University of Illinois Fall 2018

ECE 544NA: Pattern Recognition
Lecture 19: October 30

Lecturer: Alexander Schwing Scribe: Dongwei Shi

1 Goals

- Getting to know Structured Latent Variable Models
- Learning about Markrov Hidden Models

2 Recaps

In previous lectures, we have gone through two types of machine learning models, Generative models and Discriminative models. Generative models model the distribution of individual class. Discriminative models, on the other hand, learn the boundary between classes. The two models can be written as follows

Generative:

$$\ln p_\theta(x^i) = \ln \sum_n p_\theta(x^i, z) \tag{1}$$

Discriminative:

$$\ln p_w(y|x^i) \tag{2}$$

There are several observations between these two models. In Generative model, variable z is a categorical variable and never observed. The variable z are not directly observed but are rather inferred from other variables that are observed. In Discriminative model, however, the variable y is fully observed. Now a question might came up, how to train a model with latent variables or variables without fully observed.

2.1 Learning with full observations

Given a dataset $D = \{x^i, y^i\}$, where x^i is the data and y^i is the ground truth (labeled). For example, we are given an image segmentation dataset in Figure 1. From the dataset above, the

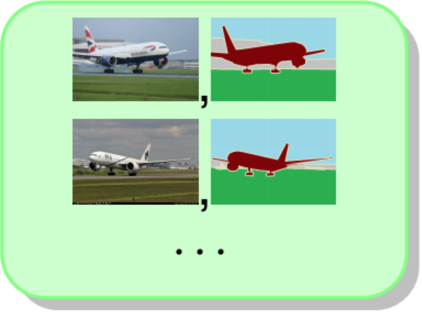


Figure 1: Training Dataset with full observations

plane images on the left side indicate the data sample x^i, and the ground truth palette image on the right side indicate label y^i. In the palette image y^i, we observed that the sky, plane and ground are fully labeled with different palette, which means our model could fully observed those information in the training process. With fully observed dataset, the inference process of our model is simple represented as:

$$\arg\max_{\hat{y}} F(w, x^i, \hat{y}) = \arg\max_{\hat{y}} \sum_r f_r(w, x^i, \hat{y}) \tag{3}$$

From previous lecture, the scoring function $F(w, x^i, \hat{y})$ of the image segmentation problem, could be decomposed to summation of scoring function of a restricted set $r \in R$. Each $f_r(w, x^i, \hat{y})$ is represented as an unary term or a pairwise. The more details are included in lecture 13 and lecture 14. The learning target could be written as:

$$\max_{\hat{y}} F(w, x^i, \hat{y}) \leq F(w, x^i, y^i)) \tag{4}$$

Since we want to make our predicted scoring approximate to the ground truth scoring. Thus, we could use hinge loss to penalize whenever maximum is with in a margin $L(\hat{y}, y^i)$ of the data (x^i, y^i) score. Now our learning target has transformed to :

$$\max_{\hat{y}}(F(w, x^i, \hat{y}) + L(\hat{y}, y^i)) \geq F(w, x^i, y^i)) \tag{5}$$

Thus our learning framework with full observation could be written as:

$$\min_w \frac{C}{2}||w||_2^2 + \sum_{i \in D}(\max_{\hat{y}}(F(w, x^i, \hat{y}) + L(\hat{y}, y^i)) - F(w, x^i, y^i))) \tag{6}$$

Now we given a weakly labeled dataset, we could observe in figure 2 that some objects are not labeled. For example, the land and the sky is not labeled in ground truth palette image. Those variables will not be observed by the learning framework in the training process. Unlike the previous fully observed data sample, now our data sample within the weakly labeled dataset can be completed as $y = (s, z)$, where s is observed data and z is latent variable (not observed). We want to penalize whenever best overall prediction exceeds best prediction with annotation being clamped, thus the weakly labeled hinge loss:

