

Discriminatively Trained Markov Model for Sequence Classification

Oksana Yakhnenko

Adrian Silvescu

Vasant Honavar

Artificial Intelligence Research Lab

Iowa State University

ICDM 2005

Outline

- Background
- Markov Models
- Generative vs. Discriminative Training
- Discriminative Markov model
- Experiments and Results
- Conclusion

Sequence Classification

Σ – alphabet

s in Σ^* - sequence

$C = \{c_1, c_2, \dots, c_n\}$ - a set of class labels

Goal: Given $D = \{ \langle s^i, c^i \rangle \}$ produce a hypothesis

$h: \Sigma^* \rightarrow C$ and assign $c = h(s)$ to an unknown sequence s from Σ^*

Applications

computational biology

- protein function prediction, protein structure classification...

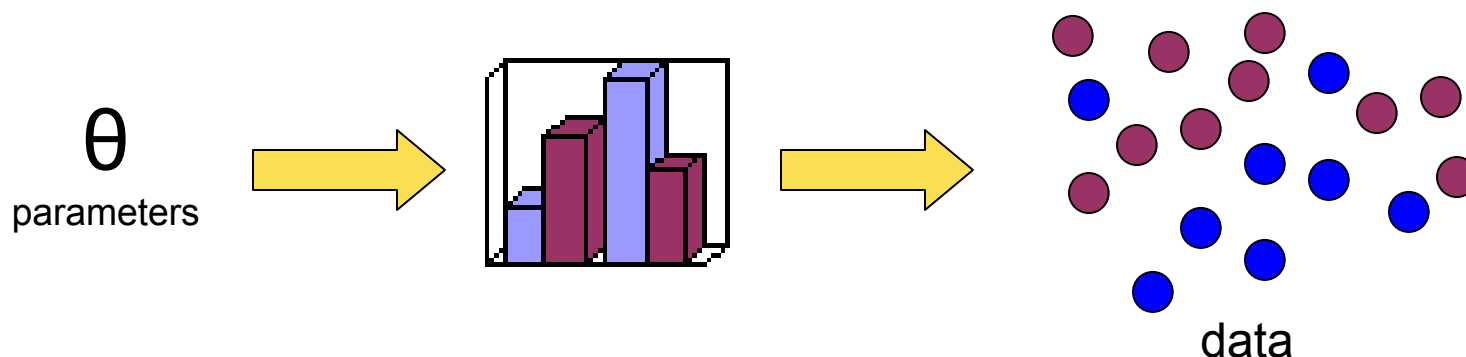
natural language processing

- speech recognition, spam detection...

etc.

Generative Models

- Learning phase:
 - Model the process that generates the data
 - assumes the parameters specify probability distribution for the data



- learns the parameters that maximize joint probability distribution of example and class: $P(x,c)$

Generative Models

Classification phase:

Assign the most likely class to a novel sequence s

$$c = \arg \max_{c_j} P(s, c_j)$$

Simplest way – Naïve Bayes assumption:

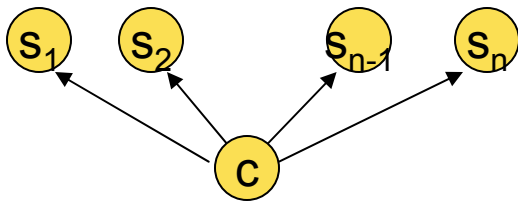
- assume all features in s are *independent* given c_j ,
- estimate $P(s_i|c_j), 0 \leq i \leq \text{length}(s)$ $P(c_j)$
- $P(s, c_j) = P(c_j) \prod_{i=1}^{\text{length}(s)} P(s_i|c_j)$

Outline

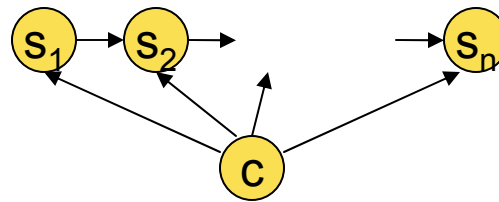
- Background
- **Markov Models**
- Generative vs. Discriminative Training
- Discriminative Markov model
- Experiments and Results
- Conclusion

Markov Models

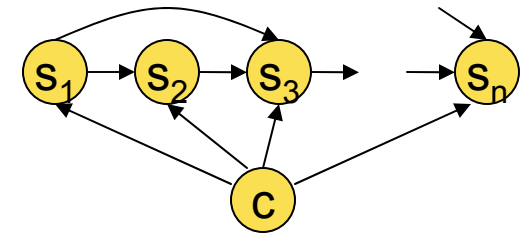
- Capture dependencies between elements in the sequence



