CSCI567
Practice Exam2 Review

Ke Zhang, Jeremy Hsu, Chaoyang He

Mixture Models

Tips

You ALWAYS have the joint PDF when you are dealing with generative models such as

- NB
- GMM
- HMM

→ Start from the **joint PDF** and get what the problem asks you for.

2 Mixture Models (20 points)

The general Expectation-Maximization (EM) algorithm is summarized as follow:

Algorithm 1: General EM algorithm

Step 0

Initialize $\theta^{(1)}$, $t = 1$

Step 1 (E-Step)

1-1 Update the posterior of latent variables

$$q_n^{(t)}(\cdot) = p\left(\cdot|\mathbf{x}_n; \boldsymbol{\theta}^{(t)}\right)$$

1-2 Obtain **Expectation** of complete likelihood

$$Q\left(\boldsymbol{\theta}; \boldsymbol{\theta}^{(t)}\right) = \sum_{n=1}^{N} \mathbb{E}_{z_n \sim q_n^{(t)}} \left[\ln p\left(\mathbf{x}_n, z_n; \boldsymbol{\theta}\right)\right]$$

Step 2 (M-Step)

Update the model parameter via **Maximization**

$$\boldsymbol{\theta}^{(t+1)} \leftarrow \arg\max_{\boldsymbol{\theta}} Q\left(\boldsymbol{\theta}; \boldsymbol{\theta}^{(t)}\right)$$

Step 3

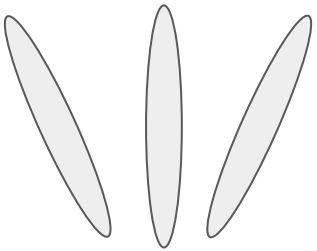
$t \leftarrow t + 1$ and return to Step 1 if not converged

Consider a GMM with $K \times L$ components. One latent variable, $z_1 \in \{1, 2, ..., K\}$, governs the mean; the other, $z_2 \in \{1, 2, ..., L\}$ governs the covariance. The two latent variables are independent and gives the following PDF

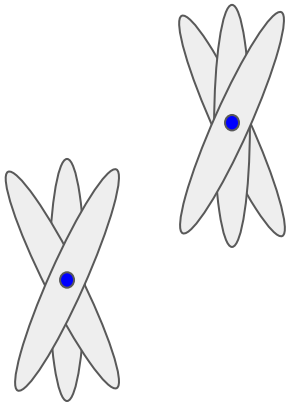
Mixture Models

Consider a GMM with $K \times L$ components. One latent variable, $z_1 \in \{1, 2, ..., K\}$, governs the mean; the other, $z_2 \in \{1, 2, ..., L\}$ governs the covariance. The two latent variables are independent and give the following PDF

$$p(\mathbf{x}; \boldsymbol{\theta}) = \sum_{k=1}^{K} \sum_{l=1}^{L} \omega_k \nu_l \mathcal{N}\left(\mathbf{x} | \boldsymbol{\mu}_k, \Sigma_l\right). \tag{1}$$



3 possible covariances,
2 possible means.



Find the posterior

10. Express $p(\mathbf{z}_n = [k,\ l]^T |\ \mathbf{x}_n; \boldsymbol{\theta}^{(t)})$, which is the posterior of latent variables for the E-Step, with model parameters. You may omit $\boldsymbol{\theta}$, the superscript t, and the subscript n for simplicity during derivation; there is no need to expand $\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \Sigma_l)$.

Find the posterior

10. Express $p(\mathbf{z}_n = [k,\ l]^T |\ \mathbf{x}_n; \boldsymbol{\theta}^{(t)})$, which is the posterior of latent variables for the E-Step, with model parameters. You may omit $\boldsymbol{\theta}$, the superscript t, and the subscript n for simplicity during derivation; there is no need to expand $\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \Sigma_l)$.

Find the joint probability first

- If it is not given, you must be able to find it (according to the factorization rule of the model).
- Apply Bayes' rule, do marginalization, etc.

