1(a).

1. False

Given $K(u,v) = \alpha K_1(u,v) + \beta K_2(u,v)$

To prove a function 'K' to be a kernel function, it must be symmetric and positive definite.
As $K_1 \; and \; K_2$ are kernel functions, by the property of symmetricity, linear combination of two symmetric functions is symmetric. And linear combination of two positive semi definite function is a positive semi definite function but because they are multiplied by coefficients whether or not K is a positive semi definite function depends on the coefficients $\alpha \; and \; \beta$. So whether or not K is a valid kernel function depends on the coefficients.

Alternatively, a function K is said to be valid kernel if it satisfies weak Mercer's condition which is given by

$$\iint_{uv} K(u,v)f(u)f(v)dudv \geq 0$$

$$\iint_{uv} K(u,v)f(u)f(v)dudv = \iint_{uv}(\alpha K_1(u,v) + \beta K_2(u,v))f(u)f(v)dudv$$

$$= \iint_{uv} \alpha K_1(u,v)f(u)f(v)dudv + \iint_{uv} \beta K_2(u,v)f(u)f(v)dudv$$

$$= \alpha \iint_{uv} K_1(u,v)f(u)f(v)dudv + \beta \iint_{uv} K_2(u,v)f(u)f(v)dudv$$

$= \alpha(function1 \geq 0) + \beta(function2 \geq 0)$ as K_1 and K_2 are kernels. The function above will be greater than 0 depending on the coefficients.

2. True

Given $K(u,v) = K_1(f(u),f(v))$

A function can be a valid kernel if it can be expressed as inner product of functions of u and v.
K_1 is valid kernel if can be expressed as Hilbert/ inner space as $< \Phi(u), \Phi(v) >$
Because f is a transformation in same domain, the Hilbert/inner space of K can be expressed as $< \Phi(f(u)), \Phi(f(v)) >$ and $\Phi(f(x)) \; can \; be \; expressed \; as \; \Phi_f(x)$ and the Hilbert space can be expressed as $< \Phi_f(u), \Phi_f(v) >$. So K is a valid kernel function.

3. False

Given $K(u,v) = 1 \; if \; ||u-v||_2 \leq 1 \; \; else \; 0$

If K is to be a kernel then it must satisfies weak Mercer's condition which can be expressed as

$$\iint_{uv} a_u a_v K(u,v) \geq 0$$

Or in discrete terms it can be expressed as $\sum_{u,v} a_u a_v K(u,v) \geq 0$

And the function $K(u,v)$ takes value either 0 or 1. So the function can be reduced to

$\sum_{u,v} a_u a_v$ where $u, v \; \forall \quad ||u-v||_2 \leq 1$

So whether or not the function is positive semi definite depends on the coefficients $a_u,\ a_v$.

4. True

Given $K(u,v) = \frac{K'(u,v)}{\sqrt{K'(u,u)\,K'(v,v)}}$ and K' is a kernel function.

The function K can be expressed as

$$K(u,v) = \sum_{i,j=1}^{n} \frac{c_i c_j K'(u_i,v_j)}{\sqrt{K'(u_i,u_i)\,K'(v_j,v_j)}} = \sum_{i,j=1}^{n} \frac{c_i c_j \langle \Phi(u_i),\Phi(u_j)\rangle}{\|\Phi(u_i)\|_H \|\Phi(u_j)\|_H} \geq 0$$

K is a positive semi definite and symmetric, so is a kernel function.

1(b).

Kernelized Fisher Linear Discriminant Analysis:

Linear Discriminant can be represented as, maximize the class separation function while minimizing the in-class variance. The class separation function is given by

$$J(w) = \frac{w^T S_B w}{w^T S_W w}$$

Where w is the weight function and S_B, S_W are between class covariance matrix and within-class covariance matrix.

The objective of the LDA is to maximize J(w) with respect to w.

