Reducing The Ambiguity Of Parikh Matrices

LATA 2020 & 2021

J. Dick L. K. Hutchinson Robert Mercaş D. Reidenbach

23 September 2021

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 matrices

- Histograms
- Parikh vectors [Parikh, 1966] represent a type histograms specific to the analysis of sequences of symbols

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

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

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.

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:

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

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.

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

A big mess

Parikh vectors are not injective.

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

Parikh matrices are not injective

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

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]
- We focus on the word, instead of the mapping

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]

- We focus on the word, instead of the mapping
- Alter word itself: projection morphisms and conjugates

1. Preliminaries
2. ℙ-Parikh Matrices
3. 𝕃-Parikh Matrices
4. Conclusion

$\mathbb{P}$-Parikh Matrix Definition

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

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 $w = bacbebda$.

ℙ-Parikh Matrix Definition

Definition

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

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

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 $w = bacbebda$.

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

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

$$= \varepsilon \; a \; \varepsilon \; \varepsilon \; c \; \varepsilon \; b \; a = acba$$

$\mathbb{P}$-Parikh Matrix Definition

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 $w = bacbebda$.

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

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

$$= \varepsilon\ a\ \varepsilon\ \varepsilon\ c\ \varepsilon\ b\ a = acba$$

$$\Psi_S(bacbebda) = \Psi_S(\pi_{\{a,d,e\}}(bacbebda)) = \Psi_S(acba) = \left\langle \begin{matrix} 2 & 1 & 0 \\ & 1 & 0 \\ & & 1 \end{matrix} \right\rangle$$

When is ambiguity reduced?

Proposition

For any word $w \in \Sigma_n^*$ with a factor $a_i a_j$ where $|i - j| > 1$, we can reduce the ambiguity of $\Psi(w)$ using the $\mathbb{P}$-Parikh matrix of w.

When is ambiguity reduced?

Proposition

For any word $w \in \Sigma_n^*$ with a factor $a_i a_j$ where $|i - j| > 1$, we can reduce the ambiguity of $\Psi(w)$ using the $\mathbb{P}$-Parikh matrix of w.

Example

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

When is ambiguity reduced?

Proposition

For any word $w \in \Sigma_n^*$ with a factor $a_i a_j$ where $|i - j| > 1$, we can reduce the ambiguity of $\Psi(w)$ using the $\mathbb{P}$-Parikh matrix of w.

Example

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

$$\pi_S(abd) = bd$$

When is ambiguity reduced?

Proposition

For any word $w \in \Sigma_n^*$ *with a factor* $a_i a_j$ *where* $|i - j| > 1$*, we can reduce the ambiguity of* $\Psi(w)$ *using the* $\mathbb{P}$*-Parikh matrix of* w*.*

Example

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

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

When is ambiguity reduced?

Proposition

For any word $w \in \Sigma_n^*$ with a factor $a_i a_j$ where $|i-j| > 1$, we can reduce the ambiguity of $\Psi(w)$ using the $\mathbb{P}$-Parikh matrix of w.

Example

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

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

$$\pi_S(adb) = db$$

When is ambiguity reduced?

Proposition

For any word $w \in \Sigma_n^*$ with a factor $a_i a_j$ where $|i-j|>1$, we can reduce the ambiguity of $\Psi(w)$ using the $\mathbb{P}$-Parikh matrix of w.

Example

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

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

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

When is ambiguity not reduced?

Proposition

Let $w, w' \in \Sigma_n^*$ with $w = u_1 a_i a_j v a_j a_i u_2$ and $w' = u_1 a_j a_i v a_i a_j u_2$, where $a_i \leq a_j \in \Sigma_n$ and $u_1, u_2 \in \Sigma_n^*$. If $v \in \{a_k \in \Sigma_n | a_i \leq a_k \leq a_j\}^*$, then for all $S \subseteq \Sigma_n$, we have $\Psi_S(w) = \Psi_S(w')$.

When is ambiguity not reduced?

Proposition

Let $w, w' \in \Sigma_n^*$ with $w = u_1 a_i a_j v a_j a_i u_2$ and $w' = u_1 a_j a_i v a_i a_j u_2$, where $a_i \leq a_j \in \Sigma_n$ and $u_1, u_2 \in \Sigma_n^*$. If $v \in \{a_k \in \Sigma_n | a_i \leq a_k \leq a_j\}^*$, then for all $S \subseteq \Sigma_n$, we have $\Psi_S(w) = \Psi_S(w')$.

