Maestría en Ciencia de Datos del ITAM

# Teoría de Grafos para Análisis de Datos

Ricardo Mansilla

15 de febrero de 2016

# Índice

# Introducción

Con la rápida expansión de internet y aparición reciente de nuevas tecnologías que engendrán un volumen masivo de datos, tanto a nivel personal como colectivo, la industria ha reconocido la necesidad de dedicar cada vez mas recursos a la creación de tecnologías y técnicas de análisis de datos para procesar estos grandes volúmenes con baja latencia.

La aplicación de teorías y métodos de uso limitado al estudio de problemas puramente académicos en casos de aplicación real son cada vez mas frecuentes. Parte importante de estos problemas consiste en capturar en estructuras tan compactas y eficientes como sea posible, la interacción y correlación de los elementos (datos) que forman parte de nuestro sistema a estudiar. Las gráficas han demostrado ser en más de un campo de la matemática abstracta una de dichas estucturas extremadamente eficiente. Es por eso que algunos profesionales de la Ciencia de Datos han reconocido el poder que implica su uso y cada vez con más frecuencia las involucran en sus modelos. Problemas importantes del *machine learning* y de la ciencia de datos en general se basan en el uso de estructuras de grafos.

En este curso pretende establecer un camino posible (puesto que en la literatura no existe) para definir y establecer la teoría necesaria en el análisis de datos y el machine learning usando estructuras gráficas.

# 1. Teoría de Grafos

En muchos problemas que aparecen como casos de estudios en el mundo real, es necesario modelar la interacción entre los entes que forman parte del sistema a estudiar. Por ejemplo, algunos de estos casos podrían consistir en como se relacionan usuarios de un servicio *online* cualquiera. La manera en que se conectan los autores de artículos académicos a través de las citas mutuas que hacen en los mismos. O en un caso mas general, la proximidad bajo alguna medida de distancia que pueden tener dos puntos en algún espacio métrico. En todos ellos, tenemos una colección de entes, que llamamos vértices o nodos, y realciones entre ellos, que pueden verse como una colección de segmentos abstractos que unen a estos entes y modelan su relación, estas son llamadas aristas o flechas. El estudio de las estructuras abstractas de este tipo es lo que llamamos Teoría de Grafos.

Las relaciones entre nuestros entes pueden ser simétricas o no. Es decir, dado un conjunto $V$ y una relación

$$R = \{(a,b) \mid a,b \in V\}$$

se dice que esta es simétrica si para todo $(a,b) \in R \Rightarrow (b,a) \in R$. En otras palabras si la relación es mutua. Por ejemplo, la relación "*padre de*" entre un conjunto de familiares no es simétrica. Puesto que si *A es padre de B*, entonces *B no es padre de A*. Sin embargo la relación "*amigo de*" obviamente lo es. La estructura de grafos que modela una relación no simétrica es una **gráfica dirigida**.

## 1.1. Grafos y subgrafos

### 1.1.1. Grafos

De manera simple y suficientemente formal podemos definir una grafo como

**Definición 1.1.** *Sea un conjunto de elementos $V$ y un conjunto de pares $E$ de elementos de $V$, entonces definimos una grafo como el par ordenado $G = (V,E)$.*

*Dicho de otra forma, sea el conjunto $V$ y el conjunto $E = \{x,y \mid x,y \in V\}$. Entonces definimos $G = (V,E)$.*

*Al conjunto $V$ se le llama conjunto de vértices y a $E$ el conjunto de aristas.*

En general tomaremos el conjunto $V$ como un conjunto finito de elementos a menos que se indique lo contrario.

Los elemento de cada par ordenado que forma una arista se llaman los **extremos** de la arista. Notemos entonces que cada arista está definida por sus extremos unívocamente. Y que además, no podemos tener aristas que tengan extremos fuera del conjunto $V$.

En la literatura se hace referencia a que la palabra "*grafo*" proviene del hecho de que la estructura anterior puede ser representada de forma gráfica dibujando un punto por cada vértice en $V$ y uniéndolos a través de una linea por cada arista que existe en $E$. Es posible además encontrar referencias a los términos **gráfica** o **red**.

### 1.1.2. Subgrafos

Usando la terminología anterior podemos definir lo que es un subgrafo.

**Definición 1.2.** *Sea $G = (V, E)$, una gráfica (lo que implica que $V$ es un conjunto de vértices y $E$ de aristas). Si tomamos $V_s \subset V$ y $E_s \subset E$ de manera que*

$$\forall e \in E_s, e = (x, y) \Rightarrow x, y \in V_s$$

*entonces $G_s = (V_s, E_s)$ es un subgrafo de $G$.*

*Dicho de otra manera, si tomamos un subconjunto $V_s$ del conjunto de vértices y un subconjunto $E_s$ de aristas de forma que cada arista en $E_s$ contenga sus extremos en $V_s$, entonces el grafo formado por $G_s = (V_s, E_s)$ es un subgrafo de $G$.*

Es bastante claro que el requerimiento de que los extremos de cada elemento en $E_s$ estén contenidos en $V_s$ puesto que esta definición necesita ser compatible con la aterior.

Por otro lado, es importante notar, que para que un grafo $S$ sea subgrafo de $G$ (que denotamos como $S \hookrightarrow G$), es necesario que $\forall e \in S \Rightarrow e \in G$. Es decir, un subgrafo no puede tener aristas que no existan en el grafo original.

### 1.1.3. Matrices de adyacencia e inicidencia

Existen otras representaciones de los grafos que son bastante comunes en la literatura. Estas representaciones son menos intuitivas pero bastante mas eficiente cuando uno esta haciendo cómputo con grafos. Ambas son representaciones matriciales. La primera de ellas es conocida como la **matriz de adyacencia**. La matriz de adyacencia es básicamente una matriz cuadrada, de entradas binarias, con tantas filas como nodos. Las entradaas de esta matriz son **1**'s si los vértices correspondientes estan conectados y **0**'s en caso contrario.

Dicho de manera mas formal

**Definición 1.3.** *Sea $G = (V, E)$ un grafo. Entonces puesto que $V$ es finito podemos ordenar sus elementos en una secuencia $v_1, v_2, \ldots, v_n$, lo que nos permite construir la matriz cuadrada $M(G)$ de dimensión $n \times n$. En dicha matriz tenemos que*

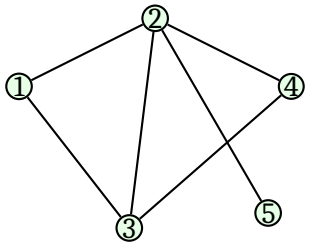
$$M(G)(v_i v_j) = 1 \Leftrightarrow (v_i, v_j) \in E$$

*y $M(G)(v_i v_j) = 0$ en caso contrario. De esta forma tenemos una definición biunívoca de una matriz "equivalente" al grafo $G$.*

La palabra equivalencia debe usarse con cuidado. En general lo anterior es cierto, pero hay que tener en cuenta el límite del significado de equivalencia en la teoría que se está trabajando.

Es importante notar que cada fila de esta matriz tiene tantos **1**'s como aristas inciden en el vértice, es decir, la fila $i$ tiene tantos **1**'s como elementos de $E$ tengan a $v_i$ como extremo. Esta propiedad es fundamental para el desarrollo posterior de nuestra teoría.

Supongamos que tenemos el siguiente grafo




Entonces tenemos una matriz de adyacencia equivalente

$$\mathrm{M(G)} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \end{pmatrix} \end{array}$$

La matriz que se muestra arriba tiene todos los elementos de su diagonal iguales a **1**. Esto es un convenio que se establece en la teoría de grafos usual, significando que cada vértice esta unido consigo mismo. A veces sin embargo es conveniente establecer este elemento en la diagonal como **0**. Es decir, escribiríamos

$$\mathrm{M(G)} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{pmatrix} 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{pmatrix} \end{array}$$

Nótese también que estas matrices son simétricas ($M(G) = M(G)^t$), ya que la relación codificada en el grafo también lo es, es decir el grafo no es dirigida.

