

# A reevaluation and benchmark of hidden Markov Models

› Jean-Paul van Oosten Prof. Lambert Schomaker



# Hidden Markov model fields & variants

- Automatic speech recognition
- Gene sequence segmentation
- Handwriting recognition
- ...

- Pseudo-2D HMMs
- Markov random fields
- Explicit duration modelling
- Nested HMMs
- ...



# Hidden Markov model fields & variants

- All these variants have in common that:
- They are essentially HMMs at the core (i.e., have an initial state distribution $\pi$, a transition matrix $\boldsymbol{A}$, and an observation probability distribution $\boldsymbol{B}$)
- They are usually trained using Baum-Welch

- Widely used…
  But do we understand them sufficiently?



# Goal: Investigate core aspects of HMMs

- I. How can we test and benchmark our implementations?
- II. How reliable is the Baum-Welch algorithm? Do we find the underlying Markov parameters?
- III. What is the role of the transition matrix and how important is temporal modelling?



# I. BENCHMARK



# Benchmark

› Initially a test for a fresh implementation

› There is no real benchmark for HMM implementations available with a gradual scale of increasing difficulty

› Real-world data sets exist (see for example Siddiqi, Gordon & Moore, 2007) but the underlying Markov parameters are unknown!



# Benchmark idea:

- Discrete observations (we inspect the temporal aspects of HMMs first).
- Varying degrees of symbol lexicon overlap between classes:
  - $\delta = 0$: $L_1 = L_2 = \{a, b, c\}$
  - $\delta = 1$: $L_1 = \{a, b, c\}, L_2 = \{b, c, d\}$
  - …
- Compare several implementations (jpHMM, dHMM, GHMM and HTK)



# Benchmark experiments

- Generate 100 classes for each difficulty (i.e., separability δ), randomly initialised Bakis models with $N_{states} = 10$ states and $N_{symbols} = 20$.
- Sequence length was fixed at $|\vec{O}| = 10$ observations, 300 sequences per class (i.e., 30 000 sequences in total)
- Discrete, single dimension observations (same procedure can be applied to continuous observations with more dimensions).



# Classification accuracy

| | Separability $\delta$ | jpHMM | dHMM | GHMM | HTK |
|---|---|---|---|---|---|
| Hard | 0 | 1% | 1% | 1% | 1% |
| | 1 | 41% | 40% | 37% | 41% |
| | 2 | 66% | 64% | 61% | 66% |
| | 3 | 81% | 78% | 76% | 80% |
| | 5 | 95% | 93% | 92% | 94% |
| | 10 | 100% | 100% | 100% | 100% |
| Easy | 20 | 100% | 100% | 100% | 100% |

university of groningen



# Classification accuracy

| | Separability $\delta$ | jpHMM | dHMM | GHMM | HTK |
|---|---|---|---|---|---|
| Hard | 0 | 1% | 1% | 1% | 1% |
| | 1 | 41% | 40% | 37% | 41% |
| | 2 | 66% | 64% | 61% | 66% |
| | 3 | 81% | 78% | 76% | 80% |
| | 5 | 95% | 93% | 92% | 94% |
| | 10 | 100% | 100% | 100% | 100% |
| Easy | 20 | 100% | 100% | 100% | 100% |

No essential differences between implementations!

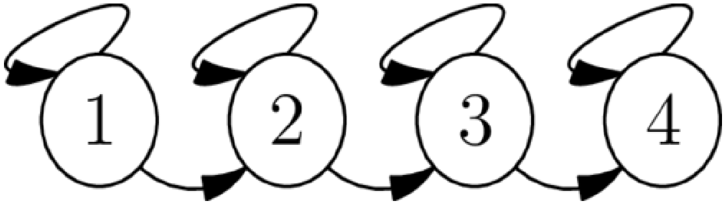
# Benchmark

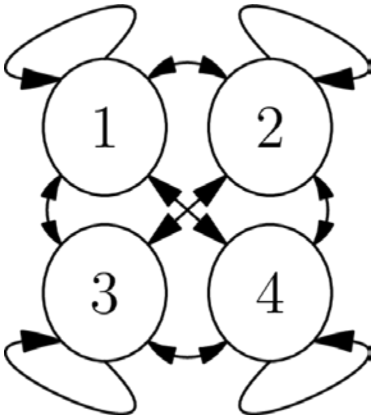
- Gauging the difficulty of any dataset
- 95% performance accuracy?
  Implies that 5 tokens must be different between classes on a 20 token alphabet.