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 is ambiguity not reduced?

Proposition

Let $w, w' \in \Sigma_n^*$ with $w = u_1 a_i a_j v a_j a_i u_2$ and $w' = u_1 a_j a_i v a_i a_j u_2$, where $a_i \leq a_j \in \Sigma_n$ and $u_1, u_2 \in \Sigma_n^*$. If $v \in \{a_k \in \Sigma_n | a_i \leq a_k \leq a_j\}^*$, then for all $S \subseteq \Sigma_n$, we have $\Psi_S(w) = \Psi_S(w')$.

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 $S \subseteq \Sigma_3$.

When is ambiguity not reduced?

Proposition

Let $w, w' \in \Sigma_n^*$ with $w = u_1 a_i a_j v a_j a_i u_2$ and $w' = u_1 a_j a_i v a_i a_j u_2$, where $a_i \leq a_j \in \Sigma_n$ and $u_1, u_2 \in \Sigma_n^*$. If $v \in \{a_k \in \Sigma_n | a_i \leq a_k \leq a_j\}^*$, then for all $S \subseteq \Sigma_n$, we have $\Psi_S(w) = \Psi_S(w')$.

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 $S \subseteq \Sigma_3$.

$$S = \{1\}, S = \{2\}, S = \{3\} : \ \Psi_S(w) = \left\langle \begin{smallmatrix} 1 & 2 \\ 1 & 1 \end{smallmatrix} \right\rangle = \Psi_S(w')$$

When is ambiguity not reduced?

Proposition

Let $w, w' \in \Sigma_n^*$ with $w = u_1 a_i a_j v a_j a_i u_2$ and $w' = u_1 a_j a_i v a_i a_j u_2$, where $a_i \leq a_j \in \Sigma_n$ and $u_1, u_2 \in \Sigma_n^*$. If $v \in \{a_k \in \Sigma_n | a_i \leq a_k \leq a_j\}^*$, then for all $S \subseteq \Sigma_n$, we have $\Psi_S(w) = \Psi_S(w')$.

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 $S \subseteq \Sigma_3$.

$$S = \{1\}, S = \{2\}, S = \{3\} : \ \Psi_S(w) = \left\langle \begin{smallmatrix} 1 & 2 \\ 1 & 1 \end{smallmatrix} \right\rangle = \Psi_S(w')$$

$$S = \{1, 2\}, S = \{1, 3\} : \ \Psi_S(w) = \left\langle \begin{smallmatrix} 1 & 2 & 2 \\ & 1 & 2 \\ & & 1 \end{smallmatrix} \right\rangle = \Psi_S(w')$$

When is ambiguity not reduced?

Proposition

Let $w, w' \in \Sigma_n^*$ with $w = u_1 a_i a_j v a_j a_i u_2$ and $w' = u_1 a_j a_i v a_i a_j u_2$, where $a_i \leq a_j \in \Sigma_n$ and $u_1, u_2 \in \Sigma_n^*$. If $v \in \{a_k \in \Sigma_n | a_i \leq a_k \leq a_j\}^*$, then for all $S \subseteq \Sigma_n$, we have $\Psi_S(w) = \Psi_S(w')$.

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 $S \subseteq \Sigma_3$.

$$S = \{1\}, S = \{2\}, S = \{3\} : \ \Psi_S(w) = \left\langle \begin{smallmatrix} 1 & 2 \\ 1 & 1 \end{smallmatrix} \right\rangle = \Psi_S(w')$$

$$S = \{1,2\}, S = \{1,3\} : \ \Psi_S(w) = \left\langle \begin{smallmatrix} 1 & 2 & 2 \\ & 1 & 2 \\ & & 1 \end{smallmatrix} \right\rangle = \Psi_S(w')$$

$$S = \{2,3\} : \ \pi_S(w) = cbbc = \pi_S(w')$$

$\mathbb{L}$-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.

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 $\Psi_{\ell ex}(x) = \Psi(L(w))$.

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-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 $\Psi(w) = \Psi(u) = \Psi(v)$.

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

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 $\Psi(w) = \Psi(u) = \Psi(v)$.

