Bounds on the Rate of 2-D Bit-Stuffing Encoders

Ido Tal Ron M. Roth

Computer Science Department
Technion, Haifa 32000, Israel

2-D constraints

The square constraint – our running example

- A binary $M \times N$ array satisfies the square constraint iff no two '1' symbols are adjacent on a row, column, or diagonal.
- Example:

$$\begin{matrix} 1 & \mathbf{0} & 0 & 0 & \mathbf{0} & 1 & \mathbf{0} & 0 \\ \mathbf{0} & \mathbf{0} & 0 & 0 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ 0 & 0 & 0 & 0 & 0 & \mathbf{0} & 1 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & 0 & 0 & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & 1 & \mathbf{0} & 0 & 0 & 0 & 0 & 0 \end{matrix}$$

  - If a bold-face **0** is changed to 1, then the square constraint does not hold.

Notation for the general case

- Let $\mathbb{S}$ be a constraint over an alphabet Σ.
- Denote by $\mathbb{S} \cap \Sigma^{M \times N}$ all the $M \times N$ arrays satisfying the constraint.

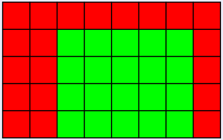
Bit stuffing encoders

Encoder Definition

$$\mathsf{E} = \left(\Psi, \mu, \boldsymbol{\delta} = (\delta_{M,N})_{M,N>0}\right) .$$

$\delta_{M,N}$ and $\partial_{M,N}$

- $\partial_{M,N} = \partial_{M,N}(E)$ is the border index set of the array we wish to encode into. $\bar{\partial}_{M,N}$ is the complementary set.



Bit stuffing encoders

Encoder Definition

$$\mathsf{E} = (\Psi, \mu, \boldsymbol{\delta} = (\delta_{M,N})_{M,N>0}) \ .$$

$\delta_{M,N}$ and $\partial_{M,N}$

- $\partial_{M,N} = \partial_{M,N}(E)$ is the border index set of the array we wish to encode into. $\bar{\partial}_{M,N}$ is the complementary set.
- $\delta_{M,N}$ is a probability distribution on all valid borders,

$$\delta_{M,N} : \mathbb{S}[\partial_{M,N}] \to [0,1] \ .$$

1	0	0	0	0	1	0	0
0	0						0
0	0						0
0	0						0
0	1						0

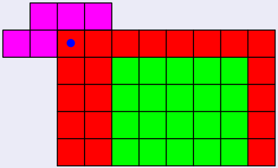
Encoder

Ψ and μ

- Let $\sigma_{\alpha,\beta}$ be the index shifting operator:

$$\sigma_{\alpha,\beta}(U) = \{(i+\alpha, j+\beta) : (i,j) \in U\} \ .$$

- Encoding into $\bar{\partial}_{M,N}$ is done in raster fashion.
- When encoding to position $(i,j) \in \bar{\partial}_{M,N}$, we only look at positions $\sigma_{i,j}(\Psi)$: the neighborhood of (i,j).
- The probability distribution of entry (i,j) is given by the function $\mu(\cdot|\cdot)$.



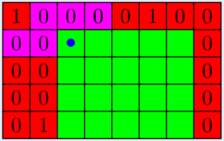
Encoder

Ψ and μ

- Let $\sigma_{\alpha,\beta}$ be the index shifting operator:

$$\sigma_{\alpha,\beta}(U) = \{(i+\alpha, j+\beta) : (i,j) \in U\} \ .$$

- Encoding into $\bar{\partial}_{M,N}$ is done in raster fashion.
- When encoding to position $(i,j) \in \bar{\partial}_{M,N}$, we only look at positions $\sigma_{i,j}(\Psi)$: the neighborhood of (i,j).
- The probability distribution of entry (i,j) is given by the function $\mu(\cdot|\cdot)$.



$$\mu\left(0 \left| \begin{matrix} & 0 & 0 & 0 \\ 0 & 0 & \bullet & \end{matrix} \right.\right) = 0.258132$$

Encoder

Ψ and μ

- Let $\sigma_{\alpha,\beta}$ be the index shifting operator:

$$\sigma_{\alpha,\beta}(U) = \{(i+\alpha, j+\beta) : (i,j) \in U\} \ .$$

- Encoding into $\bar{\partial}_{M,N}$ is done in raster fashion.
- When encoding to position $(i,j) \in \bar{\partial}_{M,N}$, we only look at positions $\sigma_{i,j}(\Psi)$: the neighborhood of (i,j).
- The probability distribution of entry (i,j) is given by the function $\mu(\cdot|\cdot)$.