Existe otra matriz de gran importancia y es la **matriz de incidencia**. Esta matriz nos dice como se relacionan los vértices y las aristas, es decir, nos permite codificar en una estructura algebráica única las etiquetas de ambos conjuntos. Usando el mismo ejemplo de la gráfica anterior y suponiendo que tenemos sus aristas ordenadas, la matriz correspondiente sería

$$\mathrm{I(G)} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{cccccc} e1 & e2 & e3 & e4 & e5 & e6 \end{array} \\ \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \end{array}$$

Es importante ver que esta matriz debe tener en cada columna dos y solamente dos **1**'s puesto que cada arista tiene solo dos extremos. De nuevo, el numero de **1**'s en cada fila nos dice la cantidad de aristas que tienen a dicho vértice como extremo.

Ambas matrices son extremadamente útiles para codificar una estructura de grafo en una unidad de cómputo. La equivalencia entre cada grafo y sus matrices correspondientes es de gran utilidad pues muchos de los resultados mas conocidos y usados han sido probados a través de las matrices de adyacencia correspondientes. El área que estudia esto es conocida como Teoría Algebraica de Grafos.

### 1.1.4. Caminos y ciclos

Los grafos son buenos entre otras cosas para modelar la dinámica de algunos sistemas. Muchas de estas técnicas exigen la existencia de definiciones formales que permitan establecer a los nodos como "estados" y a las aristas como la posibilidad de "cambiar de estado" lo que se encarga de generar la dinámica buscada.

Definamos primero una simple función que nos será de utilidad mas adelante

**Definición 1.4.** *Sea $\phi_G : E \longrightarrow V$, de manera que dada $e = (v_1, v_2) \in E$, $\phi_G(e) = \{v_1, v_2\} \subset V$. Es decir la función que envía cada aristas en sus extremos. A esta la llamaremos* **aplicación de extremos** *en G.*

**Definición 1.5.** *Sea $G = (V, E)$ un grafo, y $\alpha = e_1 e_2 \dots e_k$ una secuencia ordenada de aristas de G tales que $\forall e_i$,*

$$\phi_G(e_i) \cap \phi_G(e_{i-1}) \neq \emptyset,$$
$$\phi_G(e_i) \cap \phi_G(e_{i+1}) \neq \emptyset$$

*y*

$$\phi_G(e_i) \cap \phi_G(e_{i-1}) \neq \phi_G(e_i) \cap \phi_G(e_{i+1}).$$

*Esto es lo mismo que decir que en la secuencia $\alpha$ cada arista comparte exactamente un vértice con la próxima en la secuencia, distinto del que comparte con la anterior (si existe). A esto lo llamamos un* **camino** *en G.*

*La longitud de un camino es la antidad de aristas que forman parte de él, y se denota como $l(\alpha)$*

**Definición 1.6.** *Si en la definición anterior todas las $e_i$'s son distintas, entonces se dice que $\alpha$ es un* **recorrido** *o* **circuito**.

En general no haremos diferencia entre caminos y circuitos teniendo en cuenta que para fines prácticos solo nos interesan los circuitos (caminos sin repeticiones), que nosotros llamamos caminos.

**Definición 1.7.** *Sea $G = (V, E)$ un grafo y $\alpha = e_1 e_2 \dots e_k$ un camino en G, tomemos*

$$\{o_\alpha\} = \phi(e_1) - \phi(e_2)$$

*y*

$$\{l_\alpha\} = \phi(e_k) - \phi(e_{k-1})$$

*entonces se dice que el camino $\alpha$ "***conecta** $o_\alpha$ **con** $l_\alpha$ *" o que "***va desde** $o_\alpha$ **hasta** $l_\alpha$ *". Lo anterior podemos denotarlo como*

$$\alpha : e_1 \sim e_k$$

Nos podemos referir de muchas formas a lo anterior, pero la idea básica es que cada camino empieza en un vértice y termina en otro. Es claro que puede existir más de un camino que conecten dos vértices. Por otro lado, hay un hecho importantísimo que tiene que ver con el conjunto de dichos caminos

**Definición 1.8.** *Sean $G = (V, E)$ un grafo, $v_1, v_2 \in E$ y $C = \{\alpha \mid \alpha : v_1 \sim v_2\}$. Entonces existe $\alpha_0$, tal que $\forall \alpha \in C$, $l(\alpha_0) \leq \alpha$. Este camino $\alpha_0$ es llamado* **camino mínimo** *de $v_1$ a $v_2$.*

Más adelante veremos un algoritmo para calcular el camino mínimo, por ahora pensemos en la posibilidad de que un camino regrese en algún momento a su origen.

**Definición 1.9.** *Sea $G = (V, E)$ un grafo y $\alpha = e_1 e_2 \dots e_k$ un camino en $G$, entonces si $o_\alpha = l_\alpha$ se dice que $\alpha$ es un* **ciclo**.

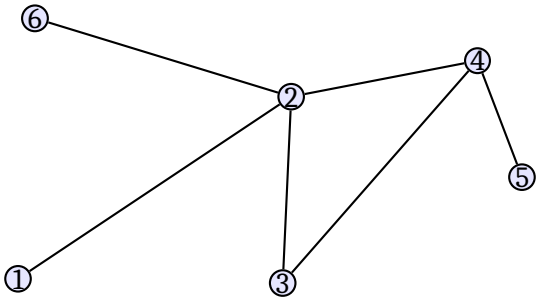
Por último daremos una definición estructural de suma importancia aprovechando la de ciclo

**Definición 1.10.** *Sea $G$ un grafo. Si $G$ no tiene ciclos entonces se dice que $G$ es un* **árbol**.

Para enunciar algunas de las propiedades básicas de un grafo necesitamos definir un concepto de suma importancia conocido como el **grado de un vértice**.

**Definición 1.11.** *Sea $G = (V, E)$ un grafo y $v \in V$. Sea además $k = |\{e \in E \; v \in \phi(e)\}|$, es decir el número de aristas que tienen a $v$ como extremo. Se dice entonces que $k$ es el* **grado** *de $v$, y se denota como $\sigma(v) = k$.*

Examinemos el siguiente grafo y veámos algunas de las propiedades que hemos definido




El vértice **2** tiene grado 4, mientras que el **5** tiene solo grado 1. Es interesante notar también, por ejemplo que existen solo dos caminos que comiencen en el vértice **1** y terminen en **5**. El camino $\alpha = \{(2,3),(3,4),(4,2)\}$ es un ciclo. Por último, si quitáramos la arista $(3,4)$ del grafo tendríamos un árbol.

### 1.1.5. Propiedades y resultados básicos

Comencemos esta sección con algunas propiedades estructurales de los grafos.

**Definición 1.12.** *Sea $G = (V, E)$ un grafo, se define la* **distancia** *entre dos vértices $v_1, v_2 \in V$ como la longitud del camino mínimo(1.8) entre $v_1$ y $v_2$. Esta distancia se denota como $d(v_1, v_2)$.*

La distancia es de hecho una métrica en el grafo puesto que el camino mínimo es invariante ya que solo recorremos a cada camino en el conjunto posible en dirección contraria, la distancia de cualquier vértice a si mismo es nula a través del camino trivial y la desigualdad del triángulo es consecuencia de la minimalidad de la longitud del camino que da la distancia.

**Definición 1.13.** *Sea $G = (V, E)$ un grafo y $v \in V$. Se define la* **excentricidad** *de $v$ como*

$$max_w\{d(v, w)\}$$

*donde $w \in V - \{v\}$. Es decir, la distancia máxima de $v$ al resto de los vértices. Esto se denota como $\epsilon(v)$.*

Intuitivamente uno podría decir que si un vértice tiene la excentricidad mas chica está cerca del "centro" del grafo.

**Definición 1.14.** *La menor de las excentricidades es el* **radio** *del grafo. Esto es*

$$\rho(G) = min_v\{\epsilon(v)\}$$

*con $v \in V$. Por tanto el* **centro** *del grafo es el vértice donde se alcanza el* **radio**, *en otras palabras, si*

$$\epsilon(v_0) = \rho(G)$$

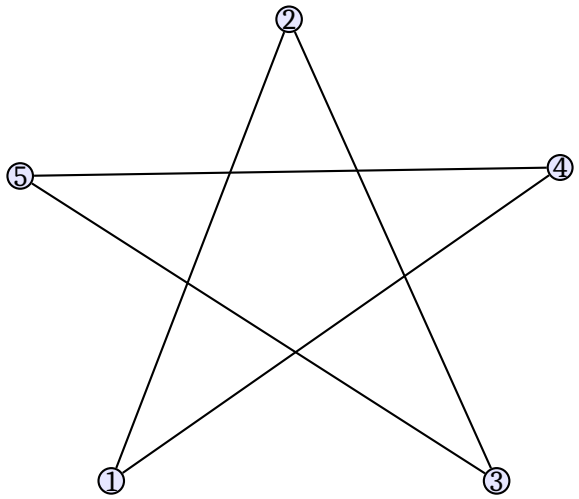
*entonces a $v_0$ se le llama* **centro**.

