A Toolkit for Parikh Matrices

CIAA 2022

L. K. Hutchinson Robert Mercaş D. Reidenbach

30th June 2022

Loughborough University

Compression

Compression

Wikipedia

In signal processing, data compression, source coding, or bit-rate reduction involves encoding information using fewer bits than the original presentation.

Compression

Wikipedia

In signal processing, data compression, source coding, or bit-rate reduction involves encoding information using fewer bits than the original presentation. Compression can be either lossy or lossless.

- Lossless compression reduces bits by identifying and eliminating statistical redundancy (no information is lost in compression).
- Lossy compression reduces bits by removing unnecessary or less important information.

Compression

Wikipedia

In signal processing, data compression, source coding, or bit-rate reduction involves encoding information using fewer bits than the original presentation. Compression can be either lossy or lossless.

- Lossless compression reduces bits by identifying and eliminating statistical redundancy (no information is lost in compression).
- Lossy compression reduces bits by removing unnecessary or less important information.

Lossless techniques

- Lempel-Ziv factorization LZ77 [Lempel and Ziv, 1976, Lempel and Ziv, 1977],
- Straight Line Programs [Kieffer and Yang, 2000],
- run-length Burrows-Wheeler transform [Burrows and Wheeler, 1994],
- Compact Directed Acyclic Word Graph [Blumer et al., 1987], [Crochemore and Vérin, 1997]

Parikh matrices

- Histograms
- Parikh vectors [Parikh, 1966] represent a type histograms specific to the analysis of sequences of symbols
- compression is easily computed and guaranteed to be logarithmic in the size of the word

Parikh matrices

- Histograms
- Parikh vectors [Parikh, 1966] represent a type histograms specific to the analysis of sequences of symbols
- compression is easily computed and guaranteed to be logarithmic in the size of the word, but ambiguous (multiple words share a Parikh vector)

Parikh matrices

- Histograms
- Parikh vectors [Parikh, 1966] represent a type histograms specific to the analysis of sequences of symbols
- compression is easily computed and guaranteed to be logarithmic in the size of the word, but ambiguous (multiple words share a Parikh vector)

- Parikh matrices [Mateescu et al., 2000] are a refinement of the vectors
- same asymptotic compactness as Parikh vector and significantly smaller number of words associated to it
- but it does not normally remove ambiguity (entirely)

Example

Definition

Let $w \in \Sigma_k^*$. The *Parikh matrix*, denoted $\Psi(w)$, that w is associated with has size $(k+1) \times (k+1)$. The diagonal of the matrix is populated with 1's and all elements below it are 0. The count of all subwords that consist of consecutive letters in Σ_k and are of length n in the word are found on the n-diagonal, for $1 \leq n \leq k$, i. e., the Parikh vector is found on the 1-diagonal.

Example

Definition

Let $w \in \Sigma_k^*$. The *Parikh matrix*, denoted $\Psi(w)$, that w is associated with has size $(k+1) \times (k+1)$. The diagonal of the matrix is populated with 1's and all elements below it are 0. The count of all subwords that consist of consecutive letters in Σ_k and are of length n in the word are found on the n-diagonal, for $1 \leq n \leq k$, i. e., the Parikh vector is found on the 1-diagonal.

For $\Sigma_3 = \{a < b < c\}$ we consider all of the factors of abc.

Thus, for $w \in \Sigma_3$ the Parikh matrix is of size 4×4 and we count all occurrences of the above factors as (scattered) subwords:

$$\Psi(aabcbaa) = \begin{pmatrix} 1 & |w|_a & |w|_{ab} & |w|_{abc} \\ 0 & 1 & |w|_b & |w|_{bc} \\ 0 & 0 & 1 & |w|_c \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 4 & 4 & 2 \\ 0 & 1 & 2 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Example

Consider the word $w = abca$ defined over the alphabet $\Sigma_3 = \{a < b < c\}$.

Example

Consider the word $w = abca$ defined over the alphabet $\Sigma_3 = \{a < b < c\}$. The Parikh matrix is of size 4×4 and we have:

$$\Psi(abca) = \begin{pmatrix} 1 & 2 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Example

Consider the word $w = abca$ defined over the alphabet $\Sigma_3 = \{a < b < c\}$. The Parikh matrix is of size 4×4 and we have:

$$\Psi(abca) = \begin{pmatrix} 1 & 2 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$= \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Example

Consider the word $w = abca$ defined over the alphabet $\Sigma_3 = \{a < b < c\}$. The Parikh matrix is of size 4×4 and we have:

$$\begin{aligned}\Psi(abca) &= \begin{pmatrix} 1 & 2 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \\ &= \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \\ &= \Psi(a) \cdot \Psi(b) \cdot \Psi(c) \cdot \Psi(a)\end{aligned}$$

Example

Consider the word $w = abca$ defined over the alphabet $\Sigma_3 = \{a < b < c\}$. The Parikh matrix is of size 4×4 and we have:

$$\begin{aligned}\Psi(abca) &= \begin{pmatrix} 1 & 2 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \\ &= \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \\ &= \Psi(a) \cdot \Psi(b) \cdot \Psi(c) \cdot \Psi(a)\end{aligned}$$

$$\Psi(abca) = \left\langle \begin{matrix} 2 & 1 & 1 \\ & 1 & 1 \\ & & 1 \end{matrix} \right\rangle$$

A big mess

A big mess

Parikh vectors are not injective.