The above function for nonlinear mappings and adapting kernel function, kernel discriminant analysis can be produced as:

For a n-class classifier $y_i = w_i^T \phi(x)$ for i = ,,..,n-1 and in matrix form $y = w^T \phi(x)$ and weights can be represented as $w = \sum \alpha_i \phi(x_i)$

Between class covariance matrix

$$S_B^{\Phi} = \sum_{i=1}^{n} l_i (m_i^{\Phi} - m^{\Phi})(m_i^{\Phi} - m^{\Phi})^T$$

$$S_W^{\Phi} = \sum_{i=1}^{n} \sum_{j=1}^{l_i} l_i (\phi(x_j^i) - m_i^{\Phi})(\phi(x_j^i) - m_i^{\Phi})^T$$

$$\text{And } m_i^{\Phi} = \frac{1}{l_i} \sum_{j=1}^{l_i} \phi(x_j^i)$$

$$w^T m_i^{\Phi} = \frac{1}{l_i} \sum_{j=1}^{l} \sum_{k=1}^{l_i} \alpha_j k(x_j, x_k^i) = \alpha^T M_i$$

$$M_i = \frac{1}{l_i} \sum_{k=1}^{l_i} k(x_j, x_k^i)$$

And the class separation function is given by

$$J(w) = \frac{w^T S_B^{\Phi} w}{w^T S_W^{\Phi} w}$$

$$w^T S_W^{\Phi} w = (\alpha \phi^T(x)) \left(\sum_{i=1}^{n} \sum_{j=1}^{l_i} l_i (\phi(x_j^i) - m_i^{\Phi})(\phi(x_j^i) - m_i^{\Phi})^T \right) (\alpha \phi(x))$$

$$= \sum_{i=1}^{n} \sum_{l=1}^{L} \sum_{j=1}^{l_j} \sum_{k=1}^{L} \alpha_k \, \phi^T(x_k) \, (\phi(x_j^i) - m_i^{\phi}) \, (\phi(x_j^i) - m_i^{\phi})^T \, \alpha_l \, \phi(x_l)$$

$$= \sum_{i=1}^{n} \sum_{l=1}^{L} \sum_{j=1}^{l_j} \sum_{k=1}^{L} (\alpha_k k(x_k, x_j^i) - \frac{1}{l_j} \sum_{p=1}^{l_j} (\alpha_k k(x_k, x_p^i)) \,)(\alpha_l k(x_l, x_j^i) - \frac{1}{l_j} \sum_{p=1}^{l_j} (\alpha_l k(x_l, x_p^i)) \,)$$

$$= \sum_{i=1}^{n} (\sum_{l=1}^{L} \sum_{j=1}^{l_j} \sum_{k=1}^{L} \alpha_k k(x_k, x_j^i) \alpha_l k(x_l, x_j^i) - \frac{2\alpha_k \alpha_l}{l_j} \sum_{p=1}^{l_j} k(x_k, x_p^i) k(x_l, x_j^i) + \frac{\alpha_k \alpha_l}{l_j l_j} \sum_{p=1}^{l_j} k(x_k, x_p^i) k(x_l, x_j^i))$$

$$= \sum_{i=1}^{n} (\sum_{l=1}^{L} \sum_{j=1}^{l_j} \sum_{k=1}^{L} \alpha_k k(x_k, x_j^i) \alpha_l k(x_l, x_j^i) - \frac{\alpha_k \alpha_l}{l_j} \sum_{p=1}^{l_j} k(x_k, x_p^i) k(x_l, x_j^i)) \,)$$

$$= \sum_{i=1}^{n} (\alpha^T K_k K_l \alpha - \alpha^T K_k l_j K_l \alpha) = \alpha^T N \alpha$$

$$\text{Where } N = \sum_{i=1}^{n} K_k K_l - K_k l_j K_l$$

Similarly,

$$w^T S_B^{\phi} w = \alpha^T M \alpha \text{ where } M = \sum_{j=1}^{c} l_j (M_j - M_*)(M_j - M_*)^T,$$