# II.
# LEARNING THE TOPOLOGY OF A TRANSITION MATRIX

# Learning the topology of a transition matrix



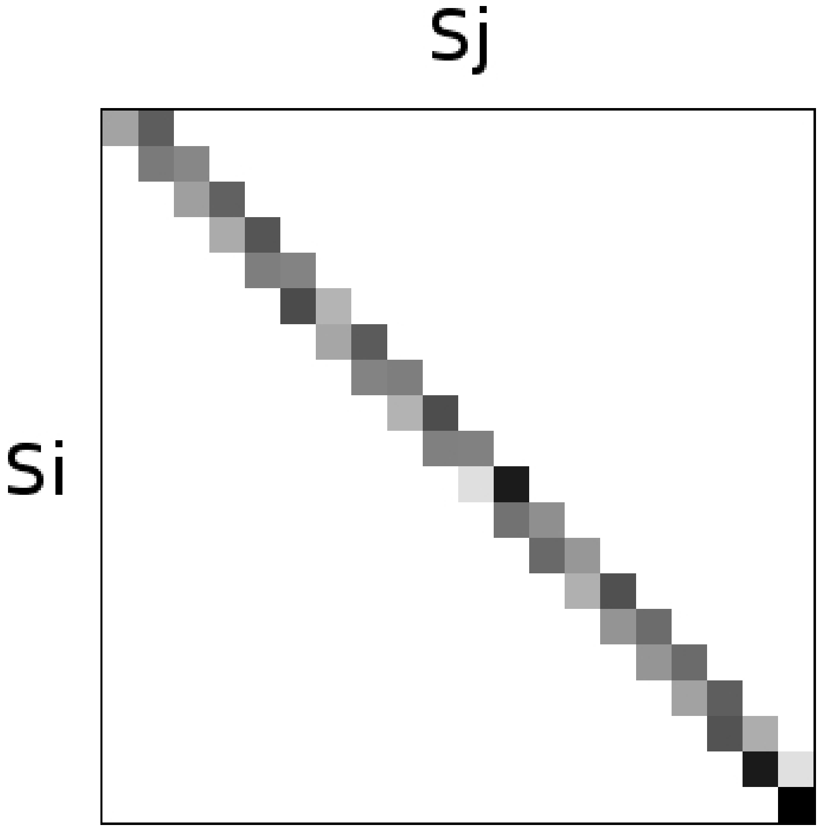




- How reliable is the Baum-Welch algorithm?
- Figueiredo and Jain (2002) have already shown that EM algorithms can be brittle.
- Will HMM detect an underlying Bakis topology?

