

Introducción a la Optimización, Tipos de Métodos y Convexidad

Dr. Ing. Claudio Risso (crisso@fing.edu.uy)

Instituto de Computación (FING - UDELAR)

Curso Optimización Continua y Aplicaciones (OCA)
(Agosto 2023)

¿Qué es la optimización?

Podemos pensar que: *“la optimización busca la gestión eficiente de recursos escasos mediante un enfoque cuantitativo”*.

Existe un objetivo (eventualmente pueden ser dos o más) que se busca maximizar o minimizar. Por ejemplo, se busca minimizar los costos de distribución de mercaderías.

El problema depende de un conjunto de variables de control, que puedo fijar para cambiar el valor del objetivo. Por ejemplo, cómo organizar la logística de la distribución: recorridos de los camiones, el cronograma de entregas, etc.

También existe un conjunto de restricciones técnicas a cumplir. En el ejemplo anterior serían: límites de carga en los camiones, tiempos máximos de entrega en los distintos clientes, etc.

¿Qué es la optimización?

Para plantear un problema de optimización, hay que poder cuantificar los elementos anteriores, idealmente expresarlos en forma algebraica.

El proceso puede llegar a comprender los siguientes pasos:

- 1 El análisis matemático del sistema, así como la construcción y validación de un modelo abstracto
- 2 Encontrar una solución para el problema que surge del modelo general en conjunto con los datos concretos de la aplicación
- 3 Aplicar y hacer el seguimiento de los cambios propuestos

En este curso, se repasan propiedades teóricas y técnicas de optimización que tienen por fin encontrar soluciones a distintos tipos de problemas (paso 2).

También se ven aplicaciones concretas como ejemplo del paso 1. Especialmente en los obligatorios.

¿En qué se usa?

Un esquema estándar de los pasos/etapas hacia la excelencia en la gestión de un problema (negocio o similar) son:

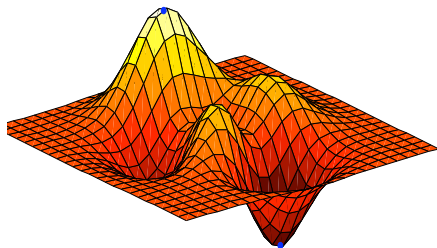
- 1 **Descriptivo** - Consiste en explorar y entender el problema. Busca determinar lo que realmente está pasando. El análisis de datos y la estadística son las principales herramientas.
- 2 **Predictivo** - Consiste en generar modelos para estimar qué podría suceder ante cambios en las variables de control o en las condiciones del problema. Las herramientas usuales son la inferencia estadística y el aprendizaje automático.
- 3 **Prescriptivo** - Con algún modelo como referencia, busca determinar cuáles son los pasos/decisiones a seguir con el fin de lograr algún objetivo.

La optimización aparece como herramienta del análisis descriptivo y predictivo, pero donde realmente es central es en el prescriptivo.

El problema general de optimización

Definición

Dado un dominio X para un conjunto de n variables (e.g. $X = \mathbb{R}^n$ o $X = \mathbb{N}^n$), una función objetivo $f : X \rightarrow \mathbb{R}$, y un conjunto $F \subseteq X$ de soluciones técnicamente viables para el problema, el planteo general de optimización (G) consiste en hallar $\bar{x} \in F$, tal que $f(\bar{x})$ es el mínimo (o máximo) valor posible de f .



$$(G) \begin{cases} \min f(x) \\ x \in F \end{cases}$$

Nos referimos a F como la *región factible del problema* (G) .

El problema general de optimización

El formato anterior es el más general, y permite expresar el problema aun cuando no se conozcan expresiones algebraicas (analíticas) para algunos de los objetos.

Supongamos que nuestro problema fuera coordinar las frecuencias y fases de los semáforos de una avenida, con fin de conseguir el tránsito más fluido posible en ella y su vecindad.

Podríamos elegir minimizar el tiempo medio entre que los vehículos ingresan y abandonan la avenida, y pedir además que los largos de las filas de ingreso a esa avenida desde calles conexas no superen ciertos umbrales con cierta probabilidad.

El problema (G) está bien definido (si tenemos datos históricos de tráfico), pero no es posible usar una expresión algebraica para calcular f y F . No obstante, podríamos simular sobre un conjunto grande de casos independientes (Monte-Carlo) y usar los promedios resultantes como estimadores.

El problema de Programación Matemática

Definición

Diremos que un problema de optimización (G) es de programación matemática (P), si la región factible F puede expresarse como la intersección de un conjunto de restricciones algebraicas en la variable x (combinado con el dominio X de las variables) y la función f también tiene expresión analítica.

$$(P) \begin{cases} \min f(x) \\ g_i(x) \leq 0, & 1 \leq i \leq p, \\ h_j(x) = 0, & 1 \leq j \leq q, \\ x \in X \end{cases}$$

El problema (P) tiene n variables y $m = p + q$ restricciones.

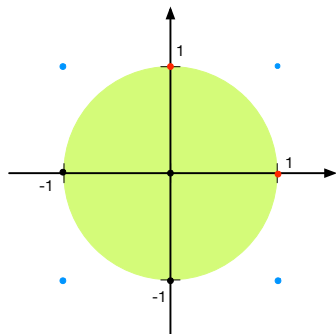
Cuando no se especifica X , se asume que $X \subseteq \mathbb{R}^n$.

El problema de Programación Matemática

El conjunto X tiene un impacto fundamental para el problema.

Cuando $X \subseteq \mathbb{Z}^2$ ese punto ya no es factible. El nuevo óptimo es 1 y se alcanza en: $(1, 0)$ y $(0, 1)$.

$$(P) \begin{cases} \max x + y \\ x^2 + y^2 \leq 1, \\ (x, y) \in \mathbb{Z}^2 \end{cases}$$



De hecho, sólo hay 5 puntos factibles en el nuevo problema.

Variantes del problema de Programación Matemática

- Cuando el conjunto $X \subseteq \mathbb{R}^n$, decimos que el problema es de optimización continua.
- Decimos que es de optimización combinatoria (Integer Programming, o IP) cuando $X \subseteq \mathbb{Z}^n$, y que es mixto (Mixed Integer Programming, o MIP) cuando parte de las variables son reales y las otras enteras.
- Cuando el conjunto de restricciones es vacío tenemos un problema de optimización sin restricciones.
- Cuando sí hay restricciones, la formulación de programación matemática supone que el conjunto es cerrado.
- Si la frontera no es parte de conjunto (restricciones de $<$), la teoría y las técnicas son más parecidas a las de optimización sin restricciones.

Propiedades y Técnicas

Este curso se centra en el optimización continua, y en propiedades teóricas o métodos para resolver esos problemas.

Ejemplo (adelanto) de propiedades conocidas:

- toda función continua en un compacto tiene máximo y mínimo
- si un punto \bar{x} es un óptimo local interior a la región factible de un problema (P) y f es diferenciable, entonces $\nabla f(\bar{x}) = \vec{0}$

Entre los tipos de métodos (algoritmos) destacamos:

Exactos - Tienen por objetivo alcanzar el óptimo, aunque eso implique hacerlo en una cantidad infinita de pasos.

Heurísticos - Buscan construcciones de buena calidad en pocas iteraciones, sin asegurar la convergencia al óptimo.

Este curso apunta a las técnicas exactas, estableciendo límites a la cantidad de iteraciones o buscando una precisión objetivo.

Propiedades y Técnicas

Entre los métodos exactos diferenciamos (no excluyente):

Directos - No conseguimos una solución al problema hasta terminar. Llegan a la solución en un número finito de pasos para ciertos problemas (asumiendo precisión absoluta).

Reductivos - Se construye una sucesión finita de instancias equivalentes hasta alcanzar una trivial (son directos).

Recursivos - La solución se construye incrementalmente, a partir de soluciones de instancias más pequeñas del mismo problema (también son métodos directos).

Iterativos - Se generan aproximaciones sucesivas para la solución empezando desde una estimación inicial. En general, el óptimo no se alcanza en una cantidad finita de pasos.

Estocásticos - Algunos algoritmos iterativos no son determinísticos, y su convergencia es estadística.

Escalación Gaussiana (Directo, Reductivo)

Se busca resolver el sistema $Ax = b$, $A \in \mathbb{R}^{n \times n}$, $b \in \mathbb{R}^n$.