Where $(M_*)_j = \frac{1}{l} \sum_{k=1}^{l} k(x_j, x_k)$

The class separation function is now expressed as $J(\alpha) = \frac{\alpha^T M \alpha}{\alpha^T N \alpha}$

$$\alpha^* = argmax(\alpha) = \left| \frac{\alpha^T M \alpha}{\alpha^T N \alpha} \right|$$

By differentiating the above function with respect to α and equating it to zero, α^* can be found using n-1 eigenvectors of $N^{-1}M$.

And projection of new input x_u is given by

$$y(x_u) = \alpha^{*T} K \quad where\ i\ element\ of\ K\ is\ k(x_i, x_t)$$

Class of new input label is determined as $f(x) = argmin_j(distance(y(x), y_j))$ where y_j is projected mean for class j, y(x) is predicted y.

2(a).

$\psi(a, b = 1, c) = \phi_1(a, b = 1)\phi_2(b = 1, c)$

a b $\phi_1(a, b = 1)$

0 1 3

1 1 1

b c $\phi_2(b = 1, c)$

1 0 4

1 1 1

1 2 3

a b c $\psi(a, b = 1, c)$

0 1 0 12

0 1 1 3
0 1 2 9
1 1 0 4
1 1 1 1
1 1 2 3

$\psi(a, b = 1, c)$ =12+3+9+4+4=**32**

2(b).

$P(a = 1, b = 1, c) = \frac{1}{z}\psi(a = 1, b = 1, c)$

$\psi(a = 1, b = 1, c) = 4 + 1 + 3 = 7$

$z = \psi(a, b, c)$

a	b	c	$\psi(a, b, c)$
0	0	0	12
0	0	1	8
0	0	2	4
0	1	0	12
0	1	1	3
0	1	2	9
1	0	0	9
1	0	1	6
1	0	2	3
1	1	0	4
1	1	1	1
1	1	2	3

$z = \psi(a, b, c) = 74$

$P(a = 1, b = 1, c) = 8/74 = 0.108$

2(c).

Difference between conditional random fields and hidden markov models:

- Models: Conditional Random Fields are discriminative models as the generation of data doesn't matter and they simply categorize signals. And Hidden Markov Models are generative models as they categorize signals based on how it is generated.

- Objective function: The difference between CRFs and HMMs in terms of objective function is that a HMM uses per-state models for the conditional probabilities of next states given the current state, while a CRF has a single exponential model for the joint probability of the entire sequence of labels given the observation sequence

- Require normalization constant: CRF requires normalization constant to make conditional probabilities sum to 1 and HMMs don't. require

3a).

Probability $P(x_1,..,x_n,z_1,..,z_n)$ in hidden markov models is given by

$$P(x_1,..,x_n,z_1,..,z_n) = P(z_1)P(x_1|z_1)\prod_{i=2}^{n} P(z_i|z_{i-1})P(x_i|z_i)$$

$P(O) = P(H,T,H) = \sum_{i=1}^{3}\sum_{j=1}^{3}\sum_{k=1}^{3} P(H,T,H,z_i,z_j,z_k)$ where z_i represents state of choosing coin *i.*