Markov Model of order 0
(Naïve Bayes)



Markov Model of order 1



Markov Model of order 2

- Joint probability can be decomposed as a product of an element given its predecessors

Markov Models of order $k-1$

- In general, for k dependencies full likelihood is

$$P(S = s_1 \dots s_n, c_j) = P(s_1 \dots s_{k-1} | c_j) \prod_{i=1}^n P(s_i | s_{i-1} \dots s_{i-k+1} | c_j)$$

- Two types of parameters that have closed-form solution and can be estimated in one pass through data

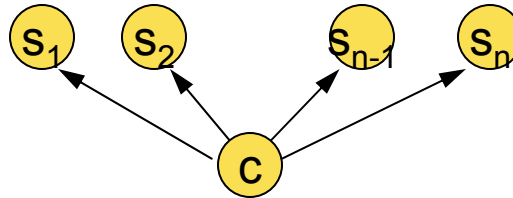
$$\left. \begin{array}{l} P(s_1 s_2 \dots s_{k-1}, c) \\ P(s_i | s_{i+1} \dots s_{i+k-1}, c) \end{array} \right\} \text{ sufficient statistics}$$

- Good accuracy and expressive power in protein function prediction tasks [Peng & Schuurmans, 2003], [Andorf et. al 2004]

Outline

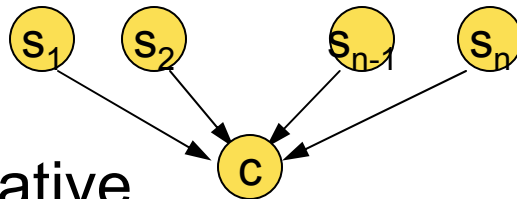
- Background
- Markov Models
- **Generative vs. Discriminative Training**
- Discriminative Markov model
- Experiments and Results
- Conclusion

Generative vs. discriminative models



- Generative

- Parameters are chosen to maximize full likelihood of features and class
- Less likely to overfit



- Discriminative

- Solve classification problem directly
 - Model a class given the data (least square error, maximum margin between classes, most-likely class given data, etc)
- More likely to overfit

How to turn a generative trainer into discriminative one

- Generative models give joint probability

$$P(x, c_j)$$

- Find a function that models the class given the data

$$P(c_j|x) = \frac{P(x, c_j)}{\sum_{c_k \in C} P(x, c_k)}$$

- No closed form solution to maximize class-conditional probability
 - use optimization technique to fit the parameters

Examples

- Naïve Bayes \leftrightarrow Logistic regression [Ng & Jordan, 2002]
 - With sufficient data discriminative models outperform generative
- Bayesian Network \leftrightarrow Class-conditional Bayesian Network [Grossman & Domingos, 2004]
 - Set parameters to maximize full likelihood (closed form solution), use class-conditional likelihood to guide structure search
- Markov Random Field \leftrightarrow Conditional Random Field [Lafferty et. al, 2001]