Es un método de reducción basado en que las operaciones siguientes no cambian la solución: i) multiplicar una ecuación por un escalar no nulo; ii) intercambiar de posición dos ecuaciones; y iii) sumar a una ecuación un múltiplo de otra.

$$\text{Ej: } \begin{bmatrix} 0 & 2 & 3 \\ 1 & -1 & 1 \\ 1 & 1 & -1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 8 \\ 4 \\ 2 \end{bmatrix} \Rightarrow \left[\begin{array}{ccc|c} 0 & 2 & 3 & 8 \\ 1 & -1 & 1 & 4 \\ 1 & 1 & -1 & 2 \end{array} \right].$$

Escalerización Gaussiana (Directo, Reductivo)

Se busca resolver el sistema $Ax = b$, $A \in \mathbb{R}^{n \times n}$, $b \in \mathbb{R}^n$.

$$\text{Ej: } \begin{bmatrix} 0 & 2 & 3 \\ 1 & -1 & 1 \\ 1 & 1 & -1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 8 \\ 4 \\ 2 \end{bmatrix} \Rightarrow \begin{bmatrix} 0 & 2 & 3 & | & 8 \\ 1 & -1 & 1 & | & 4 \\ 1 & 1 & -1 & | & 2 \end{bmatrix}.$$

Intercambiamos la primera y la tercera fila.

$$\begin{bmatrix} 1 & 1 & -1 & | & 2 \\ 1 & -1 & 1 & | & 4 \\ 0 & 2 & 3 & | & 8 \end{bmatrix}$$

Escalerización Gaussiana (Directo, Reductivo)

Se busca resolver el sistema $Ax = b$, $A \in \mathbb{R}^{n \times n}$, $b \in \mathbb{R}^n$.

$$\text{Ej: } \begin{bmatrix} 0 & 2 & 3 \\ 1 & -1 & 1 \\ 1 & 1 & -1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 8 \\ 4 \\ 2 \end{bmatrix} \Rightarrow \left[\begin{array}{ccc|c} 0 & 2 & 3 & 8 \\ 1 & -1 & 1 & 4 \\ 1 & 1 & -1 & 2 \end{array} \right].$$

Restamos la primera fila a la segunda.

$$\left[\begin{array}{ccc|c} 1 & 1 & -1 & 2 \\ 1 & -1 & 1 & 4 \\ 0 & 2 & 3 & 8 \end{array} \right] \quad \left[\begin{array}{ccc|c} 1 & 1 & -1 & 2 \\ 0 & -2 & 2 & 2 \\ 0 & 2 & 3 & 8 \end{array} \right]$$

Escalerización Gaussiana (Directo, Reductivo)

Se busca resolver el sistema $Ax = b$, $A \in \mathbb{R}^{n \times n}$, $b \in \mathbb{R}^n$.

$$\text{Ej: } \begin{bmatrix} 0 & 2 & 3 \\ 1 & -1 & 1 \\ 1 & 1 & -1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 8 \\ 4 \\ 2 \end{bmatrix} \Rightarrow \begin{bmatrix} 0 & 2 & 3 & | & 8 \\ 1 & -1 & 1 & | & 4 \\ 1 & 1 & -1 & | & 2 \end{bmatrix}.$$

Sumamos la segunda a la tercera.

$$\begin{bmatrix} 1 & 1 & -1 & | & 2 \\ 1 & -1 & 1 & | & 4 \\ 0 & 2 & 3 & | & 8 \end{bmatrix} \begin{bmatrix} 1 & 1 & -1 & | & 2 \\ 0 & -2 & 2 & | & 2 \\ 0 & 2 & 3 & | & 8 \end{bmatrix} \begin{bmatrix} 1 & 1 & -1 & | & 2 \\ 0 & -2 & 2 & | & 2 \\ 0 & 0 & 5 & | & 10 \end{bmatrix}$$

Escalerización Gaussiana (Directo, Reductivo)

Se busca resolver el sistema $Ax = b$, $A \in \mathbb{R}^{n \times n}$, $b \in \mathbb{R}^n$.

$$\text{Ej: } \begin{bmatrix} 0 & 2 & 3 \\ 1 & -1 & 1 \\ 1 & 1 & -1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 8 \\ 4 \\ 2 \end{bmatrix} \Rightarrow \begin{bmatrix} 0 & 2 & 3 & | & 8 \\ 1 & -1 & 1 & | & 4 \\ 1 & 1 & -1 & | & 2 \end{bmatrix}.$$

Dividimos la segunda fila por -2.

$$\begin{bmatrix} 1 & 1 & -1 & | & 2 \\ 1 & -1 & 1 & | & 4 \\ 0 & 2 & 3 & | & 8 \end{bmatrix} \begin{bmatrix} 1 & 1 & -1 & | & 2 \\ 0 & -2 & 2 & | & 2 \\ 0 & 2 & 3 & | & 8 \end{bmatrix} \begin{bmatrix} 1 & 1 & -1 & | & 2 \\ 0 & -2 & 2 & | & 2 \\ 0 & 0 & 5 & | & 10 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 1 & -1 & | & 2 \\ 0 & 1 & -1 & | & -1 \\ 0 & 0 & 5 & | & 10 \end{bmatrix}$$

Escalerización Gaussiana (Directo, Reductivo)

Se busca resolver el sistema $Ax = b$, $A \in \mathbb{R}^{n \times n}$, $b \in \mathbb{R}^n$.

$$\text{Ej: } \begin{bmatrix} 0 & 2 & 3 \\ 1 & -1 & 1 \\ 1 & 1 & -1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 8 \\ 4 \\ 2 \end{bmatrix} \Rightarrow \left[\begin{array}{ccc|c} 0 & 2 & 3 & 8 \\ 1 & -1 & 1 & 4 \\ 1 & 1 & -1 & 2 \end{array} \right].$$

Dividimos la tercera fila por 5.

$$\left[\begin{array}{ccc|c} 1 & 1 & -1 & 2 \\ 1 & -1 & 1 & 4 \\ 0 & 2 & 3 & 8 \end{array} \right] \quad \left[\begin{array}{ccc|c} 1 & 1 & -1 & 2 \\ 0 & -2 & 2 & 2 \\ 0 & 2 & 3 & 8 \end{array} \right] \quad \left[\begin{array}{ccc|c} 1 & 1 & -1 & 2 \\ 0 & -2 & 2 & 2 \\ 0 & 0 & 5 & 10 \end{array} \right]$$

$$\left[\begin{array}{ccc|c} 1 & 1 & -1 & 2 \\ 0 & 1 & -1 & -1 \\ 0 & 0 & 5 & 10 \end{array} \right] \quad \left[\begin{array}{ccc|c} 1 & 1 & -1 & 2 \\ 0 & 1 & -1 & -1 \\ 0 & 0 & 1 & 2 \end{array} \right]$$

Escalación Gaussiana (Directo, Reductivo)

Se busca resolver el sistema $Ax = b$, $A \in \mathbb{R}^{n \times n}$, $b \in \mathbb{R}^n$.

$$\text{Ej: } \begin{bmatrix} 0 & 2 & 3 \\ 1 & -1 & 1 \\ 1 & 1 & -1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 8 \\ 4 \\ 2 \end{bmatrix} \Rightarrow \left[\begin{array}{ccc|c} 0 & 2 & 3 & 8 \\ 1 & -1 & 1 & 4 \\ 1 & 1 & -1 & 2 \end{array} \right].$$

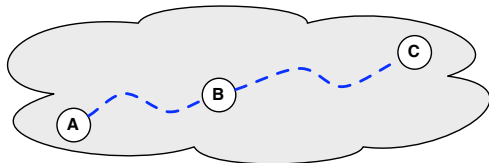
$$\left[\begin{array}{ccc|c} 1 & 1 & -1 & 2 \\ 1 & -1 & 1 & 4 \\ 0 & 2 & 3 & 8 \end{array} \right] \quad \left[\begin{array}{ccc|c} 1 & 1 & -1 & 2 \\ 0 & -2 & 2 & 2 \\ 0 & 2 & 3 & 8 \end{array} \right] \quad \left[\begin{array}{ccc|c} 1 & 1 & -1 & 2 \\ 0 & -2 & 2 & 2 \\ 0 & 0 & 5 & 10 \end{array} \right]$$

$$\left[\begin{array}{ccc|c} 1 & 1 & -1 & 2 \\ 0 & 1 & -1 & -1 \\ 0 & 0 & 5 & 10 \end{array} \right] \quad \left[\begin{array}{ccc|c} 1 & 1 & -1 & 2 \\ 0 & 1 & -1 & -1 \\ 0 & 0 & 1 & 2 \end{array} \right]$$

Ahora podemos despejar: $z = 2$ (última fila), $y = -1 + z = 1$ (segunda fila) y $x = 2 - y + z = 3$ (primera fila).

Programación Dinámica (Directo, Recursivo)

Muchos de los problemas tratables en forma directa cumplen el *Principio de Optimalidad* (Bellman), que dice: “Dada una secuencia óptima de decisiones, toda subsecuencia de ella es a su vez óptima para el subproblema correspondiente”.



En el problema del camino más corto en una red (costos positivos), el Principio de Optimalidad representa que si se conoce el camino más corto entre dos nodos (A y C), y el nodo B pertenece al mismo, no hay un camino más corto para conectar a A y B (o B y C), que el subcamino correspondiente en el original.