Find the posterior

10. Express $p(\mathbf{z}_n = [k,\ l]^T |\ \mathbf{x}_n; \boldsymbol{\theta}^{(t)})$, which is the posterior of latent variables for the E-Step, with model parameters. You may omit $\boldsymbol{\theta}$, the superscript t, and the subscript n for simplicity during derivation; there is no need to expand $\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \Sigma_l)$.

$$
\begin{aligned}
p(\mathbf{x};\boldsymbol{\theta}) &= \sum_{k=1}^{K}\sum_{l=1}^{L} p(\boldsymbol{x}_n, \boldsymbol{z}_n) && \text{marginalization} \\
&= \sum_{k=1}^{K}\sum_{l=1}^{L} p(\boldsymbol{z}_n)p(\boldsymbol{x}_n|\boldsymbol{z}_n) && \text{(according to GMM factorization)} \\
&= \sum_{k=1}^{K}\sum_{l=1}^{L} p(z_1 = k)p(z_2 = l)p(\boldsymbol{x}_n|\boldsymbol{z}_n) && \text{(according to independence)} \\
&= \sum_{k=1}^{K}\sum_{l=1}^{L} \omega_k \nu_l \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \Sigma_l) && \text{(from problem descriptions)}
\end{aligned}
$$

Find the posterior

10. Express $p(\mathbf{z}_n = [k,\ l]^T |\ \mathbf{x}_n; \boldsymbol{\theta}^{(t)})$, which is the posterior of latent variables for the E-Step, with model parameters. You may omit $\boldsymbol{\theta}$, the superscript t, and the subscript n for simplicity during derivation; there is no need to expand $\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \Sigma_l)$.

$$p(\mathbf{x}; \boldsymbol{\theta}) = \sum_{k=1}^{K}\sum_{l=1}^{L} p(z_1 = k)p(z_2 = l)p(\boldsymbol{x}_n|\boldsymbol{z}_n) = \sum_{k=1}^{K}\sum_{l=1}^{L} \omega_k \nu_l \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \Sigma_l)$$

$$p(z_1 = k) = \omega_k,\ \sum_{k=1}^{K} \omega_k = 1$$

$$p(z_2 = l) = \nu_l,\ \sum_{l=1}^{L} \nu_l = 1$$

Find the posterior

10. Express $p(\mathbf{z}_n = [k,\ l]^T |\ \mathbf{x}_n; \boldsymbol{\theta}^{(t)})$, which is the posterior of latent variables for the E-Step, with model parameters. You may omit $\boldsymbol{\theta}$, the superscript t, and the subscript n for simplicity during derivation; there is no need to expand $\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \Sigma_l)$.

$$p(\boldsymbol{x}_n, \boldsymbol{z}_n) = \omega_k \nu_l \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \Sigma_l)$$

$$p\left(\mathbf{z}_n = [k,\ l]^T |\ \mathbf{x}_n; \boldsymbol{\theta}^{(t)}\right) = \frac{\omega_k^{(t)} \nu_l^{(t)} \mathcal{N}\left(\mathbf{x}_n|\boldsymbol{\mu}_k^{(t)}, \Sigma_l^{(t)}\right)}{\sum_{k'=1}^{K} \sum_{l'=1}^{L} \omega_{k'}^{(t)} \nu_{l'}^{(t)} \mathcal{N}\left(\mathbf{x}_n|\boldsymbol{\mu}_{k'}^{(t)}, \Sigma_{l'}^{(t)}\right)}$$

Optimizing Model Parameters

11. Recall that the solution of the simplex optimization problem

$$\begin{aligned} &\arg\max_{\mathbf{q}} \sum_{k=1}^{K} a_k \ln q_k \\ &\text{s.t. } q_k \geq 0 \\ &\sum_{k=1}^{K} q_k = 1 \end{aligned}$$

with $a_1, \cdots a_k \geq 0$ is $q_k^* = \frac{a_k}{\sum_{k'=1}^{K} a_{k'}}$. Let $a_{nkl} = p(\mathbf{z}_n = [k,\ l]^T |\ \mathbf{x}_n; \boldsymbol{\theta}^{(t)})$. Derive the update of parameters ω_k, ν_l for the M-Step.

Optimizing Model Parameters

Find the optimal update for component weights.