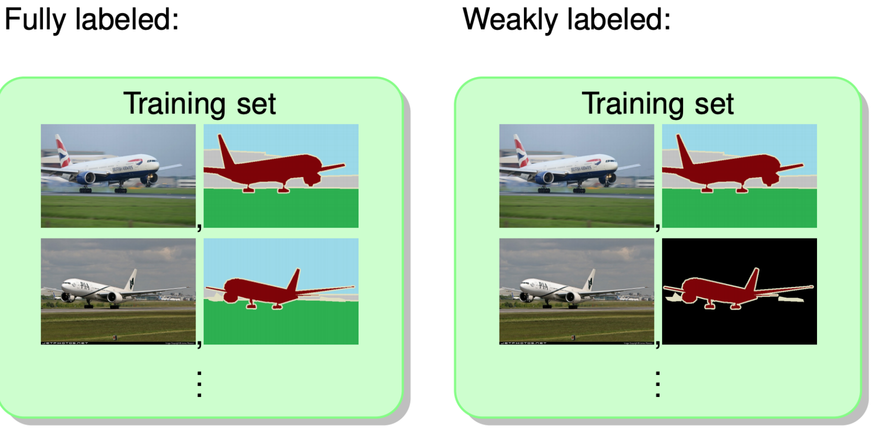


Figure 2: weakly labeled dataset

$$\max_{\hat{s}, \hat{z}} F(w, x^i, \hat{s}, \hat{z}) \geq \max_{\hat{z}} F(w, x^i, s^i, \hat{z}) \tag{7}$$

2.2 Latent SSVM and Hidden CRFs

Now we have Latent SSVM (LSSVM):

$$\min_{w} \frac{C}{2}||w||_2^2 + \sum_{i \in D}(\max_{\hat{s},\hat{z}} F(w, x^i, \hat{s}, \hat{z})) - \sum_{i \in D}(\max_{\hat{z}} F(w, x^i, s^i, \hat{z})) \tag{8}$$

As the ϵ approximate to 0, the soft-max function is given below:

$$\epsilon \ln \sum_{\hat{s},\hat{z}} \exp \frac{F(w, x^i, \hat{s}, \hat{z})}{\epsilon} \xrightarrow{\epsilon \to 0} \max_{\hat{s},\hat{z}} F(w, x^i, \hat{s}, \hat{z}) \tag{9}$$

By plugging the soft-max function (9) in to equation (8), we can obtain a learning framework for structured prediction with latent variables:

$$\min_{w} \frac{C}{2}||w||_2^2 + \sum_{i \in D} \epsilon \ln \sum_{\hat{s},\hat{z}} \exp \frac{F(w, x^i, \hat{s}, \hat{z})}{\epsilon} - \sum_{i \in D} \epsilon \ln \sum_{\hat{s},\hat{z}} \exp \frac{F(w, x^i, \hat{s}, \hat{z})}{\epsilon} \tag{10}$$

Note that as $\epsilon = 0$, the above learning framework is LSSVM with hinge-loss and max-margin. As $\epsilon \to 1$, the above learning framework is Hidden CRFs (HCRF) with log-loss and max-likelihood. By adding the margin L we now have obtained:

$$\min_{w} \frac{C}{2}||w||_2^2 + \sum_{i \in D} \epsilon \ln \sum_{\hat{s},\hat{z}} \exp \frac{F(w, x^i, \hat{s}, \hat{z}) + L_i(\hat{s}, \hat{z})}{\epsilon} - \sum_{i \in D} \epsilon \ln \sum_{\hat{s},\hat{z}} \exp \frac{F(w, x^i, \hat{s}, \hat{z}) + L_i^c(s^i, \hat{z})}{\epsilon} \tag{11}$$