Método de Programación Dinámica (Bellman 1940-1953)

- Es un *método recursivo* para resolver problemas complejos mediante su descomposición en problemas más simples
- Durante la recursión se almacenan los resultados encontrados al momento, y luego se usan en los problemas más complejos
- Deben existir “pasos base”, instancias de tamaño reducido para las que la solución es trivial o conocida
- Se usa en problemas de: matemáticas, management, economía, computer science, machine learning y bioinformática
- Requiere una estructura particular del problema, con un conjunto discreto de etapas y estados en cada etapa
- El problema debe cumplir el Principio de Optimalidad entre sus etapas, lo que permite un planteo *greedy*
- Debe ser Markoviano cuando es estocástico

Método de Programación Dinámica (ejemplo en juegos)

Suponga una mesa con 30 cerillos. En cada turno, un jugador puede tomar 1, 2 o 3 cerillos. Se continúa hasta que un jugador levante el último, en cuyo caso pierde ¿Cómo jugar para ganar?

Método de Programación Dinámica (ejemplo en juegos)

Suponga una mesa con 30 cerillos. En cada turno, un jugador puede tomar 1, 2 o 3 cerillos. Se continúa hasta que un jugador levante el último, en cuyo caso pierde ¿Cómo jugar para ganar?

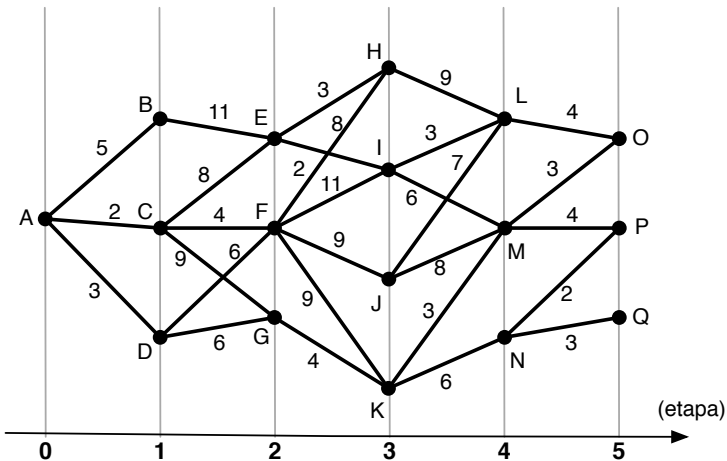
Hay que forzar al otro a llegar a una mano con un cerillo. Gano si llego a una mano con $\{2,3,4\}$ cerillos. La anterior entonces debe ser de 5 para el oponente. Puedo forzar esto (y ganar) si llego a una mano con $\{6,7,8\}$ cerillos. Su mano anterior entonces debe ser de 9. Puedo forzar esa mano si llego a una con $\{10,11,12\}$.

La secuencia (él,yo) sigue así: $(13,\{14,15,16\})$, $(17,\{18,19,20\})$, $(21,\{22,23,24\})$, $(25,\{26,27,28\})$. El conjunto prohibido es $\{1,5,9,13,17,21,25,29\}$.

¡¡Gana siempre el que saca un solo cerillo al empezar!!

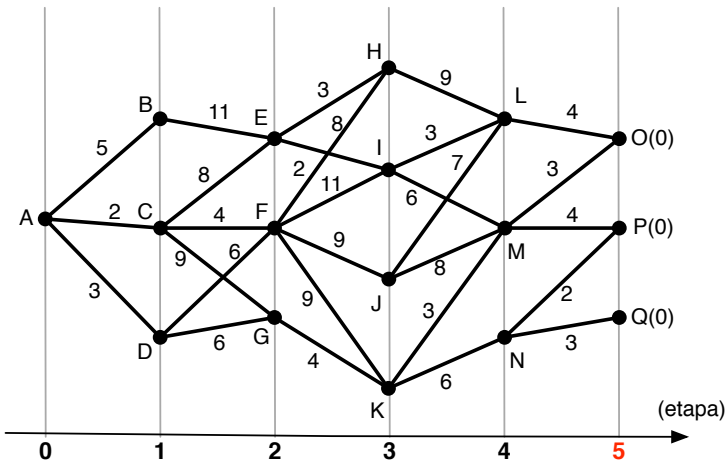
Método de Programación Dinámica (en optimización)

¿Como recorrer las etapas para llegar desde A hasta O, P, o Q (indistintamente) acumulando la menor distancia posible?



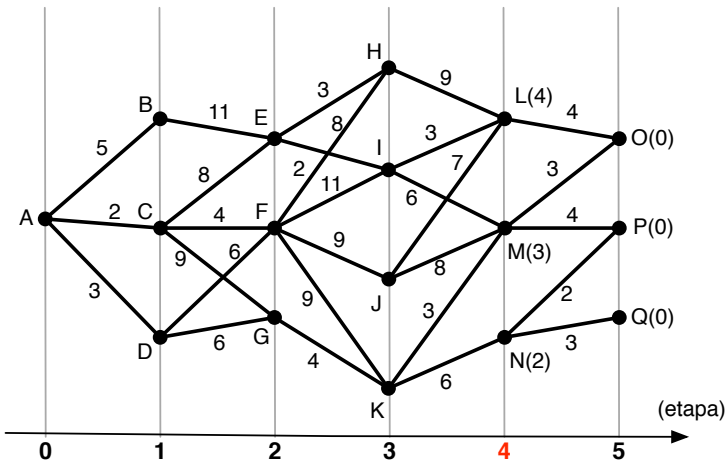
Método de Programación Dinámica (en optimización)

¿Como recorrer las etapas para llegar desde A hasta O, P, o Q (indistintamente) acumulando la menor distancia posible?



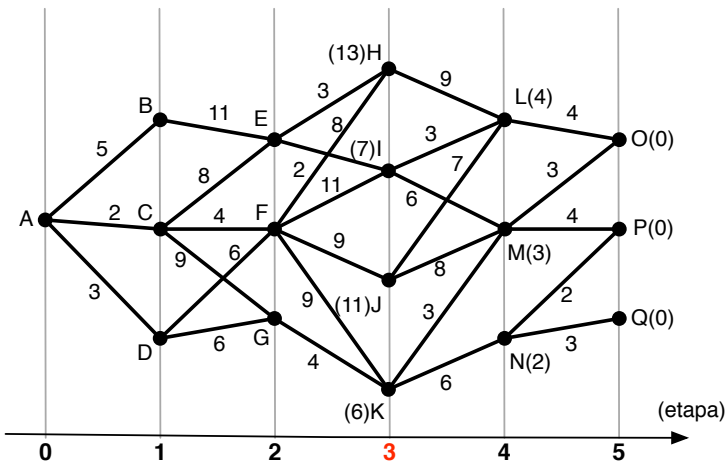
Método de Programación Dinámica (en optimización)

¿Como recorrer las etapas para llegar desde A hasta O, P, o Q (indistintamente) acumulando la menor distancia posible?



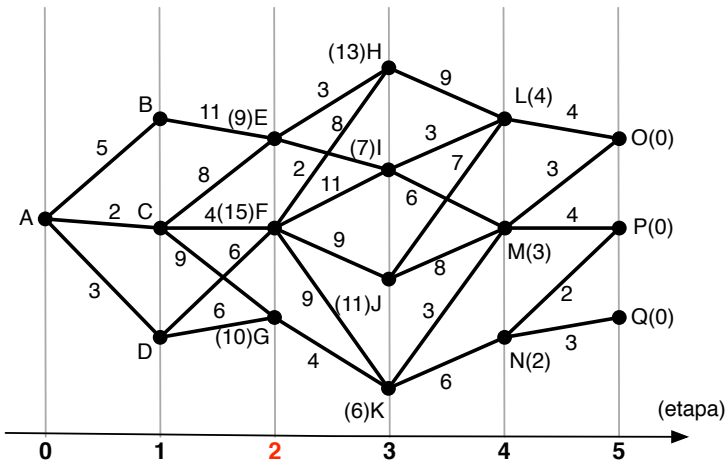
Método de Programación Dinámica (en optimización)

¿Como recorrer las etapas para llegar desde A hasta O, P, o Q (indistintamente) acumulando la menor distancia posible?



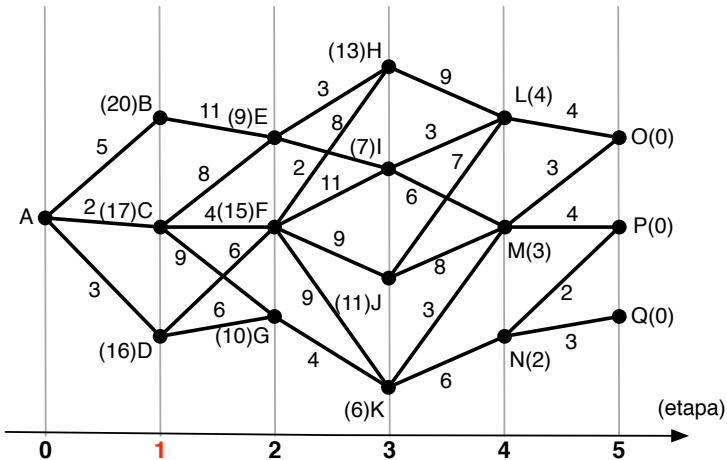
Método de Programación Dinámica (en optimización)

¿Como recorrer las etapas para llegar desde A hasta O, P, o Q (indistintamente) acumulando la menor distancia posible?