**Definición 1.15.** *Sea $G = (V, E)$ un grafo, entonces*

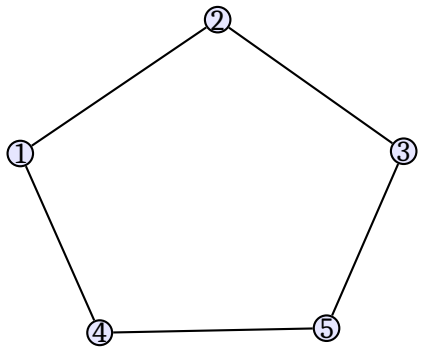
$$\delta(G) = max_v\{\epsilon(v)\}$$

*con $v \in V$ es conocido como el* **diámetro** *del grafo. La* **circunferencia** *es la longitud del ciclo más largo en $G$.*

Observemos el grafo siguiente



Este es un ejemplo de grafo **2-regular**, lo que significa que todos sus vértices tienen el mismo grado igual a 2. La excentricidad de cualquiera de sus vértices es 2, por tanto el radio del grafo es también igual a 2, lo que convierte a todos su vértices en centro del grafo. Finalmente la circunferencia es igual a 5, ya que de hecho el grafo es un ciclo de longitud 5, es decir el grafo anterior es "exactamente" el grafo




En matemáticas cuando decimos que dos estrucutras abstractas son la misma, en general se refiere a que son equivalentes bajo alguna aplicación que preserva su estructura. Estas aplicaciones son conocidas como isomorfismos. De manera similar definimos los isomorfimos de grafos

**Definición 1.16.** *Sean $G_1 = (V_1, E_1)$ y $G_2 = (V_2, E_2)$ dos grafos, entonces se dice que son* **isomorfos** *si existen funciones $\gamma$ y $\psi$ que cumplen*

$$\gamma : V_1 \longrightarrow V_2$$

*y $\gamma$ es una biyección de conjuntos,*

$$\psi : E_1 \longrightarrow E_2$$

*y $\psi$ es tal que*

$$e = (v, w),\ \psi(e) = (\gamma(v), \gamma(w))$$

*con $(\gamma(v), \gamma(w)) \in E_2$. Cuando $G_1 = G_2$ entonces a $(\gamma, \psi)$ se le llama* **automorfismo**.

En este sentido podemos reescribir lo anterior, y decir que los grafos **2**-regulares exhibidos arriba son isomorfos entre ellos (**ejercicio**). Uno de los problemas mas importantes de la teoría de la computación (por no decir el que más al momento), es equivalente al problema de decidir si dos grafos son isomorfos o no. Lo que sugiere que en general ésta no es tarea fácil.

Además de nuestras propiedades básicas tenemos algunos resultados que relacionan las definiciones que hemos estado estudiando.

**Teorema 1.1.** *Sea $G = (V, E)$ un grafo, entonces*

$$\sum_{v \in V} \sigma(v) = 2|E|$$

*donde $\sigma(v)$ es el grado del vértice $v$.*

En otras palabras, la suma de los grados de un grafo es exactamente dos veces la cardinalidad del conjunto de aristas. La intuición de la prueba es que básicamente cuando contamos el grado de los vértices contamos cada arista exactamente dos vértices, puesto que la arista tiene dos extremos.

De este resultado podemos derivar algunos corolarios

**Corolario 1.1.** *Sea $G = (V, E)$ un grafo, entonces el número de vértices de grado impar es par.*

**Demostración:** La demostración es bastante simple usando el resultado anterior. Puesto que

$$\sum_{v \in V} \sigma(v)$$

es un número par, el número de términos impares en la suma debe ser par. □

**Corolario 1.2.** *Sea $G = (V, E)$ un grafo regular, es decir $\forall v \in V$, $\sigma(v) = k$, con $k$ una constante, entonces*

$$k|V| = 2|E|.$$

□

De la misma forma tenemos cotas superiores e inferiores para el número de aristas basados en los grados de los vértices

**Corolario 1.3.** *Sea $G = (V, E)$ un grafo, entonces si*

$$max_{v \in V}\{\sigma(v)\} \leq k$$

*se cumple*

$$k|V| \geq 2|E|.$$

*La relación simétrica también es cierta, si tenemos que*

$$max_{v \in V}\{\sigma(v)\} \geq k$$

*se cumple*

$$k|V| \leq 2|E|.$$

□

Sobre los árboles tenemos dos teoremas importantes

**Teorema 1.2.** *Si $G$ es un grafo sin ciclos (un árbol), entonces entre dos vértices cualesquiera solo existe un camino.* □

**Teorema 1.3.** *Si $G$ es un árbol entonces se cumple que $|E| = |V| - 1$*

**Demostración:** La demostración es simple si usamos el principio de inducción sobre el número de vértices. Es decir, supongamos que $|V| = 1$, entonces el resultado es obvio.

Ahora supongamos que el resultado se cumple cuando $|V| = n$. Tomamos un grafo sin ciclos $A = (V_0, E_0)$, con $|V_0| = n+1$ y dos vértices $u, v \in V_0$, tales que $(u,v) \in E_0$. De esta manera, al hacer $E_0 - (u,v)$ tenemos dos subgrafos $A_1, A_2$, que son a su vez árboles puesto que $\alpha = (u,v)$ es el único camino $\phi(\alpha) = \{u,v\}$ (ya que $A$ es un árbol). Así se cumple

$$E(A_1) = V(A_1) - 1$$

$$E(A_2) = V(A_2) - 1$$

de lo que se deduce

$$E(A) = E(A_1) + E(A_2) + 1 = V(A_1) + V(A_2) - 1 = V(A) - 1.$$

□

### 1.1.6. Aplicaciones: El problema del camino mínimo

En esta sección veremos una aplicación de los conceptos definidos hasta el momento. La aplicación hace uso de un algoritmo diseñado por Dijkstra en 1959 para encontrar caminos mínimos en un grafo con pesos.

**Definición 1.17.** *Sea $G = (V,E)$ un grafo y sea $\omega : E \longrightarrow \mathbb{N}$, una función sobre las aristas de $G$. Entonces a la triada $G_\omega = (V,E,\omega)$ se le llama* **grafo con pesos** *o* **grafo ponderado**.

En general el codominio de $\omega$ puede ser cualquier conjunto numérico, pero tomarlo como $\mathbb{N}$ es suficiente. Dicho lo anterior reformulemos un poco algunos conceptos ya definidos.

**Definición 1.18.** *Sea $G_\omega = (V,E,\omega)$ un grafo ponderado y $\alpha$ un camino en $G_\omega$. Entonces definimos $l_\omega(\alpha)$, la longitud de $\alpha$ como la suma de los pesos de cada arista en $\alpha$, esto es*

$$l_\omega(\alpha) = \sum_{e \in \alpha} \omega(e).$$

*De esta manera $l_\omega$ define una métrica nueva en $G_\omega$, que es básicamente el camino de longitud mínima (peso mínimo) que una a dos vértices. Escrito formalmente, si*

$$C_{uv} = \{\alpha | \phi(\alpha) = \{u,v\}\}$$

*entonces*

$$d_\omega(u,v) = min_{\alpha \in C_{uv}}\{l_\omega(\alpha)\}.$$

**Definición 1.19.** *Sea $G_\omega = (V,E,\omega)$ un grafo ponderado y $S \subsetneq V$, esto es, un subconjunto propio de los vértices. Si tomamos $u \in V - S$, entonces definimos la distancia de $u$ a $S$ como*

$$d_\omega(u,S) = min_{v \in S}\{d_\omega(u,v)\}.$$

*Un camino mínimo de $u$ a $S$ es $\alpha = u \ldots v$, con $v \in S$, tal que*

$$l_\omega(\alpha) = d_\omega(u,S).$$

El problema del camino mínimo es encontrar, dados $u, v \in V$, el camino $\alpha$ con $l_w$ mínimo tal que $\alpha = u \dots v$. O dicho de otra manera, encontrar $\alpha$ tal que

$$\phi(\alpha) = \{u, v\}$$

y

$$l_\omega(\alpha) = d_\omega(u, v).$$

La idea del algoritmo es la siguiente. Supongamos que tenemos un conjunto $S \subsetneq V$, tomemos $u_0 \in S$ y denotemos $\overline{S} = V - S$. Entonces si $P = u_0 \dots u\overline{v}$ es un camino mínimo de $u_0$ a $\overline{S}$, es obvio que $u \in S$ y el camino $u_0 \dots u$ es un camino mínimo de $u_0$ a $u$ (por la definición todos los subcaminos de un camino mínimo son mínimos). Entonces es claro que

$$d_\omega(u_0, \overline{v}) = d(u_0, u) + \omega(u\overline{v})$$

por lo que la distancia de $u_0$ a $\overline{S}$ se puede escribir como

$$d_\omega(u_0, \overline{S}) = min_{u \in S, v \in \overline{S}} \{d(u_0, u) + \omega(u\overline{v})\}.$$