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

$\mathbb{L}$-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 $\Psi(w) = \Psi(u) = \Psi(v)$.

$$L(w) = aababbb, \Psi_{\ell ex}(w) = \langle {}^{3\ 11}_{\ \ 4} \rangle$$
$$L(u) = abababb, \Psi_{\ell ex}(u) = \langle {}^{3\ 9}_{\ \ 4} \rangle$$
$$L(v) = aaabbbb, \Psi_{\ell ex}(v) = \langle {}^{3\ 12}_{\ \ 4} \rangle$$

When do 𝕃-Parikh matrices NOT reduce ambiguity? - Case 1

When do $\mathbb{L}$-Parikh matrices NOT reduce ambiguity? - Case 1

Proposition

For a word $w \in \Sigma^*$, if all words amiable with w belong to the same conjugacy class, then $\mathbb{L}$-Parikh matrices do not reduce the ambiguity of $\Psi(w)$.

When do $\mathbb{L}$-Parikh matrices NOT reduce ambiguity? - Case 1

Proposition

For a word $w \in \Sigma^*$, if all words amiable with w belong to the same conjugacy class, then $\mathbb{L}$-Parikh matrices do not reduce the ambiguity of $\Psi(w)$.

Example

Let $w = aababa$ and $w' = abaaab$. They are amiable with only each other.

$$L(w) = aaabab = L(w')$$

When do $\mathbb{L}$-Parikh matrices NOT reduce ambiguity? - Case 1

Proposition

For a word $w \in \Sigma^*$, if all words amiable with w belong to the same conjugacy class, then $\mathbb{L}$-Parikh matrices do not reduce the ambiguity of $\Psi(w)$.

Example

Let $w = aababa$ and $w' = abaaab$. They are amiable with only each other.

$$L(w) = aaabab = L(w')$$

Proposition

Let $w \in \Sigma_2^*$. Then all words amiable with w belong to the same conjugacy class, if and only if $L(w) \in \{aabb, ababbb, aababb, aabbab, aaabab\}$.

When do $\mathbb{L}$-Parikh matrices NOT reduce ambiguity? - Case 2

Proposition

For a word $w \in \Sigma^*$, if all words associated to $\Psi(w)$ are Lyndon conjugates, then $\mathbb{L}$-Parikh matrices do not reduce the ambiguity of $\Psi(w)$.

When do $\mathbb{L}$-Parikh matrices NOT reduce ambiguity? - Case 2

Proposition

For a word $w \in \Sigma^*$, if all words associated to $\Psi(w)$ are Lyndon conjugates, then $\mathbb{L}$-Parikh matrices do not reduce the ambiguity of $\Psi(w)$.

Example

Both $w = aaabbabb$ and $w' = aabaabbb$ are Lyndon conjugates and only amiable with each other. Hence $\Psi(w) = \Psi(w') = \Psi_{\ell ex}(w) = \Psi_{\ell ex}(w')$.

When do $\mathbb{L}$-Parikh matrices NOT reduce ambiguity? - Case 2

Proposition

For a word $w \in \Sigma^*$, if all words associated to $\Psi(w)$ are Lyndon conjugates, then $\mathbb{L}$-Parikh matrices do not reduce the ambiguity of $\Psi(w)$.

Example

Both $w = aaabbabb$ and $w' = aabaabbb$ are Lyndon conjugates and only amiable with each other. Hence $\Psi(w) = \Psi(w') = \Psi_{\ell ex}(w) = \Psi_{\ell ex}(w')$.

Proposition

All words, w, associated to a Parikh matrix are Lyndon conjugates if and only if $w = a^* v b^*$ 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? - Case 2

Proposition

For a word $w \in \Sigma^*$, if all words associated to $\Psi(w)$ are Lyndon conjugates, then $\mathbb{L}$-Parikh matrices do not reduce the ambiguity of $\Psi(w)$.

Example

Both $w = aaabbabb$ and $w' = aabaabbb$ are Lyndon conjugates and only amiable with each other. Hence $\Psi(w) = \Psi(w') = \Psi_{\ell ex}(w) = \Psi_{\ell ex}(w')$.

Proposition

All words, w, associated to a Parikh matrix are Lyndon conjugates if and only if $w = a^* v b^*$ and for $n = |v|_{ba}$ we have that $|v|_a = 2n$ and $|v|_b = n + 1$.

Example

$w_1 = aaabbabb, v_1 = aaabbab, w_2 = aabaabbb, v_2 = aabaabb$

When do $\mathbb{L}$-Parikh matrices NOT reduce ambiguity? - Case 2

Proposition

For a word $w \in \Sigma^*$, if all words associated to $\Psi(w)$ are Lyndon conjugates, then $\mathbb{L}$-Parikh matrices do not reduce the ambiguity of $\Psi(w)$.