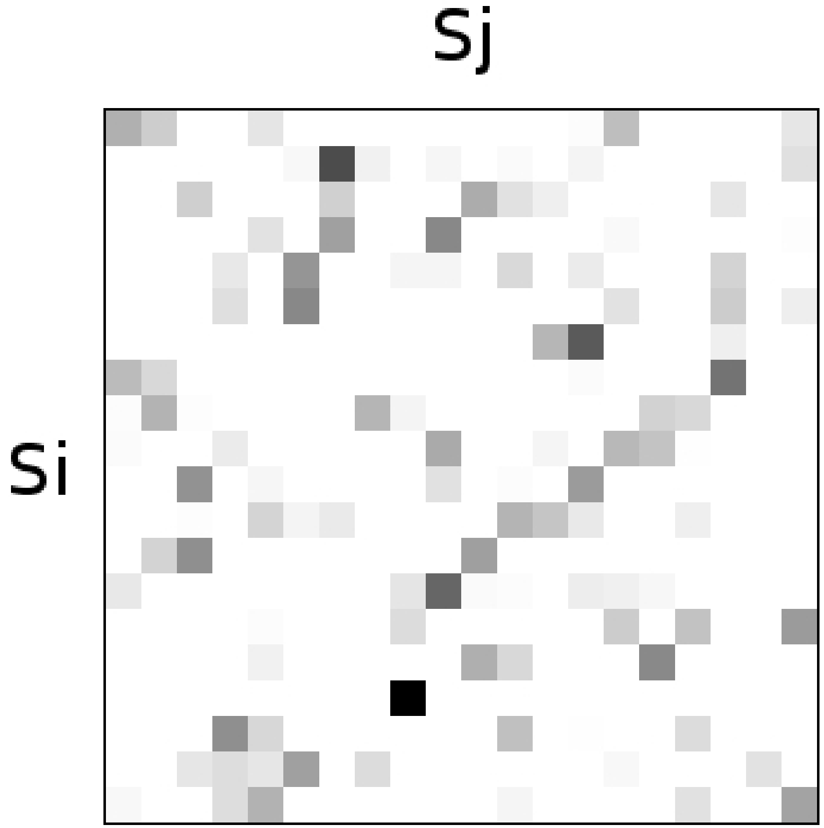
# General setup:

- Generate data using a Bakis topology, so we know the exact Markov parameters.
- Train models without restrictions (i.e., ergodic)
- Align hidden state order by permuting all state orderings and selecting the one with smallest $\chi^2$ distance to original (B-W can create a state order that is not necessarily the same as the original)
- Compare the transition matrices:

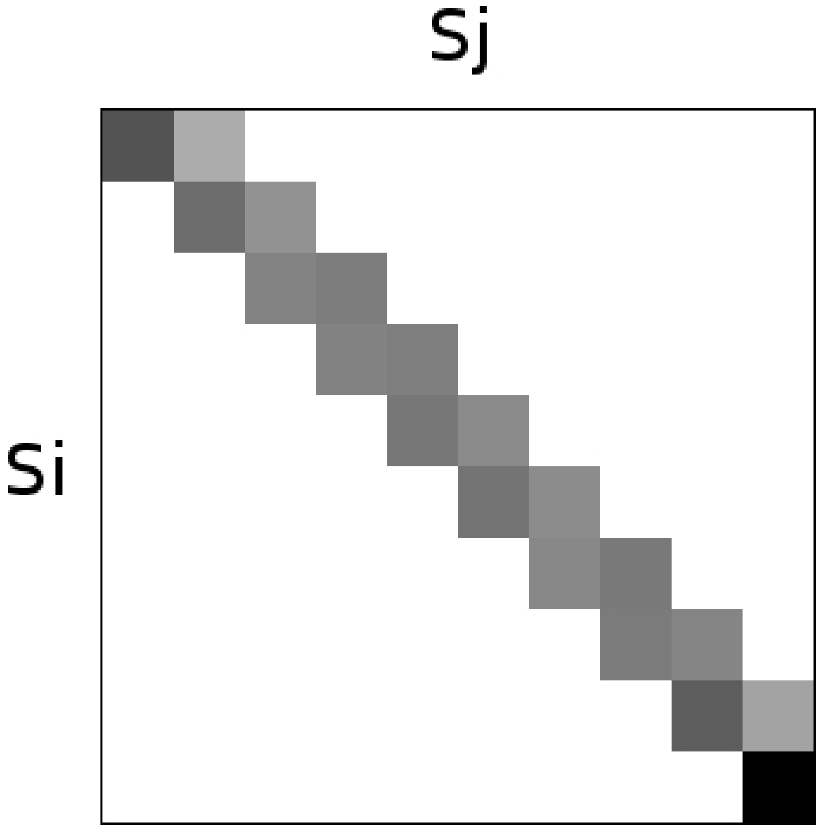
## Target (Bakis) model



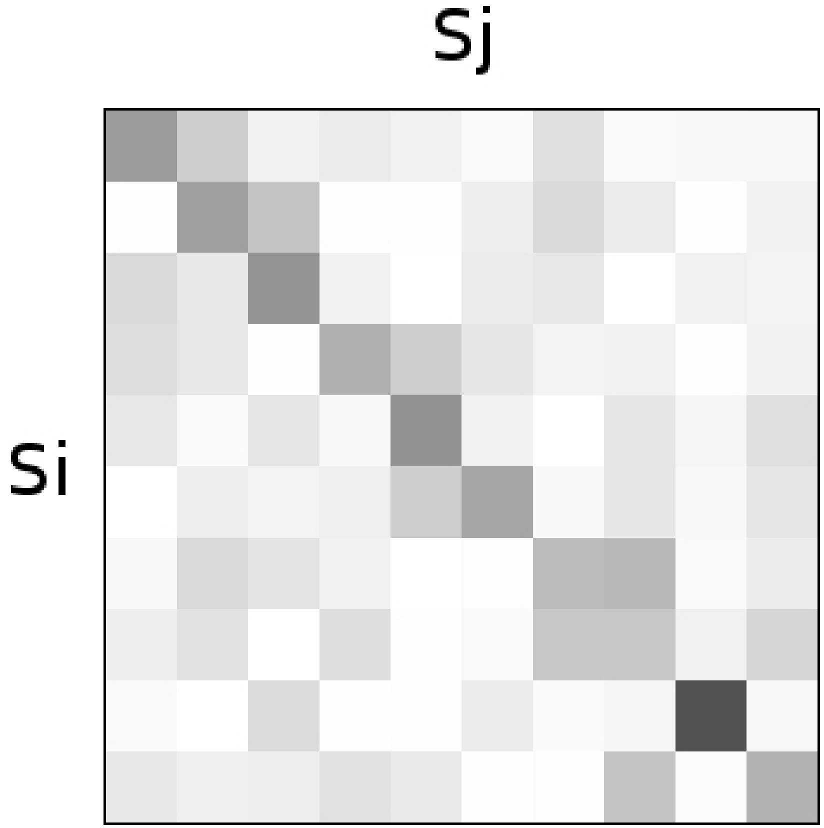

## Learned (ergodic) model



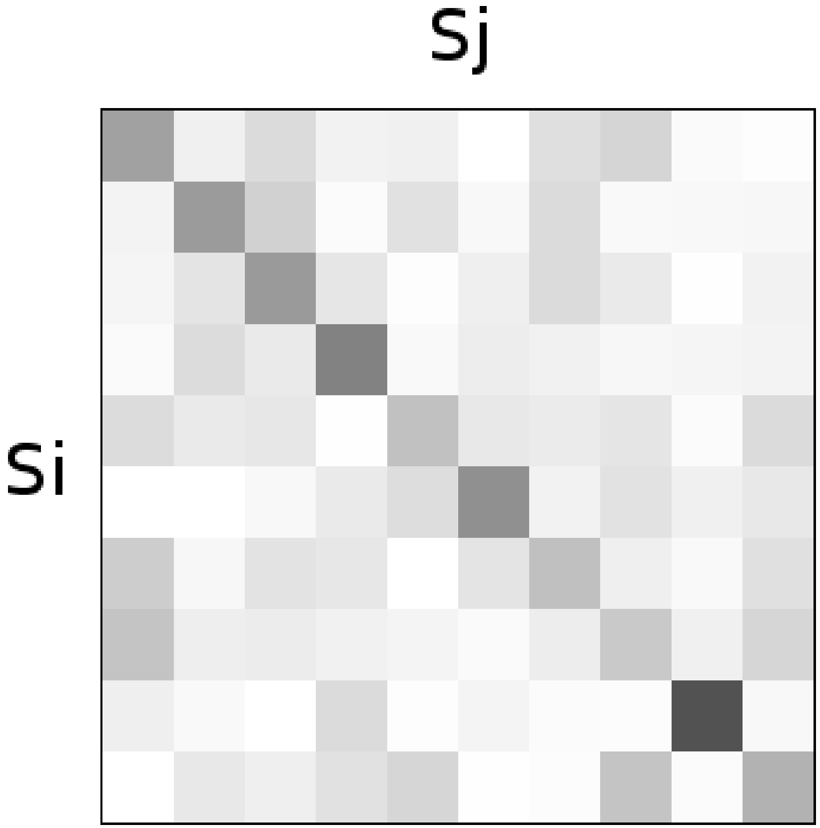
**Target (Bakis) model**



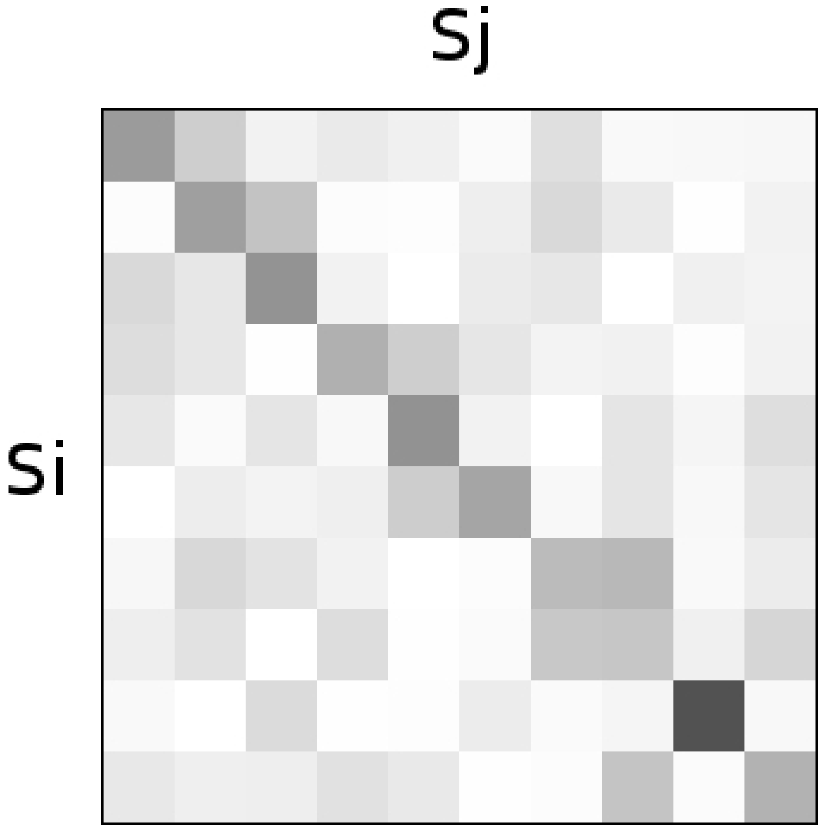

**Learned (ergodic) model**



## Unaligned model




## Aligned model





# Learning the topology of a transition matrix

- It is amazing that we don’t find (an approximation) of the Bakis topology, given the amount of effort we put into this.
- How can the performance in applications of HMMs be attractive if we see that the real properties are not found?



# III.
# THE IMPORTANCE OF TEMPORAL MODELLING



# The importance of temporal modelling

- What happens to performance if we remove the temporal aspect from an HMM?
- Flat topology ("Orderless bag of states"): $a_{ij} = \frac{1}{N}$
- Compare Flat vs Bakis vs Ergodic
- Handwritten cursive words; two features:
  - $FCO^3$ (4900D, 130 classes, 31k instances, 3 states)
  - Sliding window, discretized using SOFM (625D, 20 classes, 5k instances, 27 states)



# What happens when we remove temporal modelling?

- The temporal aspect is probably important
- We expect the performance of a flat HMM to drop drastically compared to ergodic or Bakis models.

$$FCO^3$$

| Topology | Accuracy |
|---|---|
| Flat | $59.1\% \pm 0.8$ |
| Bakis | $59.9\% \pm 0.9$ |
| Ergodic | $59.5\% \pm 0.9$ |

**$FCO^3$**

| Topology | Accuracy |
| --- | --- |
| Flat | 59.1% $\pm$ 0.8 |
| Bakis | 59.9% $\pm$ 0.9 |
| Ergodic | 59.5% $\pm$ 0.9 |

**Sliding window**

| Topology | Accuracy |
| --- | --- |
| Flat | 71.1%$\pm$1.3 |
| Bakis | 75.2%$\pm$2.0 |
| Ergodic | 78.5%$\pm$1.2 |



# The importance of temporal modelling

- The performance drop is not so drastic as expected
- The temporal aspect seems to be less important than the observation probabilities
- Design of features is still important!

