLR)*, 5:819–844.

[Jebara et al., 2007] Jebara, T., Song, Y., and Thadani, K. (2007). Spectral Clustering and Embedding with Hidden Markov Models. In *Proc. of the 18th European Conference on Machine Learning (ECML)*, Warsaw, Poland.

[Jin et al., 2004] Jin, G., Tao, L., and Xu, G. (2004). Hidden Markov Model Based Events Detection in Soccer Video. In *Proc. International Conference on Image Analysis and Recognition*, pages 605–612.

[Juang and Rabiner, 1985] Juang, B. and Rabiner, L. (1985). A Probabilistic Distance Measure for Hidden Markov Models. *AT&T Technical Journal*, 64(2):391–408.

[Kohavi, 1995] Kohavi, R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Proceedings of the International Joint Conference on Artificial Intelligence*.

[Kullback and Leibler, 1951] Kullback, S. and Leibler, R. (1951). On Information and Sufficiency. *Annals of Mathematical Statistics*, 22:79–86.

[Li and Biswas, 2000] Li, C. and Biswas (2000). A Bayesian Approach to Temporal Data Clustering using Hidden Markov Models. In *International Conf. on Machine Learning*, pages 543–550.

[Liang and Ouhyoung, 2002] Liang, R. and Ouhyoung, M. (2002). A real-time continuous gesture recognition system for sign language. In *Automatic Face and Gesture Recognition, 1998. Proceedings. Third IEEE International Conference on*, pages 558–567. IEEE.

[Liao, 2005] Liao, T. W. (2005). Clustering of time series data–a survey. *Pattern Recognition*, 38(11):1857 – 1874.

[Liese and Vajda, 2006] Liese, F. and Vajda, I. (2006). On divergences and information in statistics and information theory. *IEEE Transactions on Information Theory*, 52(10):4394–4412.

[L.J.Savage, 1971] L.J.Savage (1971). Elicitation of personal probabilities and expectations. *Journal of the American Statistical Association*, 66(336):783–801.

[Louradour et al., 2007] Louradour, J., Daoudi, K., and Bach, F. (2007). Feature space mahalanobis sequence kernels: Application to svm speaker verification. *IEEE Transactions on Audio, Speech and Language Processing*, 15(8):2465–2475.

[Lyu, 2005] Lyu, S. (2005). A kernel between unordered sets of data: the Gaussian mixture approach. In *Proc. ECML*.

[Mahalanobis, 1936] Mahalanobis, P. (1936). On the generalized distance in statistics. *Proceedings of the National Institute of Sciences of India*, 2:49–55.

[Malioutov and Barzilay, 2006] Malioutov, I. and Barzilay, R. (2006). Minimum cut model for spoken lecture segmentation. In *In Proceedings of the Annual Meeting of the Association for Computational Linguistics (COLING-ACL 2006*, volume 2006, pages 25–32.

[Marcel and Millan, 2007] Marcel, S. and Millan, J. d. R. (2007). Person authentication using brainwaves (eeg) and maximum a posteriori model adaptation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29:743–752.

[Marchand and Taylor, 2003] Marchand, M. and Taylor, J. S. (2003). The set covering machine. *J. Mach. Learn. Res.*, 3:723–746.

[McKinney and Breebart, 2003] McKinney, M. F. and Breebart, J. (2003). Features for audio and music classification. In *Proc. Intl. Symp. Music Information Retrieval (ISMIR)*, pages 151–158.

[Meng et al., 2007] Meng, A., Ahrendt, P., Larsen, J., and Hansen, L. K. (2007). Temporal feature integration for music genre classification. *IEEE Trans. Audio Speech and Lang. Process.*, 15:1654–1664.

[Microsoft, 2011] Microsoft (2011). Kinect. `http://www.xbox.com/kinect`.

[Mitchell, 1997] Mitchell, T. (1997). *Machine Learning*. McGraw-Hill Book Company.

[Müller, 2007] Müller, M. (2007). *Information Retrieval for Music and Motion*. Springer.

[Mullin and Sukthankar, 2000] Mullin, M. and Sukthankar, R. (2000). Complete cross-validation for nearest neighbor classifiers. In *Proceedings of the International Conference on Machine Learning*.

[Murphy, 2002] Murphy, K. (2002). *Dynamic Bayesian Networks: Representation, Inference and Learning*. PhD thesis, UC Berkeley, Computer Science Division.

[Ng et al., 2002] Ng, A., Jordan, M., and Weiss, Y. (2002). On Spectral Clustering: Analysis and an Algorithm. In *Advances in Neural Information Processing Systems*.

