Algorítmica

Curso 2023-2024

Grupo Viterbi

PRÁCTICA 5- PROGRAMACIÓN DINÁMICA

Integrantes:

Miguel Ángel De la Vega Rodríguez	miguevrod@correo.ugr.es
Alberto De la Vera Sánchez	joaquinrojo724@correo.ugr.es
Joaquín Avilés De la Fuente	adelaveras01@correo.ugr.es
Manuel Gomez Rubio	e.manuelgmez@go.ugr.es
Pablo Linari Perez	e.pablolinari@go.ugr.es

Facultad de Ciencias UGR
Escuela Técnica Ingeniería Informática UGR
Granada
2023-2024

Índice general

Apartado 1

Autores

- **Miguel Ángel De la Vega Rodríguez:** 20%
  - Estructura del documento
  - Ejemplos de Uso
- **Joaquín Avilés De la Fuente:** 20%
  - Explicación del Modelo Oculto de Markov (HMM)
  - Explicación del Algoritmo de Viterbi (incluyendo sus pasos)
  - Documentación de la Introducción
- **Alberto De la Vera Sánchez:** 20%
  - RELLENAR
- **Manuel Gomez Rubio** 20%
  - RELLENAR
- **Pablo Linari Pérez:** 20%
  - RELLENAR

Apartado 2

Equipo de trabajo

- **Miguel Ángel De la Vega Rodríguez:** (Ordenador donde se ha realizado el computo)
  - AMD Ryzen 7 2700X 8-Core
  - 16 GB RAM DDR4 3200 MHz
  - NVIDIA GeForce GTX 1660 Ti
  - 1 TB SSD NvMe
  - Debian 12 Bookworm
  - Compilador GCC 12.2.0

Apartado 3

Introducción

El algoritmo de Viterbi es un algoritmo de programación dinámica utilizado para encontrar la secuencia de estados más probable en un **modelo oculto de Markov (HMM, Hidden Markov Model)**. El algoritmo de Viterbi es ampliamente utilizado en diversos campos, como el lenguaje, la ingeniería de comunicaciones, la robótica, la biología, la medicina, la meteorología, etc.

En este documento, presentamos una descripción detallada del algoritmo de Viterbi, su implementación y ejemplos de uso en diferentes campos, haciendo sobre todo énfasis en el uso de la programaciónn dinámica.

Apartado 4

Descripción del Algoritmo

Como bien hemos comentado en la introducción el algoritmo de Viterbi es un algoritmo de programación dinámica utilizado para encontrar la secuencia de estados más probable en un modelo oculto de Markov (HMM, Hidden Markov Model), por lo que en primer lugar pasaremos a explicar dicho modelo así como los elementos que lo componen y su sintaxis matemática, para poder comprender mejor el algoritmo de Viterbi y sus ejemplos.

4.1 Modelo Oculto de Markov (HMM)

Un modelo oculto de Markov (HMM) es un modelo estadístico que describe la secuencia de estados a través de la cual pasa un proceso estocástico. En un HMM, el proceso estocástico es un proceso de Markov, lo que significa que la probabilidad de que el sistema pase a un estado futuro depende únicamente del estado actual y no de los estados anteriores, de forma que el calculo del estado más probables se hace mediante la probabilidad de transición entre estados y de la probabilidad de emisión de observaciones.

Un HHM se representa mediante una tupla $(Q, V, \pi, A, B,)$ y explicaremos a continuación sus componentes:

- Un conjunto de estados $Q = \{q_1, q_2, \ldots, q_N\}$.
- Un conjunto de estados observables $V = \{v_1, v_2, \ldots, v_L\}$.
- Un conjunto de probabilidades de transición entre estados $A = \{a_{ij}\}$, donde $a_{ij} = P(q_t = j | q_{t-1} = i)$, es decir, la probabilidad de pasar de un estado i en el instante de tiempo $t-1$ a un estado j en el instante de tiempo t.
- Un conjunto de probabilidades de las observaciones $B = \{b_j(v_k)\}$, donde $b_j(v_k) = P(o_t = v_k | q_t = j)$, es decir, la probailidad de observar v_k en el estado j en el instante de tiempo t.
- Un conjunto de probabilidades de estados iniciales $\pi = \{\pi_i\}$, donde $\pi_i = P(q_1 = i)$, es decir, la probabilidad de que en el instante 1 (al inicio) se tenga el estado i.