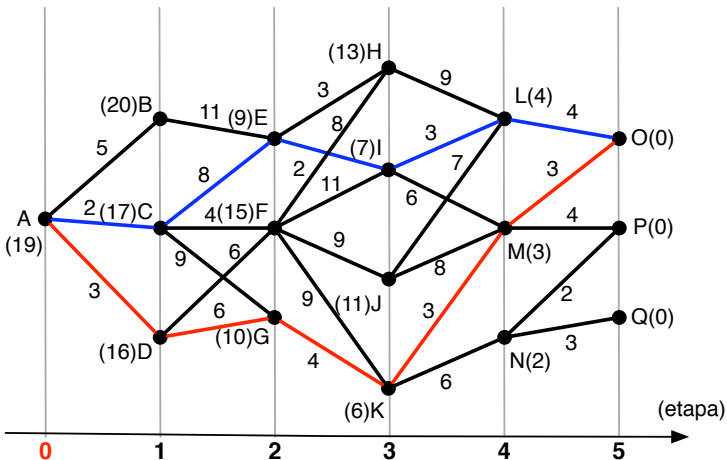
Método de Programación Dinámica (en optimización)

¿Como recorrer las etapas para llegar desde A hasta O, P, o Q (indistintamente) acumulando la menor distancia posible?



Método de Programación Dinámica (en optimización)

¿Como recorrer las etapas para llegar desde A hasta O, P, o Q (indistintamente) acumulando la menor distancia posible?



Algoritmo de Programación Dinámica

Suponemos que el problema consta de T etapas $(1, \dots, T)$, que en cada etapa hay n estados $(1, \dots, n$, los mismos por simplicidad) y que conocemos la función de costo final $f(i, T)$, con $i = 1, \dots, n$.

Se conoce por último la función de costo $c(i, j, t + 1)$, con el costo incurrido por pasar del estado i en t , al estado j en $t + 1$.

La recursión queda: $f(i, t) = \min_{1 \leq j \leq n} \{c(i, j, t + 1) + f(j, t + 1)\}$.
 El valor óptimo es $f(i, 1)$ para el(los) estado(s) inicial(es).

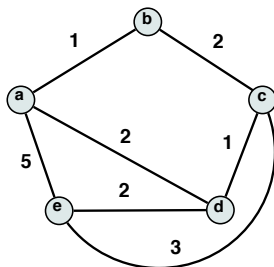
Es sumamente eficiente (usa a lo sumo Tn^2 cálculos). Constituye una de las pocas excepciones de interés dentro de los problemas combinatorios, en las que hay algoritmos eficientes para resolverlo.

Enriqueciendo el número de estados se pueden modelar problemas más complejos (a costa de la eficiencia). Un caso importante es el de la Programación Dinámica Estocástica.

Contraejemplo a la Programación Dinámica

La programación dinámica carece de generalidad ante un problema de optimización cualquiera. Se basa en una recursión greedy que requiere etapas, estados y cumplir además con el *principio de optimalidad* entre esas etapas

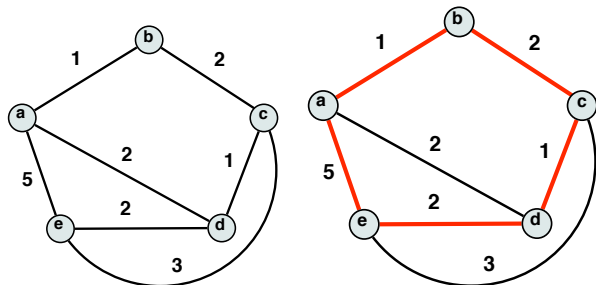
Los problemas *computacionalmente duros* no aceptan soluciones greedy. Uno de los ejemplos más clásicos es el TSP:



Contraejemplo a la Programación Dinámica

La programación dinámica carece de generalidad ante un problema de optimización cualquiera. Se basa en una recursión greedy que requiere etapas, estados y cumplir además con el *principio de optimalidad* entre esas etapas

Los problemas *computacionalmente duros* no aceptan soluciones greedy. Uno de los ejemplos más clásicos es el TSP:

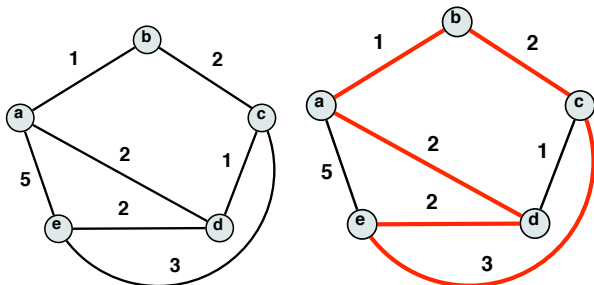


Solución greedy de costo 11.

Contraejemplo a la Programación Dinámica

La programación dinámica carece de generalidad ante un problema de optimización cualquiera. Se basa en una recursión greedy que requiere etapas, estados y cumplir además con el *principio de optimalidad* entre esas etapas

Los problemas *computacionalmente duros* no aceptan soluciones greedy. Uno de los ejemplos más clásicos es el TSP:



Solución óptima de costo 10.

Método de Bipartición (Iterativo)

Un método iterativo popular para encontrar raíces de funciones (o extremos relativos) es el de bipartición o bisección.

Dada una función continua en un intervalo cerrado, $f[a, b] \rightarrow \mathbb{R}$, con $f(a) \cdot f(b) < 0$, existe $c \in (a, b)$ tal que $f(c) = 0$ (TM Bolzano).

El método de bipartición consiste en generar tres sucesiones $\{a_n\}$, $\{b_n\}$, $\{r_n\}$, definidas por las siguientes reglas:

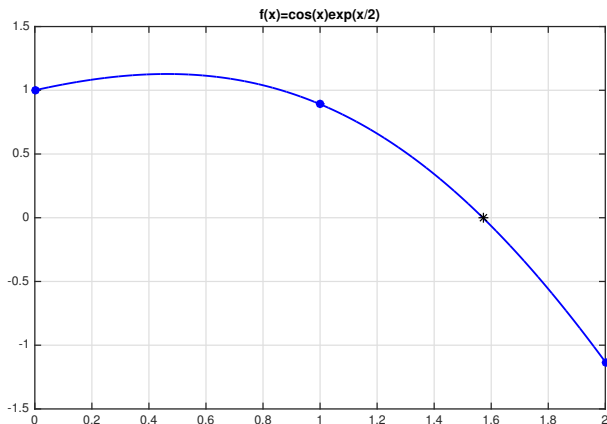
$$1 \quad a_0 = a, a_{n+1} = \begin{cases} a_n, & \text{si } f(a_n) \cdot f(r_n) < 0 \\ r_n, & \text{si } f(a_n) \cdot f(r_n) > 0 \end{cases}$$

$$2 \quad b_0 = b, b_{n+1} = \begin{cases} r_n, & \text{si } f(a_n) \cdot f(r_n) < 0 \\ b_n, & \text{si } f(a_n) \cdot f(r_n) > 0 \end{cases}$$

$$3 \quad r_n = \frac{a_n + b_n}{2}$$

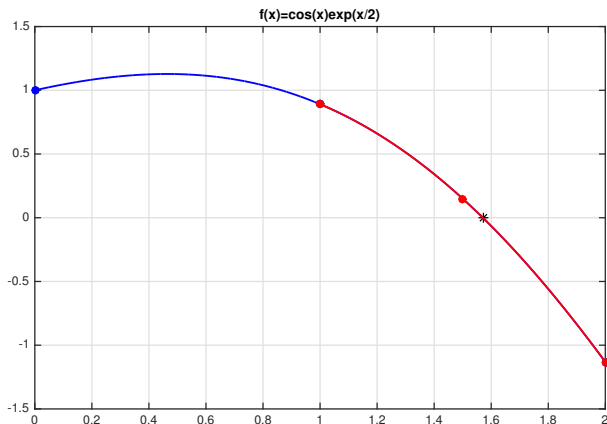
4 Salir cuando $f(r_n) = 0$ o se alcance cierto error meta

Método de Bipartición (Iterativo)



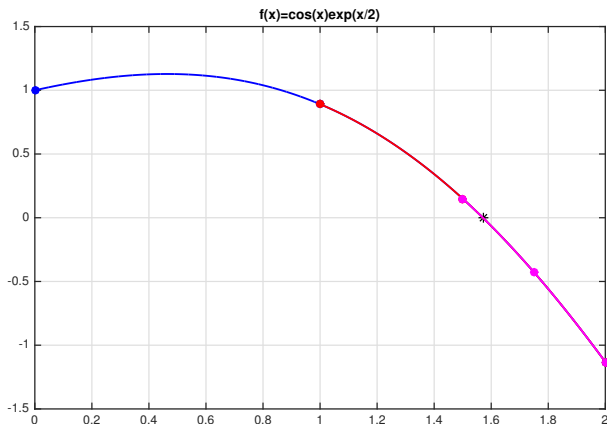
En la función $f(x) = \cos(x)e^{x/2}$ (la raíz es $\pi/2$). Tomando $a_0 = 0$ y $b_0 = 2$, queda $r_0 = \frac{0+2}{2} = 1$.

