Machine Learning

4771

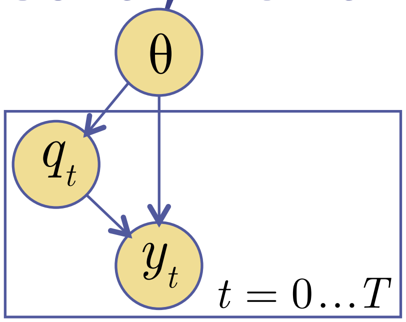
Instructor: Tony Jebara

Topic 19

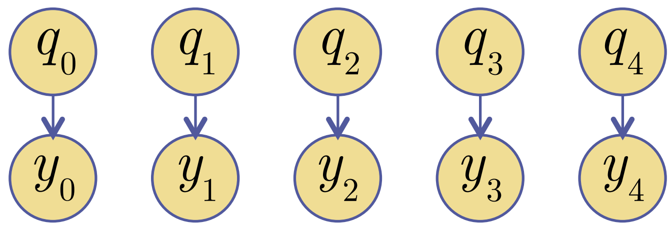
- Hidden Markov Models
- HMMs as State Machines & Applications
- HMMs Basic Operations
- HMMs via the Junction Tree Algorithm

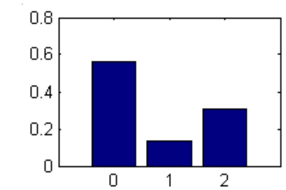
Hidden Markov Models

- A great application of Junction Tree Algorithm with EM
- So far, we have dealt with mixture models with IID:



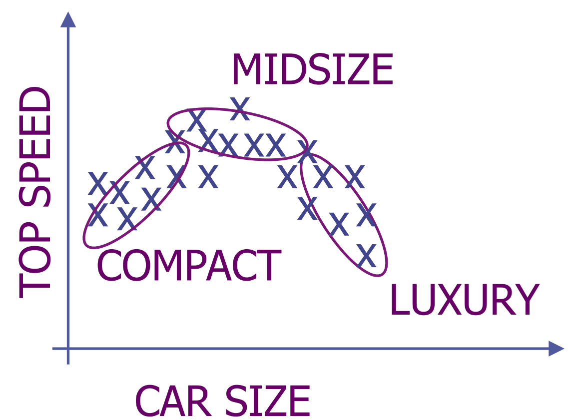
$\Uparrow$





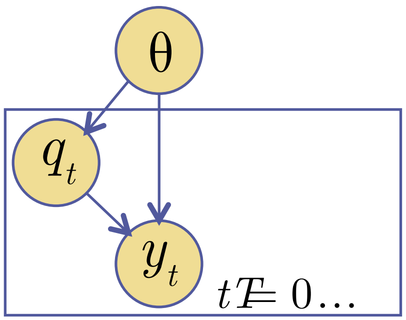
$$p\left(q_t \mid \theta\right) = \begin{bmatrix} .5 & .2 & .3 \end{bmatrix}$$

- Recall mixture of Gaussians and EM…
- Variable q was a multinomial
- Roll a die to determine sub-population:
  q={compact,midsize,luxury}
- Then sample appropriate Gaussian mean and covariance to get
  y=(speed,size)
- Can consider other mixtures too, multinomials, Poisson…