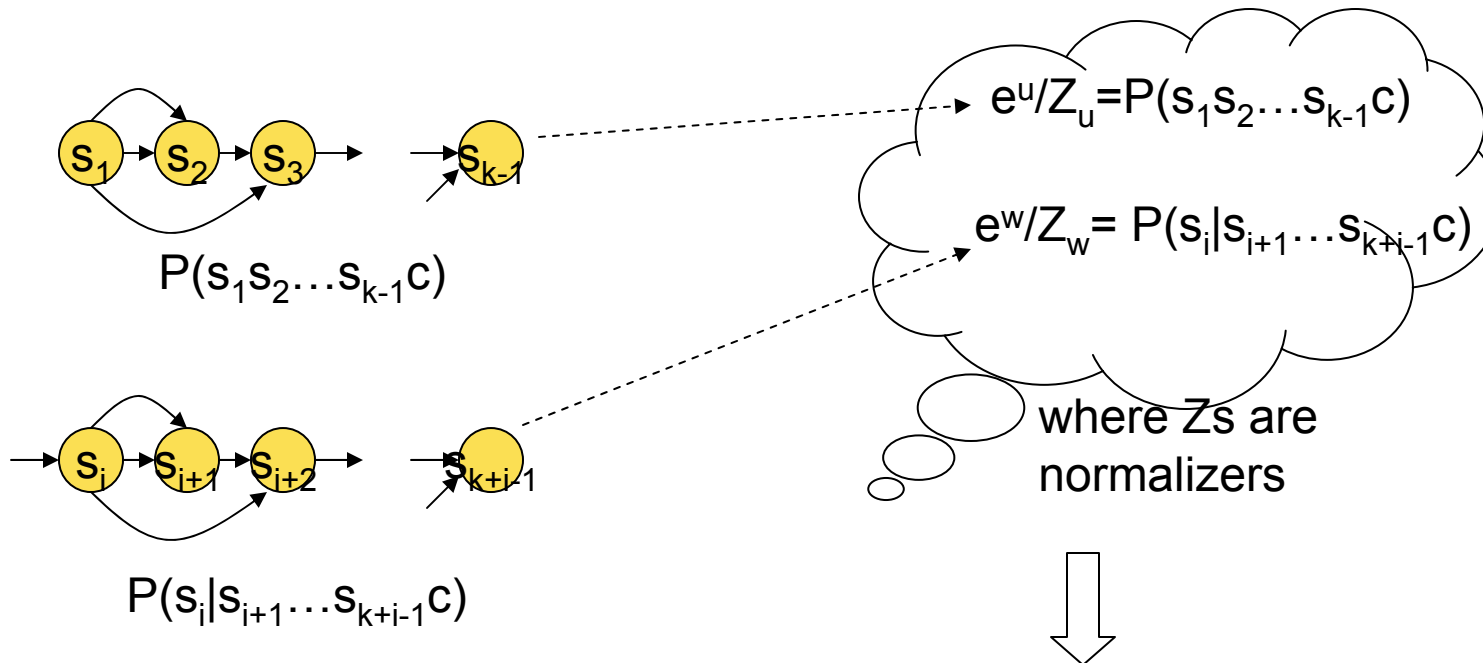
Outline

- Background
- Markov Models
- Generative vs. Discriminative Training
- Discriminative Markov model
- Experiments and Results
- Conclusion

Discriminative Markov Model

1. Initialize parameters with full likelihood maximizers
2. Use gradient ascent to chose parameters to maximize $\log P(c|S)$:
$$P(\text{k-gram})_{t+1} = P(\text{k-gram})_t + \alpha \nabla \text{CLL}$$
3. Reparameterize P's in terms of weights
 - probabilities need to be in $[0, 1]$ interval
 - probabilities need to sum to 1
4. To classify - use weights to compute the most likely class

Reparameterization



1. Initialize by joint likelihood estimates,
2. Use gradient updates for w 's and u 's instead of probabilities

$$w^{t+1} = w^t + \partial \text{CLL} / \partial w$$

$$u^{t+1} = u^t + \partial \text{CLL} / \partial u$$

Parameter updates

On-line, per sequence updates

The final updates are:

$$\frac{\partial CLL}{\partial w_{i_1 \dots i_k c_q}} = \text{count}[i_1 \dots i_{k-1} : S] \left(\frac{\text{count}[i_1 \dots i_k : S]}{\text{count}[i_1 \dots i_{k-1} : S]} - \frac{e^{w_{i_1 \dots i_k c_q}}}{Z_w} \right) (\delta(c_j, c_q) - P(c_q | S))$$

$$\frac{\partial CLL}{\partial u_{i_1 \dots i_{k-1} c_q}} = \left(\delta([i_1 \dots i_{k-1} : S], [S_1 \dots S_{k-1}]) - \frac{e^{u_{i_1 \dots i_{k-1} c_q}}}{Z_u} \right) (\delta(c_j, c_q) - P(c_q | S))$$

CLL is maximized when:

- weights are close to probabilities
- probability of true class given the sequence is close to 1

Algorithm

Training:

1. Initialize parameters with estimates according to generative model
2. Until termination condition met
 - for each sequence s in the data
 - update the parameters w and u with gradient updates ($d\text{CLL}/dw$ and $d\text{CLL}/du$)

Classification:

- Given new sequence S , use weights to compute $c = \text{argmax}_{c_j} P(c_j|S)$

Outline

- Background
- Markov Models
- Generative vs. Discriminative Training
- Discriminative Markov model
- Experiments and Results
- Conclusion