$$P(H,T,H) = P(H,T,H,z_1,z_1,z_1) + P(H,T,H,z_1,z_1,z_2) + P(H,T,H,z_1,z_1,z_3) + P(H,T,H,z_1,z_2,z_1) +$$
$$P(H,T,H,z_1,z_2,z_2) + P(H,T,H,z_1,z_2,z_3) + P(H,T,H,z_1,z_3,z_1) + P(H,T,H,z_1,z_3,z_2) +$$
$$P(H,T,H,z_1,z_3,z_3) + P(H,T,H,z_2,z_1,z_1) + P(H,T,H,z_2,z_1,z_2) + P(H,T,H,z_2,z_1,z_3) +$$
$$P(H,T,H,z_2,z_2,z_1) + P(H,T,H,z_2,z_2,z_2) + P(H,T,H,z_2,z_2,z_3) + P(H,T,H,z_2,z_3,z_1) +$$
$$P(H,T,H,z_2,z_3,z_2) + P(H,T,H,z_2,z_3,z_3) + P(H,T,H,z_3,z_1,z_1) + P(H,T,H,z_3,z_1,z_2) +$$
$$P(H,T,H,z_3,z_1,z_3) + P(H,T,H,z_3,z_2,z_1) + P(H,T,H,z_3,z_2,z_2) + P(H,T,H,z_3,z_2,z_3)\ P(H,T,H,z_3,z_3,z_1)$$
$$+ P(H,T,H,z_3,z_3,z_2) + P(H,T,H,z_3,z_3,z_3)$$

Transition Probability matrix:

	1	2	3
1	0.9	0.05	0.05
2	0.45	0.1	0.45
3	0.45	0.45	0.1

Emission probabilities:

	1	2	3
H	0.5	0.75	0.25
T	0.5	0.25	0.75

$P(H,T,H,z_1,z_1,z_1) = p(z_1)p(H|z_1)p(z_1|z_1)p(T|z_1)p(z_1|z_1)p(H|z_1)$ = $\frac{1}{3} * \frac{1}{2} * \frac{9}{10} * \frac{1}{2} * \frac{9}{10} * \frac{1}{2}$ = 0.03375

$P(H,T,H,z_1,z_1,z_2)$ = $p(z_1)p(H|z_1)p(z_1|z_1)p(T|z_1)p(z_2|z_1)p(H|z_2)$ = $\frac{1}{3} * \frac{1}{2} * \frac{9}{10} * \frac{1}{2} * 0.05 * 0.75$ = 0.0028125

By calculating all the terms as shown above we get

$P(O) = P(H,T,H) = 0.138515625$

3b).

Baum-Welch algorithm can be used to find out unknown A,B,Π parameters. The algorithm uses EM algorithm to find the maximum likelihood estimate of the parameters of a hidden Markov model given a set of observed feature vectors.

Let X_t be a discrete hidden random variable with N possible values. $p(X_t|X_{t-1})$ is independent of time so the stochastic transition matrix is given by

$$A = \{a_{ij}\} = p(X_t = i|X_{t-1} = j)$$

Initial State distributions are given by $\pi_i = p(X_1 = i)$

Observed variable Y_t can take one of K possible values.The probability of a certain observation at time t for state j is given by

$$b_j(y_t) = P(Y_t = y_t|X_t = j)$$

All possible values of $\{b_j(y_t)\}$ produce emission matrix of size KxN

Thus we can describe a hidden Markov chain by θ = (A , B , π). The Baum–Welch algorithm finds a local maximum for $\theta * = \arg\max_\theta P(Y|\theta)$ (i.e. the HMM parameters θ that maximise the probability of the observation.

Algorithm:

Set θ = (A , B , π) with random initial conditions.

Forward procedure:

Let $\alpha_i(t) = P(Y_1 = y_1, ..., Y_t = y_t, X_t = i|\theta)$. This is found recursively.

1. $\alpha_1(t) = \pi_i b_i(y_1)$
2. $\alpha_j(t+1) = b_i(y_{t+1}) \sum_{i=1}^{N} \alpha_i(t) a_{ij}$

Backward procedure:

Let $\beta_i(t) = P(Y_{t+1} = y_{t+1}, ..., Y_T = y_T, X_t = i, \theta)$. This is found recursively.