$\phi(abbcabac) = (3, 3, 2) = \phi(aaabbbcc)$

A big mess

Parikh vectors are not injective.

$$\phi(abbcabac) = (3,3,2) = \phi(aaabbbcc)$$

Parikh matrices are not injective

$$\Psi(abca) = \left\langle \begin{smallmatrix} 2 & 1 & 1 \\ & 1 & 1 \\ & & 1 \end{smallmatrix} \right\rangle = \Psi(abac)$$

A big mess

Parikh vectors are not injective.

$$\phi(abbcabac) = (3,3,2) = \phi(aaabbbcc)$$

Parikh matrices are not injective, but also not surjective.

$$\Psi(abca) = \left\langle \begin{smallmatrix} 2 & 1 & 1 \\ & 1 & 1 \\ & & 1 \end{smallmatrix} \right\rangle = \Psi(abac)$$

there exists no w with $\Psi(w) = \left\langle \begin{smallmatrix} 1 & 0 & 1 \\ & 1 & 0 \\ & & 1 \end{smallmatrix} \right\rangle$

A big mess

Parikh vectors are not injective.

$$\phi(abbcabac) = (3,3,2) = \phi(aaabbbcc)$$

Parikh matrices are not injective, but also not surjective.

$$\Psi(abca) = \left\langle \begin{smallmatrix} 2 & 1 & 1 \\ & 1 & 1 \\ & & 1 \end{smallmatrix} \right\rangle = \Psi(abac)$$

there exists no w with $\Psi(w) = \left\langle \begin{smallmatrix} 1 & 0 & 1 \\ & 1 & 0 \\ & & 1 \end{smallmatrix} \right\rangle$

Note

If two words share a Parikh matrix, we say that they are *amiable*.

History

History

- alternative to the Parikh matrix concept that would make a mapping from a word injective, or less ambiguous in general [Şerbănuţă, 2004], [Egecioglu, 2004], [Egecioglu and Ibarra, 2004], [Egecioglu and Ibarra, 2007], [Alazemi and Černý, 2011], [Alazemi and Černý, 2013], [Bera and Mahalingam, 2016],

History

- alternative to the Parikh matrix concept that would make a mapping from a word injective, or less ambiguous in general [Şerbănuţă, 2004], [Egecioglu, 2004], [Egecioglu and Ibarra, 2004], [Egecioglu and Ibarra, 2007], [Alazemi and Černý, 2011], [Alazemi and Černý, 2013], [Bera and Mahalingam, 2016],
- for Parikh matrices the focus was on investigating uniqueness on binary [Atanasiu et al., 2001], [Atanasiu et al., 2002], [Atanasiu et al., 2008], [Salomaa and Yu, 2010], [Atanasiu and Teh, 2016]

History

- alternative to the Parikh matrix concept that would make a mapping from a word injective, or less ambiguous in general [Şerbănuţă, 2004], [Egecioglu, 2004], [Egecioglu and Ibarra, 2004], [Egecioglu and Ibarra, 2007], [Alazemi and Černý, 2011], [Alazemi and Černý, 2013], [Bera and Mahalingam, 2016],
- for Parikh matrices the focus was on investigating uniqueness on binary [Atanasiu et al., 2001], [Atanasiu et al., 2002], [Atanasiu et al., 2008], [Salomaa and Yu, 2010], [Atanasiu and Teh, 2016] and ternary [Şerbănuţă, 2009], [Mahalingam and Subramanian, 2012], [Atanasiu, 2014], [Poovanandran and Chean Teh, 2017]

History

- alternative to the Parikh matrix concept that would make a mapping from a word injective, or less ambiguous in general [Şerbănuţă, 2004], [Egecioglu, 2004], [Egecioglu and Ibarra, 2004], [Egecioglu and Ibarra, 2007], [Alazemi and Černý, 2011], [Alazemi and Černý, 2013], [Bera and Mahalingam, 2016],
- for Parikh matrices the focus was on investigating uniqueness on binary [Atanasiu et al., 2001], [Atanasiu et al., 2002], [Atanasiu et al., 2008], [Salomaa and Yu, 2010], [Atanasiu and Teh, 2016] and ternary [Şerbănuţă, 2009], [Mahalingam and Subramanian, 2012], [Atanasiu, 2014], [Poovanandran and Chean Teh, 2017]
- *reducing* ambiguity, investigation was based on either gathering more information about the specific word by altering the order of the alphabet (dual order) or by considering the reverse image of the word [Mateescu et al., 2000], [Atanasiu et al., 2002]