1	0	0	0	0	1	0	0
0	0	0					0
0	0						0
0	0						0
0	1						0

$$\mu\left(0 \,\middle|\, \begin{matrix} & 0 & 0 & 0 \\ 0 & 0 & \bullet & \end{matrix}\right) = 0.258132$$

Encoder

Ψ and μ

- Let $\sigma_{\alpha,\beta}$ be the index shifting operator:

$$\sigma_{\alpha,\beta}(U) = \{(i+\alpha, j+\beta) : (i,j) \in U\} \ .$$

- Encoding into $\bar{\partial}_{M,N}$ is done in raster fashion.
- When encoding to position $(i,j) \in \bar{\partial}_{M,N}$, we only look at positions $\sigma_{i,j}(\Psi)$: the neighborhood of (i,j).
- The probability distribution of entry (i,j) is given by the function $\mu(\cdot|\cdot)$.

1	0	0	0	0	1	0	0
0	0	0	•				0
0	0						0
0	0						0
0	1						0

$$\mu\left(0 \middle| \begin{matrix} & 0 & 0 & 0 \\ 0 & 0 & \bullet & \end{matrix}\right) = 0.258132$$

Encoder

Ψ and μ

- Let $\sigma_{\alpha,\beta}$ be the index shifting operator:

$$\sigma_{\alpha,\beta}(U) = \{(i+\alpha, j+\beta) : (i,j) \in U\} \ .$$

- Encoding into $\bar{\partial}_{M,N}$ is done in raster fashion.
- When encoding to position $(i,j) \in \bar{\partial}_{M,N}$, we only look at positions $\sigma_{i,j}(\Psi)$: the neighborhood of (i,j).
- The probability distribution of entry (i,j) is given by the function $\mu(\cdot|\cdot)$.

1	0	0	0	0	1	0	0
0	0	0	1				0
0	0						0
0	0						0
0	1						0

$$\mu\left(0 \,\middle|\, \begin{matrix} & 0 & 0 & 0 \\ 0 & 0 & \bullet & \end{matrix}\right) = 0.258132$$

Encoder

Ψ and μ

- Let $\sigma_{\alpha,\beta}$ be the index shifting operator:

$$\sigma_{\alpha,\beta}(U) = \{(i+\alpha, j+\beta) : (i,j) \in U\} \ .$$

- Encoding into $\bar{\partial}_{M,N}$ is done in raster fashion.
- When encoding to position $(i,j) \in \bar{\partial}_{M,N}$, we only look at positions $\sigma_{i,j}(\Psi)$: the neighborhood of (i,j).
- The probability distribution of entry (i,j) is given by the function $\mu(\cdot|\cdot)$.

1	0	0	0	0	1	0	0
0	0	0	1	•			0
0	0						0
0	0						0
0	1						0

$$\mu\left(0 \,\middle|\, \begin{array}{cccc} & 0 & 0 & 1 \\ 0 & 1 & \bullet & \end{array}\right) = 1$$

Encoder

Encoder?

- Q: So, why is this an “encoder”?
- A: The “coins” are in fact (invertible) probability transformers, the input of which is the information we wish to encode.

Encoder

Encoder?

- Q: So, why is this an “encoder”?
- A: The “coins” are in fact (invertible) probability transformers, the input of which is the information we wish to encode.