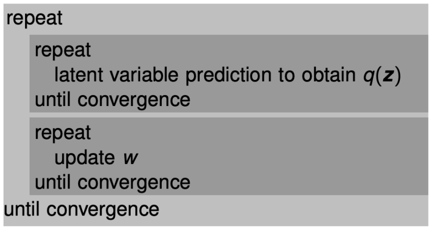
Recall in lecture 18, we use Jensen's inequality established the variational expression for partition function.

$$\epsilon \ln \sum_{z} \exp \frac{F(w, x^i, s^i, z)}{\epsilon} = \max_{q(z) \in \Delta} (\sum_{z} q(z) F(w, x^i, s^i, z) + \epsilon H(q(z))) \tag{12}$$

Due to the similarity with the structured prediction, then similar algorithm could be employed:

$$\min_{w} \frac{C}{2}||w||_2^2 + \sum_{i \in D} \epsilon \ln \sum_{\hat{s},\hat{z}} \exp \frac{F(w, x^i, \hat{s}, \hat{z}) + L_i(\hat{s}, \hat{z})}{\epsilon} - \max_{q(z) \in \Delta} (\sum_{z} q(z)(F(w, x^i, s^i, z) + L_i^c(s^i, \hat{z})) + \epsilon H(q(z))) \tag{13}$$

To train this model, we can use EM approach to optimize $q(z)$ and w alternatively. The algorithm structure presented in Figure 3



```
repeat
    repeat
        latent variable prediction to obtain $q(z)$
    until convergence
    repeat
        update $w$
    until convergence
until convergence
```

Figure 3: Algorithm structure

3 Hidden Markrov Model

Hidden Markrov Model is an example with latent variable. As with Gaussian Mixture Model (GMM), we have a series of observed random variables $(X_1, X_2, ..., X_n)$, and a series of latent variables $(Y_1, Y_2, ..., Y_n)$. Like Gaussian Mixture Model, the Hidden Markrov model also has conditional independence of observations given latent variables:

$$p(X_i|X_1, ...X_t, Y_1, ..., Y_t) = p(X_i|Y_i) \tag{14}$$

However, unlike GMM model, the HMM latent variables have dependencies. That is markrov assumption: Y_{i-1} depends only on Y_i. The hidden Markrov model can be represented as a graphical model in figure 4.

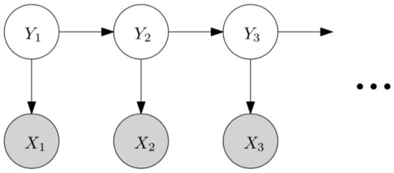


Figure 4: HMM Graphical Model

Then the likelihood of this graphical Model above is:

$$\begin{aligned} p(X_1, ..., X_n, Y_1, ..., Y_n) &= p(Y_1)p(X_1, ..., X_n, Y_2, ..., Y_n|Y_1) \\ &= p(Y_1)p(X_1|Y_1)p(X_2, ..., X_n, Y_2, ..., Y_n|Y_1) \\ &= p(Y_1)p(X_1|Y_1)p(X_2|Y_2)p(Y_2|Y_1)p(X_3, ..., X_n, Y_2, ..., Y_n|Y_2, Y_1) \\ &= p(Y_1)p(X_1|Y_1)p(X_2|Y_2)p(Y_2|Y_1)p(X_3, ..., X_n, Y_2, ..., Y_n|Y_2) \\ &= p(Y_1)(\prod_{i=1}^{n} p(X_i|Y_i))(\prod_{i=2}^{n} p(Y_i|Y_{i-1})) \end{aligned} \tag{15}$$