3. $\beta_i(T) = 1$
4. $\beta_i(t) = \sum_{j=1}^{N} \beta_i(t+1) a_{ij} b_j(y_{t+1})$

Update

Now temporary variables can be calculated using Bayes theorem:

The probability of being in state i at time t given the observed sequence Y and the parameters θ is given by

$$\gamma_i(t) = P(X_t = i|Y, \theta) = \frac{\alpha_i(t)\beta_i(t)}{\sum_{j=1}^{N} \alpha_j(t)\beta_j(t)}$$

The probability of being in state i and j at times t and t + 1 respectively given the observed sequence Y and parameters θ is given by

$$P(X_t = i, X_{t+1} = j|Y, \theta) = \frac{\alpha_i(t) a_{ij} \beta_j(t+1) b_j(y_{t+1})}{\sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_j(t) a_{ij} \beta_j(t+1) b_j(y_{t+1})}$$

Now θ can be updated as

- $\pi_i^* = \gamma_i(1)$,this is the expected frequency spent in state i at time 1

- $a_{ij}^{*} = \frac{\sum_{t=1}^{T-1} \xi_{ij}(t)}{\sum_{t=1}^{T} \gamma_i(t)}$, this is the expected number of transitions from state *i* to state *j* compared to the expected total number of transitions away from state *i*.
- $b_i^{*}(v_k) = \frac{\sum_{t=1}^{T} 1_{y_t=v_k}\gamma_i(t)}{\sum_{t=1}^{T} \gamma_i(t)}$,where $1_{y_t=v_k}$ = 1 if $y_t = v_k$, 0 otherwise

This is the expected number of times the output observations have been equal to v k while in state i over the expected total number of times in state i
These steps are repeated iteratively until a desired level of convergence.

3c).
Let's assume we started with initial θ_0.
Posterior distribution of latent variables is given by $p(z|X,\theta)$ where X are observable states.
Log likelihood of the posterior distribution is given by

$$Q(\theta,\theta_0) = \sum_z p(z|X,\theta_0) \ln p(X,z|\theta)$$

Now Q function is to be maximized with respect to θ
Let $\gamma(z_n) = p(z_n|X,\theta_0)$ be the marginal posterior distribution of z_n
And $\xi(z_{n-1},z_n) = p(z_{n-1},z_n|X,\theta_0)$ be the joint posterior of two successive latent variables

$\gamma(z_{nk})$ *denotes conditional probability of* $z_{nk} = 1$ and $\xi(z_{n-1,j},z_{nk})$ denoted joint probability when $z_{nk} = 1$
Because the expectation of a binary random variable is the probability that it takes value 1

$$\gamma(z_{nk}) = E[z_{nk}] = \sum_z \gamma(z)\, z_{n-1,j} z_{nk}$$

$$Q(\theta,\theta_0) = \sum_z p(z|X,\theta_0) \ln p(X,z|\theta)$$

$$p(X,z|\theta) = p(z_1|\pi) \prod_{n=2}^{N} p(z_n|z_{n-1},A) \prod_{m=1}^{N} p(x_m|z_{m,}\varphi)$$

$$= \sum_{k=1}^{K} \gamma(z_{1k}) ln(\pi_k) + \sum_{n=2}^{N}\sum_{j=1}^{K}\sum_{k=1}^{K} \xi(z_{n-1,j},z_{nk}) ln\, A_{jk} + \sum_{n=1}^{N}\sum_{k=1}^{K} \gamma(z_{nk})\, lnp(x_n|\phi_k)$$

$p(z|X,\theta_0)$ is independent of θ. So to maximize Q with respect to θ, $p(X,z|\theta)$ w.r.t to θ must be maximized.
Maximizing Q w.r.t π using lagrangian multiplier gives

$$\pi_k^{*} = \frac{\gamma(z_{1k})}{\sum_{j=1}^{K} \gamma(z_{1j})}$$

Maximizing Q w.r.t A using lagrangian multiplier gives

$$A_{jk}^{*} = \frac{\sum_{n=2}^{N} \xi(z_{n-1,j}, z_{nk})}{\sum_{l=1}^{K} \sum_{n=2}^{N} \xi(z_{n-1,j}, z_{nl})}$$