Método de Bipartición (Iterativo)



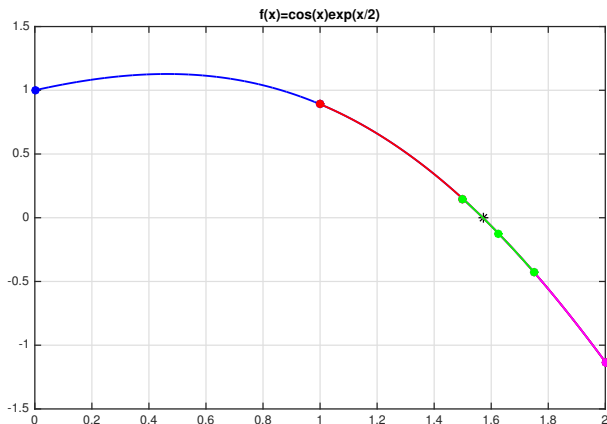
En la siguiente iteración: $a_1 = 1$, $b_1 = 2$ y $r_1 = \frac{3}{2}$.

Método de Bipartición (Iterativo)



En la segunda: $a_2 = \frac{3}{2}$, $b_2 = 2$ y $r_2 = \frac{7}{4}$.

Método de Bipartición (Iterativo)



En la segunda: $a_3 = \frac{3}{2}$, $b_3 = \frac{7}{4}$ y $r_3 = \frac{13}{8}$.

Método de Bipartición (Iterativo)

Observar que $a_n \leq a_{n+1} < r_{n+1} < b_{n+1} \leq b_n$. Se prueba por inducción: (paso base) $a_0 < r_0 < b_0$, (inductivo) si $a_n < r_n < b_n$:

- $a_{n+1} = a_n$ o $a_{n+1} = r_n$, luego $a_{n+1} \geq a_n$ ($\{a_n\}$ creciente).
- $b_{n+1} = b_n$ o $b_{n+1} = r_n$, luego $b_{n+1} \leq b_n$ ($\{b_n\}$ decreciente).
- Los cambios en $\{a_n\}$ y $\{b_n\}$ son excluyentes (si aumenta a_n no disminuye b_n), entonces $a_{n+1} < b_{n+1}$.
- Finalmente, $a_{n+1} < r_{n+1} < b_{n+1}$, porque r_{n+1} es la semisuma de a_{n+1} y b_{n+1} , y se verificaba que $a_{n+1} < b_{n+1}$.
- De existir r_n con $f(r_n) = 0$, al método para, si no, pasa a $n + 1$.

Como la sucesión $\{a_n\}$ es monótona creciente y acotada superiormente por b_0 , entonces tiene límite: $a_n \rightarrow \bar{a}$.

Como la sucesión $\{b_n\}$ es monótona decreciente y acotada inferiormente por a_0 , entonces tiene límite: $b_n \rightarrow \bar{b}$.

Método de Bipartición (Iterativo)

Propiedad

Dada $f[a, b] \rightarrow \mathbb{R}$ continua, con $f(a) \cdot f(b) < 0$, el método de bipartición converge a: $r \in (a, b)$ tal que $f(r) = 0$.

Prueba: Si existe n tal que $f(r_n) = 0$, tomo $r = r_n$ (se cumple).

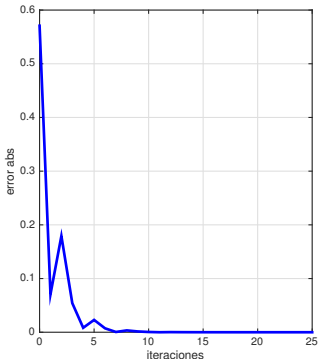
Si no, por construcción $f(a_n) \cdot f(b_n) < 0$, y por el TM de Bolzano existe $c_n \in (a_n, b_n)$ tal que $f(c_n) = 0$.

También por construcción, $b_n - a_n = \frac{b-a}{2^n}$, y $a_n < r_n < b_n$. En caso de nunca parar: $a_n < \bar{a} \leq \bar{b} < b_n$, pero $b_n - a_n \rightarrow 0$, entonces $a_n \rightarrow \bar{a}$ y $b_n \rightarrow \bar{b}$ con $\bar{a} = \bar{b}$, de donde $r_n \rightarrow \bar{r} = \bar{a} = \bar{b}$.

Además, como $c_n \in (a_n, b_n)$, $c_n \rightarrow \bar{c} = \bar{r}$, con $f(c_n) = 0$ para todo n .

Por la continuidad de f , se cumple $f(r_n) \rightarrow f(\bar{r}) = f(\bar{c}) = 0$, y se prueba que el método converge a una raíz.

Método de Bipartición (Convergencia)



El error del método de bipartición converge a 0 exponencialmente.

El logaritmo del error aproxima una función lineal.

El método es muy robusto, pero no escala en las variables, ni es el de convergencia más rápida.

Método de Jacobi (Iterativo)

Se resuelve el sistema $Ax = b$, construyendo una sucesión $\{x^{(k)}\}$. Tomo $D = \text{diag}(A)$ y $M = A - D$. Se cumple que si la sucesión definida por $Dx^{(k+1)} + Mx^{(k)} = b$ converge (i.e. $x^{(k)} \rightarrow \bar{x}$), debe hacerlo a la solución del sistema, ya que $D\bar{x} + (A - D)\bar{x} = b$.

$$\text{Ej: } A = \begin{bmatrix} 4 & 1 & 0 \\ 0 & 3 & 2 \\ 1 & 1 & 5 \end{bmatrix} \text{ y } b = \begin{bmatrix} 5 \\ 5 \\ 7 \end{bmatrix} \Rightarrow M = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 2 \\ 1 & 1 & 0 \end{bmatrix} \text{ y se}$$

cumple $x^{(k+1)} = D^{-1}(b - Mx^{(k)})$, esto es: $x_i^{(k+1)} = \frac{1}{a_{ii}}(b - Mx^{(k)})$.

Método de Jacobi (Iterativo)

Se resuelve el sistema $Ax = b$, construyendo una sucesión $\{x^{(k)}\}$. Tomo $D = \text{diag}(A)$ y $M = A - D$. Se cumple que si la sucesión definida por $Dx^{(k+1)} + Mx^{(k)} = b$ converge (i.e. $x^{(k)} \rightarrow \bar{x}$), debe hacerlo a la solución del sistema, ya que $D\bar{x} + (A - D)\bar{x} = b$.

$$\text{Ej: } A = \begin{bmatrix} 4 & 1 & 0 \\ 0 & 3 & 2 \\ 1 & 1 & 5 \end{bmatrix} \text{ y } b = \begin{bmatrix} 5 \\ 5 \\ 7 \end{bmatrix} \Rightarrow M = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 2 \\ 1 & 1 & 0 \end{bmatrix} \text{ y se}$$

cumple $x^{(k+1)} = D^{-1}(b - Mx^{(k)})$, esto es: $x_i^{(k+1)} = \frac{1}{a_{ii}}(b - Mx^{(k)})$.

$$x^{(0)} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, x^{(1)} = \begin{bmatrix} 1.250 \\ 1.667 \\ 1.400 \end{bmatrix}, x^{(2)} = \begin{bmatrix} 0.833 \\ 0.733 \\ 0.817 \end{bmatrix}, x^{(3)} = \begin{bmatrix} 1.067 \\ 1.122 \\ 1.087 \end{bmatrix}.$$

La solución es $\bar{x} = [1, 1, 1]$, así que en 3 iteraciones alcanzamos un 9.5% de error.

¿Por qué usar Métodos Iterativos en Problemas Lineales?

En varios problemas (especialmente en los de gran tamaño, muchos datos), la mayoría de los elementos de $A \in \mathbb{R}^{n \times n}$ son 0 (es *sparse*).

El método de Gauss termina *poblando* la matriz con muchos elementos no-nulos, lo que consume mucha más memoria. Si al inicio, A tenía 99.9% de ceros, es probable que luego de la diagonalización, el valor se aproxime a 50%.

Si $A \in \mathbb{R}^{n \times n}$, el número de operaciones con Gauss aproxima $\frac{n^3}{3}$. En una matriz con 99.9% de ceros, cada iteración de Jacobi requeriría $\frac{n^2}{1000}$ operaciones, un buen iterativo alcanza precisión razonable en pocas iteraciones, y muchas veces se pueden paralelizar.

Además, Gauss es más sensible a los errores numéricos, así que tampoco es exacto en problemas grandes.

En este curso buscamos escalabilidad en los algoritmos y nos centraremos en los Métodos Iterativos (excepción Programación Dinámica). No se ven Heurísticas.