En dicho modelo, notaremos a la secuencia de observaciones como el conjunto $O = \{o_1, o_2, \ldots, o_L\}$ y al conjunto de estados ocultos con mayor probabilidad como $S = \{s_1, s_2, \ldots, s_L\}$, donde L es el número de observaciones, por lo que como se podía deducir se tiene el mismo número de observaciones que estados ocultos en la secuencia con mayor probabilidad, pues estos estados serán consecuencia (en parte) de las observaciones dadas.

4.2 Algoritmo de Viterbi

En esta sección procederemos a explicar el **Algoritmo de Viterbi**, donde daremos una explicación detallada de su funcionamiento y del uso de programación dinámica en el mismo.

El algoritmo de Viterbi se basa en la idea de que la probabilidad de la secuencia de estados más probable hasta el instante de tiempo t se puede calcular a partir de la probabilidad de la secuencia de estados más probable hasta el instante de tiempo $t-1$.

El algoritmo de Viterbi se puede resumir en los siguientes pasos:

- Inicialización: Calcular la probabilidad de la secuencia de estados más probable hasta el instante de tiempo $t = 1$, es decir, obtener el estado inicial más probable
- Recursión: Calcular la probabilidad de la secuencia de estados más probable hasta el instante de tiempo t a partir de la probabilidad de la secuencia de estados más probable hasta el instante de tiempo $t-1$.
- Terminación: Calcular la probabilidad de la secuencia de estados más probable hasta el instante de tiempo T.
- Reconstrucción de la secuencia de estados: Reconstruir la secuencia de estados más probable a partir de las probabilidades calculadas.

El uso de programación dinámica en este caso es claro, tener una matriz de probabilidades donde se almacenen la probabilidad máxima de cada estado hasta el instante de tiempo t, lo que nos permitirá calcular la probabilidad máxima de cada estado hasta el instante de tiempo $t+1$ de forma eficiente. Por tanto tendremos una matriz $\mathcal{M}$ de N filas (una para cada estado q_k donde $k \in N$) y $T = L$ columnas (una para cada instante de tiempo $t \in T = L$). Recordemos que teníamos la secuencia de observaciones $O = \{o_1, o_2, \ldots, o_L\}$.

A continuación procederemos a explicar los pasos del algoritmo de Viterbi de forma más detallada.

4.2.1 Inicialización

La matriz $\mathcal{M}$ se inicializa de la siguiente forma:

- $\mathcal{M}[k][1] = \pi_k \cdot b_k(o_1) \forall k \in N$, es decir, la probabilidad de que el estado k sea el estado inicial multiplicado por la probabilidad de que se observe la primera observación en el estado k. Destacar que N representa la cantidad de estados posibles ocultos en el modelo oculto de Markov.

4.2.2 Recursión

Una vez inicializada la matriz $\mathcal{M}$, se procede a calcular las probabilidades de la secuencia de estados más probable hasta el instante de tiempo t a partir de la probabilidad de la secuencia de estados más probable hasta el instante de tiempo $t-1$, es decir, para calcular los valores de la columnas t usaremos los de la columna $t-1$, es aquí donde hacemos un uso claro de la programación dinámica. Para ello, se utiliza la siguiente fórmula:

- $\mathcal{M}[j][t] = \max_{i=1}^{N} \left\{ \mathcal{M}[i][t-1] \cdot a_{ij} \cdot b_j(o_t) \right\}$, es decir, se multiplica la probabilidad máxima del estado i en el instante $t-1$ por la probabilidad de transición de i a j por la probabilidad de observar o_t en el estado j, todo esto para todo $i \in N$ y se selecciona el máximo de estos valores. Tenemos así la máxima probabilidad de llegar al estado $j \in N$ en el instante de tiempo t.

