



Compresión de datos

Block sorting
Move to Front
Modelo de Shannon
Modelo estructurado



Block sorting



Block Sorting

- También llamada Burrows-Wheeler
- Es una transformación que no comprime, incluso expande la fuente
- Obtiene una permutación de caracteres para **mejorar la localidad**
- La salida, con más localidad, es apropiada para ingresar al algoritmo Move to Front



Block sorting: transformación

- Se colocan todas las permutaciones de la fuente en una matriz, desplazando un carácter más en cada fila
- Se ordenan alfabéticamente las filas
- Se emiten la última columna y el índice de la fila correspondiente a la fuente original



Block sorting: ejemplo

SALALALASARAS

Entrada

S A L A L A L A S A R A S

Salida

S L L S L R A A A S A A 10

←BS→

0	S	A	L	A	L	A	L	A	S	A	R	A	S
1	S	S	A	L	A	L	A	L	A	S	A	R	A
2	A	S	S	A	L	A	L	A	L	A	S	A	R
3	R	A	S	S	A	L	A	L	A	L	A	S	A
4	A	R	A	S	S	A	L	A	L	A	L	A	S
5	S	A	R	A	S	S	A	L	A	L	A	L	A
6	A	S	A	R	A	S	S	A	L	A	L	A	L
7	L	A	S	A	R	A	S	S	A	L	A	L	A
8	A	L	A	S	A	R	A	S	S	A	L	A	L
9	L	A	L	A	S	A	R	A	S	S	A	L	A
10	A	L	A	L	A	S	A	R	A	S	S	A	L
11	L	A	L	A	L	A	S	A	R	A	S	S	A
12	A	L	A	L	A	L	A	S	A	R	A	S	S

A	L	A	L	A	L	A	S	A	R	A	S	S	A
A	L	A	L	A	S	A	R	A	S	S	A	L	A
A	L	A	S	A	R	A	S	S	A	L	A	L	A
A	R	A	S	S	A	L	A	L	A	L	A	S	A
A	S	A	R	A	S	S	A	L	A	L	A	S	A
A	S	S	A	L	A	L	A	L	A	S	A	R	A
L	A	L	A	L	A	S	A	R	A	S	S	A	A
L	A	L	A	S	A	R	A	S	S	A	L	A	A
L	A	S	A	R	A	S	S	A	L	A	L	A	A
R	A	S	S	A	L	A	L	A	L	A	S	A	A
S	A	L	A	L	A	L	A	S	A	R	A	S	A
S	A	R	A	S	S	A	L	A	L	A	L	A	A
S	S	A	L	A	L	A	L	A	S	A	R	A	A



Block sorting: antitransformación

- Se intenta reconstruir la matriz, primero se escribe la última columna

A
A
A
A
A
A
L
L
L
R
S
S
S



Block sorting: antitransformación

- Se intenta reconstruir la matriz, primero se escribe la última columna
- Se la reordena y se obtiene la primera

S
L
L
S
L
R
A
A
A
S
A
A

A
A
A
A
A
A
L
L
L
R
S
S
S



Block sorting: antitransformación

- Se intenta reconstruir la matriz, primero se escribe la última columna
- Se la reordena y se obtiene la primera
- Se concatena la primera y segunda, se la reordena y se obtiene la tercera

A	L
A	L
A	L
A	R
A	S
A	S
L	A
L	A
L	A
R	A
S	A
S	A
S	S

A
A
A
A
A
A
L
L
L
R
S
S
S



Block sorting: antitransformación

- Se intenta reconstruir la matriz, primero se escribe la última columna
- Se la reordena y se obtiene la primera
- Se concatena la primera y segunda, se la reordena y se obtiene la tercera
- Se repiten los pasos hasta obtener toda la matriz
- La fila indicada por el índice es la fuente original



Move to Front



Move to Front

- Es una transformación que no comprime
- Reduce la entropía de la fuente aumentando la probabilidad de unos símbolos por sobre la de otros
- Sólo funciona bien si la fuente tiene mucha localidad
- El índice recibido desde el MTF no es transformado



Move to Front

- Si su entrada es apropiada, su salida es mucho más adecuada para ser comprimida por algunos compresores estadísticos, y en algunos casos, por un compresor half coding
- Tiene la propiedad de transformar repeticiones de cualquier carácter en números bajos. Cuanto más cerca la repetición, más bajo el número



Move to Front: transformación

- Se codifica el alfabeto de la forma estándar (sin comprimir), por lo que a cada símbolo le corresponde un número. Se arma un array con los números.
- Se van leyendo símbolos de la fuente según la codificación. Por cada símbolo que se lee, se emite su posición en el array, y se mueve ese símbolo a la posición 0 del array (el frente)



Move to Front: ejemplo

SALALALASARAS, luego del BS se transforma en SLLSRAAAASAA-10

Equivalencias

A 0
L 1
R 2
S 3

0	3	1	1	3	1	2	0	0	0	0	3	0
1	0	3	3	1	3	1	2	2	2	2	0	3
2	1	0	0	0	0	3	1	1	1	1	2	2
3	2	2	2	2	2	0	3	3	3	3	1	1

3	2	0	1	1	3	3	0	0	0	3	1	0
↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓

3	1	1	3	1	2	0	0	0	0	3	0	0
↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓

Entrada

←MTF→

Salida



Move to Front: antitransformación

- Se vuelve a construir el array con las codificaciones del alfabeto
- Se va leyendo a la entrada, donde cada lectura es una posición. Se emite el símbolo encontrado en la posición leída, y se pasa el símbolo al frente del array.