History

- alternative to the Parikh matrix concept that would make a mapping from a word injective, or less ambiguous in general [Şerbănuţă, 2004], [Egecioglu, 2004], [Egecioglu and Ibarra, 2004], [Egecioglu and Ibarra, 2007], [Alazemi and Černý, 2011], [Alazemi and Černý, 2013], [Bera and Mahalingam, 2016],
- for Parikh matrices the focus was on investigating uniqueness on binary [Atanasiu et al., 2001], [Atanasiu et al., 2002], [Atanasiu et al., 2008], [Salomaa and Yu, 2010], [Atanasiu and Teh, 2016] and ternary [Şerbănuţă, 2009], [Mahalingam and Subramanian, 2012], [Atanasiu, 2014], [Poovanandran and Chean Teh, 2017]
- *reducing* ambiguity, investigation was based on either gathering more information about the specific word by altering the order of the alphabet (dual order) or by considering the reverse image of the word [Mateescu et al., 2000], [Atanasiu et al., 2002]
- *additional* matrices used to remove ambiguity, by altering the word itself: projection morphisms and conjugates [Dick et al., 2021]

$\mathbb{P}$-Parikh Matrix

Definition

Let $S \subset \Sigma_n$ such that $S = \{a_{k_1}, a_{k_2}, \ldots a_{k_m}\}$. We define the $\mathbb{P}$-Parikh matrix of the word w with respect to S as $\Psi_S(\pi_S(w))$, where the morphism $\pi : \Sigma_n^* \to \Sigma_m^*$ is defined as

$$\pi_S(a_i) := \begin{cases} a_i & : a_i \in S, \\ \varepsilon & : a_i \notin S. \end{cases}$$

$\mathbb{P}$-Parikh Matrix

Definition

Let $S \subset \Sigma_n$ such that $S = \{a_{k_1}, a_{k_2}, \ldots a_{k_m}\}$. We define the $\mathbb{P}$-Parikh matrix of the word w with respect to S as $\Psi_S(\pi_S(w))$, where the morphism $\pi : \Sigma_n^* \to \Sigma_m^*$ is defined as

$$\pi_S(a_i) := \begin{cases} a_i & : a_i \in S, \\ \varepsilon & : a_i \notin S. \end{cases}$$

Example

Let $\Sigma_5 = \{a, b, c, d, e\}$, $S = \{a, d, e\}$, and $w = bacbebda$.

$\mathbb{P}$-Parikh Matrix

Definition

Let $S \subset \Sigma_n$ such that $S = \{a_{k_1}, a_{k_2}, \ldots a_{k_m}\}$. We define the $\mathbb{P}$-Parikh matrix of the word w with respect to S as $\Psi_S(\pi_S(w))$, where the morphism $\pi : \Sigma_n^* \to \Sigma_m^*$ is defined as

$$\pi_S(a_i) := \begin{cases} a_i & : a_i \in S, \\ \varepsilon & : a_i \notin S. \end{cases}$$

Example

Let $\Sigma_5 = \{a, b, c, d, e\}$, $S = \{a, d, e\}$, and $w = bacbebda$.

$$\pi_S(a) = a, \qquad \pi_S(d) = b, \qquad \pi_S(e) = c$$

$\mathbb{P}$-Parikh Matrix

Definition

Let $S \subset \Sigma_n$ such that $S = \{a_{k_1}, a_{k_2}, \ldots a_{k_m}\}$. We define the $\mathbb{P}$-Parikh matrix of the word w with respect to S as $\Psi_S(\pi_S(w))$, where the morphism $\pi : \Sigma_n^* \to \Sigma_m^*$ is defined as

$$\pi_S(a_i) := \begin{cases} a_i & : a_i \in S, \\ \varepsilon & : a_i \notin S. \end{cases}$$

Example

Let $\Sigma_5 = \{a, b, c, d, e\}$, $S = \{a, d, e\}$, and $w = bacbebda$.

$$\pi_S(a) = a, \qquad \pi_S(d) = b, \qquad \pi_S(e) = c$$

$$\pi_S(w) = \pi_S(b) \cdot \pi_S(a) \cdot \pi_S(c) \cdot \pi_S(b) \cdot \pi_S(e) \cdot \pi_S(b) \cdot \pi_S(d) \cdot \pi_S(a)$$

$\mathbb{P}$-Parikh Matrix

Definition

Let $S \subset \Sigma_n$ such that $S = \{a_{k_1}, a_{k_2}, \ldots a_{k_m}\}$. We define the $\mathbb{P}$-Parikh matrix of the word w with respect to S as $\Psi_S(\pi_S(w))$, where the morphism $\pi : \Sigma_n^* \to \Sigma_m^*$ is defined as

$$\pi_S(a_i) := \begin{cases} a_i & : a_i \in S, \\ \varepsilon & : a_i \notin S. \end{cases}$$

Example

Let $\Sigma_5 = \{a, b, c, d, e\}$, $S = \{a, d, e\}$, and $w = bacbebda$.

$$\pi_S(a) = a, \qquad \pi_S(d) = b, \qquad \pi_S(e) = c$$

$$\begin{aligned} \pi_S(w) &= \pi_S(b) \cdot \pi_S(a) \cdot \pi_S(c) \cdot \pi_S(b) \cdot \pi_S(e) \cdot \pi_S(b) \cdot \pi_S(d) \cdot \pi_S(a) \\ &= \varepsilon\ a\ \varepsilon\ \varepsilon\ c\ \varepsilon\ b\ a = acba \end{aligned}$$