Now formally, we have an HMM mdoel for which we have a series of observed outputs $X = \{x_1, x_2, ..., x_T\}$ drawn from a set of observed states $V = \{v_1, v_2, ..., v_{|V|}\}$, where $x_t \in V$ and a series of latent outputs $Y = \{y_1, y_2, ..., y_T\}$ drawn from a set of unobserved states $S = \{s_1, s_2, ..., s_{|S|}\}$. For $(y_2, ..., y_T)$, we can have transition probabilities $p(y_{i+1} = i|y_i = j)$ .The transition probabilities between states i and j can be represented as a matrix $A \in [0, 1]^{k \times k}$. Similarly, according to the output independence assumption, we can define$P(x_t = v_j|y_t = s_k) = P(x_t = v_j|x_1, ..., x_T, y_1, ..., y_T) = B_{kj}$. The matrix B encodes the probability of hidden state generating the output v_j given that the state at corresponding time was s_k. With matrix A and matrix B, we can simplify our likelihood expression as:

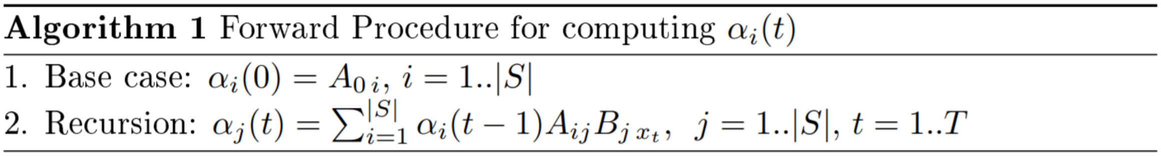
$$\begin{aligned} P(X; A, B) &= \sum_{Y} P(X|Y; A, B)P(Z; A, B) \\ &= \sum_{Y} (\prod_{t=1}^{T} P(x_t|y_t; B))(\prod_{t=1}^{T} P(y_t|y_{t-1}; A)) \\ &= \sum_{Y} (\prod_{t=1}^{T} B_{y_t x_t})(\prod_{t=1}^{T} A_{y_{t-1} y_t}) \end{aligned} \tag{16}$$

4 Forward Procedure

Now we have our Markrov Model represented in terms of parameters using A and B. However, the latent space y is very large, because y_t can take one of the $|S|$ possible values at each time step, evaluate this directly will require $O(|S|^T)$ operations. A faster way of computing $P(X; A, B)$ is possible using dynamic programming algorithm called the Forward Procedure. We can define a quantity $\alpha_i(t) = P(x_1, x_2, .., x_t, y_t = s_i; A, B)$. $\alpha_i(t)$ represents the total probability of all the observations up through time t and that we are in state s_i at time t. Thus we can have:

$$\begin{aligned} P(X; A, B) &= \sum_{i=1}^{|S|} P(x_1, x_2, .., x_T, y_t = s_i; A, B) \\ &= \sum_{i=1}^{|S|} \alpha_i(T) \end{aligned} \tag{17}$$

So now our computation complexity has dramatically reduced from $O(|S|^T)$ to $O(|S|\dot{T})$. The Forward Procedure for computing $\alpha_i(t)$ is presented in Figure 5:



```
**Algorithm 1** Forward Procedure for computing $\alpha_i(t)$
1. Base case: $\alpha_i(0) = A_{0\,i}$, $i = 1..|S|$
2. Recursion: $\alpha_j(t) = \sum_{i=1}^{|S|} \alpha_i(t-1) A_{ij} B_{j\,x_t}$,  $j = 1..|S|$, $t = 1..T$
```

Figure 5: Forward Procedure

5 The Viterbi Algorithm

For a Hidden Markrov Model, our concern is trying to find the most likely series of hidden states $z \in S^T$ given an observed series of data $x \in V^T$. From Bayes Rule, we could simply get:

$$\arg\max_z P(z|x; A, B) = \arg\max_z \frac{P(x, z; A, B)}{\sum_z P(x, z; A, B)} \tag{18}$$