Encoder rate

- Let $A = A(\mathsf{E}, M, N)$ be the random variable corresponding to the array we produce.
- The rate of our encoder is

$$R(\mathsf{E}) \triangleq \liminf_{M,N\to\infty} \frac{H(A[\bar{\partial}_{M,N}] | A[\partial_{M,N}])}{M \cdot N} \; .$$

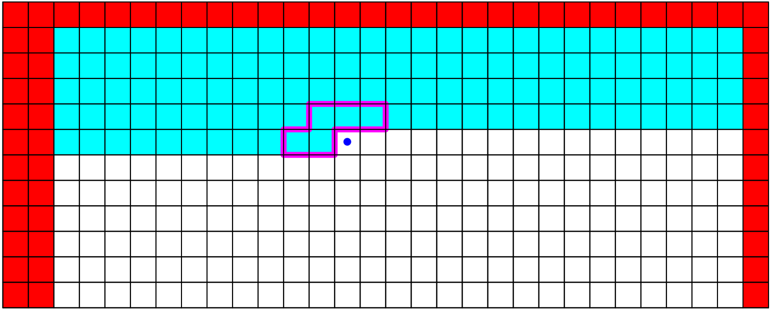
- Problem: How does one calculate the rate…

First fact: Locality of conditional entropy

Let $T_{i,j}$ be all the indices preceding (i,j) in the raster scan.

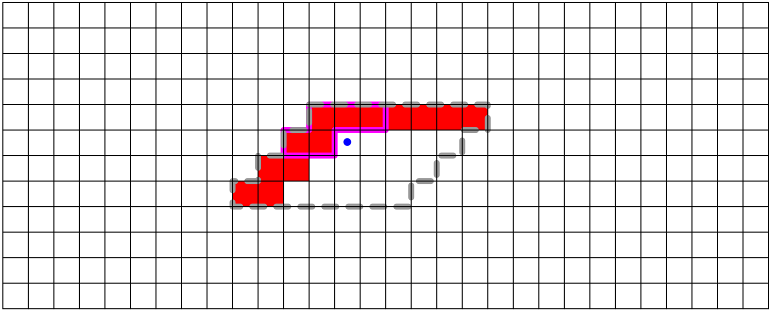
$$R(\mathsf{E}) = \liminf_{M,N\to\infty} \sum_{(i,j)\in\bar{\partial}_{M,N}} \frac{H(a_{i,j}|A[\partial_{M,N}] \cup A[T_{i,j}])}{M \cdot N}$$

$$= \liminf_{M,N\to\infty} \sum_{(i,j)\in\bar{\partial}_{M,N}} \frac{H(a_{i,j}|A[\sigma_{i,j}(\Psi)])}{M \cdot N}$$



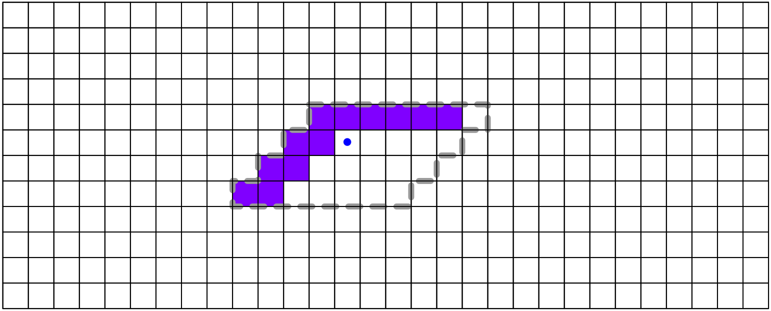
Second fact: If we know the border's distribution, then we know the whole distribution

- Consider a (relatively small) patch Λ with border Γ.
- If we know the probability distribution of $A[\Gamma]$, then we know the probability distribution of $A[\Lambda]$.



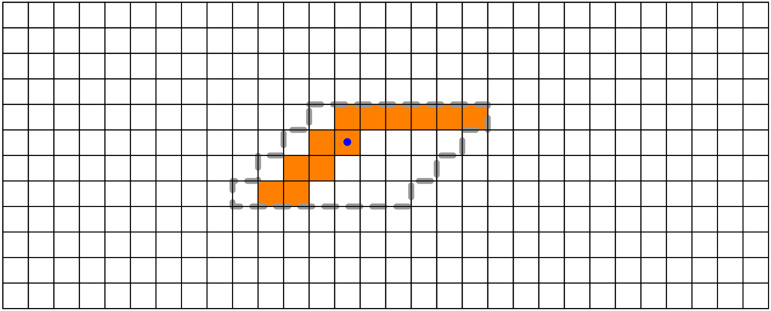
Third fact: Stationarity inside the patch

- Let Γ' be Γ, without the last column.
- We will prove later that w.l.o.g., the probability distributions of $A[\Gamma']$ is equal to the probability distribution of $A[\sigma_{0,1}(\Gamma')]$.