Maximizing Q w.r.t ϕ_k implies maximizing $\sum_{n=1}^{N} \sum_{k=1}^{K} \gamma(z_{nk}) \, lnp(x_n|\phi_k)$ w.r.t ϕ_k

Given the model follows Gaussian distribution. So, $p(x_n|\phi_k) \sim N(x|\mu_k, \Sigma_k)$

Differentiating the above function w.r.t ϕ_k and equating it to 0, we get

$$\mu_k = \frac{\sum_{n=1}^{N} \gamma(z_{nk}) x_n}{\sum_{n=1}^{N} \gamma(z_{nk})}$$

$$\Sigma_k = \frac{\sum_{n=1}^{N} \gamma(z_{nk})(x_n - \mu_k)(x_n - \mu_k)^T}{\sum_{n=1}^{N} \gamma(z_{nk})}$$

For the case of discrete multinomial observed variables, the conditional distribution of the observations takes the form

$$p(x|z) = \prod_{i=1}^{D} \prod_{k=1}^{K} \mu_{ik}^{x_i z_k}$$

3d).

- False

  All the incoming edges contribute to a probability of a state which need not be 1. Sum of all outgoing edges will sum up to 1.

- False

  An edge from state s to state t in an HMM denotes the conditional probability of going to state t given that we are currently at state s and not the otherwise.

- True

  A HMM is said to follow Markov property if it's distribution is solely determined by the current state

- False

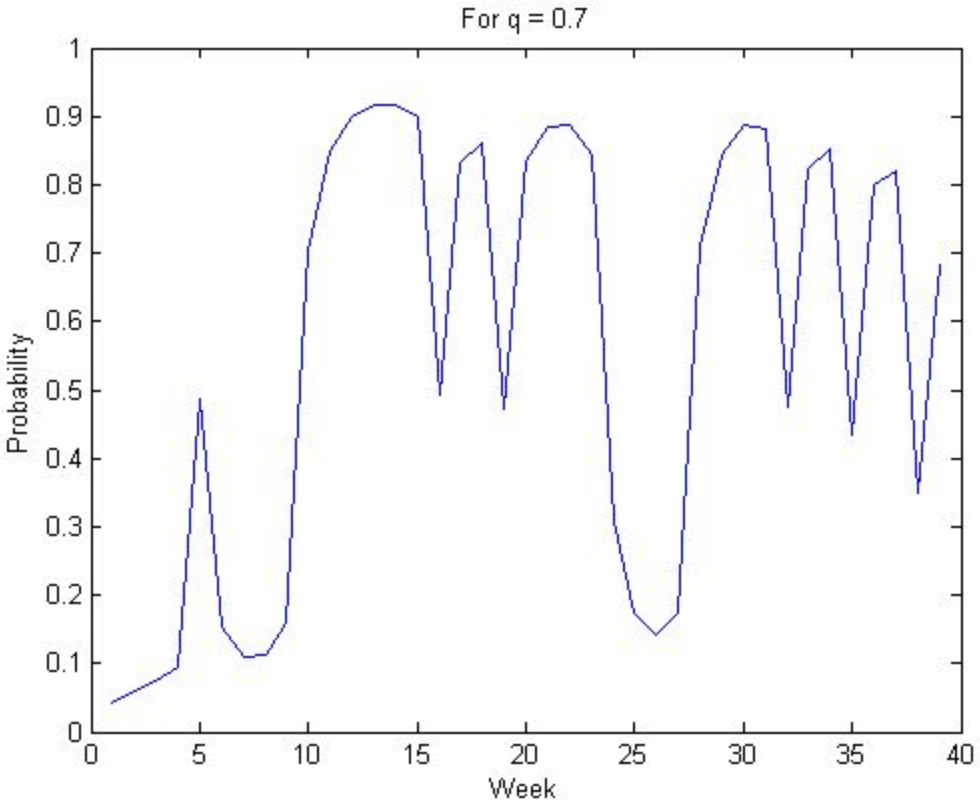
  The Baum-Welch algorithm is a type of an EM algorithm but it is not guaranteed to converge to the (globally) optimal solution. It converges to local maximum.