El algoritmo de Dijkstra se basa en la ecuación de arriba para $d_\omega(u_0, \overline{S})$ y hace uso de un proceso iterativo para calcular el camino mínimo de $u_0$ a todos los vértices.

Empezamos con un conjunto $S_0 = \{u_0\}$ y calculamos $d_\omega(u_0, \overline{S}_0)$, a través de un camino $P_1 = u_0 \dots u_1$, con $u_1 \in \overline{S}_0$. En este punto vale la pena notar que $u_0 u_1$ es un camino mínimo de $u_0$ a $u_1$, es decir, $u_1$ es el otro extremo de la arista de menor peso adyacente a $u_0$. Dicho de otra forma

$$d_\omega(u_0, \overline{S}_0) = min_{u \in S_0, u_1 \in \overline{S}_0} \{d(u_0, u) + \omega(u u_1)\}$$

pero $\forall u \in S_0$, $d_\omega(u_0, u) = 0$, entonces

$$d_\omega(u_0, \overline{S}_0) = min_{u_1 \in \overline{S}_0} \{\omega(u_0 u_1)\}.$$

Una vez que tengamos tal $u_1$, hacemos $S_1 = \{u_0, u_1\}$, tomamos $P_1$ y eso completa el primer paso del proceso iterativo.

En general tendremos en el paso $k$, conjuntos $S_0 \subset \dots \subset S_k$ y caminos $P_1, \dots, P_k$, con $S_k = \{u_0, \dots, u_k\}$. Para calcular $d_\omega(u_0, \overline{S}_k)$, escribimos

$$d(u_0, \overline{S}_k) = min_{u \in S_k, u_{k+1} \in \overline{S}_k} \{d(u_0, u) + \omega(u u_{k+1})\}$$

lo que nos da un procedimiento para encontrar $u_{k+1}$. Más aún, vemos en la ecuación de arriba que $u \in S_k$, es decir, $u = u_j$ para algún $0 \leq j \leq k$. Entonces, como ya tenemos $P_j = u_0 \dots u_j$ que es camino mínimo, hacemos $P_{k+1} = u_0 \dots u_j u_{k+1} = P_j + (u_j, u_{k+1})$ y $S_{k+1} = \{u_0, \dots, u_k, u_{k+1}\}$.

Computacionalmente hablando, algunos de los cálculos anteriores pueden ser repetidos, el algoritmo es básicamente una forma eficiente de hacer el procedimiento anterior. Para esto se le asignan etiquetas $l(v)$ para cada $v \in V$, donde cada una de estas es una cota superior de la longitud del camino mínimo de $u_0$ a $v$. A medida que el computo de los $S_i$ avanza estas etiquetas se modifican de forma que en el paso $k$, se cumple que $\forall v \in S_k$, $l(v) = d\omega(u_0, v)$ y

$$l(v) = min_{u \in S_{k-1}} \{d(u_0, u) + \omega(uv)\}$$

para $v \in \overline{S}_k$. Como eventualmente $S_k = V$ para algún $k$, entonces todas las etiquetas $l(v)$, con $v \in V$, tendrán los calores correctos. Veamos la descripción del algoritmo en pseudo código. Haremos uso de dos elementos importantes, un número $M$ que debe ser suficientemente grande, para esto podemos tomar $M = \sum_{e \in E} \omega(e)$, y de el hecho de que si $(u,v) \notin E$, entonces $\omega(u,v) = M$.

**Algoritmo 1** (de Dijkstra para el camino mínimo)

**Require:** $i = 0, S_0 = \{u_0\}, l(u_0) = 0$ y $l(v) = M, \forall v \in V - \{u_0\}$
  **while** $i \leq |V| - 1$ **do**
    **for** $v \in \overline{S}_i$ **do**
      $l(v) \leftarrow min\{l(v), l(u_i) + \omega(u_i v)\}$
    **end for**
    $u_{i+1} \leftarrow u \in \{v | l(v) = min_{v \in \overline{S}_i}\{l(v)\}\}$
    $S_{i+1} \leftarrow S_i \cup \{u_{i+1}\}$
    $i \leftarrow i + 1$
  **end while**

Cuando el algoritmo se detiene, las etiquetas $l(v)$ contienen la longitud del camino mínimo de $u_0$ al resto de los vértices. Uno podría ir almacenando los $P_j$ en cada iteración si quisiera tener un registro de los vértices que forman el camino.

## 1.2. Grafos dirigidos

Un grafo dirigido dicho coloquialmente es una estructura como las ya definidas, pero con la diferencia de que los elementos en el conjunto de aristas $E$ (que en el caso dirigido denotamos $A$) es un par ordenado donde si $e \in E$, $e = (u,v)$ y $\overline{e} = (v,u)$, entonces $e \neq \overline{e}$. De esta forma tenemos la definición

**Definición 1.20.** *Sea $D = (V,A)$ un par ordenado de dos conjuntos $V$ y $A$, donde $V$ es un conjunto abstracto de elementos y $A$ un conjunto de pares de $V$, con $A = \{(u,v) | u,v \in V\}$ de manera que $(u,v) \neq (v,u)$. Entonces a $D$ se le llama* **grafo dirigido** *o* **digrafo**. *Al conjunto $A$ se le llaman las* **flechas** *de $D$.*

*Si $a = (u,v) \in A$, entonces se dice que $a$* **empieza** *en $u$ y* **termina** *en $v$, o que $u$ es el* **origen** *o* **inicio** *de $a$ y que $v$ es el* **final** *de $a$.*

Existe la visión de que los grafos dirigidos son una extensión del caso no dirigido en el sentido de que todo grafo no dirigido puede pensarse (modelarse) como un grafo dirigido. Es decir, sea $\mathscr{P}_G$ la colección de todos los grafos no dirigidos posibles y $\mathscr{P}_D$ la de los digrafos existe una aplicación $e_G = (e_{G1}, e_{G2})$

$$e_G : \mathscr{P}_G \longrightarrow \mathscr{P}_D$$

donde $e_G(G) = e_G(V,E) = (e_{G1}(V), e_{G2}(E)) = (V,A) = D$. Esta aplicación cumple

$$e_{G1}(V) = V$$

y que $e_{G2} : E \longrightarrow A$ es tal que

$$e_{G2}(\{(u,v) \in E\}) = \{(u,v) \in E\} \cup \{(v,u) \in E\}$$

Esta asiganción de conjuntos esta bien definida, puesto que lo que hacemos es "completar" el conjunto $E$ con los pares inversos que representan la dirección contraria. De esta forma $A$ contiene ambas flechas y podemos decir que es "equivalente" a tener un grafo no dirigido.

Si en el ejemplo de arriba pensamos que a $A$ le podemos "quitar" una arista $a_0$, entonces es imposible encontrar un $G \in \mathscr{P}_G$ tal que $e_G(G) = (V, A - \{a_0\})$. Por tanto $\mathscr{P}_D$ "contiene" mas elementos que $\mathscr{P}_G$. En términos prácticos esto sugiere que la riqueza de los sistemas basados en digrafos es bastante mas intrincada que la de los no dirigidos.