Data

- Protein function data: families of human kinases. 290 examples, 4 classes [Andorf et. al 2004]
- Subcellular localization [Hua & Sun, 2001]
 - Prokaryotic 997 examples, 3 classes
 - Eukaryotic 2427 examples, 4 classes
- Reuters-21578 text categorization data: 10 classes that have the highest number of examples [Lewis, 1997]

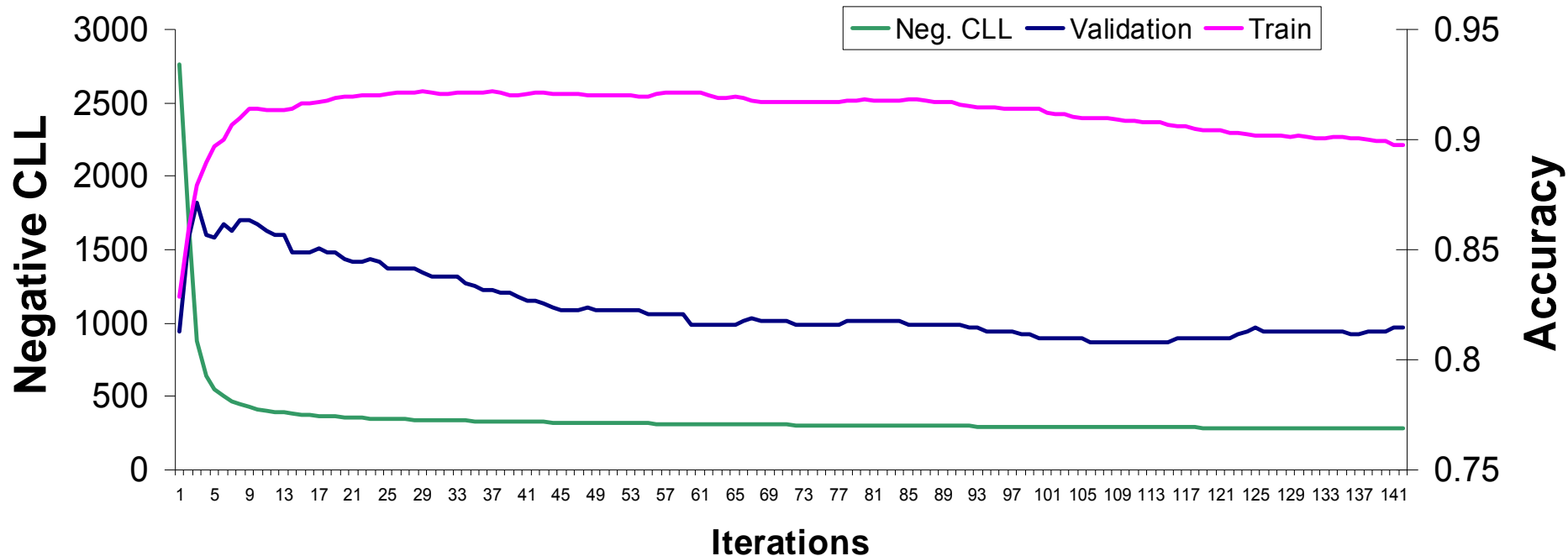
Experiments

- Overfitting?
 - 90% for training, 10% for validation
 - Record accuracies on training and validation data; and value of negative CLL at each iteration
- Performance comparison
 - compare with SVM that uses k-grams as feature inputs (equivalent to string kernel) and generative Markov model
 - 10-fold cross-validation
 - collective classification for kinase data
 - one-against-all for localization and text data

