Compression et comparaison de séquences par la transformée étendue de Burrows-Wheeler

Michel Chilowicz

15 mars 2006

Plan

1. Introduction
2. Transformée de Burrows-Wheeler pour un mot
3. Extension de la transformée de Burrows-Wheeler à un ensemble de mots
4. Comparaison de séquences
5. Compression de textes avec la transformée étendue
6. Conclusion

Introduction

Transformation de Burrow et Wheeler (BWT)

- Proposé par Burrows et Wheeler en 1994 [1].
- Permutation réversible sur les mots.
- Utile pour la compression de données.

Extension de BWT sur un ensemble fini de mots

- Extension proposée par Mantaci, Restivo, Rosone et Sciortino.
- Généralisation de BWT à un ensemble de k mots.
- Meilleures performances pour la compression de données.
- Utilisée pour introduire une fonction de distance pour la comparaison de génômes.

Transformée de Burrows-Wheeler

Soit $w = a_1 \cdots a_n$ un mot primitif sur l'alphabet ordonné A.
Calcul de $T(w) = ((L), i)$:

1. Tri des $|w|$ conjugués de w par ordre lexicographique.
2. Mémorisation de la séquence $(L)_{0 \leq i \leq |w-1|}$ des dernières lettres des conjugués triés.
3. Mémorisation de l'indice i de la position de w dans la liste triée des conjugués.

Complexité :

- Par tri par bacs : $O\left(|w|^2\right)$
- Par tri des suffixes du mot de Lyndon w (tableau, arbre de suffixes) : $O\left(|w|\right)$.

Transformée de Burrows-Wheeler : exemple sur $w = abraca$

	F					L
1	a	a	b	r	a	c
→2	a	b	r	a	c	a
3	a	c	a	a	b	r
4	b	r	a	c	a	a
5	c	a	a	b	r	a
6	r	a	c	a	a	b

$T(abraca) = (caraab, 2)$

Interprétation combinatoire

Permutation π

π permutation de $1, \ldots, n$ avec $\pi_i = j$ ssi F_i et L_j correspondent au même caractère

$\longrightarrow$ cycle de conjugués décalés.

Exemple avec $w = abraca$

$F = aaabcr$ et $L = caraab$

$$\pi = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 2 & 4 & 5 & 6 & 1 & 3 \end{pmatrix}$$

Cycle obtenu : $(2, 4, 6, 3, 5, 1)$

Injectivité de la transformée de Burrows-Wheeler

BWT n'est pas surjective...

Certains mots de $A*$ ne sont l'images d'aucun mot par BWT.

... mais est injective

BWT est donc une transformation réversible.

Transformée inverse de Burrows-Wheeler

$rang(i, y)$: nombre d'occurrences de b_i dans $pref_i(y)$

- $rang(i, F) = rang(\pi(i), L) \implies (i < j \text{ et } F_i = F_j \implies \pi(i) < \pi(j)$
- Reconstruction possible de π à partir de L.

Algorithme de reconstruction de π

1. Tri de L : obtention de F.
2. Pour chaque F_i, recherche de j tel que $F_i = L_j$ et $rang(i, F) = rang(j, L)$: $\pi(i) = j$.

Transformée inverse de Burrows-Wheeler (2)

$\pi \rightarrow$ cycle des positions $\imath, \pi(i), \cdot, \pi^{n-1}(i)$ des conjugués de décalage $1, \ldots, n$ de π.
$\rightarrow L_{\pi(i)} \cdots (L_{\pi^n(i)} = L_i) = a_1 \cdots a_n$.
Pour $i = 1$, obtention du mot de Lyndon de la classe de conjugaison.

Complexité

Tri de L en temps $O(n)$
Reconstitution de π en temps $O(n)$

Exemple de transformation inverse

$$T(w) = (caraab, 2) \rightarrow w =?$$

- Détermination de F (tri de L) : $F = aaabcr$.
- Obtention de π :

$$\pi = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 2 & 4 & 5 & 6 & 1 & 3 \end{pmatrix}$$

- Cycle obtenu : $(2, 4, 6, 3, 5, 1)$
- $w = a_1 \cdot a_6 = L_4 L_6 L_3 L_5 L_1 L_2 = abraca$

π est une permutation valide pour $T(w) \Leftrightarrow \pi$ cyclique.

Application à la compression de données

Une remarque

- Consécutivité des conjugués triés avec prefixes communs.
- Conjugués avec long préfixe commun
  $\longrightarrow$ forte probabilité que la lettre précédente (L) soit identique.
- Exemple pour un texte en anglais : conjugués partageant le préfixe *he*· triés consécutivement et précédés généralement (lettre L) par t (mot *the*).