Conceptos de Cadenas de Markov

Definición

*Un Proceso Estocástico es una familia $X = \{X_t\}_{t \in T}$ de variables aleatorias, donde el conjunto $T \neq \emptyset$ se denomina **parámetro tiempo**. Denominamos **espacio de estados** a $E = \bigcup_{t \in T} R_{X_t}$, esto es, al mínimo conjunto de todos los valores que las vv.aa. pueden tomar a lo largo de todos los instantes. El valor $e \in E$ representa un estado posible del sistema en algún instante, y notamos con $(X_t = e)$ al evento que indica que en el instante t el proceso se encuentra en el estado e .*

Una clasificación usual de los procesos estocásticos (PE) es según la continuidad de sus espacios.

Las *cadenas* son los PE de espacio de estados discreto.

Si además el parámetro tiempo es discreto, decimos que el PE es una Cadena de Tiempo Discreto.

Conceptos de Cadenas de Markov

Entre los valores fundamentales que se buscan para una Cadena de Tiempo Discreto está el calcular la probabilidad de una evolución:

$$P(X_n=e_n, X_{n-1}=e_{n-1}, X_{n-2}=e_{n-2}, \dots, X_1=e_1, X_0=e_0),$$

esto es, que la cadena esté en e_0 en $t = 0$, y en e_1 en $t = 1$, etc.

Para calcular lo anterior, el insumo son las probabilidades condicionales: $P(X_i = e_i | X_{i-1} = e_{i-1}, \dots, X_0 = e_0)$.

Definición

Una Cadena de Tiempo Discreto es de Markov (CMTD) si y sólo si $P(X_n=e_n | X_{n-1}=e_{n-1}, X_{n-2}=e_{n-2}, \dots, X_1=e_1, X_0=e_0) = P(X_n=e_n | X_{n-1}=e_{n-1}) = p_{e_{n-1}, e_n}^n$, siendo la última la probabilidad de transición de e_{n-1} a e_n en el instante $n - 1$.

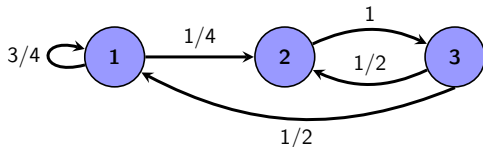
Conceptos de Cadenas de Markov

Definición

Una CMTD es Homogénea (CMTDH) si y sólo si las probabilidades de transición no dependen del tiempo, es esto, si se cumple que:
 $P(X_{n+1} = j | X_n = i) = p_{ij}$ (fijo), $\forall i, j \in E, \forall n \geq 0$.

Las CMTDH quedan determinadas por su matriz de transición P o (equivalentemente) por el grafo asociado a la cadena.

$$P = \begin{bmatrix} 3/4 & 1/4 & 0 \\ 0 & 0 & 1 \\ 1/2 & 1/2 & 0 \end{bmatrix}$$

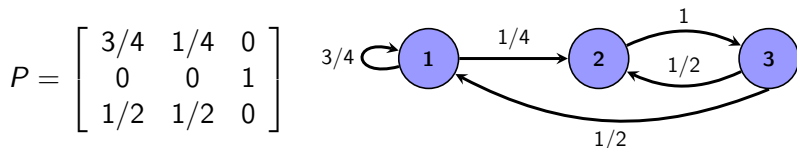


Conceptos de Cadenas de Markov

Definición

Decimos que un estado i en una CMTDH es Recurrente cuando partiendo del estado hay certeza de volver al mismo.

Decimos que es Transitorio cuando hay probabilidad no-nula de abandonarlo para siempre.



En este ejemplo, todos los estados son recurrentes.

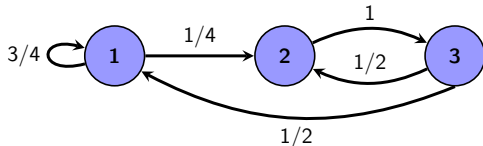
Conceptos de Cadenas de Markov

Definición

Decimos que dos estados i y j en una CMTDH se Comunican cuando hay caminos para ir y volver entre ellos.

Decimos que una cadena es Irreducible cuando todos sus estados se comunican en sí.

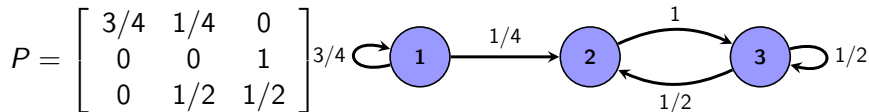
$$P = \begin{bmatrix} 3/4 & 1/4 & 0 \\ 0 & 0 & 1 \\ 1/2 & 1/2 & 0 \end{bmatrix}$$



En este ejemplo, la cadena es irreducible y todos sus estados son recurrentes.

Conceptos de Cadenas de Markov

En este ejemplo, hay dos componentes en el cadena.



El estado $\{1\}$ es transitorio, mientras que $\{2, 3\}$ son recurrentes.

Propiedad

Si dos estados se comunican, ambos comparten la condición de recurrencia o de transitoriedad.

Toda CMTDH finita tiene al menos un estado recurrente.

Esta propiedad permite clasificar rápidamente la cadena de la transparencia anterior, ya que un cadena finita irreducible tiene que estar compuesta exclusivamente de estados recurrentes.

Conceptos de Cadenas de Markov

Definición

Llamamos vector de distribución de estados en el instante n o $Q(n)$, al vector estocástico que contiene las probabilidades de estar en cualquier estado j en la n -ésima etapa. Siendo $E = \{1, 2, \dots, r\}$ el espacio de estado y $\pi_j(n) = P(X_n = j)$, se cumple:
 $Q(n) = [\pi_1(n), \dots, \pi_r(n)]$, con $\sum_{j=1}^r \pi_j(n) = 1$.

Propiedad

Dado $Q(0)$ cualquiera para una CMTDH de matriz P , la propiedad de Chapman-Kolmogorov permite calcular $Q(n) = Q(0)P^n$, o equivalentemente $Q(n+1) = Q(n)P$.

Conceptos de Cadenas de Markov

Definición

Dada una CMTDH con espacio de estados E , y un estado $i \in E$ cualquiera, se define $d(i)$: “período del estado i ” como $d(i) = \text{MCD}\{n : p_{ii}^{(n)} > 0, n \geq 0\}$, esto es, el Máximo Común Divisor del conjunto “ n ” de pasos que llevan el proceso del estado i a sí mismo. Decimos que un estado i es aperiódico si $d(i) = 1$, y que es periódico en caso contrario.

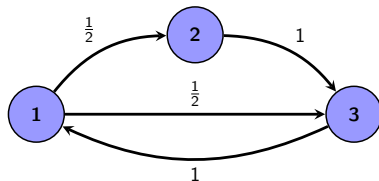
Propiedad

En toda CMTDH finita, irreducible y aperiódica (i.e. Ergódica), se cumple que existe el límite $\lim_n Q(n) = \pi$, independientemente de la distribución inicial, y ese valor verifica $\pi P = \pi$, lo que se denomina Distribución Estacionaria, que en este caso es única.

Motores de Búsqueda

Imaginemos que Internet solamente constara de tres páginas y los usuarios las recorrieran siguiendo (al azar) los links de una a otra. Entonces, las probabilidades de transición serían las siguientes:

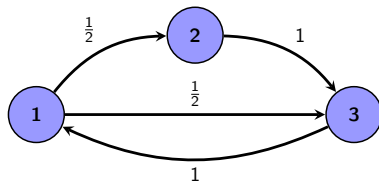
$$P = \begin{bmatrix} 0 & 1/2 & 1/2 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}$$



Por resultados de Cadenas de Markov, la fracción del tiempo que un *random walker* estaría en cada página es π , la solución estacionaria: $\pi \cdot P = \pi$, con $\pi_1 + \pi_2 + \pi_3 = 1$, que vale $[2/5, 1/5, 2/5]$. En ese *ranking*, las páginas 1 y 3 están primeras, y luego está 2.

Motores de Búsqueda

$$P = \begin{bmatrix} 0 & 1/2 & 1/2 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}$$



Partiendo de $\pi(0) = [1, 0, 0]$ y usando $\pi(n+1) = \pi(n)P$, resulta:
 $\pi(1) = [0, 0.5, 0.5]$, $\pi(2) = [0.5, 0, 0.5]$, $\pi(3) = [0.5, 0.25, 0.25]$,
 $\pi(4) = [0.25, 0.25, 0.5]$, $\pi(5) = [0.5, 0.1250, 0.3750]$,
 $\pi(6) = [0.375, 0.25, 0.375]$, $\pi(7) = [0.375, 0.1875, 0.4375]$,
 $\pi(8) = [0.4375, 0.1875, 0.375]$, $\pi(9) = [0.375, 0.2188, 0.4062]$,
 $\pi(10) = [0.4062, 0.1875, 0.4062]$.

En tan solo 10 iteraciones: $\frac{\|\pi(10) - \pi\|}{\|\pi\|} \approx 2.55\%$.

El PageRank de Google

Supongamos ahora que tenemos los datos de las N páginas de Internet, y generamos la matriz Q de hyperlinks:

$$Q_{ij} = \begin{cases} 1/k_i, & \text{si hay un link de } i \text{ a } j \\ 0, & \text{en otro caso} \end{cases},$$

donde k_i es la cantidad de links que salen de la página i hacia otras. El valor de N es varios miles de millones ¿Cómo podemos resolver $\pi \cdot Q = \pi$? ¿Debe existir solución y ser única?