Naively, we could try every possible assignment to latent variable z and find the maximum among every assignments. However, this approach (exhaustive search) requires $O(|S|^T)$ operations, which is not very efficient. We could use dynamic programming or message pass to estimate the individual $q(z)$. Recall the algorithm structure that presented in Figure 3, we can naively apply the EM (Figure 3) algorithm structure to HMMs. This naive application is called VITERBI ALGORITHM. It is just like the forward procedure except that instead of searching all possibility of generating the observation exhaustively, we need only track that maximum probability and record its corresponding state sequence. The Naive application of EM to HMMs is shown in Figure 6. In our algorithm structure, the $q(z)$ is represented as $Q(z)$ and the w is two transition probabilities A and B.

```
Algorithm 2 Naive application of EM to HMMs
Repeat until convergence {
(E-Step) For every possible labeling $\vec{z} \in S^T$, set
        $Q(\vec{z}) \;:=\; p(\vec{z}|\vec{x}; A, B)$
(M-Step) Set
        $A, B \;:=\; \arg\max_{A,B} \sum_{\vec{z}} Q(\vec{z}) \log \frac{P(\vec{x}, \vec{z}; A, B)}{Q(\vec{z})}$
        $s.t. \;\; \sum_{j=1}^{|S|} A_{ij} = 1,\ i = 1..|S|;\ \ A_{ij} \geq 0,\ \ i, j = 1..|S|$
        $\sum_{k=1}^{|V|} B_{ik} = 1,\ i = 1..|S|;\ \ B_{ik} \geq 0,\ \ i = 1..|S|, k = 1..|V|$
}
```

Figure 6: Algorithm Structure to HMMs

6 Quiz

6.1 Variational expression of partition function?

Answer:

$$\epsilon \ln \sum_z \exp \frac{F(w, x^i, s^i, z)}{\epsilon} = \max_{q(z)\in\Delta} (\sum_z q(z) F(w, x^i, s^i, z) + \epsilon H(q(z))) \tag{19}$$

How to decompose? Answer: using Jensens inequality

$$\begin{aligned} \epsilon \ln \sum_z \exp \frac{F(w, x^i, s^i, z)}{\epsilon} &= \epsilon \ln \sum_z q(z) \exp \frac{F(w, x^i, s^i, z)}{\epsilon q(z)} \\ &= \max_{q(z)\in\Delta} (\sum_z q(z) F(w, x^i, s^i, z) + \epsilon H(q(z))) \end{aligned} \tag{20}$$

6.2 Structured prediction with latent variables?

Answer:

$$\min_w \frac{C}{2} ||w||_2^2 + \sum_{i\in D} \epsilon \ln \sum_{\hat{s},\hat{z}} \exp \frac{F(w, x^i, \hat{s}, \hat{z}) + L_i(\hat{s}, \hat{z})}{\epsilon} - \max_{q(z)\in\Delta} (\sum_z q(z)(F(w, x^i, s^i, z) + L_i^c(s^i, \hat{z})) + \epsilon H(q(z))) \tag{21}$$

Recall that we have introduce a temperature parameter ϵ. ϵ defines an entire family of structured prediction tasks with latent variable. For $\epsilon = 1$, we could obtain the maximum likelihood (HCRF) framework. While $\epsilon = 0$ results in the max-margin formulation for latent variables(LSSVM):

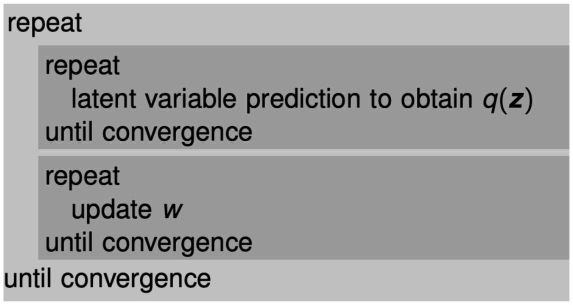
$$\min_w \frac{C}{2} ||w||_2^2 + \sum_{i\in D} \epsilon \ln \sum_{\hat{s},\hat{z}} \exp \frac{F(w, x^i, \hat{s}, \hat{z}) + L_i(\hat{s}, \hat{z})}{\epsilon} - \sum_{i\in D} \epsilon \ln \sum_{\hat{s},\hat{z}} \exp \frac{F(w, x^i, \hat{s}, \hat{z}) + L_i^c(s^i, \hat{z})}{\epsilon} \tag{22}$$