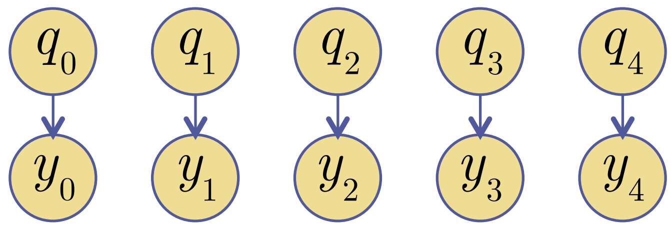


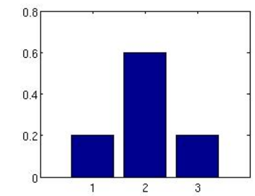
Hidden Markov Models

- Consider mixture of multinomials (dice) y={1,2,3,4,5,6}

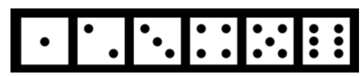


$\Uparrow$

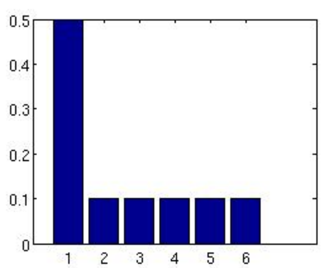




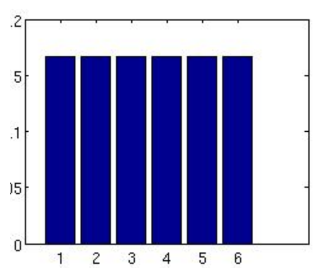
$$p\left(q_t \mid \theta\right) = \left[\begin{array}{ccc} .2 & .6 & .2 \end{array}\right]$$



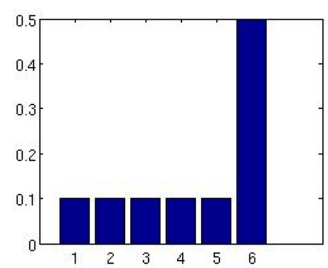
- Example: a crooked casino croupier using mixture of dice.
- You win if he rolls 1,2,3. You lose he rolls 4,5,6.
- Croupier has three dice (one fair & two weighted):



1=helpful



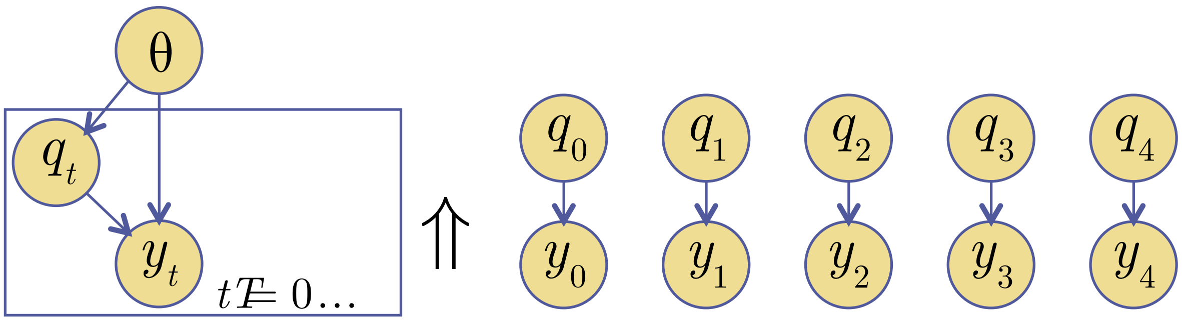
2=fair

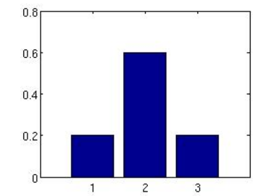


3=adversarial

Hidden Markov Models

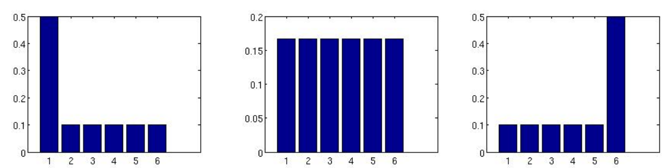
- Consider mixture of multinomials (dice) y={1,2,3,4,5,6}





$$p(q_t \mid \theta) = \begin{bmatrix} .2 & .6 & .2 \end{bmatrix}$$

q={1=helpful,2=fair,3=adversarial}

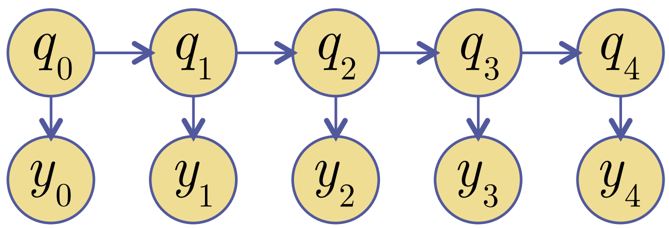


- What if the dealer has a memory or mood? Not IID!

  5646166166 43215341614341634 1113114121
- Dealer might start to like you and roll the helpful die…
- Dealer has a memory of his mood and last type of die q_{t-1}
- Will often use same die for q^t as was rolled before…
- Now, order of $y_0,\ldots,y_T$ matters (if IID order doesn't matter)

Hidden Markov Models

- Since next choice of the dice depends on previous one…



Order of $y_0,\ldots,y_T$ matters
Temporal or sequence model!