4.2.3 Terminación

En esta última fase el objetivo será obtener cual es la máxima probabilidad de llegar a un estado $k \in N$ en el instante de tiempo $t = T$, es decir, la probabilidad de la secuencia de estados más probable hasta el instante de tiempo $t = T$. Para ello, se utiliza la siguiente fórmula:

- $P = \max_{i=1}^{N} \{\mathcal{M}[i][T]\}$, es decir, se selecciona el máximo de las probabilidades de llegar a cada estado en el instante de tiempo $t = T$.

donde tenemos que P almacena la probabilidad máxima de todos los posibles estados $k \in N$ en el instante de tiempo $t = T$.

4.2.4 Reconstrucción de la secuencia de estados

Una vez obtenida la probabilidad máxima de la secuencia de estados más probable hasta el instante de tiempo $t = T$, se procede a reconstruir la secuencia de estados más probable. La idea a desarrolar es la siguiente:

- En el instante $t = T$ se selecciona el estado k que maximiza la probabilidad de llegar a dicho estado en el instante de tiempo $t = T$, es decir, se selecciona el estado k tal que $\mathcal{M}[k][T] = P$, ya obtenido dicho valor obtendremos el estado s en el instante $t-1$ que maximiza la probabilidad de llegar a k en el instante de tiempo $t = T$, es decir, se selecciona el estado s tal que $\mathcal{M}[s][T-1] \cdot a_{sk} \cdot b_k(o_T) = \mathcal{M}[k][T] = P$, y así sucesivamente hasta llegar al instante de tiempo $t = 1$.

Tenemos ahora la idea con la que obtendremos dicha secuencia más probable de estados, es decir, la secuencia de estados $S = \{s_1, s_2, \ldots, s_L\}$ que maximiza la probabilidad de la secuencia de observaciones $O = \{o_1, o_2, \ldots, o_L\}$. Veamos por tanto como vamos a hacerlo:

- En el instante de tiempo $t-1$ se selecciona el estado $i \in N$ y se multiplica dicho valor por la probabilidad de transición de i a k y por la probabilidad de observar o_t en el estado k, si dicho valor nuevo obtenido coincide con el valor de la matriz $\mathcal{M}[k][T] = P$, entonces se selecciona el estado i como el estado anterior a k en la secuencia de estados más probable. En caso contrario se hace dicha comprobación para todos los estados $i \in N$, obteniendo así el estado s en el instante $t-1$ de la secuencia de estados ocultos con mayor probabilidad.
- Este proceso se repite para todo $t \in T$ hasta llegar al instante de tiempo $t = 1$, obteniendo así la secuencia de estados ocultos con mayor probabilidad.

Apartado 5

Ejemplos de Uso

El algoritmo de Viterbi es ampliamente utilizado en diversos campos tales como el lenguaje, la ingeniería de comunicaciones, la robótica, la biología, la medicina, la meteorología, etc. A continuación presentamos algunas de las aplicaciones específicas de este algoritmo en distintos campos.

5.1 Lenguaje

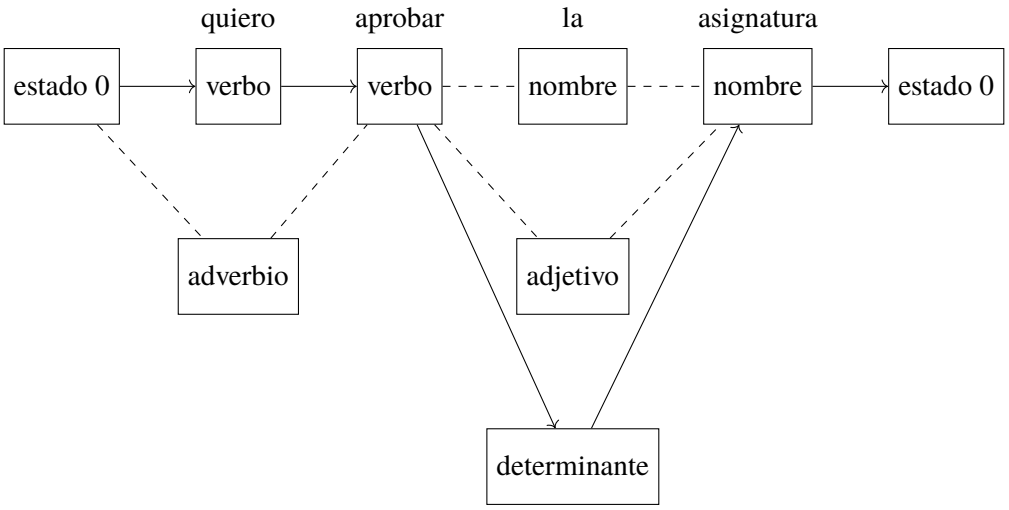
El algoritmo de Viterbi es utilizado en el reconocimiento de voz, en el reconocimiento de escritura a mano, en la corrección de errores en el texto, en la traducción automática, en la generación de texto, en la síntesis de voz, en la transcripción de audio, etc. Veamos un ejemplo de uso de Viterbi en el proceso de desambiguación de palabras en un texto.