Si $D = (V,A)$ es un digrafo y $a = (u,v) \in A$ definamos las funciones $\phi_-, \phi_+$, tales que

$$\phi_-(a) = u$$

$$\phi_+(a) = v$$

de igual manera la función extremo $\phi$ existe en el caso dirigido, es decir $\phi(u,v) = \{u,v\}$. Así, tenemos las siguientes definiciones correspondientes

**Definición 1.21.** *Sea $D = (V,A)$ un digrafo y tomemos $u \in V$. Entonces definimos*

- $\sigma^+(u)$ *como el* **ingrado** *de u, es decir* $|\{a | a \in A, \phi_+(a)\}|$
- $\sigma^-(u)$ *como el* **exgrado** *de u, es decir* $|\{a | a \in A, \phi_-(a)\}|$
- $\sigma(u)$ *como el* **grado** *de u, donde* $\sigma(u) = \sigma^+(u) + \sigma^-(u)$

### 1.2.1. Caminos y cliclos dirigidos

Como es de esperarse en los digrafos tendremos caminos de forma similar a los definidos en el caso no dirigido. Es decir, la función $\phi$ nos permite definir el caso de **camino no dirigo** exactamente igual a la Definición 1.5. Sin embargo en los digrafos tenemos la definición de caminos dirigidos

**Definición 1.22.** *Sea $D = (V,A)$ un digrafo y $\alpha = a_1 a_2 \dots a_k$ una secuencia de flechas en $A$ tales que $\forall i \in \{1, \dots, k-1\}$, $\phi_+(a_i) = \phi_-(a_{i+1})$. Se dice entonces que $\alpha$ es un* **camino dirigido** *en $A$.*

De igual manera damos la definición de **ciclos dirigidos**

**Definición 1.23.** *Sea $D$ un digrafo y $\alpha = a_1 a_2 \dots a_k$ un camino dirigido en $D$. Si se cumple que*

$$\phi_+(a_k) = \phi_-(a_1)$$

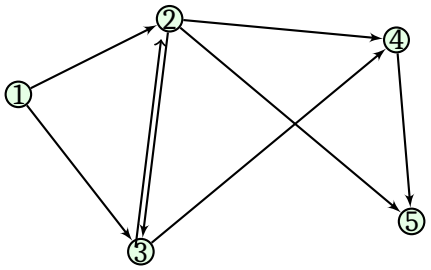
*entonces se dice que $\alpha$ es un ciclo dirigido en $D$.*

Los conceptos de excentricidad, centralidad y demás que vimos en la sección anterior tambien pueden ser replicados de manera natural para digrafos usando las funciones $\phi$, $\phi_+$ y $\phi_-$, con los conceptos de camino dirigido y no dirigido.

### 1.2.2. Matrices de adyacencia e inicidencia

Como ya hemos mencionado los digrafos son estructuras mas generales que los grafos no dirigidos. Este hecho se puede ver de manera un poco mas clara en las propiedades de sus matrices de adyacencia e incidencia.

Puesto que en los digrafos las aristas poseen dirección, la matriz de incidencia necesita codificar de alguna forma la dirección en la que inciden las flechas. Supongamos por ejemplo, que tenemos el siguiente grafo




Entonces nuestra matriz de incidencia deberia capturar, por ejemplo, ambas flechas: $(3,2)$ y $(2,3)$. Es por eso que la codificación de la incidencia es un tanto distinta en el caso dirigido.

Sea $D$ la grafica de arriba, escribamos la matrices

$$I^+(D) = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{cccccccc} (1,2) & (1,3) & (2,3) & (2,4) & (2,5) & (3,2) & (3,4) & (4,5) \end{array} \\ \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \end{pmatrix} \end{array}$$

$$I^-(D) = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{cccccccc} (1,2) & (1,3) & (2,3) & (2,4) & (2,5) & (3,2) & (3,4) & (4,5) \end{array} \\ \begin{pmatrix} -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & -1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \end{array}$$

$$I(D) = I^+(D) + I^-(D) = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{cccccccc} (1,2) & (1,3) & (2,3) & (2,4) & (2,5) & (3,2) & (3,4) & (4,5) \end{array} \\ \begin{pmatrix} -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & -1 & -1 & -1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & -1 & -1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \end{pmatrix} \end{array}$$

A la matriz $I(D)$ se le denomina **matriz de incidencia** y define como

**Definición 1.24.** *Sea $D=(V,A)$ un digrafo y sean $\{v_1,v_2,\ldots,v_n\}$ y $\{a_1,a_2,\ldots,a_m\}$, ordenaciones de sus vértices y sus aristas respectivamente. Entonces se define la matriz de incidencia $I(D)$ como*

$$I(D)_{ij}=\begin{cases}1 & a_j=(v_l,v_i)\in A\\ -1 & a_j=(v_i,v_l)\in A\\ 0 & e.o.c\end{cases}$$

La matriz de incidencia nos permite identificar que vértices son **focos** (de ingrado 0) o **atractores** (de exgrado 0). Por ejemplo, en la matriz $I(D)$ de el ejemplo anterior observamos que el exgrado del vértice **5** es cero puesto que la quinta fila de la matriz no tiene números negativos, por tanto este vértice es un atractor. De la misma forma el vértice **1** es un foco. Existen algunas porpiedades interesantes de esta matriz en términos espectrales que notaremos en secciones posteriores.

La matriz de adyacencia en el caso dirigido tiene un significado crucial. Se define formalmente como

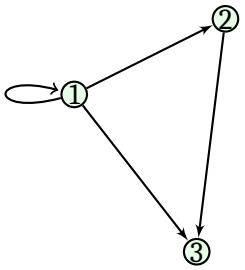
**Definición 1.25.** *Sea $D=(V,A)$ un digrafo y sea $\{v_1,\ldots,v_n\}$ una ordenación de sus vértices. Definimos la* **matriz de adyacencia** *de la siguiente forma*

$$M(D)_{ij}=\begin{cases}1 & (v_i,v_j)\in A\\ 0 & e.o.c\end{cases}$$

La matriz de adyacencia del grafo anterior (1.2.2), se escribe

$$\mathrm{M(G)}=\begin{array}{c} \\ 1\\2\\3\\4\\5\end{array}\begin{array}{c}\begin{array}{ccccc}1&2&3&4&5\end{array}\\ \begin{pmatrix}0&1&1&0&0\\0&0&1&1&1\\0&1&0&1&0\\0&0&0&0&1\\0&0&0&0&0\end{pmatrix}\end{array}$$

Lo primero a notar de esta definición es que la matriz que resulta no es simétrica ($M(G)\neq M(G)^t$). Como ya mencionamos esto codifica el mismo hecho escrito en la estructura del grafo de que estamos hablando de una relación entre nodos que no es simétrica. Más aún, como el grafo es simple, todos los elementos de su diagonal son nulos, es decir, no estamos permitiendo *loops* en los nodos, lo que significa situaciones del tipo siguiente




*Grafo no simple: Si $D$ es el grafo de arriba entonces $M(D)_{11}=1$.*

Otra cosa importante a notar es que el hecho de que el vértice **5** es atractor y el vértice **1** es foco esta codificado en que las fila 5 y la columna 1 son identicamente nulas respectivamente. Es decir, la suma sobre las filas y las columnas nos refieren el exgrado e ingrado de los vértices correspondientes. Esto esta relacionado y es origen del estudio de lo que sucede con las potencias de esta matriz y su significado en la topología del grafo. Si examinamos de nuevo el digrafo que tomamos como ejemplo (1.2.2) y su matriz de adyacencia

$$\mathrm{M(G)} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{pmatrix} 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \end{array}$$

Notamos que si pensamos por ejemplo en si existe el camino

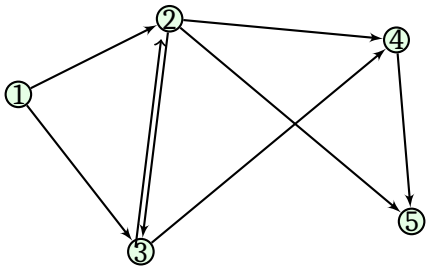
$$\alpha = (2,3)(3,4)(4,5)$$

la respuesta se puede visualizar de manera muy simple usando su matriz. Puesto que en cada uno de las entradas correspondientes a las "transiciones de estados"[1] en $M(D)$ hay **1**'s en cada una de estas entradas, i.e: $M(D)_{23}$, $M(D)_{34}$, $M(D)_{45}$. Estos **1**'s muestran que en efecto existe dicha arista.