- Add left-right arrows. This is a hidden Markov model
- Markov: $future \perp\!\!\!\perp past \mid present$

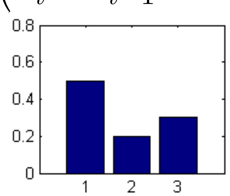
$$p\left(q_t \mid q_{t-1}, q_{t-2}, \ldots, q_1, q_0\right) = p\left(q_t \mid q_{t-1}\right)$$

- From graph, have the following general pdf:

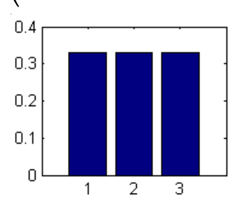
$$p\left(X_U\right) = p\left(q_0\right)\prod_{t=1}^{T} p\left(q_t \mid q_{t-1}\right)\prod_{t=0}^{T} p\left(y_t \mid q_t\right)$$

- So $p(q_t)$ depends on previous state q_{t-1} …

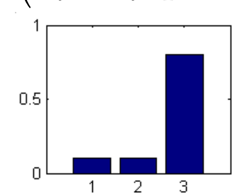
$p\left(q_t \mid q_{t-1} = 1\right)$



$p\left(q_t \mid q_{t-1} = 2\right)$

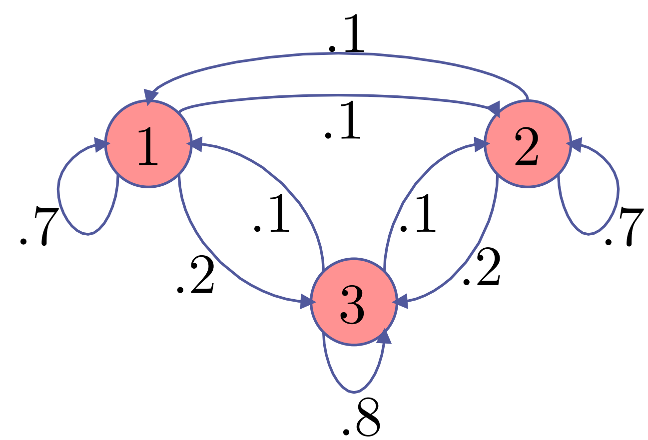


$p\left(q_t \mid q_{t-1} = 3\right)$

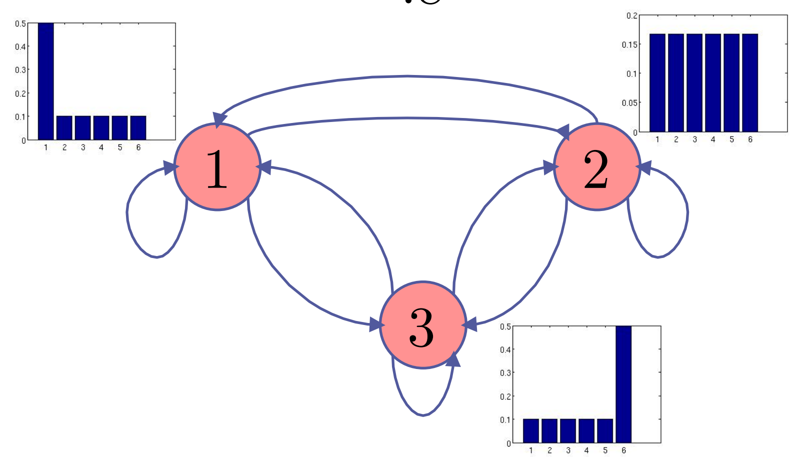


HMMs as State Machines

- HMMs have two variables: state q and emission y
- Typically, we don't know q (hidden variable 1,2,3,?)
- HMMs are like stochastic automata or finite state machines… next state depends on previous one… (helpful, fair, adversarial)

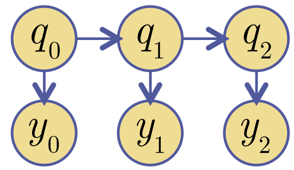


- Can't observe state q directly, just a random related emission y outcome (dice roll) so… doubly-stochastic automaton