# CONCLUSIONS

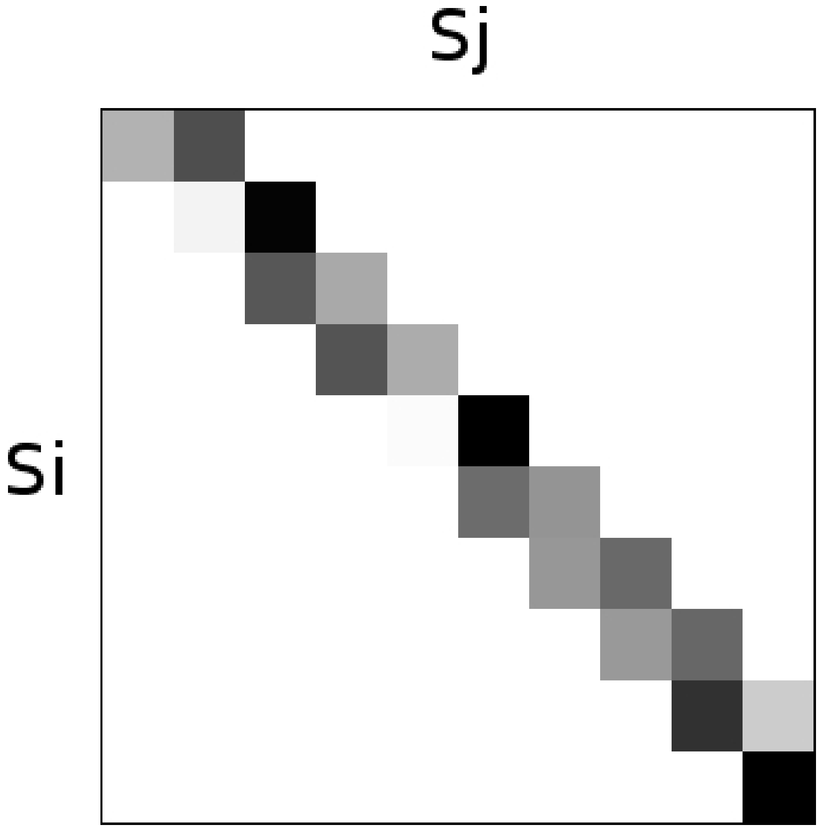
# Conclusions

- Why stress the temporal state modeling of HMMs when the hidden state sequence plays a relatively minor role?
- Baum-Welch is brittle (also see Figueiredo and Jain (2002))
- “Bag of states” (including dynamic programming) and the Markov assumption are strong principles
- There are many tricks of the trade, many of which badly documented in the literature (see also the appendix of the paper).