$\mathbb{P}$-Parikh Matrix

Definition

Let $S \subset \Sigma_n$ such that $S = \{a_{k_1}, a_{k_2}, \ldots a_{k_m}\}$. We define the $\mathbb{P}$-Parikh matrix of the word w with respect to S as $\Psi_S(\pi_S(w))$, where the morphism $\pi : \Sigma_n^* \to \Sigma_m^*$ is defined as

$$\pi_S(a_i) := \begin{cases} a_i & : a_i \in S, \\ \varepsilon & : a_i \notin S. \end{cases}$$

Example

Let $\Sigma_5 = \{a, b, c, d, e\}$, $S = \{a, d, e\}$, and $w = bacbebda$.

$$\pi_S(a) = a, \qquad \pi_S(d) = b, \qquad \pi_S(e) = c$$

$$\begin{aligned} \pi_S(w) &= \pi_S(b) \cdot \pi_S(a) \cdot \pi_S(c) \cdot \pi_S(b) \cdot \pi_S(e) \cdot \pi_S(b) \cdot \pi_S(d) \cdot \pi_S(a) \\ &= \varepsilon\ a\ \varepsilon\ \varepsilon\ c\ \varepsilon\ b\ a = acba \end{aligned}$$

$$\Psi_S(bacbebda) = \Psi_S(\pi_{\{a,d,e\}}(bacbebda)) = \Psi_S(acba) = \begin{pmatrix} 2 & 1 & 0 \\ & 1 & 0 \\ & & 1 \end{pmatrix}$$

When do $\mathbb{P}$-Parikh Matrices reduce ambiguity, and when not?

Example

Let $w = abd$, and $w' = adb$ in Σ_4. Then w and w' are amiable. Note that w contains the factor bd, and we choose $S = \{b, d\}$.

When do $\mathbb{P}$-Parikh Matrices reduce ambiguity, and when not?

Example

Let $w = abd$, and $w' = adb$ in Σ_4. Then w and w' are amiable. Note that w contains the factor bd, and we choose $S = \{b, d\}$.

$$\pi_S(abd) = bd$$

When do $\mathbb{P}$-Parikh Matrices reduce ambiguity, and when not?

Example

Let $w = abd$, and $w' = adb$ in Σ_4. Then w and w' are amiable. Note that w contains the factor bd, and we choose $S = \{b, d\}$.

$$\pi_S(abd) = bd \implies \Psi_S(w) = \Psi(\pi_S(abd)) = \Psi(bd) = \left\langle \begin{smallmatrix} 1 & 1 & 1 \\ & 1 & 1 \\ & & 1 \end{smallmatrix} \right\rangle$$

When do $\mathbb{P}$-Parikh Matrices reduce ambiguity, and when not?

Example

Let $w = abd$, and $w' = adb$ in Σ_4. Then w and w' are amiable. Note that w contains the factor bd, and we choose $S = \{b, d\}$.

$$\pi_S(abd) = bd \implies \Psi_S(w) = \Psi(\pi_S(abd)) = \Psi(bd) = \left\langle \begin{smallmatrix} 1 & 1 & 1 \\ & 1 & 1 \\ & & 1 \end{smallmatrix} \right\rangle$$

$$\pi_S(adb) = db$$

When do $\mathbb{P}$-Parikh Matrices reduce ambiguity, and when not?

Example

Let $w = abd$, and $w' = adb$ in Σ_4. Then w and w' are amiable. Note that w contains the factor bd, and we choose $S = \{b, d\}$.

$$\pi_S(abd) = bd \implies \Psi_S(w) = \Psi(\pi_S(abd)) = \Psi(bd) = \begin{pmatrix} 1 & 1 & 1 \\ & 1 & 1 \\ & & 1 \end{pmatrix}$$

$$\pi_S(adb) = db \implies \Psi_S(w') = \Psi(\pi_S(adb)) = \Psi(db) = \begin{pmatrix} 1 & 1 & 0 \\ & 1 & 0 \\ & & 1 \end{pmatrix}$$

When do $\mathbb{P}$-Parikh Matrices reduce ambiguity, and when not?

Example

Let $w = abd$, and $w' = adb$ in Σ_4. Then w and w' are amiable. Note that w contains the factor bd, and we choose $S = \{b, d\}$.

$$\pi_S(abd) = bd \implies \Psi_S(w) = \Psi(\pi_S(abd)) = \Psi(bd) = \left\langle \begin{smallmatrix} 1 & 1 & 1 \\ & 1 & 1 \\ & & 1 \end{smallmatrix} \right\rangle$$

$$\pi_S(adb) = db \implies \Psi_S(w') = \Psi(\pi_S(adb)) = \Psi(db) = \left\langle \begin{smallmatrix} 1 & 1 & 0 \\ & 1 & 0 \\ & & 1 \end{smallmatrix} \right\rangle$$

Example

Consider $w = acbbca$ and $w' = cabbac$. Then $\Psi(w) = \Psi(w') = \left\langle \begin{smallmatrix} 2 & 2 & 2 \\ & 2 & 2 \\ & & 2 \end{smallmatrix} \right\rangle$.

When do $\mathbb{P}$-Parikh Matrices reduce ambiguity, and when not?