Pensemos en que pasa cuando hacemos $M(D)^2 = M(D) * M(D)$, el computo sería

$$M(D)^2 = M(D) * M(D) = \begin{pmatrix} 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} * \begin{pmatrix} 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 1 & 1 & 2 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 2 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Ahora si examinamos el digrafo




notamos que por ejemplo existen 2 caminos de longitud 2, de **1** a **4**, lo que se manifiesta en $M(D)_{14}$. De la misma forma existen 2 caminos de longitud 2, de **3** a **5**, que es $M(D)_{35}$.

[1] Usamos a la ligera el lenguaje de sistemas dinámicos puesto que resulta intuitivo para los propósitos del curso

Y existe 1 camino de longitud 1, de **1** a **4**, es decir $M(D)_{14}$. La idea es bastante clara, de la misma forma que la matriz nos dice los caminos de longitud 1 que hay de un vértice a otro (las flechas), la segunda potencia nos explica la cantidad de caminos de longitud 2 que hay de un vértice a otro. Y así sucesivamente.

En general se cumple que si $\alpha_{ij}$ denota un camino de $v_i$ a $v_j$ y $l(\alpha_{ij})$ la longitud de dicho camino, entonces

$$M(D)^k_{ij} = |\{\alpha_{ij} | l(\alpha_{ij}) = k, \alpha_{ij} \hookrightarrow D\}|$$

es decir, la $k$-ésima potencia dice en la entrada $(ij)$, la cantidad de caminos de longitud $k$ que existen de $v_i$ a $v_j$ en $D$. Este resultado es de lo mas útil que existe en la teoría de digrafos.

### 1.2.3. Aplicaciones: PageRank

Una de las aplicaciones mas influyentes de las últimas décadas de los grafos dirigidos es el famoso algoritmo de PageRank creado por Larry Page y Sergei Brin para hacer un "ranking" de los sitios en la web. El algoritmo esta en el centro de la compañia de tecnologías de internet Google, y el artículo original contiene una version preliminar del método usado actualmente. Sin embargo para efectos prácticos del curso es bastante conveniente examinar una versión mejorada de este. No tan sofisticada como la que usa Hummingbird, que es la última versión del Google Sarch Engine.

Antes de estudiar el algoritmo vale la pena notar que si en una matriz de adyacencia de un grafo dirigido tomamos una normalización sobre sus filas, obtenemos lo que es conocido como una **matriz estocástica** (o **matriz de Markov**). En otras palabras

**Definición 1.26.** *Sea $P$ una matriz de $n \times n$, tal que $M_{ij} \neq 0$, $\forall i, j \in \{1, \ldots, n\}$ y que*

$$\sum_{j=1\ldots n} M_{ij} = 1$$

*entonces $P$ se conoce como* **matriz estocástica** *o* **de Markov**.

*Si $v$ es una matriz de $n \times 1$ (vector de dimensión $n$), entonces se conoce como* **vector estocástico**.

El sentido del uso de la palabra "Markov" para nombrar estas matrices es que básicamente modelan un proceso de Markov sobre un conjunto de estados finito. Vale la pena notar también que como estas son justo las probabilidades de transición, entonces la potencia $k$-ésima de la matriz $P$, dice la probabilidad de moverse después de $k$ pasos de un estado a otro. Lo que sería equivalente a (como vimos antes) contar los caminos posibles y sus probabilidades respectivas. De esta manera las potencias $P^k$, también son matrices estocásticas.

**Teorema 1.4.** *Si $P$ es una matriz estocástica y $v$ un vector estocástico entonces $vP$ es un vector estocástico.*

La prueba puede verse haciendo el producto y factorizando cada entrada de $v$ por filas de $P$, eso permite hacer la suma de los valores de $v$.

**Teorema 1.5.** *Si $P$ es una matriz estocástica, entonces tiene un valor propio $\lambda = 1$ y todos sus valores propios son menores iguales a 1 en valor absoluto.*

**Demostración:** Si $P$ es una matriz estocástica entonces como sus filas suman 1, tenemos que si $u = (1,1,\ldots,1)_n$ entonces

$$Pu^t = u$$

por tanto $u$ es un vector propio de $P$ asociado al valor propio 1. Por otro lado, si tuvieramos un valor propio $|\lambda| > 1$, entonces

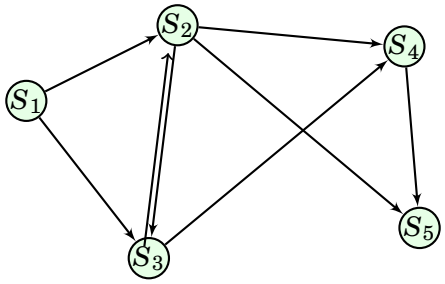
$$lim_{k\to\infty} \|P^k v^t\| = \||\lambda|^k v\| = \infty$$

por lo tanto al menos una entrada del producto $A^k v$ debe crecer exponencialmente. Como vimos en la matriz $P^k$ tenemos la probabilidad de movernos de un estado a otro en $k$ pasos del proceso estocástico, así que ninguno de estos valores puede ser mayor que 1, por tanto sus potencias también lo son. □

El vector propio asociado al valor propio 1 es conocido como el **vector de equilibrio** y representa la distribuciíon de equilibrio de la matriz $P$.

### PageRank

Ahora, lo anterior nos permite comprender un poco mejor el algoritmo de PageRank. Supongamos que tenemos un conjunto de nodos (que en la práctica representan websites), donde algunos de ellos apuntan a otros. Para modelar el ejemplo usemos el mismo grafo de antes pero pensemos en los nodos como los sitios: $S_1, S_2, \ldots, S_5$.




De esta forma el grafo captura el hecho de que, por ejemplo, el sitio $S_3$ contiene dos links que apuntan a $S_2$ y a $S_4$.

Ahora, supongamos que vemos los sitios y links como el digrafo $D = (V, A)$, entonces tenemos la siguiente definición

**Definición 1.27.** *Sea $D = (V, A)$ un digrafo, definimos los conjuntos*

$$A^-(S_k) = \{S_j | (S_k, S_j) \in A\}$$

*y*

$$A^+(S_k) = \{S_k | (S_j, S_k) \in A\}$$

es decir el conjunto de vértices que apuntan a $S_k$ o a los que este apunta respectivamente. En este caso, los sitios que contienen una referencia a $S_k$ o al revés.

Usando estos conjuntos definimos el PageRank como el volumen proporcional a la suma del PageRank de cada uno de los sitios en $A^-(S_k)$ entre su PageRank respectivo. En otras palabras

**Definición 1.28.** *Sea $D = (V, A)$ un digrafo. Definimos el* **PageRank** *de $v \in V$ como*

$$PR(v) = \sum_{w \in A^-(v)} \frac{PR(w)}{|A^+(w)|}$$

En la práctica la magnitud anterior es algo inconveniente puesto que si nos fijamos en nuestro ejemplo, el sitio $S_1$ luego del cómputo anterior tendría $PR(S_1) = 0$. Por otro lado, puesto que $S_5$ es un atractor, entonces el valor del PageRank de $S_5$ incrementaría [2]. Es por eso que se toma una magnitud denominada *damping factor* y se reformula el modelo anterior como

$$PR(v) = \frac{1-d}{|V|} + d\left(\sum_{w \in A^-(v)} \frac{PR(w)}{|A^+(w)|}\right)$$

El modelo esencialmente trata de implementar el proceso en el cual la probabilidad de que un usuario que navega aleatoriamente la red llegue a determinado sitio, induzca la "importancia" de tal sitio en la red. Esto esta relacionado con el *tiempo de paro* de determinado suceso, puesto que, teniendo en cuenta que es un proceso de Markov, el valor esperado de clics para recurrir a un sitio web, cuando el número de clics $T \to \infty$ es igual al PageRank. En otras palabras

$$PR(S) = \frac{1}{\mathbb{E}(T)}$$

Y $T$ es el tiempo de paro del evento en el cual uno parte del sitio $S$ y regresa a el mismo.

Es obvio que la definición de tal magnitud es recursiva puesto que cada $PR(w)$, con $w \in A^-(v)$ depende de $PR(v)$. Sin embargo existe un algoritmo iterativo que permite hacer dicho cálculo.