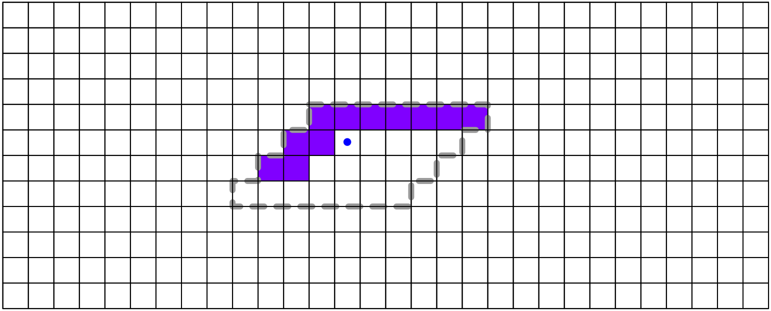
Third fact: Stationarity inside the patch

- Let Γ' be Γ, without the last column.
- We will prove later that w.l.o.g., the probability distributions of $A[\Gamma']$ is equal to the probability distribution of $A[\sigma_{0,1}(\Gamma')]$.



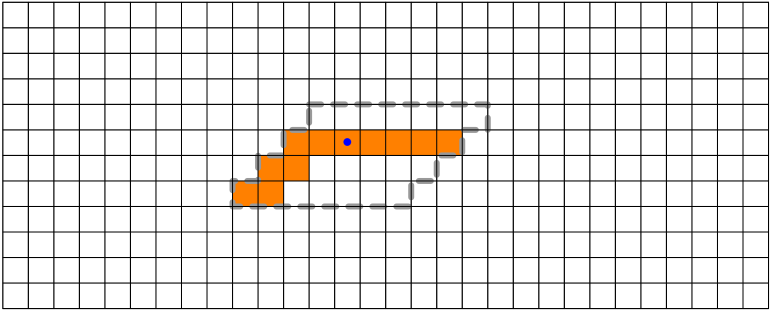
Third fact (assumption): Stationarity inside the patch

- Let Γ'' be Γ, without the last row.
- We will prove later that w.l.o.g., the probability distributions of $A[\Gamma'']$ is equal to the probability distribution of $A[\sigma_{1,-1}(\Gamma'')]$.



Third fact (assumption): Stationarity inside the patch

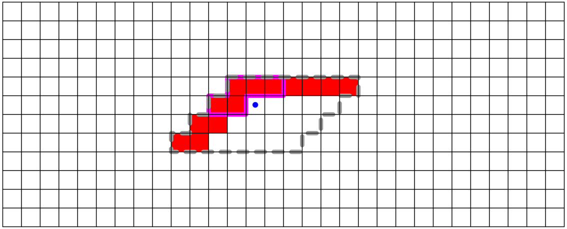
- Let Γ'' be Γ, without the last row.
- We will prove later that w.l.o.g., the probability distributions of $A[\Gamma'']$ is equal to the probability distribution of $A[\sigma_{1,-1}(\Gamma'')]$.



The bound

Recall that $R(\mathsf{E}) = \liminf_{M,N\to\infty} \sum_{(i,j)\in\bar{\partial}_{M,N}} \frac{H(a_{i,j}|A[\sigma_{i,j}(\Psi)])}{M\cdot N}$.

- Consider all patch border probabilities which result in a stationary patch.
- For each such probability, look at $H(a_{i,j}|\sigma_{i,j}(\Psi))$.
- The smallest (largest) value is an lower (upper) bound on the rate of our encoder.
- The above minimization (maximization) problem is a linear program. It gets more accurate, but harder, as we enlarge the patch.