Note that $\epsilon \to 0$ smoothly approximates the max-function via the soft-max:

$$\epsilon \ln \sum_{\hat{s},\hat{z}} \exp \frac{F(w, x^i, \hat{s}, \hat{z})}{\epsilon} \xrightarrow{\epsilon \to 0} \max_{\hat{s},\hat{z}} F(w, x^i, \hat{s}, \hat{z}) \tag{23}$$

6.3 Algorithmic structure of general framework?

Answer:



```
repeat
    repeat
        latent variable prediction to obtain $q(\boldsymbol{z})$
    until convergence
    repeat
        update $\boldsymbol{w}$
    until convergence
until convergence
```

Figure 7: Algorithm structure

7 A Concrete Example: Occasionally dishonest casino

Dealer repeatedly flips a coin. Sometimes the coin is fair, with P(heads) = 0.5, sometimes its loaded, with P(heads) = 0.8. Dealer occasionally switches coins, invisibly to you. We could use a graphical model to address this problem (in Figure 8). After each flip, dealer switches coins with probability 0.4. Given the above information, we can obtain probabilities transition matrix A and E Now, assume we could observe the latent space and we flipped the coins six times, we might get the following situation shown in Figure 10 and the joint probability are obvious given the two

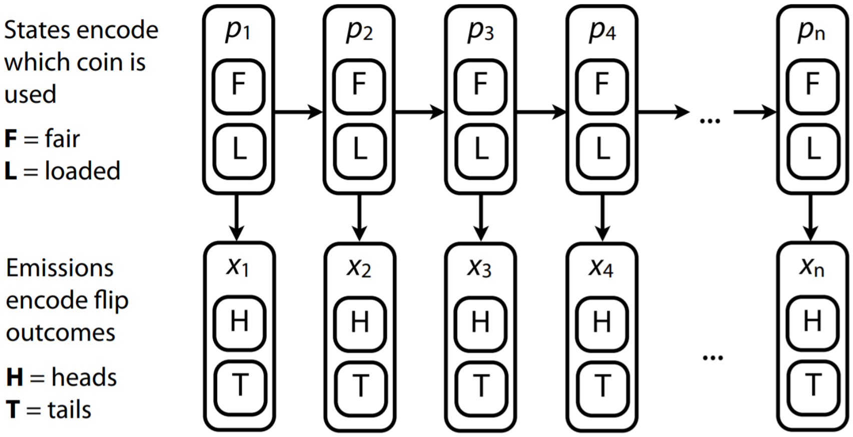
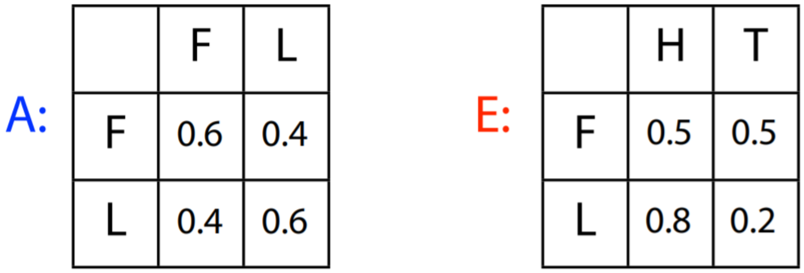


Figure 8: The Graphical Model



A:

	F	L
F	0.6	0.4
L	0.4	0.6

E:

	H	T
F	0.5	0.5
L	0.8	0.2

Figure 9: Transition Probabilities Matrix A and E

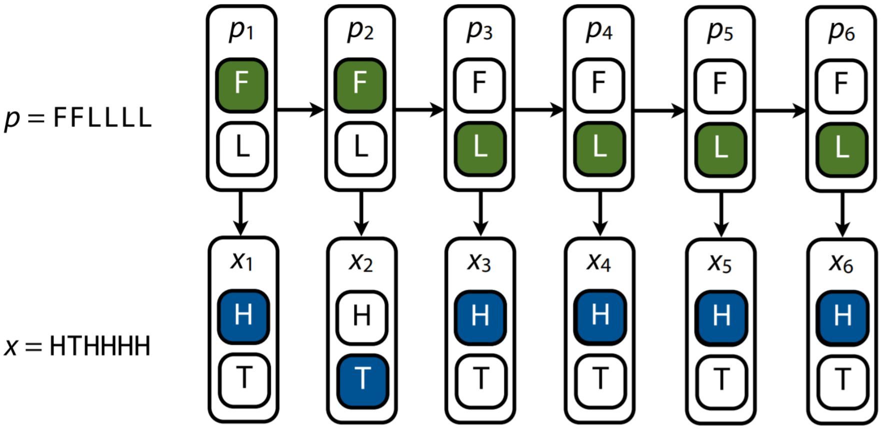
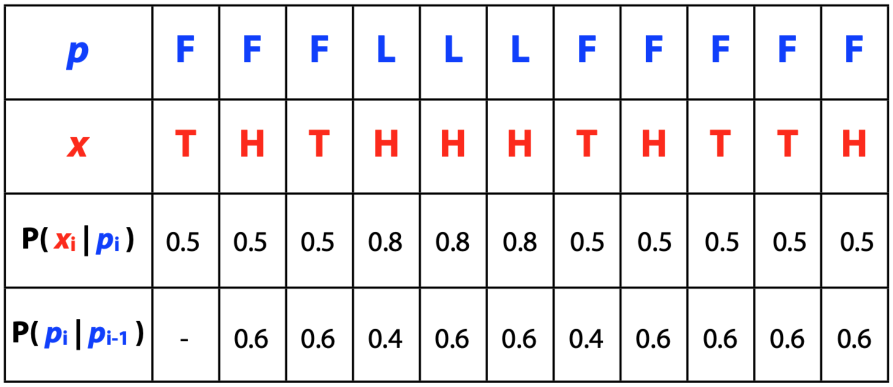


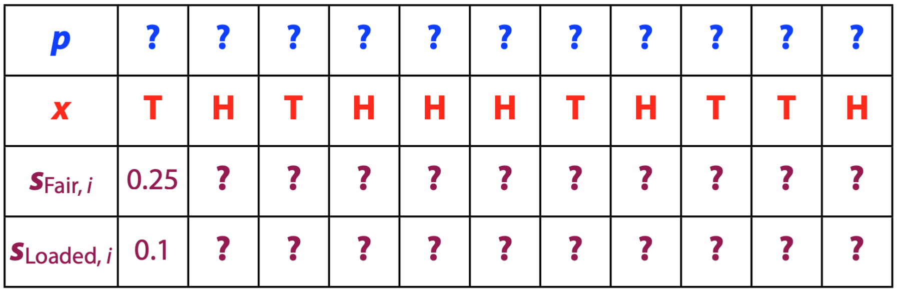
Figure 10: Simulation with full observation



p	F	F	F	L	L	L	F	F	F	F	F
x	T	H	T	H	H	H	T	H	T	T	H
$P(x_i \mid p_i)$	0.5	0.5	0.5	0.8	0.8	0.8	0.5	0.5	0.5	0.5	0.5
$P(p_i \mid p_{i-1})$	-	0.6	0.6	0.4	0.6	0.6	0.4	0.6	0.6	0.6	0.6