Antes de exhibir el proceso de cómputo vale la pena notar que básicamente lo que se quiere calcular es el valor del vector solución $PR$ de la siguiente ecuación

$$PR(t+1) = \frac{1-d}{|V|}\begin{pmatrix} 1 \\ 1 \\ 1 \\ \dots \\ 1 \end{pmatrix} + d\begin{pmatrix} I(v_1,v_1) & I(v_1,v_2) & I(v_1,v_3) & \dots & I(v_1,v_n) \\ I(v_2,v_1) & I(v_2,v_2) & I(v_2,v_3) & \dots & I(v_2,v_n) \\ I(v_3,v_1) & I(v_3,v_2) & I(v_3,v_3) & \dots & I(v_3,v_n) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ I(v_n,v_1) & I(v_n,v_2) & I(v_n,v_3) & \dots & I(v_n,v_n) \end{pmatrix}^{\mathrm{t}} PR(t)$$

[2] Cuando esto pasa el valor de tal sitio se divide entre el resto de la red.

donde

$$PR(t) = \begin{pmatrix} PR(v_1)(t) \\ PR(v_2)(t) \\ PR(v_3)(t) \\ \dots \\ PR(v_n)(t) \end{pmatrix}$$

y $I(v)$ es una función

$$I(v,w) = \begin{cases} \frac{1}{|A^+(v)|} & si\ (v,w) \in A \\ 0 & e.o.c \end{cases}$$

Por otro lado, si

$$E = \begin{pmatrix} |A^+(v_1)| \\ |A^+(v_2)| \\ |A^+(v_3)| \\ \dots \\ |A^+(v_n)| \end{pmatrix}^{\mathsf{t}} \begin{pmatrix} 1 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \end{pmatrix} = \begin{pmatrix} |A^+(v_1)| & 0 & 0 & \dots & 0 \\ 0 & |A^+(v_2)| & 0 & \dots & 0 \\ 0 & 0 & |A^+(v_3)| & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & |A^+(v_n)| \end{pmatrix}$$

y $A$ es la matriz de adyacencia, entonces

$$E^{-1}A = \begin{pmatrix} I(v_1,v_1) & I(v_1,v_2) & I(v_1,v_3) & \dots & I(v_1,v_n) \\ I(v_2,v_1) & I(v_2,v_2) & I(v_2,v_3) & \dots & I(v_2,v_n) \\ I(v_3,v_1) & I(v_3,v_2) & I(v_3,v_3) & \dots & I(v_3,v_n) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ I(v_n,v_1) & I(v_n,v_2) & I(v_n,v_3) & \dots & I(v_n,v_n) \end{pmatrix}$$

y lo anterior se puede escribir en notación matricial como

$$PR(t+1) = \frac{1-d}{|V|} \begin{pmatrix} 1 \\ 1 \\ 1 \\ \dots \\ 1 \end{pmatrix} + d\left(E^{-1}A\right)^{\mathsf{t}} PR(t)$$

El sistema anterior converge rápidamente (según el paper original de Page y Brin en alrededor de 54 iteraciones). El proceso iterativo se detiene cuando $\|c\|$ es suficientemente chico, donde

$$\epsilon = PR(t+1) - PR(t) = \begin{pmatrix} PR(v_1)(t+1) \\ PR(v_2)(t+1) \\ PR(v_3)(t+1) \\ \dots \\ PR(v_n)(t+1) \end{pmatrix} - \begin{pmatrix} PR(v_1)(t) \\ PR(v_2)(t) \\ PR(v_3)(t) \\ \dots \\ PR(v_n)(t) \end{pmatrix}$$

Veamos un ejemplo de pseudocódigo que implemente dicho proceso iterativo

**Algoritmo 2** (PageRank)

**Require:** $t = 0$, $it = 60$, $d = 0.85$ y $PR(v) = \frac{1}{N}, \forall v \in V$
  **while** $t \leq it$ **do**
    **for** $v \in V$ **do**
      $PR(v) \leftarrow \frac{1-d}{|V|} + d\left(\sum_{w \in A^-(v)} \frac{PR(w)}{|A^+(w)|}\right)$
    **end for**
    $t \leftarrow t + 1$
  **end while**

En el algoritmo anterior solo recorremos sobre las iteraciones $it = 60$. Pero se podría agregar otra condicion de paro usando $\epsilon = PR(t+1) - PR(t)$. En cualquier caso 60 iteraciones han demostrado ser suficientes para encontrar una aproximación sobrada del valor de equilibrio del vector $PR$.

## 1.3. Conectividad

Entre las propiedades topológicas mas importantes de un grafo está la conectividad. Básicamente se refiere a la característica estructural de los grafos poseer "conexión" entre sus nodos basándose en la existencia de caminos entre estos

**Definición 1.29.** *Sea $G = (V, A)$ un grafo entonces se dice que $G$ es* **conexo** *si $\forall v, w \in V$ existe un camino $\alpha = v \ldots w$.*

La definición anterior también se aplica para digrafos, sin embargo, es obvio que si se define la conexidad de caminos como arriba entonces uno puede pensar en caminos dirigidos. Esto sugiere que en grafos dirigidos podría existir una definición de conexidad dirigida más fuerte. Tenemos la siguiente

**Definición 1.30.** *Sea $D = (V, A)$ un digrafo. Se dice que $D$ es fuertemente conexo si para todos $u, v \in V$, $\exists \alpha$, $\beta$ caminos en $D$ tal que*

$$\alpha : u \sim v$$

*y*

$$\beta : v \sim u$$

donde $\sim$ denota la existencia de un camino. Es conveniente definir el camino inverso para algunas demostraciones por tanto lo establecemos como

**Definición 1.31.** *Sea $\alpha$ un camino en un grafo $G = (V, A)$ definido como*

$$\alpha = e_1 e_2 \ldots e_n$$

*donde $\{e_1, e_2, \ldots, e_n\} \subset A$. Definimos entonces $\alpha^{-1}$ el* **camino inverso**

$$\alpha = e_n e_{n-1} \ldots e_1$$

### 1.3.1. Componentes

Usando los caminos y sus inversos podemos estrablecer una relación mas general entre los nodos. De echo esta relación, por ser mas general, cumple lo que la relación cofidicada originalmente en las aristas no tiene por que cuplir

**Teorema 1.6.** *La relación entre nodos de "estar conectados" es de equivalencia, i.e $u \sim v$ es una relación de equivalencia.*

**Demostración:** Sean $u, v, w \in V$ y sean $\alpha, \beta$ caminos tales que $\alpha : u \sim v$ y $\beta : v \sim w$, y sea $\alpha_0 : u \sim u$ el camino trivial. Entonces

- Reflexividad: es obvio que $\alpha_0 : u \sim u$ entonces $u \sim u$.
- Simetría: si $\alpha : u \sim v$ entonces $\alpha^{-1} : v \sim u$ es una relación $v \sim u$.
- Transitividad: si $\alpha : u \sim v$ y $\beta : v \sim w$ entonces $\alpha\beta : u \sim w$ y se da la relación.

Puesto que la relación de conexidad entre nodos es de equivalencia, esta define clases de equivalencia en el conjunto de vértices de grafos, tenemos la siguiente definición

**Definición 1.32.** *En un grafo las clases de quivalencia inducidas por la relación de conexidad $(\sim)$ se denominan* **clases de conexidad** *o* **componentes conexas**. *El número de componentes conexas de $G$ lo denotamos como $\omega(G)$.*

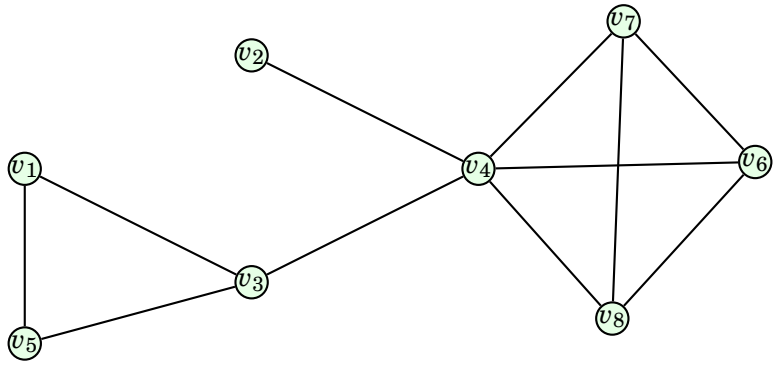
### 1.3.2. Nodos y aristas de corte

En un grafo es importante medir y definir a veces el nivel de conexidad del mismo. Existen conceptos importantes relacionados con este hecho