4.

I am using forward algorithm and backward algorithm to dynamically calculate p(x|y). I am calculating p(x,y)= $\alpha * \beta$ where α is calculated using forward algorithm and β is calculated using backward algorithm and I am normalizing it with p(y) to get p(x|y).

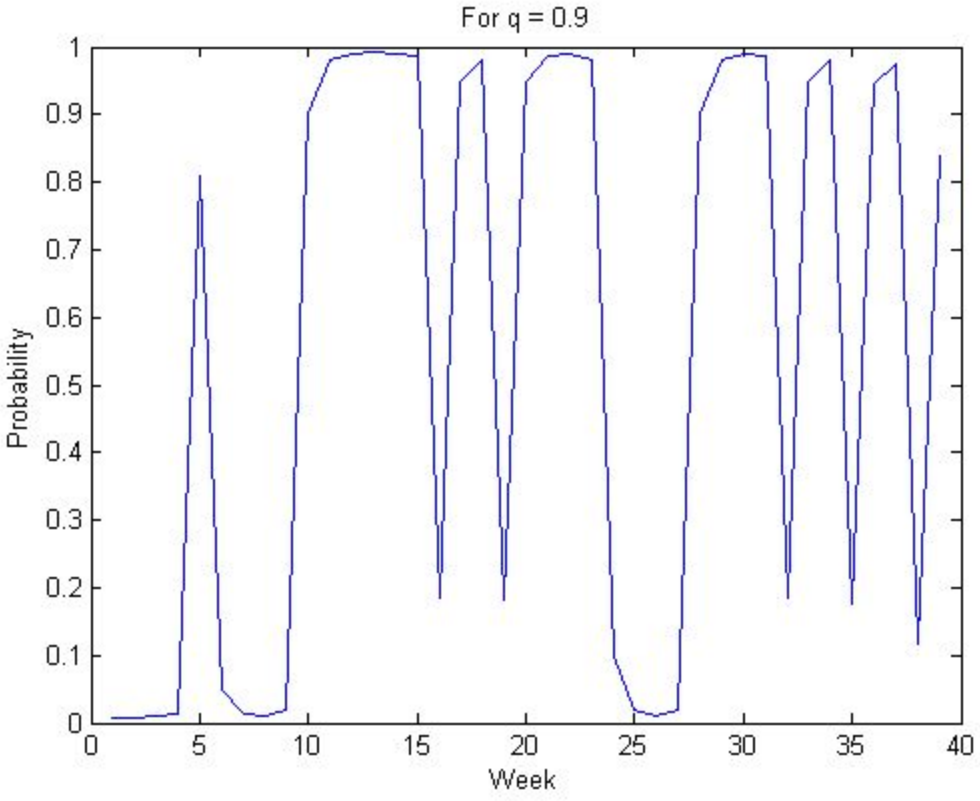
(a) When q=0.7, $P_{(X_{39}|Y)}(x_{39} = good|y)$=0.683

The plot for $P_{(X_t|Y)}(x_t = good|y)$ for t=1,2..,39 and q=0.7 is as:



(b) When q=0.9, $P_{(X_{39}|Y)}(x_{39} = good|y)$=0.8378

The plot for $P_{(X_t|Y)}(x_t = good|y)$ for t=1,2..,39 and q=0.9 is as:



Conclusions: With higher value of q, the probability of economy being in good state in the week 39 increased i.e., by increasing the probability that the price movement of SP500 is reduced when the economy is bad, the model is giving better results.