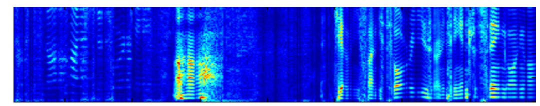


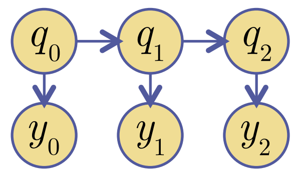
HMM Applications

- **Speech Rec** (Rabiner): phonemes from audio cepstral vectors
- **Language** (Jelinek): parts of speech from words
- **Biology** (Baldi): splice site from gene sequence
- **Gesture** (Starner): word from hand coordinates
- **Emotion** (Picard): emotion from EEG



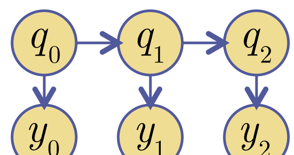
Ba-ra-kk-oo-oo-dd-ah





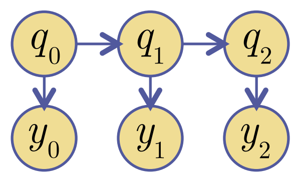
Noun Verb Noun

John Ate Pizza

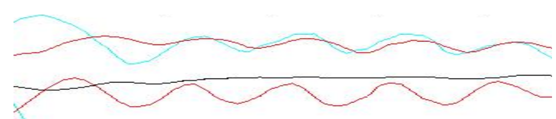


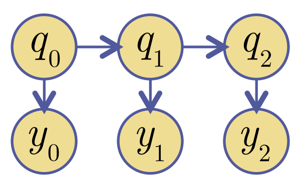
-Intron-|-Exon-|-Promoter-

GATTACATTATACCACCATACG



Pass The Salt





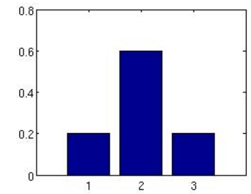
Happy Neutral Sad

Hidden Markov Models

- Graph gave: $p(X_U) = p(q_0)\prod_{t=1}^{T} p(q_t \mid q_{t-1})\prod_{t=0}^{T} p(y_t \mid q_t)$
- Haven't yet specified the types of variables or cpts…

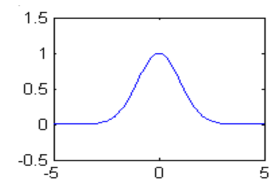
1) q can be discrete, example: finite state machine

$$p(q_t \mid q_{t-1}) = \prod_{i=1}^{M}\prod_{j=1}^{M}\left[a_{ij}\right]^{q_{t-1}^i q_t^j}$$



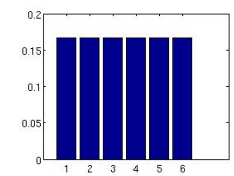
2) y can be vectors, example: time series

$$p(y_t \mid q_t) = N\left(y_t \mid \mu_{q_t}, \Sigma_{q_t}\right)$$



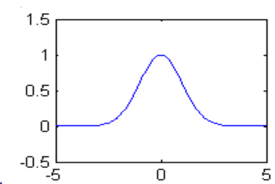
3) y can be discrete, example: strings

$$p(y_t \mid q_t) = \prod_{i=1}^{M}\prod_{j=1}^{N}\left[\eta_{ij}\right]^{q_t^i y_t^j}$$



4) q and y can be vectors, example: Kalman filter

$$p(q_t \mid q_{t-1}) = N\left(q_t \mid A q_{t-1}, Q\right) \quad and \quad N\left(y_t \mid C q_t, R\right)$$



Kalman Filters, Linear dynamical systems
Used in tracking, control (see ch. 14)

HMMs: Parameters

- We focus on HMMs with: discrete state q (of size M)
  discrete emission y (of size N)
- Input will be arbitrary length string: $y_1, \ldots, y_T$
- The pdf or (complete) likelihood is:

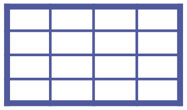
$$p\left(q,y\right) = p\left(q_0\right)\prod_{t=1}^{T} p\left(q_t \mid q_{t-1}\right)\prod_{t=0}^{T} p\left(y_t \mid q_t\right)$$