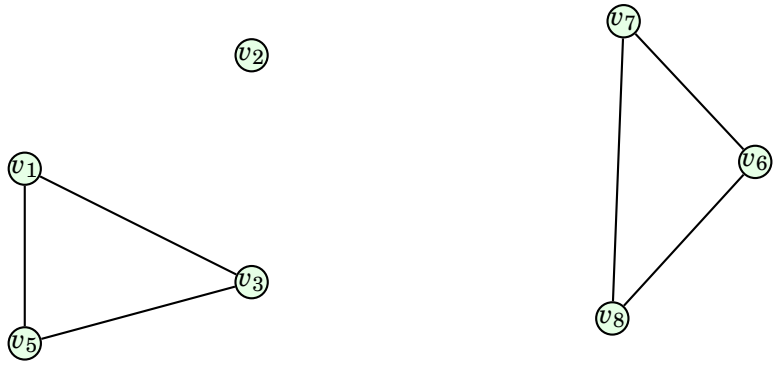
**Definición 1.33.** *Sea $G$ un grafo, se define una* **arista de corte** *de $G$ como aquella tal que*

$$\omega(G - e) > \omega(G)$$

En otras palabras, una arista de corte es aquella que al quitarla de $G$, "desconectamos" esta aún mas. Los nodos de corte tienen una definición similar, solo que vale la pena destacar que al "quitar" un nodo de un grafo es necesario quitar todas las aristas adyacentes a el. En este sentido un **nodo es de corte** si tiene una arista adyacente que sea de corte. Por ejemplo en el siguiente grafo



las aristas $(v_2, v_4)$, $(v_3, v_4)$ son de corte ya que desconectan la gráfica. Por otro lado el nodo $v_4$ es un nodo de corte puesto que al quitarlo obtenemos tres componentes conexas, es decir $G - \{v_4\}$




Tenemos algunos teoremas importantes sobre las aristas de corte

**Teorema 1.7.** *Una arista e de un grafo G es de corte sii e no esta contenida en ningún ciclo de G.*

Es facil ver que si una arista forma parte de un ciclo y la quitamos siempre existe un camino entre los nodos del ciclo aún después de quitarla. También tenemos

**Teorema 1.8.** *Un grafo G es un árbol si cada arista es de corte.*

El teorema implica entre otras cosas la definición de que un árbol no tiene cilcos. En este sentido podemos pensar en que existe un número máximo de aristas que podemos quitar a un grafo $G$ hasta hacerlo disconexo. Esto es, una serie de transformaciones $G - e_1, G - e_1, e_2, \ldots$ de manera que $T = G - e_1, e_2, \ldots, e_k$ sea tal que $\omega(T - e) > \omega(T), \forall e \in A(T)$. Es decir, de manera que si quitamos una arista más, nuestro grafo es disconexo. Esta es exactamente la definición de árbol (no tiene cilcos). La gráfica que se obtiene haciendo la transformación $G \to G - S_e = T$, donde $S_e = \{e_1, \ldots, e_k\}$, que se puede escribir como $G \to T_S$ se le llama buscar un árbol generador de $G$. Los árboles generadores no son únicos, y son centrales en el estudio de las propiedades topológicas de los grafos.

**Corolario 1.4.** *Si $T$ es un árbol generador de $G$, entonces $T - \{v\}$ es un árbol generador de $G - \{v\}$. En general si $S_v = \{v_1, \ldots, v_n\}$ entonces $T - S_v$ son árboles generadores de las componentes conexas de $G - S_v$.*

La operación anterior es conocida como **corte por n vértices**. Es decir en el caso anterior $G - S_v$ es un $n$-corte de $G$. Se dice que un grafo es **k-conexo** cuando

$$k = min_{n \in \{1,2,\ldots,N\}} n$$

tal que existe un $S_v$ con $|S_v| = n$ donde $G - S_v$ es disconexo. Es decir, el mínimo número de vértices que debemos extraer de $G$ para hacerlo disconexo. Este número se denota $K_v(G)$.

De manera similar podemos definir un **corte por n aristas** y decir que el grafo $G$ es **k-conexo por aristas**. Así denotamos la conexidad por aristas como $K_e(G)$.

**Teorema 1.9.** *Se cumple en general que $K_v(G) \leq K_e(G)$.*

Lo anterior puede verse como consecuencia de que cada vez que quitamos un nodo quitamos todas las aristas adyacentes a él.

### 1.3.3. K-Cores

Los **k-cores** de un grafo estan relacionadas con la interconectividad del mismo. Es importante por ejemplo conocer y resolver en tiempo tan eficiente como sea posible el problema de entender cuales son los nodos con mayor interconectividad[3]. Básicamente tenemos la siguiente definición

**Definición 1.34.** *Sea $G = (V, A)$ un grafo. Se definen los* **k-cores** *de un grafo como las componentes conexas de el grafo que resulta al remover de G todos los nodos con grado menor o igual que k. Es decir*

$$G - S_v^k = \{C_1^k, C_2^k, \ldots\}$$

*donde $S_v^k = \{v | v \in V,\ \sigma(v) \leq k\}$ y cada $C_j^k$ es una componente conexa.*

El concepto de **k-coreness** (o $k$-centralidad) de un vértice $v$ esta dado por el hecho de que $v$ pertenece a un $k$-core pero no a un $k+1$-core.

Otra definición equivalente de $k$-cores es la siguiente[4]

**Definición 1.35.** *Sea $G = (V, E)$ un grafo con $|V| = n$, $|E| = m$. Un subgrafo inducido (que contiene todas las aristas posibles) $S = (C, E_S)$, con $C \subset V$ es un* **k-core** *de G sii el grado de cada $v \in C$ en S es mayor o igual que k, i.e $\sigma_S(v) \geq k, \forall v \in C$.*

Existe un algoritmo en tiempo lineal para obtener los $k$-cores de un grafo $G$

**Algoritmo 3** (K-Cores)

**Require:** $k = 0$, $S_\sigma$ y $Core(v) = \sigma(v), D(v) = \sigma(v) \forall v \in V$
  **for** $v \in S_\sigma$ **do**
    $Core(v) = D(v)$
    **for** $w \in N(v)$ **do**
      **if** $D(w) > D(v)$ **then**
        $D(w) = D(w) - 1$
      **end if**
    **end for**
    resort $S_\sigma$
  **end for**

En el algoritmo $S_\sigma$ es una lista con los vértices ordenados por su grado. Al final tendremos en $Core(v)$ el $k$ del core correspondiente a $v$.

[3] Nótese que esto aunque relacionado no está exactamente descrito por el grado de los nodos

[4] V. Batagelj and M. Zaversnik, *Generalized cores*, **CoRR, cs.DS/0202039**, 2002

### 1.3.4. Aplicaciones: Usuarios relevantes en una red social

## 1.4. Redes y flujos

### 1.4.1. Redes

### 1.4.2. Flujos

### 1.4.3. Cortes

### 1.4.4. Aplicaciones

# 2. Teoría Aleatoria de Grafos

## 2.1. El modelo de grafos aleatorios

### 2.1.1. Número de links

### 2.1.2. Distribución de grados

### 2.1.3. Aplicaciones

## 2.2. Propiedades de los grafos aleatorios

### 2.2.1. Mundo pequeño

### 2.2.2. Clustering

### 2.2.3. Redes aleatorias reales

### 2.2.4. Aplicaciones

# 3. Teoría Algebraica de Grafos

## 3.1. Teoría espectral de grafos

### 3.1.1. Valores propios

### 3.1.2. Polinomio característico

### 3.1.3. Aplicaciones

## 3.2. Grafos regulares

### 3.2.1. Teoría

### 3.2.2. Aplicaciones

## 3.3. Grafos de distancia transitiva

### 3.3.1. Teoría

### 3.3.2. Aplicaciones

# 4. Teoría topológica de Grafos

## 4.1. Grafos embedidos

### 4.1.1. Triangulaciones

### 4.1.2. Simplejos

### 4.1.3. Aplicaciones: Ejemplo

## 4.2. Reconstruccion de variedades

### 4.2.1. Teoría

### 4.2.2. Aplicaciones: Ejemplo