Application à la compression de données (2)

Conséquence

- En général fort regroupement des lettres identiques dans $T(w)$:
  $\longrightarrow$ meilleur ratio de compression sur $T(w)$ que sur w.
- Méthode de compression originale (BZIP2) : *Move to Front* $\longrightarrow$ codage de Huffman.
- Influence de l'ordre de l'alphabet ?

Une nouvelle relation d'ordre total entre mots

Définition de $\leq_\omega$

$$u \leq_\omega \Leftrightarrow \begin{cases} exp(u) \leq exp(v) & \text{si } racine(u) = racine(v) \\ u^\omega \leq_{lex} v^\omega & sinon \end{cases}$$

avec u^ω mot infini $uu\cdots$.

Notes sur $\leq_\omega$

- $u \leq_\omega v$ et $v \leq_\omega u \Rightarrow u = v$.
- $u \leq_\omega v \Leftrightarrow pref_k(u^\omega) <_{lex} pref_k(v^\omega)$ avec $k = |u| + |v| - \gcd(|u|, |v|)$ (théorème de *Fine* et *Wilf*).

Exemples de comparaisons sur $\leq_\omega$

1. $abc \leq_\omega abcabc$ ($racine(abc) = racine(abcabc) = abc$ et $exp(abcabc) = 2 < (exp(abc) = 1)$
2. Comparaison de $u = abaab$ et $v = abaababa$
   - $k = |u| + |v| - \text{gcd}(|u|, |v|) = 5 + 8 - 1 = 12$
   - Comparaison lexicographique de :
     - $pref_1 2(u^\omega) = abaababaabab$
     - $pref_1 2(v^\omega) = abaababaabaa$
   - $pref(u^\omega)[1..11] = pref(v^\omega)[1..11]$ et $(pref(u^\omega)[12] = b) > (pref(u^\omega)[12] = a)$
   - $u \leq_\omega v$

Transformée de Burrows-Wheeler étendue

Idée

Étendre la transformée de Burrows-Wheeler à un ensemble fini de mots.
Utiliser $\leq_\omega$ à la place de $\leq_{lex}$ pour le tri.

Transformation : k mots $w_1, \ldots, w_k$ de longueur $l_1, \ldots, l_k$

- Trier les $\sum_i l_i$ conjugués des k mots $\rightarrow$ liste (C)
  - Coût comparaison (pire cas : l_i premiers entre-eux) : $\sum_i l_i$ - k.
  - Utilisation d'un tableau de suffixes sur *pref* $((w_i)^\omega)$
- Mémoriser :
  - (L) : suite des dernière lettre de (C).
  - (I) : suite des indices des mots $w_1, \ldots, w_k$ dans (C).

Permutation π engendrée par la transformée étendue

Permutation π

Permutation π : $(\pi(i) = j \Leftrightarrow F_i = L_j)$:
$F_i z$ conjugué de $w \Rightarrow zF_i = zL_j$ conjugué de w

Décomposition en cycles

- k mots : π se décompose en k cycles distincts.
- Toute permutation (ensemble de cycles) correspond à l'image d'un ensemble de mots $\Rightarrow$ surjectivité de BWT étendu $\Rightarrow$ bijectivité.

Transformée étendue : exemple

Calcul de $T(W)$ pour $W = \{abac, cbab, bca, cba\}$

- Tri de $pref_6(w_i^\omega)$:

Rang $\leq_\omega$	w_i^ω	w_i
1	$abacab\cdots$	$abac$
2	$abcabc\cdots$	abc
3	$abcbab\cdots$	$abcb$
4	$acabac\cdots$	$acab$
5	$acbacb\cdots$	acb
6	$babcba\cdots$	$babc$
7	$bacaba\cdots$	$baca$
8	$bacbac\cdots$	bac
9	$bcabca\cdots$	bca
10	$bcbabc\cdots$	$bcba$
11	$cabaca\cdots$	$caba$
12	$cabcab\cdots$	cab
13	$cbabcb\cdots$	$cbab$
14	$cbacba\cdots$	cba

- $T(W) = (ccbbbcacaaabba, (1, 13, 14, 9))$

Transformée étendue : exemple (2)

- Permutation π associée :

$$\pi = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 \\ 7 & 9 & 10 & 11 & 14 & 3 & 4 & 5 & 12 & 13 & 1 & 2 & 6 & 8 \end{pmatrix}$$

- Cycles de la permutation π :
  - $(1, 7, 4, 11)$: *abac*
  - $(9, 12, 2)$: *bca*
  - $(13, 6, 3, 10)$: *cbab*
  - $(14, 8, 5)$: *cba*

Comparaison de séquences avec la transformée de Burrows-Wheeler

Idée

- Tri des conjugués de v et w avec $\leq_\omega$:
  si un facteur commun s existe dans v et w, les conjugués de u et v de préfixe s sont proches.
- Introduction d'une distance quantifiant le nombre d'alternances de conjugués de u et v dans la liste triée des conjugués.