Example

Let $w = abd$, and $w' = adb$ in Σ_4. Then w and w' are amiable. Note that w contains the factor bd, and we choose $S = \{b, d\}$.

$$\pi_S(abd) = bd \implies \Psi_S(w) = \Psi(\pi_S(abd)) = \Psi(bd) = \begin{pmatrix} 1 & 1 & 1 \\ & 1 & 1 \\ & & 1 \end{pmatrix}$$

$$\pi_S(adb) = db \implies \Psi_S(w') = \Psi(\pi_S(adb)) = \Psi(db) = \begin{pmatrix} 1 & 1 & 0 \\ & 1 & 0 \\ & & 1 \end{pmatrix}$$

Example

Consider $w = acbbca$ and $w' = cabbac$. Then $\Psi(w) = \Psi(w') = \begin{pmatrix} 2 & 2 & 2 \\ & 2 & 2 \\ & & 2 \end{pmatrix}$.

Now let us find $\Psi_S(w)$ and $\Psi_S(w')$ for all $S \subseteq \Sigma_3$.

When do $\mathbb{P}$-Parikh Matrices reduce ambiguity, and when not?

Example

Let $w = abd$, and $w' = adb$ in Σ_4. Then w and w' are amiable. Note that w contains the factor bd, and we choose $S = \{b, d\}$.

$$\pi_S(abd) = bd \implies \Psi_S(w) = \Psi(\pi_S(abd)) = \Psi(bd) = \left\langle \begin{smallmatrix} 1 & 1 & 1 \\ & 1 & 1 \\ & & 1 \end{smallmatrix} \right\rangle$$

$$\pi_S(adb) = db \implies \Psi_S(w') = \Psi(\pi_S(adb)) = \Psi(db) = \left\langle \begin{smallmatrix} 1 & 1 & 0 \\ & 1 & 0 \\ & & 1 \end{smallmatrix} \right\rangle$$

Example

Consider $w = acbbca$ and $w' = cabbac$. Then $\Psi(w) = \Psi(w') = \left\langle \begin{smallmatrix} 2 & 2 & 2 \\ & 2 & 2 \\ & & 2 \end{smallmatrix} \right\rangle$.

Now let us find $\Psi_S(w)$ and $\Psi_S(w')$ for all $S \subseteq \Sigma_3$.

$$S = \{a\}, S = \{b\}, S = \{c\} : \ \Psi_S(w) = \left\langle \begin{smallmatrix} 1 & 2 \\ & 1 \end{smallmatrix} \right\rangle = \Psi_S(w')$$

When do $\mathbb{P}$-Parikh Matrices reduce ambiguity, and when not?

Example

Let $w = abd$, and $w' = adb$ in Σ_4. Then w and w' are amiable. Note that w contains the factor bd, and we choose $S = \{b, d\}$.

$$\pi_S(abd) = bd \implies \Psi_S(w) = \Psi(\pi_S(abd)) = \Psi(bd) = \left\langle \begin{smallmatrix} 1 & 1 & 1 \\ & 1 & 1 \\ & & 1 \end{smallmatrix} \right\rangle$$

$$\pi_S(adb) = db \implies \Psi_S(w') = \Psi(\pi_S(adb)) = \Psi(db) = \left\langle \begin{smallmatrix} 1 & 1 & 0 \\ & 1 & 0 \\ & & 1 \end{smallmatrix} \right\rangle$$

Example

Consider $w = acbbca$ and $w' = cabbac$. Then $\Psi(w) = \Psi(w') = \left\langle \begin{smallmatrix} 2 & 2 & 2 \\ & 2 & 2 \\ & & 2 \end{smallmatrix} \right\rangle$.

Now let us find $\Psi_S(w)$ and $\Psi_S(w')$ for all $S \subseteq \Sigma_3$.

$$S = \{a\}, S = \{b\}, S = \{c\} : \ \Psi_S(w) = \left\langle \begin{smallmatrix} 1 & 2 \\ & 1 \end{smallmatrix} \right\rangle = \Psi_S(w')$$

$$S = \{a, b\}, S = \{a, c\} : \ \Psi_S(w) = \left\langle \begin{smallmatrix} 1 & 2 & 2 \\ & 1 & 2 \\ & & 1 \end{smallmatrix} \right\rangle = \Psi_S(w')$$

When do $\mathbb{P}$-Parikh Matrices reduce ambiguity, and when not?

Example

Let $w = abd$, and $w' = adb$ in Σ_4. Then w and w' are amiable. Note that w contains the factor bd, and we choose $S = \{b, d\}$.

$$\pi_S(abd) = bd \implies \Psi_S(w) = \Psi(\pi_S(abd)) = \Psi(bd) = \left\langle \begin{smallmatrix} 1 & 1 & 1 \\ & 1 & 1 \\ & & 1 \end{smallmatrix} \right\rangle$$

$$\pi_S(adb) = db \implies \Psi_S(w') = \Psi(\pi_S(adb)) = \Psi(db) = \left\langle \begin{smallmatrix} 1 & 1 & 0 \\ & 1 & 0 \\ & & 1 \end{smallmatrix} \right\rangle$$

Example

Consider $w = acbbca$ and $w' = cabbac$. Then $\Psi(w) = \Psi(w') = \left\langle \begin{smallmatrix} 2 & 2 & 2 \\ & 2 & 2 \\ & & 2 \end{smallmatrix} \right\rangle$.