Example

Both $w = aaabbabb$ and $w' = aabaabbb$ are Lyndon conjugates and only amiable with each other. Hence $\Psi(w) = \Psi(w') = \Psi_{\ell ex}(w) = \Psi_{\ell ex}(w')$.

Proposition

All words, w, associated to a Parikh matrix are Lyndon conjugates if and only if $w = a^* v b^*$ and for $n = |v|_{ba}$ we have that $|v|_a = 2n$ and $|v|_b = n + 1$.

Example

$w_1 = aaabbabb, v_1 = aaabbab, w_2 = aabaabbb, v_2 = aabaabb$

$n = |v|_{ba} = 2, |v|_a = 2n = 4, |v|_b = n + 1 = 3$

When do $\mathbb{L}$-Parikh matrices NOT reduce ambiguity? - Case 3

Proposition

For a word $w \in \Sigma_2^*$, if the only words associated to $\Psi(w)$ are bbabbaaa and bbbaabaa, then $\mathbb{L}$-Parikh matrices do not reduce the ambiguity of $\Psi(w)$.

When do $\mathbb{L}$-Parikh matrices NOT reduce ambiguity? - Case 3

Proposition

For a word $w \in \Sigma_2^*$, if the only words associated to $\Psi(w)$ are bbabbaaa and bbbaababaa, then $\mathbb{L}$-Parikh matrices do not reduce the ambiguity of $\Psi(w)$.

Theorem [Atanasiu et al., 2008]

For a word w, we have that the set of all words amiable with w is equal to the reverse of the set of all words amiable with the reverse of w.

When do $\mathbb{L}$-Parikh matrices NOT reduce ambiguity? - Case 3

Proposition

For a word $w \in \Sigma_2^*$, if the only words associated to $\Psi(w)$ are bbabbaaa and bbbaabaa, then $\mathbb{L}$-Parikh matrices do not reduce the ambiguity of $\Psi(w)$.

Theorem [Atanasiu et al., 2008]

For a word w, we have that the set of all words amiable with w is equal to the reverse of the set of all words amiable with the reverse of w.

Example

Recall that *aaabbabb* and *aabaabbb* are amiable with only each other. We have $rev(aaabbabb) = bbabbaaa$ and $rev(abaabbb) = bbbaabaa$.

When do $\mathbb{L}$-Parikh matrices NOT reduce ambiguity? - Case 3

Proposition

For a word $w \in \Sigma_2^*$, if the only words associated to $\Psi(w)$ are bbabbaaa and bbbaabaa, then $\mathbb{L}$-Parikh matrices do not reduce the ambiguity of $\Psi(w)$.

Theorem [Atanasiu et al., 2008]

For a word w, we have that the set of all words amiable with w is equal to the reverse of the set of all words amiable with the reverse of w.

Example

Recall that *aaabbabb* and *aabaabbb* are amiable with only each other. We have $rev(aaabbabb) = bbabbaaa$ and $rev(abaabbb) = bbbaabaa$.

$$\Psi(bbabbaaa) = \Psi(bbbaabaa) = \left\langle \begin{matrix} 4 & 2 \\ & 4 \end{matrix} \right\rangle$$

The Main Result

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:*

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

The Main Result

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:

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

The Main Result

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 following 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* $|v|_b = n + 1$

The Main Result

Theorem

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

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

Conclusion

- Shown sufficient conditions for ℙ-Parikh matrices to reduce ambiguity in ternary or larger words.

Conclusion

- Shown sufficient conditions for $\mathbb{P}$-Parikh matrices to reduce ambiguity in ternary or larger words.
- Characterised when $\mathbb{L}$-Parikh matrices reduce ambiguity in binary alphabets.

Conclusion

- Shown sufficient conditions for $\mathbb{P}$-Parikh matrices to reduce ambiguity in ternary or larger words.
- Characterised when $\mathbb{L}$-Parikh matrices reduce ambiguity in binary alphabets.
- Using these matrices, we reduce the ambiguity of many words.

Future Work

Future Work

- Characterise amiable words for the ternary alphabet.

Future Work

- Characterise amiable words for the ternary alphabet.
- Count ternary amiable words.

Future Work

- Characterise amiable words for the ternary alphabet.
- Count ternary amiable words.
- Characterise Parikh matrices.

Future Work

- Characterise amiable words for the ternary alphabet.
- Count ternary amiable words.
- Characterise Parikh matrices.
- Algorithms

Questions?