- We don't know hidden states, the incomplete likelihood is:

$$p\left(y\right) = \sum_{q_0} \cdots \sum_{q_T} p\left(q,y\right)$$

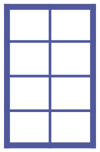
- Assume HMM is stationary, tables are repeated: $\theta = \left\{\pi, \eta, \alpha\right\}$

$$p\left(q_t \mid q_{t-1}\right) = \prod_{i=1}^{M}\prod_{j=1}^{M}\left[\alpha_{ij}\right]^{q_{t-1}^i q_t^j} \qquad \sum_{j=1}^{M} \alpha_{ij} = 1$$



$M \times M$

$$p\left(y_t \mid q_t\right) = \prod_{i=1}^{M}\prod_{j=1}^{N}\left[\eta_{ij}\right]^{q_t^i y_t^j} \qquad \sum_{j=1}^{N} \eta_{ij} = 1$$



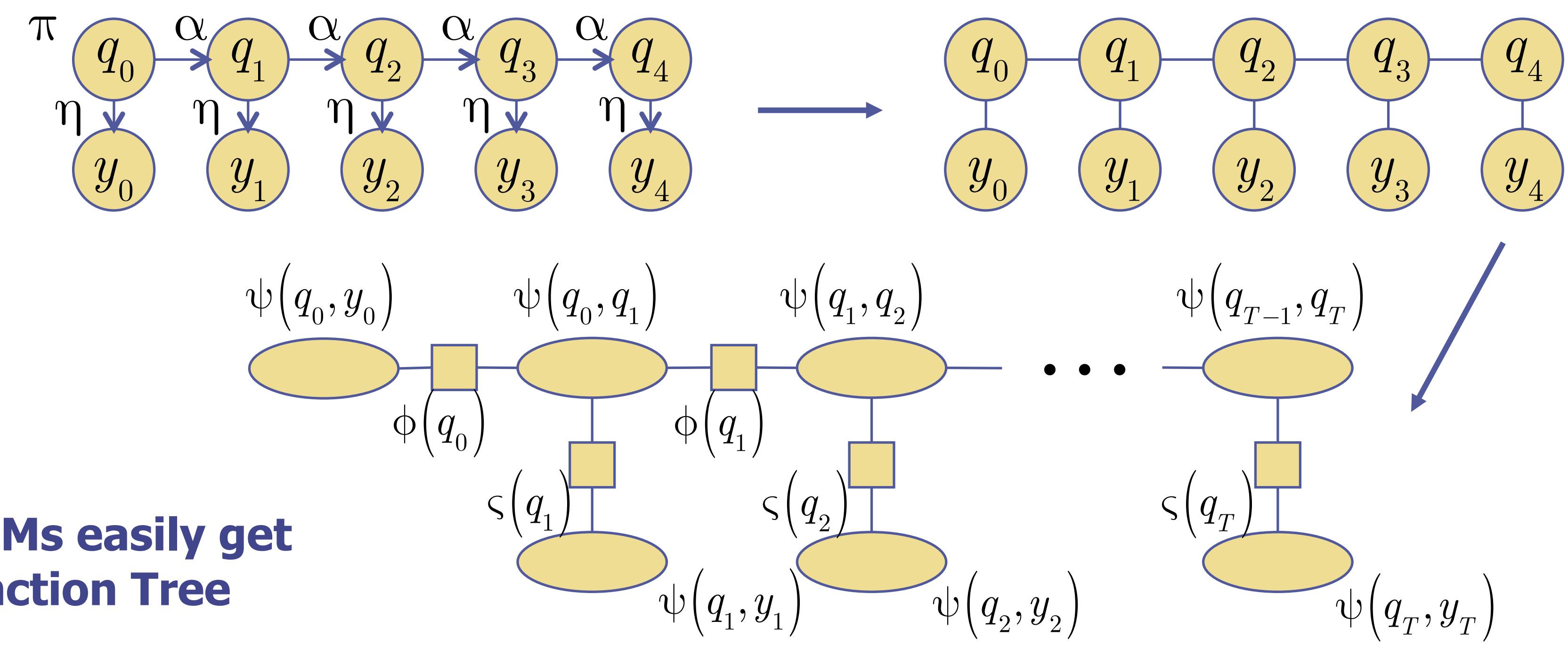
$M \times N$

$$p\left(q_0\right) = \prod_{i=1}^{M}\left[\pi_i\right]^{q_0^i} \qquad \sum_{j=1}^{M} \pi_j = 1$$