Move to Front

- Concepto de la compresión con BS y MTF:





Modelo de Shannon



Modelo de Shannon

- Es un compresor que utiliza como base el aritmético
- Se comporta muy bien si la fuente es la salida de un BS+MTF
- Utiliza 6 modelos (o contextos) numerados, de los cuales 5 matchean con un solo símbolo
- El único otro símbolo de esos modelos es el ESC



Modelo de Shannon

- Los modelos 0 y 1 matchean con el símbolo 0, es decir, emiten M si el símbolo es 0, ESC si otro
- El modelo 2 matchea con el símbolo 1
- El modelo 3 matchea con el símbolo 2
- El modelo 4 matchea con el símbolo 3
- El modelo 5 matchea con el resto de los símbolos



Modelo de Shannon

- El modelo 0 se utiliza al principio, y cuando recién hubo un match para el símbolo 0
- El modelo 1 se utiliza en el resto de los casos
- Cuando ocurre un ESC en un modelo, se busca en el modelo siguiente
- Los modelos 0 y 1 nunca se utilizan en el mismo paso

Ejemplo: 002003508

Car	M0	M1	M2	M3	M4	M5	Observaciones
0	M(1/2)						Al ser el primer caracter, se comienza desde el modelo 0. Este emite un match y se pasa al siguiente caracter. Se actualizan las probabilidades en el modelo 0
0	M(2/3)						Ahora en el modelo 0 el match se emite con probabilidad de 2/3 por haber aumentado su frecuencia en el paso anterior
2	E(1/4)		E(1/2)	M(1/2)			Se emite un escape en el modelo 0 y en el modelo 2 (no se pasa por el 1). Luego en el modelo 3 se emite un match. Se actualizan frecuencias en modelos 0, 2 y 3 (solo por los modelos en los que paso)
0		M(1/2)					Como el ultimo caracter no fue un 0, se comienza del modelo 1, que tiene al match y al escape con frecuencias iniciales 1
0	M(3/5)						Se vuelve al modelo 0 por ser un caracter 0 el ultimo comprimido
3	E(2/6)		E(2/3)	E(1/3)	M(1/2)		El caracter 3 se encuentra recién en el modelo 4
5		E(1/3)	E(3/4)	E(2/4)	E(1/3)	5(1/253)	Se llega al modelo 5, donde hay 253 simbolos con frecuencia 1.
0		M(2/4)					
8	E(3/7)		E(3/4)	E(2/4)	E(2/3)	8(1/254)	La frecuencia acumulada del modelo 5 es 254 porque el caracter 5 tiene frecuencia 2
EOF		E(2/5)	E(4/5)	E(3/5)	E(3/4)	EOF(1/255)	Fin del archivo



Modelo estructurado



Modelo estructurado

- Es un compresor que utiliza como base el aritmético
- Se comporta muy bien si la fuente es la salida de un BS+MTF
- Utiliza 9 modelos (o contextos) numerados, donde los más altos matchean con más cantidad de símbolos



Modelo estructurado

- Siempre se empieza del modelo 0
- Matches:

Modelo	Caracteres
0	0
1	1
2	2, 3
3	4, 5, 6, 7
4	8 al 15
5	16 al 31
6	32 al 63
7	64 al 127
8	128 al 255 y EOF



Modelo estructurado

Ejemplo: 002003508

Car	M0	M1	M2	M3	M4	M5	M6	M7	M8
0	0 (1/2)								
0	0 (2/3)								
2	E (1/4)	E (1/2)	2 (1/3)						
0	0 (3/5)								
0	0 (4/6)								
3	E (2/7)	E (2/3)	3 (1/4)						
5	E (3/8)	E (3/4)	E (1/5)	5 (1/5)					
0	0 (5/9)								
8	E (4/10)	E (4/5)	E (2/6)	E (1/6)	8 (1/8)				
EOF	E (5/11)	E (5/6)	E (3/7)	E (2/7)	E (1/9)	E (1/16)	E (1/32)	E (1/64)	EOF (129)



Comparando

- A la salida de BS+MTF, se pueden aplicar tres compresores que obtienen buenos resultados:
 - Modelo estructurado
 - Modelo de Shannon
 - Half Coding
- Los tres se benefician de las apariciones del carácter 0



Comparando

- Half Coding se beneficia más que los otros dos compresores con la abundancia de 0s. Los otros dos se benefician por igual
- Si hay abundancia de caracteres de valores intermedios (por ejemplo 4, 7 o 20) el modelo de Shannon comprime peor que el estructurado