An Incremental Model Selection Algorithm Based on Cross-Validation for Finding the Architecture of a Hidden Markov Model on Hand Gesture Data Sets

Aydın Ulaş[1] Olcay Taner Yıldız[2]

[1]Department of Computer Engineering
Boğaziçi University
Istanbul, Turkey

[2]Department of Computer Engineering
Işık University
Istanbul, Turkey

Outline

1. Motivation
   - Hidden Markov Models
   - Model Selection in HMM
2. MSUMO: A Meta Learning Algorithm for Architecture Selection
   - Structure Learning as State Space Search
   - MSUMO
3. Experiments
   - Experimental Setup
   - Comparison of Architectures
   - Results
4. Conclusions & Future Work

Hidden Markov Models

Graphical network

- N hidden states
- M mixture components in these hidden states
- Connectivity between hidden states

Hidden Markov Models

Probability model

- Initial state probabilities
- Observation probabilities
- State transition probabilities
- Baum-Welch algorithm using Expectation-Maximization

Hidden Markov Models

Models used in this paper

- Left-right model
- Left-right-loop model
- Full model

Hidden Markov Models

- Bioinformatics (Secondary structure prediction)
- Speech processing (Audio modelling)

Model Selection
Bias-variance tradeoff

- Error = Bias + Variance
- Small/simple model underfits (bias high, variance low)
- Large/complex model overfits (bias low, variance high)

Model Selection in HMM

- Hard to estimate number of hidden states & models in each state
- Need a methodology to optimize the model structure for novice user

Structure Learning
State Space Search

- Huge search space: All the combinations of number of hidden states and number of mixtures at each hidden state.
- Infeasible to try and evaluate all possible architectures
- Heuristic strategy visiting as few as possible states in the search space

Structure Learning

Operators

- Define operators. Example: ADD-1 from state $HMM_{4,1}$ to $HMM_{5,1}$
- Add operators for *forward search*
- Remove operators for *backward search*
- *Floating search* uses both

Structure Learning

Evaluation

- Compare performance metric of next state with current state
- Accept or reject operator based on improvement
- Takes into account both generalization error and complexity

Structure Learning
Dimensions of Search

- Initial state ($HMM_{1,1}$ or $HMM_{N,1}$)
- State transition operators (Add or remove)
- Search beam (Single or multiple operator)
- State evaluation function (AIC, BIC, or CV)
- Termination condition (No improvement or fixed number of iterations)

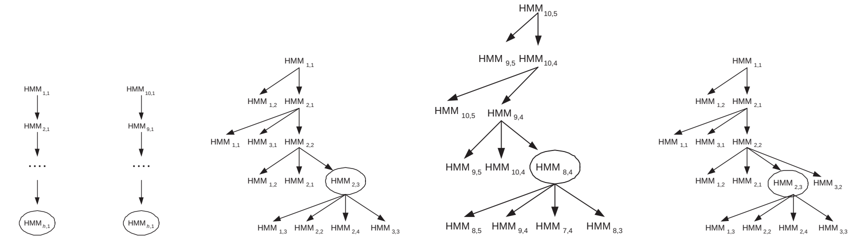
MSUMO Operators

- REMOVE-1: Remove a single hidden state from the HMM.
- ADD-1: Add a single hidden state to the HMM.
- REMOVE-L: Add a new Gaussian to the mixture.
- ADD-L: Remove a Gaussian from the mixture.

MSUMO Pseudocode

```
BEST = initial network
while BEST changed
    for each applicable operator OPER_i
        C_i ← OPER_i(BEST)
    Sort candidates C_i in the order of complexity
    for i = 1 to number of candidates
        Train and validate C_i on k folds
        if C_i is more complex than BEST
            Test H_0 : μ_BEST ≤ μ_C_i
            if H_0 is rejected
                BEST ← C_i
                break
        else
            Test H_0 : μ_C_i ≤ μ_BEST
            if H_0 is accepted
                BEST ← C_i
                break
return BEST;
```

Five MSUMO Variants



1-Fw 1-Bw Fw Bw MFw

Experimental Factors

- MSUMO variant used in search (1-Fw, 1-Bw, Fw, Bw, and MFw)
- Statistical test used in comparison (k-fold paired t test)
- Confidence level $(1 - \alpha)$ of the test (0.95)
- Correction used when applying multiple tests (Holm correction)

HMM Training

- HMM toolbox implemented by Kevin Murphy
- Retraining of the architecture when an operator is applied
- No probabilities are kept or frozen

Sign Language Datasets

Table: Properties of data sets.

	size	signs	features
IDIAP	490	7	20
eNTERFACE'06	760	19	32
British	980	91	22
Australian	6650	95	8

Optimal Architecture

- Exhaustive search over 50 architectures
- Hidden states from 1 to 10
- Gaussian mixtures with 1 to 5 components

Comparison Criteria

- The accuracy of the estimated architecture
- The complexity of the estimated architecture
- Computational complexity of the search until an architecture is found

Two measures used in comparisons

Table: An example for calculation of ranks.

	Fw	Bw	MFw	1-Fw	1-Bw
Order	3	2	1	1	2
# states	10	5	3	4	5
Rank	5	3.5	1	2	3.5

- *Order*: Has the optimal architecture in position one, second best architecture in position two, and the worst architecture in the last position
- *Rank*: Takes also into account how fast we get to the final state, which uses the number of states visited as a measure of the complexity of search

Comparison of MSUMO variants using no statistical test

Average	MFw	1-Fw	1-Bw	Fw	Bw	Best
Rank	2.8	3.9	4.2	2.6	1.6	
Order	9	37	36	8	5	
# states	14	4	2	12	5	
Error	28.18	36.71	34.28	28.48	28.00	27.17

Comparison of MSUMO variants using k-fold paired t test with no correction

Average	MFw	1-Fw	1-Bw	Fw	Bw	Best
Rank	3.2	2.9	3.6	2.9	2.3	
Order	26	35	29	25	9	
# states	5	2	7	4	15	
Error	35.57	39.67	34.94	35.26	29.57	28.27

Comparison of MSUMO variants using k-fold paired t test with Holm correction

Average	MFw	1-Fw	1-Bw	Fw	Bw	Best
Rank	3.0	2.5	3.8	2.7	3.0	
Order	18	26	24	17	10	
# states	5	2	7	4	15	
Error	36.28	39.67	34.94	35.28	29.6	31.00

Summary

- Proposed the MSUMO algorithm for model selection in HMM
- Five variants including forward/backward search with single and multiple operands
- Compared architectures w.r.t. expected error, complexity and distance to the optimal algorithm

Conclusions

- Conservativity of the selection criteria increases, the algorithm tends to find simpler models.
- If only error is important, we should use MSUMO with no statistical tests.
- Bw finds the most accurate architectures, but with an increased number of visited states compared to other algorithms.

Future Work

- Architectures where states contain different gaussian mixtures
- Jumps with multiple steps (Heuristics)

Questions?