M

HMMs: Basic Operations

- Would like to do 3 basic things with our HMMs:
  1) Evaluate: given $y_0,\ldots,y_T$ & θ compute $p(y_1,\ldots,y_T)$
  2) Decode: given $y_0,\ldots,y_T$ & θ find $q_0,\ldots,q_T$ or $p(q_0),\ldots,p(q_T)$
  3) Max Likelihood: given $y_0,\ldots,y_T$ learn parameters θ
- Typically use Baum-Welch (α-β algo)… JTA is more general:

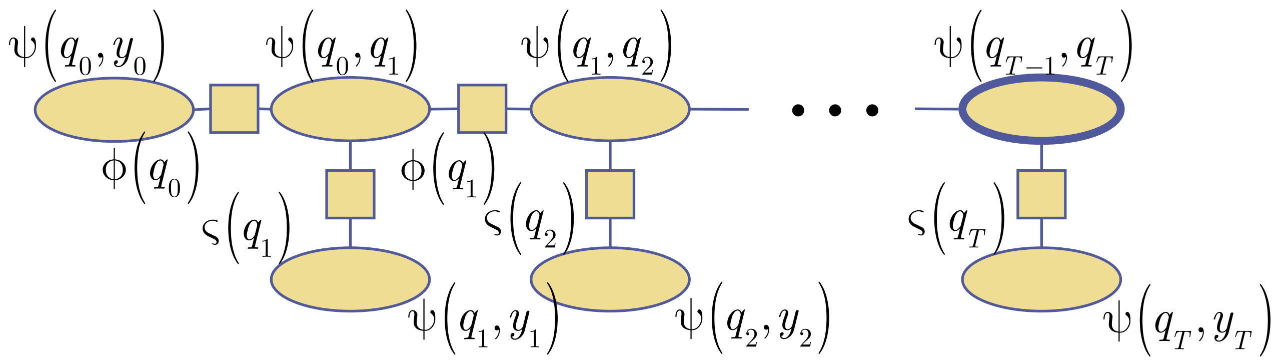


HMMs easily get Junction Tree

Tony Jebara, Columbia University

HMMs: JTA Init & Verify

- Init: $\psi(q_0,y_0)=p(q_0)\,p(y_0\,|\,q_0)$ $\quad \psi(q_t,q_{t+1})=p(q_{t+1}\,|\,q_t)=\alpha_{q_t,q_{t+1}}$ $\quad \psi(q_t,y_t)=p(y_t\,|\,q_t)$



$$Z=1$$
$$\phi(q_t)=1$$
$$\varsigma(q_t)=1$$

- Collect *up* from leaves: don't change zeta separators

$$\varsigma^*(q_t)=\sum_{y_t}\psi(q_t,y_t)=\sum_{y_t}p(y_t\,|\,q_t)=1 \qquad \psi^*(q_{t-1},q_t)=\frac{\varsigma^*}{\varsigma}\psi(q_{t-1},q_t)=\psi(q_{t-1},q_t)$$

- Collect *left-right* via phi's: change backbone to marginals

$$\phi^*(q_0)=\sum_{y_0}\psi(q_0,y_0)=p(q_0) \qquad \psi^*(q_0,q_1)=\frac{\phi^*}{\phi}\psi(q_0,q_1)=p(q_0,q_1)$$

$$\phi^*(q_t)=\sum_{q_{t-1}}\psi^*(q_{t-1},q_t)=p(q_t) \qquad \psi^*(q_{t-1},q_t)=\frac{p(q_{t-1})}{1}p(q_t\,|\,q_{t-1})=p(q_{t-1},q_t)$$

- Distribute:

$$\varsigma^{**}(q_t)=\sum_{q_{t-1}}\psi^*(q_{t-1},q_t)=\sum_{q_{t-1}}p(q_{t-1},q_t)=p(q_t)$$

$$\psi^{**}(q_t,y_t)=\frac{\varsigma^{**}}{\varsigma^*}\psi(q_t,y_t)=\frac{p(q_t)}{1}p(y_t\,|\,q_t)=p(y_t,q_t)$$

...done!