Distance δ utilisant la transformée étendue de Burrows-Wheeler

Soient $u, v \in A^*$, c_i un des m conjugués de la liste triée :

$$\gamma(c_i) = \begin{cases} U & \text{si} c_i \text{ est un conjugué de } u \\ V & \text{si} c_i \text{ est un conjugué de } v \end{cases}$$

Soit $\Gamma(u, v) = \gamma(c_1)\gamma(c_2)\cdots\gamma(c_m) = U^{n_1}V^{n_2}U^{n_3}\cdots V^{n_k}$ $(\forall i \in \{1, \ldots, k\}, n_i > 0)$:

$$\delta(u, v) = \sum_{i=1}^{k} n_i - 1$$

Généralisation possible de la distance pour deux ensembles de mots.

Propriétés de la distance δ

- δ est symétrique.
- $\delta(u, v) = 0$ ssi $\Gamma(u, v) \in V?(UV) * U?$
  - Par exemple $\delta(u, v) = 0$ si u et v sont conjugués.
    Nécessité d'un tri des conjugués égaux par identificateur de mot.
  - $\delta(u, v) = 0 \not\Rightarrow u =_{conj} v$
    Exemple : $\delta(ace, bdf) = 0$
- δ invariante sur les classes de conjugaison.
- δ ne vérifie pas l'inégalité triangulaire.
  Contre-exemple : $u = abaab, v = babab, w = abbba$:
  $\delta(u, w) + \delta(w, v) = 3 + 2 = 5 \not\geq \delta(u, v) = 6$

Matrice de distances entre mots

Généralisation de Γ pour un ensemble fini de mots

$$\Gamma(u_1, u_2, \ldots, u_k) = \gamma(w_1)\gamma(w_2)\cdots\gamma(w_m)$$

où (w) est la suite des conjugués triés des mots de u
et $\gamma(w_i) = U_j$ si w_i est un conjugué de u_j.

Distance entre paires

$$\Gamma(u_1, u_2, \ldots, u_k) \rightarrow \{\Gamma\left(u_i, u_j | i, j = 1, \ldots, k\right)\}$$

Un unique tri des m conjugués de l'ensemble de k mots pour le calcul d'une matrice de distance.

Matrice de distances entre mots : exemple

$$u_1 = abaab, u_2 = babab, u_3 = abbba$$

Alternances

$$\Gamma(u_1, u_2,_u 3) = U_1 U_3 U_1^2 U_2^2 U_3 U_1 U_3 U_1 U_2^2 U_3 U_2 U_3$$

Matrice de distances

$$\begin{bmatrix} 0 & & \\ 6 & 0 & \\ 2 & 3 & 0 \end{bmatrix}$$

Application à la génomique

- Distance δ utile pour comparaison des séquences biologiques (ADN, ARN, protéines, ...).
- Intérêt pour la réalisation d'arbres philogénétiques à partir de matrices de distances.

Compression de données avec la transformée étendue de Burrows-Wheeler

Idée

- X, Y ensembles de mots, en général pour une fonction de compression C : $|C(X \cup Y)| \leq |C(X)| + |C(Y)|$
- Plutôt que de compresser indépendamment chaque bloc : application de *EBWT* pour l'ensemble des blocs

Quelques résultats de compression avec *EBWT*

Méthodes de compression comparées

1. Application indépendante de BWT sur blocs de taille fixe (ici 64 Ko) $\rightarrow$ MTF $\rightarrow$ Huffman
2. Application de EBWT sur l'ensemble des blocs $\rightarrow$ MTF $\rightarrow$ Huffman.
3. Application de BWT sur le texte (1 bloc) $\rightarrow$ MTF $\rightarrow$ Huffman

Tests sur le corpus Calgary

Fichier	Taille (octets)	Ratio (1)	Ratio (2)	Ratio (3)
bib	111261	0,329	0,308	0,303
paper2	82199	0,365	0,348	0,347
trans	93695	0,265	0,247	0,244

Conclusion

- BWT :
  - Performant pour la compression générique sans perte (*bzip2*).
  - Point délicat : implantation de l'algorithme de tri des conjugués (complexité en temps et espace).
- EBWT :
  - Intéressant pour la constitution de matrices de distance (arbres philogénétiques).
  - Pas d'amélioration spécifique pour la compression.

Références

M. Burrows and D. J. Wheeler.
A block-sorting lossless data compression algorithm.
Technical Report 124, 1994.

Maxime Crochemore, Jacques Désarménien, and Dominique Perrin.
A note on the Burrows-Wheeler transformation.
Theoret. Comput. Sci., 332(1-3):567–572, 2005.

Sabrina Mantaci, Antonio Restivo, G. Rosone, and Marinella Sciortino.
An extension of the Burrows Wheeler transform and applications to sequence comparison and data compression.
page 178. Springer Berlin / Heidelberg, 2005.