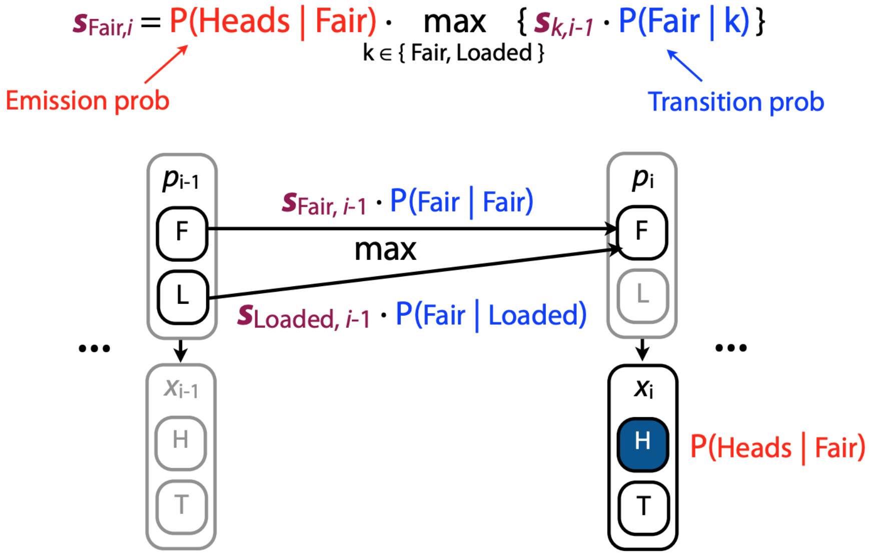
Figure 11: Joint probability table

probabilities transition matrix A and E. From Figure 11, if $P(p_1 = F) = 0.5$, then joint probability $= 0.5^9 0.8^3 0.6^8 0.4^2 = 0.0000026874$. However, in most cases, the variable p is unobserved, we more concerned about the case that latent variable p is unobserved in Figure 12. Recall in previous section, we have talked about Viterbi algorithm. The Viterbi algorithm will helped us filled all the question mark in Figure 12. Since we only care about the total probability of all the observations up through time t and that we are in states s_i at time t. Then the Forward Procedure of this problem is presented in Figure 13.



p	?	?	?	?	?	?	?	?	?	?	?
x	T	H	T	H	H	H	T	H	T	T	H
$s_{\text{Fair}, i}$	0.25	?	?	?	?	?	?	?	?	?	?
$s_{\text{Loaded}, i}$	0.1	?	?	?	?	?	?	?	?	?	?

Figure 12: Joint probability table



$s_{\text{Fair},i} = P(\text{Heads} \mid \text{Fair}) \cdot \max_{k \in \{\text{Fair, Loaded}\}} \{ s_{k,i-1} \cdot P(\text{Fair} \mid k) \}$
Emission prob
Transition prob
p_{i-1}
F
L
$s_{\text{Fair}, i-1} \cdot P(\text{Fair} \mid \text{Fair})$
max
$s_{\text{Loaded}, i-1} \cdot P(\text{Fair} \mid \text{Loaded})$
p_i
F
L
...
...
x_{i-1}
H
T
x_i
H
T
P(Heads | Fair)

Figure 13: Joint probability table

References

[1] Bishop, Christopher M. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Berlin, Heidelberg, 2006.

[2] Alexander G. Schwing, Tamir Hazan, Marc Pollefeys, Raquel Urtasun. *Efficient Structured Prediction with Latent Variables for General Graphical Models* ETH, Zurich, Switzerland

[3] ECE544na, Pattern Recognition, University of Illinois
`https://courses.engr.illinois.edu/ece544na/fa2018/secure/L19-Slides.pdf`

[4] CS229, Machine Learning, Stanford
`http://cs229.stanford.edu/section/cs229-hmm.pdf`

[5] Hidden Markov Models, John Hopkins, lecture slides
`http://www.cs.jhu.edu/ langmea/resources/lecture-notes/hidden-markov-models.pdf`