Some numerical results

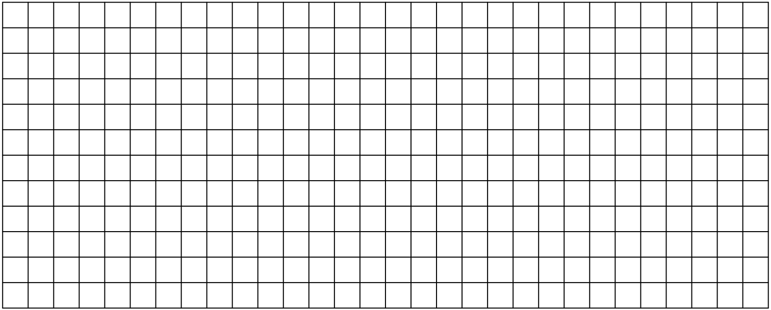
Constraint	Coins	$\mathrm{lp}^*_{\mathrm{min}}$	$\mathrm{lp}^*_{\mathrm{max}}$	[Halevy+:04]
$(2,\infty)$-RLL	1	0.440722	0.444679	0.4267
$(3,\infty)$-RLL	1	0.349086	0.386584	0.3402
n.i.b.	2	0.91773	0.919395	0.91276
$(1,\infty)$-RLL	3	0.587776	0.587785	—

Some more numerical results

Constraint	Coins	lp^*_{min}	lp^*_{max}	Others
$(2, \infty)$-RLL	5	**0.444202**	0.444997	0.4423
$(3, \infty)$-RLL	2	0.359735	0.368964	0.3641
$(0, 2)$-RLL	66	**0.815497**	0.816821	0.7736
	18	0.815013	0.816176	
	9	0.810738	0.819660	
n.i.b.	56	**0.922640**	0.923748	0.9156

$A^{(k)}$

- Define [Halevy+:04] a new random variable, $A^{(k)}$:
  Out of the k^2 contiguous $(M-k+1)\times(N-k+1)$ sub-arrays of A , pick one uniformly at random, and call it $A^{(k)}$.



$A^{(k)}$

- Define [Halevy+:04] a new random variable, $A^{(k)}$:
  Out of the k^2 contiguous $(M-k+1)\times(N-k+1)$ sub-arrays of A , pick one uniformly at random, and call it $A^{(k)}$.

0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0
0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0
1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0
0	0	0	1	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
1	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0

$A^{(k)}$

- Define [Halevy+:04] a new random variable, $A^{(k)}$: Out of the k^2 contiguous $(M-k+1)\times(N-k+1)$ sub-arrays of A , pick one uniformly at random, and call it $A^{(k)}$. $k=5$:

0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0
0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0
1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0
0	0	0	1	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
1	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0

$A^{(k)}$

- Define [Halevy+:04] a new random variable, $A^{(k)}$:
  Out of the k^2 contiguous $(M-k+1)\times(N-k+1)$ sub-arrays of A , pick one uniformly at random, and call it $A^{(k)}$. $k=5$:

0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0
0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0
1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0
0	0	0	1	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
1	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0

$A^{(k)}$

- Define [Halevy+:04] a new random variable, $A^{(k)}$:
  Out of the k^2 contiguous $(M-k+1)\times(N-k+1)$ sub-arrays of A , pick one uniformly at random, and call it $A^{(k)}$. $k=5$:

0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0
0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0
1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0
0	0	0	1	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
1	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0

$A^{(k)}$

- Define [Halevy+:04] a new random variable, $A^{(k)}$:
  Out of the k^2 contiguous $(M-k+1)\times(N-k+1)$ sub-arrays of A , pick one uniformly at random, and call it $A^{(k)}$. $k=5$:

0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0
0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0
1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0
0	0	0	1	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
1	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0

Merits of $A^{(k)}$

- Since $A^{(k)}$ is an "averaging out" of A, we have that as k grows, $A^{(k)}$ becomes more and more "locally stationary".
- We can define an encoder

$$\mathsf{E}^{(k)} = (\Psi, \mu, \boldsymbol{\delta}^{(k)})$$

  with $A(\mathsf{E}^{(k)})$ having the same probability distribution as $A^{(k)}$.
- The rate of the encoders are the same, $R(\mathsf{E}) = R(\mathsf{E}^{(k)})$.
- So, essentially, we can assume w.l.o.g. that the patch in A is stationary.

0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0
0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0
1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0
0	0	0	1	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
1	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0