Now let us find $\Psi_S(w)$ and $\Psi_S(w')$ for all $S \subseteq \Sigma_3$.

$$S = \{a\}, S = \{b\}, S = \{c\} : \ \Psi_S(w) = \left\langle \begin{smallmatrix} 1 & 2 \\ & 1 \end{smallmatrix} \right\rangle = \Psi_S(w')$$

$$S = \{a, b\}, S = \{a, c\} : \ \Psi_S(w) = \left\langle \begin{smallmatrix} 1 & 2 & 2 \\ & 1 & 2 \\ & & 1 \end{smallmatrix} \right\rangle = \Psi_S(w')$$

$$S = \{b, c\} : \ \pi_S(w) = cbbc = \pi_S(w')$$

$\mathbb{L}$-Parikh Matrices

Definition

The *Lyndon conjugate* of a word w, denoted $L(w)$, is the conjugate that is lexicographically smallest based on the order on the alphabet.

$\mathbb{L}$-Parikh Matrices

Definition

The *Lyndon conjugate* of a word w, denoted $L(w)$, is the conjugate that is lexicographically smallest based on the order on the alphabet.

Definition

Given a word w, we define its $\mathbb{L}$-*Parikh matrix* as $\Psi_{\ell ex}(x) = \Psi(L(w))$.

$\mathbb{L}$-Parikh Matrices

Definition

The *Lyndon conjugate* of a word w, denoted $L(w)$, is the conjugate that is lexicographically smallest based on the order on the alphabet.

Definition

Given a word w, we define its $\mathbb{L}$-*Parikh matrix* as $\Psi_{\ell ex}(x) = \Psi(L(w))$.

Example

Consider the words $w = babbbaa$, $u = bbababa$ and $v = bbbaaab$ with $\Psi(w) = \Psi(u) = \Psi(v)$.

$\mathbb{L}$-Parikh Matrices

Definition

The *Lyndon conjugate* of a word w, denoted $L(w)$, is the conjugate that is lexicographically smallest based on the order on the alphabet.

Definition

Given a word w, we define its $\mathbb{L}$-*Parikh matrix* as $\Psi_{\ell ex}(x) = \Psi(L(w))$.

Example

Consider the words $w = babbbaa$, $u = bbababa$ and $v = bbbaaab$ with $\Psi(w) = \Psi(u) = \Psi(v)$.

$$L(w) = aababbb, \Psi_{\ell ex}(w) = \left\langle \begin{smallmatrix} 3 & 11 \\ & 4 \end{smallmatrix} \right\rangle$$

$\mathbb{L}$-Parikh Matrices

Definition

The *Lyndon conjugate* of a word w, denoted $L(w)$, is the conjugate that is lexicographically smallest based on the order on the alphabet.

Definition

Given a word w, we define its $\mathbb{L}$-*Parikh matrix* as $\Psi_{\ell ex}(x) = \Psi(L(w))$.

Example

Consider the words $w = babbbaa$, $u = bbababa$ and $v = bbbaaab$ with $\Psi(w) = \Psi(u) = \Psi(v)$.

$$L(w) = aababbb, \Psi_{\ell ex}(w) = \left\langle \begin{smallmatrix} 3 & 11 \\ & 4 \end{smallmatrix} \right\rangle$$
$$L(u) = abababb, \Psi_{\ell ex}(u) = \left\langle \begin{smallmatrix} 3 & 9 \\ & 4 \end{smallmatrix} \right\rangle$$

$\mathbb{L}$-Parikh Matrices

Definition

The *Lyndon conjugate* of a word w, denoted $L(w)$, is the conjugate that is lexicographically smallest based on the order on the alphabet.

Definition

Given a word w, we define its $\mathbb{L}$-*Parikh matrix* as $\Psi_{\ell ex}(x) = \Psi(L(w))$.

Example

Consider the words $w = babbbaa$, $u = bbababa$ and $v = bbbaaab$ with $\Psi(w) = \Psi(u) = \Psi(v)$.

$$L(w) = aababbb, \Psi_{\ell ex}(w) = \left\langle \begin{smallmatrix} 3 & 11 \\ & 4 \end{smallmatrix} \right\rangle$$
$$L(u) = abababb, \Psi_{\ell ex}(u) = \left\langle \begin{smallmatrix} 3 & 9 \\ & 4 \end{smallmatrix} \right\rangle$$
$$L(v) = aaabbbb, \Psi_{\ell ex}(v) = \left\langle \begin{smallmatrix} 3 & 12 \\ & 4 \end{smallmatrix} \right\rangle$$

When do $\mathbb{L}$-Parikh matrices NOT reduce ambiguity?

Theorem

For the binary alphabet, the ambiguity of a Parikh matrix is not reduced by $\mathbb{L}$-Parikh matrices if and only if any of the words it describes meet at least one of the following criteria:

- $w \in \{aabb, ababbb, aababb, aabbab, aaabab, bbabbaaa, bbbaabaa\}$
- $w = a^*vb^*$ *and for* $n = |v|_{ba}$ *we have that* $|v|_a = 2n$ *and* $|v|_b = n + 1$

When do $\mathbb{L}$-Parikh matrices NOT reduce ambiguity?