CLL, accuracy on training vs. validation sets

Localization prediction data for “nuclear” class

Nuclear k=2

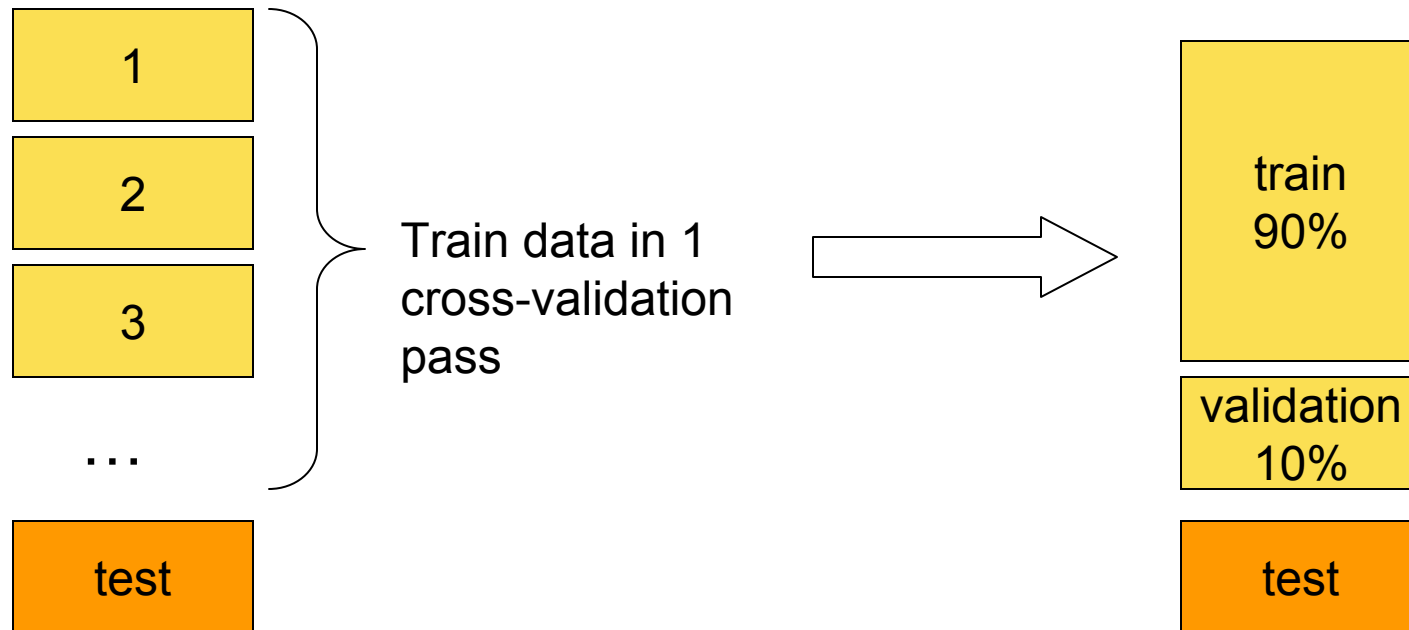


Results on overfitting

- Overfitting occurs in most cases
- Accuracy on unseen data increases and drops when accuracy on train data and CLL continue to increase
- Accuracy on validation data is at its maximum (after 5-10 iterations) not when CLL is converged (a lot longer)
- Use early termination as a form of regularization

Experiments:

- Pick the parameters that yield the highest accuracy on the validation data in the first 30 iterations or after convergence (whichever happens first)



Results

- Collective classification for Protein Function data
- One against all for Localization and Reuters
- Evaluate using different performance measures: accuracy, specificity, sensitivity, correlation coefficient

k	DMM	MM	SVM
2	89.01±5.89	87.11±7.2	90.7
3	91.55±5.82	91.13±5.5	90.3
4	78.48±6.5	78.32±8.8	x

Kinase (protein function prediction) data

Results

Class	k	Accuracy			Specificity		
		DMM	MM	SVM	DMM	MM	SVM
Peri	2	82.91	84.07	90.17	87.77	87.67	92.7
	3	86.78	87.2	88.26	97.74	97.61	97.23
	4	82.57	82.13	x	85.56	85.1	x
Extr	2	95.05	94.01	93.68	98.52	96.09	97.64
	3	94.78	94.88	94.88	99.17	99.08	99.44
	4	86.84	86.44	x	88.4	87.78	x
Ctpl	2	90.01	90.59	90.17	70.61	76.89	81.55
	3	91.35	91.21	92.47	75.47	75.92	79.94
	4	86.36	86.02	x	81.36	81.36	x
		Sensitivity			Correlation Coefficient		
	k	DMM	MM	SVM	DMM	MM	SVM
Peri	2	63.76	69.99	94.04	0.5	0.54	0.76
	3	43.66	46.23	62.97	0.54	0.55	0.6
	4	70.79	70.36	x	0.52	0.51	x
Extr	2	66.17	78.5	60.74	0.72	0.71	0.64
	3	58.32	60	57.01	0.7	0.7	0.7
	4	73.84	75.33	x	0.5	0.5	x
Ctpl	2	98.78	96.74	94.04	0.77	0.77	0.77
	3	98.49	98.08	91.11	0.79	0.79	0.82
	4	88.6	88.1	x	0.68	0.68	x