5.1.1 Desambiguación de palabras

Cuando queremos procesar el lenguaje natural, es común encontrarnos con palabras que tienen múltiples significados. Por ejemplo, la palabra *privado* está reconocida por la RAE como un adjetivo, un sustantivo y un verbo. Para desambiguar estas palabras cuando procesamos un texto, podemos utilizar el algoritmo de Viterbi, en este caso particular, los elementos del modelo oculto de Markov serían:

- El conjunto $\boldsymbol{Q}$ de estados ocultos (categorías gramaticales)
- El conjunto $\boldsymbol{V}$ de estados observables (palabras)
- El conjunto $\boldsymbol{A}$ de probabilidades de transición entre estados (probabilidades de cambio de categoría gramatical, por ejemplo, que un nombre vaya detras de un verbo)
- El conjunto $\boldsymbol{B}$ de probabilidades de emisión de observaciones (probabilidades de que una palabra pertenezca a una categoría gramatical, por ejemplo, que la palabra *perro* sea un sustantivo es mucho mayor a que sea un adjetivo)

Mostramos un ejemplo para el texto *quiero aprobar la asignatura*:



Donde, los observables son la secuencia de palabras *quiero aprobar la asignatura* y los estados ocultos son las categorías gramaticales de las palabras, que como se puede ver, contemplan sólo un conjunto limitado de categorías gramaticales. Esto se debe a que la probablidad de pertenencia de determinadas palabras a ciertas categorías gramaticales es 0, lo que simplifica y acelera el proceso de desambiguación.

Aplicación del Algoritmo de Viterbi

El algoritmo de Viterbi se utiliza para encontrar la secuencia más probable de estados ocultos (categorías gramaticales) dada una secuencia de observaciones (palabras). Para nuestro ejemplo, el proceso se desarrolla de la siguiente manera:

- Inicialización:
  - Definir los estados ocultos posibles. Por ejemplo, en nuestro caso, los estados posibles son verbo, nombre, determinante, etc.
  - Definir las probabilidades iniciales para cada estado oculto. Por ejemplo, es muy probable que la primera palabra quiero sea un verbo.
  - $\pi(\text{verb}) = P(\text{verb}|\text{estado}_0)$.
- Recursión:
  - Para cada palabra en la secuencia, calcular la probabilidad de que cada posible estado oculto (categoría gramatical) siga a cada estado anterior. Esto se hace utilizando las probabilidades de transición (A) y las probabilidades de emisión (B).
  - Ejemplo: Para la palabra aprobar, se calcularía $P(\text{verb}_2|\text{verb}_1) \cdot P(\text{aprobar}|\text{verb}_2)$ para todas las categorías posibles de aprobar.
  - Se selecciona el estado que maximiza esta probabilidad.
- Terminación:
  - Una vez procesadas todas las palabras, se selecciona la secuencia de estados que maximiza la probabilidad total de la secuencia observada.
  - Esta secuencia representará las categorías gramaticales más probables para la oración.
- Reconstrucción de la secuencia de estados:

  - Utilizando las probabilidades calculadas, se reconstruye la secuencia de categorías gramaticales más probable.
  - Por ejemplo: quiero (verbo), aprobar (verbo), la (determinante), asignatura (nombre).

Reconocimiento de voz

Problema: Dada una secuencia de observaciones acústicas, queremos predecir la secuencia de las palabras que se obtendrán. Paso 1: Definir los Componentes del Modelo Oculto de Markov (HMM)

- Estados Ocultos (Q): Fonemas (sonidos básicos del habla)
- Observaciones (V): Características acústicas extraídas de la señal de voz (vectores de características como Mel-Frequency Cepstral Coefficients - MFCC)
- Probabilidades de Transición (A): Probabilidades de transición entre fonemas.
- Probabilidades de Emisión (B): Probabilidades de emitir una característica acústica específica dado un fonema.