Theorem

For the binary alphabet, the ambiguity of a Parikh matrix is not reduced by $\mathbb{L}$-Parikh matrices if and only if any of the words it describes meet at least one of the following criteria:

- $w \in \{aabb, ababbb, aababb, aabbab, aaabab, bbabbaaa, bbbaabaa\}$
- $w = a^*vb^*$ *and for* $n = |v|_{ba}$ *we have that* $|v|_a = 2n$ *and* $|v|_b = n + 1$

When do $\mathbb{L}$-Parikh matrices NOT reduce ambiguity?

Theorem

For the binary alphabet, the ambiguity of a Parikh matrix is not reduced by $\mathbb{L}$-Parikh matrices if and only if any of the words it describes meet at least one of the following criteria:

- $w \in \{aabb, ababbb, aababb, aabbab, aaabab, bbabbaaa, bbbaabaa\}$
- $w = a^*vb^*$ *and for* $n = |v|_{ba}$ *we have that* $|v|_a = 2n$ *and* $|v|_b = n + 1$

When do $\mathbb{L}$-Parikh matrices NOT reduce ambiguity?

Theorem

For the binary alphabet, the ambiguity of a Parikh matrix is not reduced by $\mathbb{L}$-Parikh matrices if and only if any of the words it describes meet at least one of the following criteria:

- $w \in \{aabb, ababbb, aababb, aabbab, aaabab, bbabbaaa, bbbaabaa\}$
- $w = a^*vb^*$ *and for* $n = |v|_{ba}$ *we have that* $|v|_a = 2n$ *and* $|v|_b = n + 1$

MIT Licence

www.github.com/LHutch1/Parikh-Matrices-Toolkit

Binary words

If $w = aab\,aab\,aab\,aab\,aab\,aab\,aab\,aab$, with $\Psi(w) = \left\langle \begin{smallmatrix} 16 & 62 \\ 0 & 8 \end{smallmatrix} \right\rangle$,

then $|\Psi(w)| = 15,744$

Binary words

If $w = aab\,aab\,aab\,aab\,aab\,aab\,aab\,aab$, with $\Psi(w) = \left\langle \begin{smallmatrix} 16 & 62 \\ 0 & 8 \end{smallmatrix} \right\rangle$,

then $|\Psi(w)| = 15,744$ *and* $|\Psi_{\ell ex}(w)| = 1,215.$

Binary words

If $w = aab\,aab\,aab\,aab\,aab\,aab\,aab\,aab$, with $\Psi(w) = \left\langle \begin{smallmatrix} 16 & 62 \\ 0 & 8 \end{smallmatrix} \right\rangle$,

then $|\Psi(w)| = 15,744$ *and* $|\Psi_{\ell ex}(w)| = 1,215.$

If $w = abaababbabbaaabbbababaaba$, with $\Psi(w) = \left\langle \begin{smallmatrix} 13 & 69 \\ 0 & 11 \end{smallmatrix} \right\rangle$,

then $|\Psi(w)| = 56,031$

Binary words

If $w = aab\,aab\,aab\,aab\,aab\,aab\,aab\,aab$, with $\Psi(w) = \left\langle \begin{smallmatrix} 16 & 62 \\ 0 & 8 \end{smallmatrix} \right\rangle$,

then $|\Psi(w)| = 15,744$ *and* $|\Psi_{\ell ex}(w)| = 1,215.$

If $w = abaababbabbaaabbbababaaba$, with $\Psi(w) = \left\langle \begin{smallmatrix} 13 & 69 \\ 0 & 11 \end{smallmatrix} \right\rangle$,

then $|\Psi(w)| = 56,031$ *and* $|\Psi_{\ell ex}(w)| = 2,592.$

Larger alphabets

If $w = aabcd\ aabcd\ aabc$, with $\Psi(w) = \left\langle \begin{matrix} 6 & 12 & 20 & 10 \\ & 3 & 6 & 4 \\ & & 3 & 3 \\ & & & 2 \end{matrix} \right\rangle$, then

$$|\Psi(w)|=102, \quad |\Psi_{\{b,d\}}(w)|=36, \quad |\Psi_{\{a,c\}}(w)|=16, \quad |\Psi_{\{a,d\}}(w)|=1.$$

Larger alphabets

If $w = aabcd\ aabcd\ aabc$, with $\Psi(w) = \left\langle \begin{smallmatrix} 6 & 12 & 20 & 10 \\ & 3 & 6 & 4 \\ & & 3 & 3 \\ & & & 2 \end{smallmatrix} \right\rangle$, then

$$|\Psi(w)|=102, \quad |\Psi_{\{b,d\}}(w)|=36, \quad |\Psi_{\{a,c\}}(w)|=16, \quad |\Psi_{\{a,d\}}(w)|=1.$$