[Nguyen et al., 2008] Nguyen, X., Wainwright, M., and Jordan, M. (2008). Estimating divergence functionals and the likelihood ratio by penalized convex risk minimization. In Platt, J., Koller, D., Singer, Y., and Roweis, S., editors, *Advances in Neural Information Processing Systems 20*, pages 1089–1096.

[Oates et al., 2001] Oates, T., Firoiu, L., and Cohen, P. (2001). Using Dynamic Time Warping to Bootstrap HMM-Based Clustering of Time Series. In *Sequence Learning - Paradigms, Algorithms, and Applications*, pages 35–52, London, UK. Springer-Verlag.

[Panuccio et al., 2002] Panuccio, A., Bicego, M., and Murino, V. (2002). A Hidden Markov Model-Based Approach to Sequential Data Clustering. In *Proc. of the Joint IAPR International Workshop on Structural, Syntactic and Statistical Pattern Recognition*, pages 734–742.

[Pearson, 1900] Pearson, K. (1900). On the criterion that a given system of deviations from the probable in the case of correlated system of variables is such that it can be reasonable supposed to have arisen from random sampling. *Phil. Mag.*, 50:157–172.

[Perez-Cruz, 2008] Perez-Cruz, F. (2008). Estimation of information theoretic measures for continuous random variables. In *Advances in Neural Information Processing Systems 21 (NIPS*.

[Porikli, 2004] Porikli, F. (2004). Clustering Variable Length Sequences by Eigenvector Decomposition Using HMM. In *Proc. International Workshop on Structural and Syntactic Pattern Recognition*, pages 352–360, Lisbon, Portugal.

[Rabiner, 1989] Rabiner, L. (1989). A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. *Proc. of the IEEE*, 77(2):257–286.

[Rabiner and Juang, 1993] Rabiner, L. and Juang, B. H. (1993). *Fundamentals of Speech Recognition*. Prentice Hall, united states ed edition.

[Ramoni et al., 2002] Ramoni, M., Sebastiani, P., and Cohen, P. (2002). Bayesian clustering by dynamics. *Machine Learning*, 47(1):91–121.

[Reid and Williamson, 2009] Reid, M. D. and Williamson, R. C. (2009). Information, divergence and risk for binary experiments. Technical report, Australian National University.

[Reynolds, 2002] Reynolds, D. (2002). An overview of automatic speaker recognition technology. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP'02)*, volume 4, pages 4072–4075.

[Rogers and Wagner, 1978] Rogers, W. and Wagner, T. (1978). A finite sample distribution-free performance bound for local discrimination rules. *Annals of Statistics*, 6(3):506–514.

[Sakoe and Chiba, 1978] Sakoe, H. and Chiba, S. (1978). Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 26(1):43–49.

[Sanguinetti et al., 2005] Sanguinetti, G., Laidler, J., and Lawrence, N. D. (2005). Automatic determination of the number of clusters using spectral algorithms. In *IEEE Workshop on Machine Learning for Signal Processing*, pages 55–60.

[Schoelkopf and Smola, 2001] Schoelkopf, B. and Smola, A. (2001). *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond (Adaptive Computation and Machine Learning)*. The MIT Press.

[Shawe-Taylor and Cristianini, 2004] Shawe-Taylor, J. and Cristianini, N. (2004). *Kernel Methods for Pattern Analysis*. Cambridge University Press.

[Shi and Malik, 2000] Shi, J. and Malik, J. (2000). Normalized Cuts and Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 22(8):888–905.

[Shoeb and Guttag, 2010] Shoeb, A. and Guttag, J. (2010). Application of machine learning to epileptic seizure detection. In *International Conference on Machine Learning (ICML)*.

[Sigurdsson et al., 2006] Sigurdsson, S., Petersen, K. B., and Lehn-Schioler, T. (2006). Mel frequency cepstral coefficients: An evaluation of robustness of mp3 encoded music. In *Proc. Intl. Symp. Music Information Retrieval (ISMIR)*.

[Smola et al., 2007] Smola, A., Gretton, A., Song, L., and Schölkopf, B. (2007). A Hilbert space embedding for distributions. In *Algorithmic Learning Theory*, pages 13–31. Springer.

[Smyth, 1997] Smyth, P. (1997). Clustering Sequences with Hidden Markov Models. *Advances in Neural Information Processing Systems*, 9:648–654.

[Sriperumbudur et al., 2009] Sriperumbudur, B., Fukumizu, K., Gretton, A., Lanckriet, G., and Schoelkopf, B. (2009). Kernel choice and classifiability for rkhs embeddings of probability distributions. In Bengio, Y., Schuurmans, D., Lafferty, J., Williams, C. K. I., and Culotta, A., editors, *Advances in Neural Information Processing Systems 22*, pages 1750–1758.