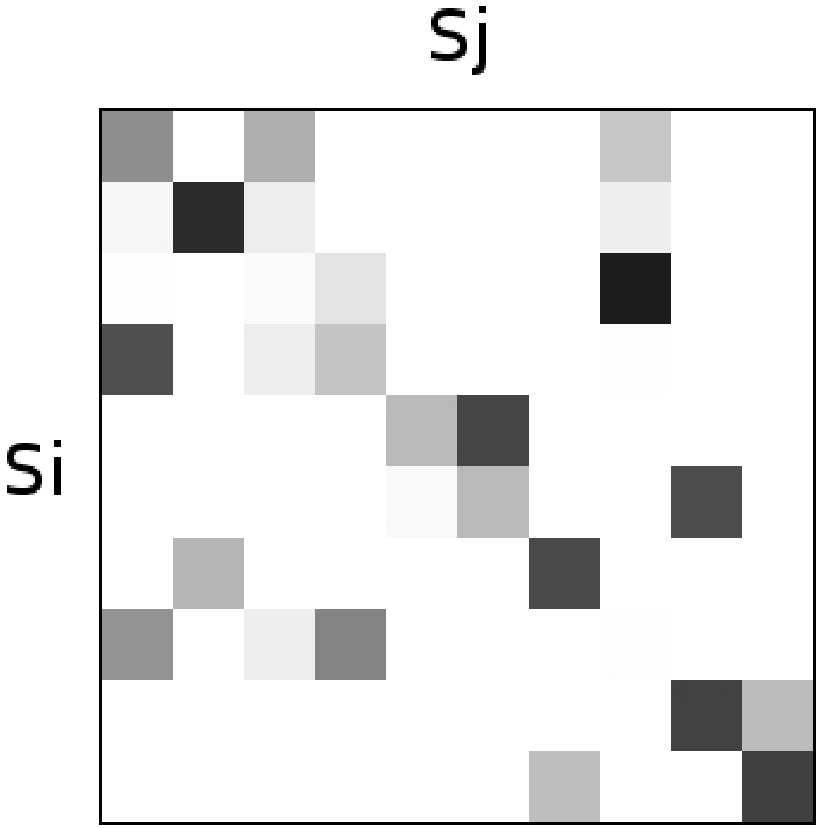
# Invitation for discussion

- The core principles such as the Markov assumption and dynamic programming seem to be working, but Baum-Welch seems to be brittle.
- Is it a problem that ergodically trained systems do not find the underlying transition probabilities?
- Is the “Bag of states” approach sufficient (for handwriting recognition purposes)?

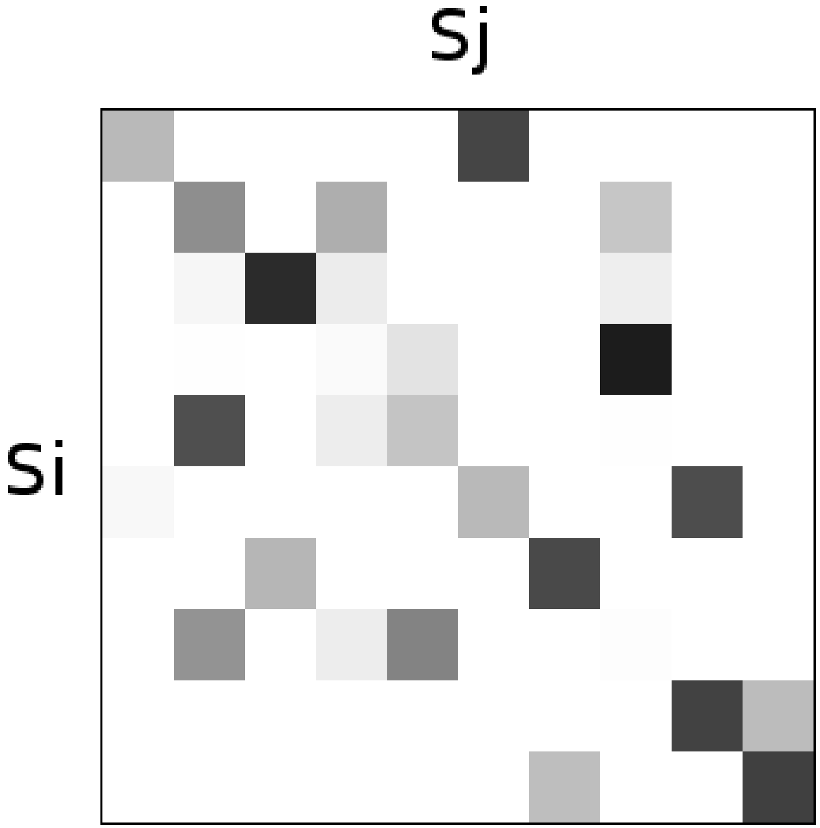
Target (Bakis) model




Learned (ergodic) model



## Unaligned model




## Aligned model