Mel-Frequency Cepstral Coefficients - MFCC estudia los diferentes tonos, acentuaciones, etc. que tienen las palabras dentro de los idiomas, lo que permite a las IAs aprender con mayor precisión los lenguajes mediante señales de audio

- Inicialización:
  Definir Estados Ocultos posibles. En este caso, son los diferentes fonemas que se pueden emplear en los idiomas

  Definir las probabilidades de cada estado oculto. {pi fonema: Probabilidad inicial de cada fonema. Estas probabilidades pueden estar basadas en la frecuencia de los fonemas al inicio de palabras en el idioma.
- Recursión:
  Seleccionado un elemento cualquiera, sin que sea el primero, de la secuencia, hay calcular la probabilidad para cada estado oculto pposible. Para cada segmento de la señal de voz (cada observación acústica), calculamos la probabilidad de que cada posible estado oculto (fonema) siga a cada estado anterior utilizando las probabilidades de transición y de emisión.

  y se seleccionará el estado que maximice la probabilidad.
- Terminación:
  Una vez procesados los fonemas que son empleados en las palabras, se selecciona la secuencia de estados que maximiza la probabilidad total de la secuencia obtenida. Obteniendo así, la secuencia de fonemas con mayor probabilidad
- Reconstrucción de la secuencia de estados:
  Se usan las probabilidades calculadas para reconstruir la secuencia de fonemas más probables.

Usamos las probabilidades calculadas para reconstruir la secuencia de fonemas más probable, y luego mapeamos estos fonemas a palabras usando un diccionario fonético. Ejemplo Práctico

Imaginemos que la secuencia de observaciones acústicas es O=o1,o2,o3y los fonemas posibles son /k/, /ae/, /t/, que corresponden a las palabras çat". Inicialización Recursión

Y así sucesivamente para /ae/ y /t/. Terminación Obtendremos la secuencia con mayor probabilidad Reconstrucción Se reconstruye esta secuencia obteniendo que la secuencia de fonemas más probable es /k/ -> /ae/ -> /t/, que mapeamos a la palabra çat".

Este ejemplo muestra cómo el algoritmo de Viterbi se utiliza para convertir una secuencia de observaciones acústicas en una secuencia de fonemas, y luego en palabras, en el reconocimiento de voz.

5.2 Predicción de genes

En esta sección, presentamos un ejemplo de uso del algoritmo de Viterbi en la predicción de genes en secuencias de ADN. La predicción de genes es un problema importante en bioinformática, En general, la predicción de genes trata de localizar en las largas secuencias de ADN, y de forma automatizada, las subsecuencias de nucleótidos que conforman los diferentes genes.

Explicación del problema

Existen cuatro tipos de nucleótidos, que se suelen representar por las letras A, C, G y T en función de la base nitrogenada que contengan: Adenina, Citosina, Guanina o Timina. Una cadena compuesta por estas cuatro letras representa la estructura primaria de una molécula de ADN. Por tanto el algoritmo se utiliza para identificar exones que poseen la información necesaria para la sintesis de proteínas .Los modelos ocultos de Markov (HMM) son una herramienta común en la predicción de genes debido a su capacidad para manejar secuencias donde los estados exones e intrones (sirven de mensajeros para el codigo de los exones) son ocultos y solo las secuencias de ADN observables están disponibles.

Definición del modelo oculto de Markov (HMM)

Primero definimos los elementos del modelo oculto de Markov:

- El conjunto Q de estados ocultos (exones e intrones)
- El conjunto V de estados observables (nucleótidos)
- El conjunto A de probabilidades de transición entre estados (probabilidades de cambio de exón a intrón y viceversa)
- El conjunto B de probabilidades de emisión de observaciones (probabildes de observar una A en un exón)

El modelo de Markov oculto dado, trata de capturar las diferencias estadísticas en exones e intrones. El modelo tiene cuatro estados, donde E_1, E_2, y E_3 se utilizan para modelar las propiedades estadísticas básicas en los exones. Cada estado E_i utiliza un conjunto de probabilidades de emisión diferentes para reflejar el símbolo en la posición i de un codón(secuencia de 3 nuecleótidos ej: ATC). El estado I se utiliza para modelizar los intrones.