[Österreicher, 2002] Österreicher, F. (2002). Csiszar's f-divergences-basic properties. Technical report, Res. Report Collection.

[Österreicher and Vajda, 1993] Österreicher, F. and Vajda, I. (1993). Statistical information and discrimination. *IEEE Transactions on Information Theory*, 39(3):1036–1039.

[Stewart and Sun, 1990] Stewart, G. and Sun, J. (1990). *Matrix Perturbation Theory*. Academic Press.

[Stone, 1977] Stone, C. (1977). Consistent nonparametric regression. *Annals of Statistics*, 5:595–645.

[Suzuki et al., 2009] Suzuki, T., Sugiyama, M., and Tanaka, T. (2009). Mutual information approximation via maximum likelihood estimation of density ratio. In *Proceedings of 2009 IEEE International Symposium on Information Theory (ISIT2009)*, pages 463–467, Seoul, Korea.

[Szlam et al., 2008] Szlam, A., Coifman, R., and Maggioni, M. (2008). A general framework for adaptive regularization based on diffusion processes. *Journal of Machine Learning Research (JMLR)*, 9(9):1711–1739.

[Tzanetakis and Cook, 2002] Tzanetakis, G. and Cook, P. (2002). Musical genre classification of audio signals. *IEEE Trans. Speech and Audio Process.*, 10:293–302.

[von Luxburg, 2007] von Luxburg, U. (2007). A Tutorial on Spectral Clustering. *Statistics and Computing*, 17(4).

[Wang et al., 2009] Wang, Q., Kulkarni, S., and Verdú, S. (2009). Divergence estimation for multidimensional densities via k-nearest-neighbor distances. *IEEE Transactions on Information Theory*, 55:2392–2405.

[Wang et al., 2005] Wang, Q., Kulkarni, S., and Verdu, S. (2005). Divergence estimation based on data-dependent partitions. *IEEE Transactions on Information Theory*, 51(9):3064–3074.

[West et al., 2006] West, K., Cox, S., and Lamere, P. (2006). Incorporating machine-learning into music similarity estimation. In *Proceedings of the 1st ACM workshop on Audio and music computing multimedia*, pages 89–96. ACM.

[Wu and Huang, 1999] Wu, Y. and Huang, T. (1999). Vision-based gesture recognition: A review. In Braffort, A., Gherbi, R., Gibet, S., Teil, D., and Richardson, J., editors, *Gesture-Based Communication in Human-Computer Interaction*, volume 1739 of *Lecture Notes in Computer Science*, pages 103–115. Springer Berlin / Heidelberg.

[Wu and Leahy, 1993] Wu, Z. and Leahy, R. (1993). An Optimal Graph-Theoretic Approach to Data Clustering: Theory and its Application to Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 15(11):1101–1113.

[X.L. Nguyen and Jordan, 2009] X.L. Nguyen, M. W. and Jordan, M. (2009). On surrogate loss functions and f- divergences. *Annals of Statistics*, 37(2):876–904.

[Xu et al., 2003] Xu, G., Ma, Y., Zhang, H., and Yang, S. (2003). A HMM based semantic analysis framework for sports game event detection. In *Proceedings. 2003 International Conference on Image Processing (ICIP'03)*, volume 1. IEEE.

[Xu et al., 2004] Xu, L., Neufeld, J., Larson, B., and Schuurmans, D. (2004). Maximum margin clustering. In *Advances in Neural Information Processing Systems (NIPS)*.

[Xu and Wunsch-II, 2005] Xu, R. and Wunsch-II, D. (2005). Survey of Clustering Algorithms. *IEEE Transactions on Neural Networks*, 16(3):645–678.

[Yin and Yang, 2005] Yin, J. and Yang, Q. (2005). Integrating Hidden Markov Models and Spectral Analysis for Sensory Time Series Clustering. In *Fifth IEEE International Conference on Data Mining*.

[Zelnik-Manor and Irani, 2001] Zelnik-Manor, L. and Irani, M. (2001). Event-based analysis of video. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition,*, 2:123.

[Zelnik-Manor and Perona, 2004] Zelnik-Manor, L. and Perona, P. (2004). Self-tuning spectral clustering. *Advances in Neural Information Processing Systems*, pages 1601–1608.

[Zhou and Chellappa, 2006] Zhou, S. K. and Chellappa, R. (2006). From sample similarity to ensemble similarity: Probabilistic distance measures in reproducing kernel hilbert space. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 28(6):917–929.