El PageRank propone usar: $P = \alpha \cdot Q + \frac{1-\alpha}{N} \times \mathbb{1}_{(N \times N)}$, donde α cumple $0 < \alpha < 1$ (es usual $\alpha = 0.85$) y la iteración $\pi_{n+1} = \pi_n \cdot P$, para algún π_0 , como $\pi_0 = [1/N, \dots, 1/N]$.

Así, la matriz P asegura que la cadena es ergódica, con lo que la solución π es única y coincide además con el límite de π_n .

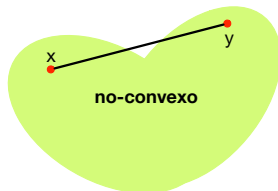
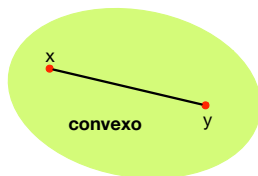
Eficiencia del PageRank

- Usemos como referencia que $N = 18 \cdot 10^9$ (tres páginas por persona), y que en la matriz Q se usan el tipo de datos `long double` (16 bytes) para los flotantes.
- La matriz completa requería casi 4500 Exabytes ($1\text{EB}=1024^6$).
- Pero si una página tuviera en promedio 1000 links, bastaría menos de un Petabyte ($1\text{PB}=1024^5=1024\text{TB}$).
- La escalerización requeriría 18 millones de veces más memoria, y de iterar 10 veces, necesitaría 600 millones de veces más operaciones, que no son paralelizables.
- Además, siendo ínfima la fracción de cambios en el conjunto de Internet, otra problema en la escalerización es no poder usar la solución del día anterior como partida.

Conjuntos Convexos

Definición

Un conjunto $S \subseteq \mathbb{R}^n$ es convexo, si y sólo si, para todo $(x, y) \in S$ y $0 \leq \alpha \leq 1$, se cumple que $[\alpha x + (1 - \alpha)y] \in S$. Bajo esas hipótesis, llamamos combinación convexa a $\alpha x + (1 - \alpha)y$.



OBS: El conjunto vacío es convexo.

Conjuntos Convexos

Definición

Un conjunto $S \subseteq \mathbb{R}^n$ es convexo, si y sólo si, para todo $(x, y) \in S$ y $0 \leq \alpha \leq 1$, se cumple que $[\alpha x + (1 - \alpha)y] \in S$. Bajo esas hipótesis, llamamos combinación convexa a $\alpha x + (1 - \alpha)y$.

PageRank se vale del hecho que tanto Q como $\frac{1}{N} \mathbb{1}_{(N \times N)}$ son elementos del espacio de matrices estocásticas de $N \times N$, y que ese espacio es convexo, así que P debe ser estocástica ($0 < \alpha < 1$).

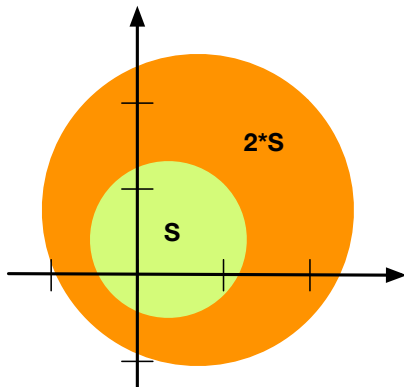
Además, la matriz $\frac{1}{N} \mathbb{1}_{(N \times N)}$ es ergódica, y como $1 - \alpha > 0$, el resultado P también debe serlo.

El ajuste de α es experimental. Cuando más alto α , más parecidas las probabilidad a las de Q , pero más lenta será la convergencia.

Conjuntos Convexos (propiedades)

Propiedad

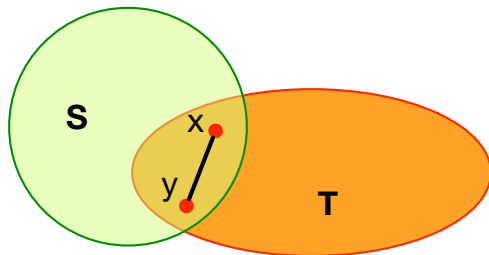
Dados cualquier $\alpha \in \mathbb{R}$ y $S \subseteq \mathbb{R}^n$ convexo, se cumple que el conjunto $\alpha S = \{\alpha x \mid x \in S\}$ es convexo.



Conjuntos Convexos (propiedades)

Propiedad

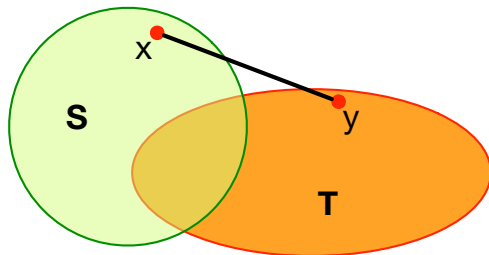
Dados $S \subseteq \mathbb{R}^n$ y $T \subseteq \mathbb{R}^n$ convexos cualesquiera, se cumple que la intersección $S \cap T = \{x \mid x \in S, x \in T\}$ es un conjunto convexo.



Conjuntos Convexos (propiedades)

Propiedad

Dados $S \subseteq \mathbb{R}^n$ y $T \subseteq \mathbb{R}^n$ convexos cualesquiera, se cumple que la intersección $S \cap T = \{x \mid x \in S, x \in T\}$ es un conjunto convexo.

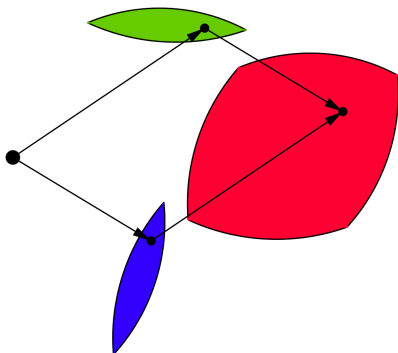


OBS: En general, la unión de convexos no es convexa.

Conjuntos Convexos (propiedades)

Propiedad

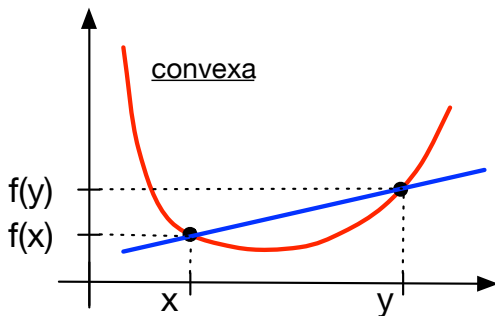
Dados $S \subseteq \mathbb{R}^n$ y $T \subseteq \mathbb{R}^n$ convexos cualesquiera, se cumple que el conjunto suma $S + T = \{x + y \mid x \in S, x \in T\}$ es convexo.



Funciones Convexas

Definición

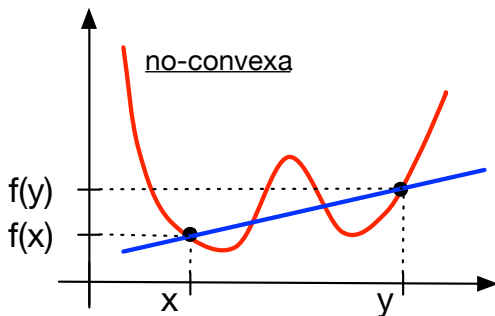
Sea $S \subseteq \mathbb{R}^n$ un conjunto convexo. Decimos que la función $f : S \rightarrow \mathbb{R}$ es convexa, si y sólo si, para todo $x, y \in S$ y $0 \leq \alpha \leq 1$, se cumple que: $f(\alpha x + (1 - \alpha)y) \leq \alpha f(x) + (1 - \alpha)f(y)$.



Funciones Convexas

Definición

Sea $S \subseteq \mathbb{R}^n$ un conjunto convexo. Decimos que la función $f : S \rightarrow \mathbb{R}$ es convexa, si y sólo si, para todo $x, y \in S$ y $0 \leq \alpha \leq 1$, se cumple que: $f(\alpha x + (1 - \alpha)y) \leq \alpha f(x) + (1 - \alpha)f(y)$.



Funciones Convexas (propiedades)

Propiedades

Sean $f : S \rightarrow \mathbb{R}$ y $g : S \rightarrow \mathbb{R}$, funciones convexas con dominio $S \subseteq \mathbb{R}^n$ convexo, y $\alpha \geq 0$. Entonces, se cumple que:

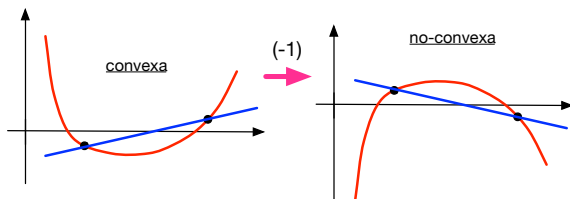
- La función $h_1(x) = f(x) + g(x)$ es convexa
- La función $h_2(x) = \alpha \cdot f(x)$ es convexa
- La función $h_3(x) = \max\{f(x), g(x)\}$ es convexa

Funciones Convexas (propiedades)

Propiedades

Sean $f : S \rightarrow \mathbb{R}$ y $g : S \rightarrow \mathbb{R}$, funciones convexas con dominio $S \subseteq \mathbb{R}^n$ convexo, y $\alpha \geq 0$. Entonces, se cumple que:

- La función $h_1(x) = f(x) + g(x)$ es convexa
- La función $h_2(x) = \alpha \cdot f(x)$ es convexa
- La función $h_3(x) = \max\{f(x), g(x)\}$ es convexa



Debe ser $\alpha \geq 0$.