Prokaryotic

Class	k	Accuracy			Specificity		
		DMM	MM	SVM	DMM	MM	SVM
Ctpl	2	77.72	73.84	81.87	77.22	69.76	90.99
	3	86	85.9	88.01	91.28	90.76	93.2
	4	89.29	88.96	x	97.76	97.41	x
Extr	2	92.13	89.7	92.45	98.76	93.82	98.47
	3	94.19	94.15	95.49	99.95	99.8	99.33
	4	95.59	95.67	x	99.8	99.7	x
Mit	2	82.37	75.44	88.46	83.76	76.83	97.86
	3	89.74	89.67	91.14	98.2	98.2	97.86
	4	90.44	90.56	x	98.6	98.6	x
Nclr	2	80.55	83.48	85.74	71.88	90.45	90.45
	3	87.52	87.97	89.25	92.4	92.48	92.03
	4	88.55	88.13	x	87.07	87.7	x
		Sensitivity			CC		
	k	DMM	MM	SVM	DMM	MM	SVM
Ctpl	2	78.94	84.21	58.63	0.52	0.49	0.53
	3	72.51	73.68	73.78	0.65	0.65	0.69
	4	67.69	67.4	x	0.73	0.72	x
Extr	2	49.23	63.08	53.54	0.56	0.45	0.63
	3	56.92	57.53	70.77	0.72	0.72	0.79
	4	70.77	69.54	x	0.81	0.8	x
Mit	2	73.2	66.36	26.79	0.45	0.32	0.37
	3	34.27	33.01	47.04	0.46	0.45	0.56
	4	36.76	37.69	x	0.5	0.51	x
Nclr	2	91.07	75.23	80.04	0.63	0.67	0.71
	3	81.59	82.5	85.87	0.75	0.75	0.71
	4	90.33	88.61	x	0.77	0.76	x

Eukaryotic

Results

Reuters data

	<i>Accuracy</i>		<i>Sensitivity</i>	
<i>data</i>	DMM	MM	DMM	MM
acq	95.27	95.82	89.53	88.01
corn	98.21	98.21	64.29	67.86
crude	97.35	97.13	90.86	89.24
earn	96.59	97.86	97.87	95.83
grain	97.49	97.5	89.86	85.81
interest	97.5	97.4	51.1	47.32
money-fx	97.07	97.34	82.12	79.89
ship	98.86	98.9	85.06	79.31
trade	97.5	98.01	80.17	75.86
wheat	98.13	98.25	70.42	76.06
	<i>Specificity</i>		<i>Correlation Coefficient</i>	
<i>data</i>	DMM	MM	DMM	MM
acq	96.87	98	0.86	0.87
corn	98.91	98.7	0.55	0.56
crude	97.75	97.6	0.79	0.77
earn	95.96	98.87	0.92	0.95
grain	97.85	98.1	0.76	0.75
interest	99.4	99.5	0.63	0.61
money-fx	97.93	98.35	0.74	0.75
ship	99.24	99.49	0.8	0.8
trade	98.13	98.8	0.69	0.72
wheat	98.75	98.75	0.62	0.65

Results - performance

Kinase

- 2% improvement over generative Markov model
- SVM outperforms by 1%

Prokaryotic

- Small improvement over generative Markov model and SVM (extracellular), other classes similar performance as SVM

Eukaryotic

- 4%, 2.5%, 7% improvement in accuracy over generative Markov model on Cytoplasmic, Extracellular and Mitochondrial
- Comparable to SVM

Results - performance

- Reuters
 - Generative and discriminate approaches have very similar accuracy
 - Discriminative show higher sensitivity, generative show higher specificity
- Performance is close to that of SVM without the computational cost of SVM

Results – time/space

- Generative Markov model needs one pass through training data
- SVM needs several passes through data
 - Needs kernel computation
 - May not be feasible to compute kernel matrix for $k > 3$
 - If kernel is computed as needed, can significantly slow down one iteration
- Discriminative Markov model needs a few passes through training data
 - $O(\text{length of sequence} \times \text{alphabet size})$ for one sequence

Conclusion

- Initializes parameters in one pass through data
- Requires few passes through data to train
- Significantly outperforms generative Markov model on large datasets
- Accuracy is comparable to SVM that uses string kernel
 - Significantly faster to train than SVM
 - Practical for larger datasets
- Combines strengths of generative and discriminative training

Future work

- Development of more sophisticated regularization techniques
- Extension of the algorithm to higher-dimensional topological data (2D/3D)
- Application to other tasks in molecular biology and related fields where more data is available

Thank you!