If $w = aabcb\ dacab\ dac$, with $\Psi(w) = \left\langle \begin{smallmatrix} 5 & 8 & 14 & 8 \\ & 3 & 6 & 4 \\ & & 3 & 3 \\ & & & 2 \end{smallmatrix} \right\rangle$, then

$$|\Psi(w)|=59, \quad |\Psi_{\{a,d\}}(w)|=26, \quad |\Psi_{\{b,d\}}(w)|=18, \quad |\Psi_{\{a,c\}}(w)|=20.$$

Larger alphabets

If $w = aabcd\ aabcd\ aabc$, with $\Psi(w) = \left\langle \begin{smallmatrix} 6 & 12 & 20 & 10 \\ & 3 & 6 & 4 \\ & & 3 & 3 \\ & & & 2 \end{smallmatrix} \right\rangle$, then

$$|\Psi(w)|=102, \quad |\Psi_{\{b,d\}}(w)|=36, \quad |\Psi_{\{a,c\}}(w)|=16, \quad |\Psi_{\{a,d\}}(w)|=1.$$

If $w = aabcb\ dacab\ dac$, with $\Psi(w) = \left\langle \begin{smallmatrix} 5 & 8 & 14 & 8 \\ & 3 & 6 & 4 \\ & & 3 & 3 \\ & & & 2 \end{smallmatrix} \right\rangle$, then

$$|\Psi(w)|=59, \quad |\Psi_{\{a,d\}}(w)|=26, \quad |\Psi_{\{b,d\}}(w)|=18, \quad |\Psi_{\{a,c\}}(w)|=20.$$

$$|\Psi_{\{a,d\}}(w) \bigcap \Psi_{\{b,d\}}(w)| = 5 \qquad |\Psi_{\{b,d\}}(w) \bigcap \Psi_{\{a,c\}}(w)| = 7$$

$$|\Psi_{\{a,d\}}(w) \bigcap \Psi_{\{a,c\}}(w)| = 9$$

Larger alphabets

If $w = aabcd\ aabcd\ aabc$, with $\Psi(w) = \left\langle \begin{smallmatrix} 6 & 12 & 20 & 10 \\ & 3 & 6 & 4 \\ & & 3 & 3 \\ & & & 2 \end{smallmatrix} \right\rangle$, then

$$|\Psi(w)|=102, \quad |\Psi_{\{b,d\}}(w)|=36, \quad |\Psi_{\{a,c\}}(w)|=16, \quad |\Psi_{\{a,d\}}(w)|=1.$$

If $w = aabcb\ dacab\ dac$, with $\Psi(w) = \left\langle \begin{smallmatrix} 5 & 8 & 14 & 8 \\ & 3 & 6 & 4 \\ & & 3 & 3 \\ & & & 2 \end{smallmatrix} \right\rangle$, then

$$|\Psi(w)|=59, \quad |\Psi_{\{a,d\}}(w)|=26, \quad |\Psi_{\{b,d\}}(w)|=18, \quad |\Psi_{\{a,c\}}(w)|=20.$$

$$|\Psi_{\{a,d\}}(w) \bigcap \Psi_{\{b,d\}}(w)| = 5 \qquad |\Psi_{\{b,d\}}(w) \bigcap \Psi_{\{a,c\}}(w)| = 7$$

$$|\Psi_{\{a,d\}}(w) \bigcap \Psi_{\{a,c\}}(w)| = 9 \qquad |\Psi_{\{a,d\}}(w) \bigcap \Psi_{\{a,c\}}(w) \bigcap \Psi_{\{b,d\}}(w)| = 2$$

$$\Psi(aabcbdaacbdca) = \Psi(aabcbdacabdac)$$

Thanks to the referees!

- matrix multiplication (in our case) can be done in quadratic time!
- shear matrices (matrices de transvection) are the answer (generators for the special linear group $SL_k(\mathbb{Z})$.
- the composition to the right of a matrix A with a shear matrix that has a value c at position (i, j) implies adding to the jth column of A the cth multiple of the column i of A.
- in our case adding to the $i+1$th column of A its values from column i, where i is the index of the letter in the alphabet.

Thanks to the referees!

- matrix multiplication (in our case) can be done in quadratic time!
- shear matrices (matrices de transvection) are the answer (generators for the special linear group $SL_k(\mathbb{Z})$.
- the composition to the right of a matrix A with a shear matrix that has a value c at position (i, j) implies adding to the jth column of A the cth multiple of the column i of A.
- in our case adding to the $i+1$th column of A its values from column i, where i is the index of the letter in the alphabet.

- when looking for the $\mathbb{L}$-Parik Matrix we were just applying Duval's algorithm
- was wrong because for repetitions it will never find the root
- we apply the algorithm on the root and take that rotation of the word

Other small speed ups

- when looking for amiable words we might need to check all m^n words
- SwitchTransformation [Lejeune et al., 2020]
- check for confluence (rewriting) to reduce complexity

Other small speed ups

- when looking for amiable words we might need to check all m^n words
- SwitchTransformation [Lejeune et al., 2020]
- check for confluence (rewriting) to reduce complexity

- Finding if a matrix is Parikh can be done just as above

Space for improvement

- there are still bugs
- the coding of the implementation can be optimised
- worst case complexity of strategies can also be improved

Questions?