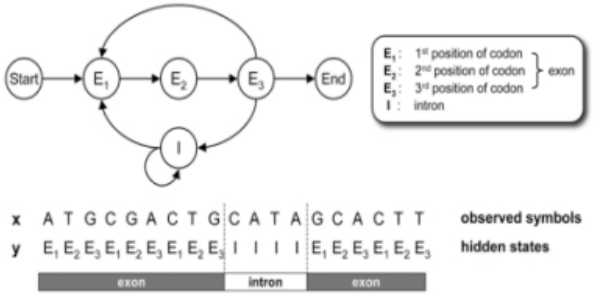


Figura 5.1: modelo de Markov

El modelo construido se utiliza ahora para analizar nuevas secuencias observadas. Por ejemplo, supongamos que tenemos una nueva secuencia de ADN $X = x_1, ..., x_19 = ATGCGACTGCATAGCACTT$. ¿Cómo podemos saber si esta secuencia de ADN es la región codificada de un gen o no? Podemos responder a la pregunta calculando la probabilidad observada de X basada en el modelo dado, que modeliza genes codificantes. Si esta probabilidad es alta, implica que esta secuencia de ADN es probablemente la de la región codificada de un gen (porción de adn que codifíca la proteina).

Aplicación del algoritmo

- Inicialización:
  - Se inicializa una matriz para almacenar las probabilidades de la secuencia más probable que termina en cada estado, para cada posición en la secuencia de ADN.
  - Se inicializa otra matriz para realizar un seguimiento de los estados previos
  - $\pi(\text{verb}) = P(\text{verb}|\text{estado}_0)$.
- Recursión:
  - Para cada posición en la secuencia de ADN y para cada estado, se calcula la probabilidad de que la secuencia más probable termine en ese estado en esa posición. Esto se hace usando las probabilidades de transición y de emisión.
  - Se actualizan las matrices de probabilidades y de estados previos.
  - Se selecciona el estado que maximiza esta probabilidad.
- Terminación:
  - Una vez que se llega al final de la secuencia de ADN, se selecciona el estado con la probabilidad más alta en la última posición de la matriz de probabilidades.
  - Se realiza un seguimiento hacia atrás usando la matriz de estados previos para determinar la secuencia de estados más probable.
- Reconstrucción de la Predicción de Genes:
  - La secuencia de estados obtenida se traduce en regiones codificantes (exones) e intrones.
  - Las predicciones se ajustan y se anotan en la secuencia de ADN.

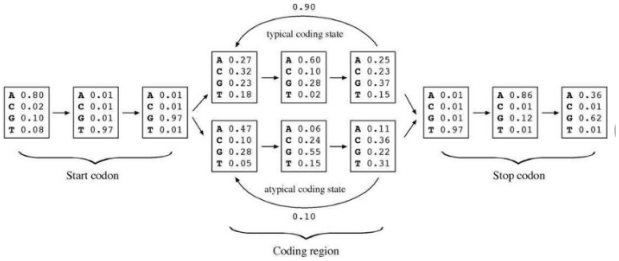


Figura 5.2: Cálculo para una procariota

5.3 Integración en el software

Existen varias herramientas bioinformáticas que implementan el algoritmo de Viterbi para la predicción de genes como GeneMark que utiliza Cadenas Ocultas de markov para identificar genes en células procariotas y eucariotas . Glimmer es otra herramienta que utiliza el algoritmo de Viterbi para la predicción de genes en genomas bacterianos .Estos programas suelen integrar datos adicionales y heurísticas para mejorar la precisión de la predicción, adaptándose a las características específicas del genoma en estudio. En resumen, el algoritmo de Viterbi es una técnica poderosa para la predicción de genes, aprovechando su capacidad para manejar secuencias y estados ocultos en un contexto biológico complejo ya que muchas veces se necesita modelizar el comportamiento futuro del ADN para detectar genes y regiones codificantes y así poder detectar enfermedades genéticas y tratarlas a tiempo.