Funciones Convexas (propiedades)

Propiedades

Sean $f : S \rightarrow \mathbb{R}$ y $g : S \rightarrow \mathbb{R}$, funciones convexas con dominio $S \subseteq \mathbb{R}^n$ convexo, y $\alpha \geq 0$. Entonces, se cumple que:

- La función $h_1(x) = f(x) + g(x)$ es convexa
- La función $h_2(x) = \alpha \cdot f(x)$ es convexa
- La función $h_3(x) = \max\{f(x), g(x)\}$ es convexa

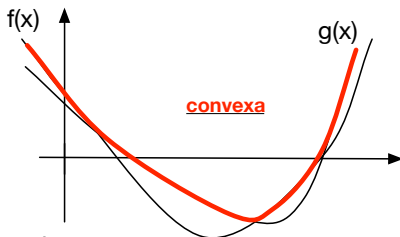
Por inducción, puede probarse que cualquier combinación lineal de funciones convexas con multiplicadores positivos también es convexa.

Funciones Convexas (propiedades)

Propiedades

Sean $f : S \rightarrow \mathbb{R}$ y $g : S \rightarrow \mathbb{R}$, funciones convexas con dominio $S \subseteq \mathbb{R}^n$ convexo, y $\alpha \geq 0$. Entonces, se cumple que:

- La función $h_1(x) = f(x) + g(x)$ es convexa
- La función $h_2(x) = \alpha \cdot f(x)$ es convexa
- La función $h_3(x) = \max\{f(x), g(x)\}$ es convexa



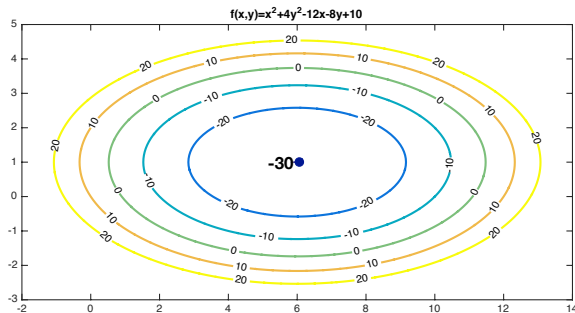
No se cumple para el mínimo.

Funciones Convexas (propiedades)

Propiedades

Sean $S \subseteq \mathbb{R}^n$ convexo, $f : S \rightarrow \mathbb{R}$ convexa y $\alpha \in \mathbb{R}$ cualquiera. Entonces, se cumple que:

- El conjunto $\{x | f(x) \leq \alpha\}$ es convexo.
- El epigrafo $\{(x, \mu) | f(x) \leq \mu\}$ es convexo.

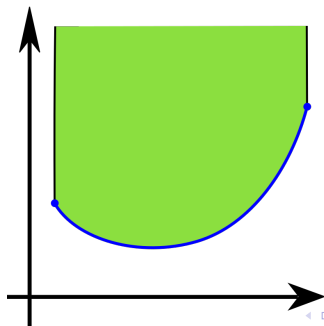


Funciones Convexas (propiedades)

Propiedades

Sean $S \subseteq \mathbb{R}^n$ convexo, $f : S \rightarrow \mathbb{R}$ convexa y $\alpha \in \mathbb{R}$ cualquiera.
Entonces, se cumple que:

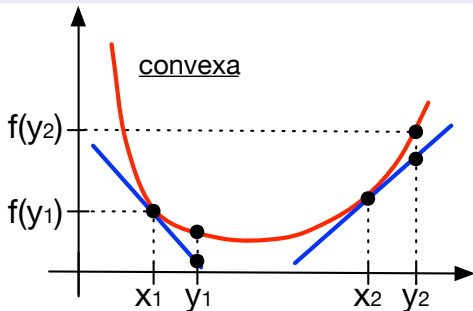
- El conjunto $\{x | f(x) \leq \alpha\}$ es convexo.
- El epigrafo $\{(x, \mu) | f(x) \leq \mu\}$ es convexo.



Identificando Funciones Convexas en \mathbb{R}

Propiedades

- *Los conjuntos convexos en \mathbb{R} son los intervalos, ya sean éstos abiertos o cerrados.*
- *Dado un intervalo $I \subseteq \mathbb{R}$, una función $f : I \rightarrow \mathbb{R}$ diferenciable es convexa sii $f(y) \geq f(x) + f'(x)(y - x)$, para todos $x, y \in I$.*



Identificando Funciones Convexas en \mathbb{R}

Propiedades

- *Los conjuntos convexos en \mathbb{R} son los intervalos, ya sean éstos abiertos o cerrados.*
- *Dado un intervalo $I \subseteq \mathbb{R}$, una función $f : I \rightarrow \mathbb{R}$ diferenciable es convexa sii $f(y) \geq f(x) + f'(x)(y - x)$, para todos $x, y \in I$.*
- *Una función diferenciable $f : I \rightarrow \mathbb{R}$ en un intervalo $I \subseteq \mathbb{R}$ es convexa si y sólo si f' es no-decreciente.*
- *Si además es diferenciable a segundo orden, f es convexa si y sólo si $f''(x) \geq 0$ para todo $x \in I$.*

OBS: Las funciones de la forma $f = (ax - b)^2$ son convexas, ya que $f'(x) = 2a(ax - b)$ y $f''(x) = 2a^2 \geq 0$.

Identificando Funciones Convexas en \mathbb{R}

Propiedades

- *Los conjuntos convexos en \mathbb{R} son los intervalos, ya sean éstos abiertos o cerrados.*
- *Dado un intervalo $I \subseteq \mathbb{R}$, una función $f : I \rightarrow \mathbb{R}$ diferenciable es convexa sii $f(y) \geq f(x) + f'(x)(y - x)$, para todos $x, y \in I$.*
- *Una función diferenciable $f : I \rightarrow \mathbb{R}$ en un intervalo $I \subseteq \mathbb{R}$ es convexa si y sólo si f' es no-decreciente.*
- *Si además es diferenciable a segundo orden, f es convexa si y sólo si $f''(x) \geq 0$ para todo $x \in I$.*

Observar que $(x - 6)^2 + (2y - 2)^2 - 30 = x^2 + 4y^2 - 12x - 8y + 10$ es convexa, porque es la suma de convexas $f(x) = (x - 6)^2 - 30$ y $g(y) = (2y - 2)^2$, y está definida en el conjunto suma de $\{(x, 0), x \in \mathbb{R}\}$ y $\{(0, y), y \in \mathbb{R}\}$, que también es convexo.

Identificando Funciones Convexas en \mathbb{R}^n

Propiedades

- Una función diferenciable $f : S \rightarrow \mathbb{R}$ en un conjunto convexo $S \subseteq \mathbb{R}^n$ es convexa sii $f(y) \geq f(x) + \nabla f(x) \cdot (y - x)$, para todo $x, y \in S$.
- Si la función f es diferenciable a segundo orden en S convexo y abierto, f es convexa sii su matriz Hessiana H es semidefinida positiva para todo $x \in S$.

$$H[f] = \begin{bmatrix} \frac{\partial^2 f(x)}{\partial x_1^2} & \cdots & \frac{\partial^2 f(x)}{\partial x_1 \partial x_n} \\ \vdots & & \vdots \\ \frac{\partial^2 f(x)}{\partial x_n \partial x_1} & \cdots & \frac{\partial^2 f(x)}{\partial x_n^2} \end{bmatrix}$$

Identificando Funciones Convexas en \mathbb{R}^n

Propiedades

- Una función diferenciable $f : S \rightarrow \mathbb{R}$ en un conjunto convexo $S \subseteq \mathbb{R}^n$ es convexa sii $f(y) \geq f(x) + \nabla f(x) \cdot (y - x)$, para todo $x, y \in S$.
- Si la función f es diferenciable a segundo orden en S convexo y abierto, f es convexa sii su matriz Hessiana H es semidefinida positiva para todo $x \in S$.

$f(x, y) = x^2 + 4y^2 - 12x - 8y + 10$, $\frac{\partial f}{\partial x} = 2x - 12$, $\frac{\partial f}{\partial y} = 8y - 8$,
 $\frac{\partial^2 f}{\partial x^2} = 2$, $\frac{\partial^2 f}{\partial x \partial y} = 0$ y $\frac{\partial^2 f}{\partial y^2} = 8$. Los valores propios de $H[f]$ son 2 y 8, así que es definida positiva y por tanto convexa.