1. Expand the objective

$$
\begin{aligned}
Q\left(\boldsymbol{\theta}; \boldsymbol{\theta}^{(t)}\right) &= \sum_{n=1}^{N} \mathbb{E}_{\mathbf{z}_n \sim p\left(\mathbf{z}_n \mid \mathbf{x}_n; \boldsymbol{\theta}^{(t)}\right)} \left[\ln p\left(\mathbf{x}_n, \mathbf{z}_n; \boldsymbol{\theta}\right)\right] \\
&= \sum_{n=1}^{N} \sum_{k=1}^{K} \sum_{l=1}^{L} p\left(\mathbf{z}_n = [k,\ l]^T \mid \mathbf{x}_n; \boldsymbol{\theta}^{(t)}\right) \ln\left[\omega_k \nu_l \mathcal{N}\left(\mathbf{x}_n \mid \boldsymbol{\mu}_k, \Sigma_l\right)\right] \\
&= \sum_{n=1}^{N} \sum_{k=1}^{K} \sum_{l=1}^{L} a_{nkl} \left[\ln(\omega_k) + \ln(\nu_l) + \ln \mathcal{N}\left(\mathbf{x}_n \mid \boldsymbol{\mu}_k, \Sigma_l\right)\right].
\end{aligned}
$$

Optimizing Model Parameters

Find the optimal update for component weights.

2. Formulate the optimization problem

arg max

w

$$\sum_{n=1}^{N}\sum_{k=1}^{K}\sum_{l=1}^{L} a_{nkl}\left[\ln(\omega_k)+\ln(\nu_l)+\ln\mathcal{N}\left(\mathbf{x}_n|\boldsymbol{\mu}_k,\Sigma_l\right)\right].$$

Excluding the terms unrelated to ω_k, the update of ω_k can be found by

$$\omega_k = \arg\max_{\omega_k} \sum_{n=1}^{N}\sum_{k=1}^{K}\sum_{l=1}^{L} a_{nkl}\ln(\omega_k)$$

$$\text{s.t. } \omega_k \geq 0$$

$$\sum_{k=1}^{K}\omega_k = 1$$

Optimizing Model Parameters

Find the optimal update for component weights.

2. Formulate the optimization problem

arg max

w

$$\sum_{n=1}^{N}\sum_{k=1}^{K}\sum_{l=1}^{L} a_{nkl}\left[\ln(\omega_k)+\ln(\nu_l)+\ln\mathcal{N}\left(\mathbf{x}_n|\boldsymbol{\mu}_k,\Sigma_l\right)\right].$$

Excluding the terms unrelated to ω_k, the update of ω_k can be found by

$$\omega_k = \arg\max_{\omega_k} \sum_{n=1}^{N}\sum_{k=1}^{K}\sum_{l=1}^{L} a_{nkl}\ln(\omega_k)$$

$$\text{s.t. } \omega_k \geq 0$$

$$\sum_{k=1}^{K}\omega_k = 1$$

Optimizing Model Parameters

Find the optimal update for component weights.

3. Solve the optimization problem

$$\omega_k = \arg\max_{\omega_k} \sum_{n=1}^{N} \sum_{k=1}^{K} \sum_{l=1}^{L} a_{nkl} \ln(\omega_k)$$

$$\text{s.t. } \omega_k \geq 0$$

$$\sum_{k=1}^{K} \omega_k = 1$$

Applying the hint (letting $q_k = \omega_k$), the solution is

$$\omega_k^{(t+1)} = \frac{\sum_{n=1}^{N} \sum_{l=1}^{L} a_{nkl}}{\sum_{n=1}^{N} \sum_{k'=1}^{K} \sum_{l=1}^{L} a_{nkl}} = \frac{1}{N} \sum_{n=1}^{N} \sum_{l=1}^{L} a_{nkl}$$

Optimizing Model Parameters

Find the optimal update for component weights.

3. Solve the optimization problem

$$\omega_k = \arg\max_{\omega_k} \sum_{n=1}^{N}\sum_{k=1}^{K}\sum_{l=1}^{L} a_{nkl} \ln(\omega_k)$$

$$\text{s.t. } \omega_k \geq 0$$

$$\sum_{k=1}^{K} \omega_k = 1$$

Applying the hint (letting $q_k = \omega_k$), the solution is

$$\omega_k^{(t+1)} = \frac{\sum_{n=1}^{N}\sum_{l=1}^{L} a_{nkl}}{\sum_{n=1}^{N}\sum_{k'=1}^{K}\sum_{l=1}^{L} a_{nkl}} = \frac{1}{N}\sum_{n=1}^{N}\sum_{l=1}^{L} a_{nkl}$$

Optimizing Model Parameters

Find the optimal update for component weights.

3. Solve the optimization problem

Similarly, the update of ν_l is

$$\nu_l^{(t+1)} = \frac{\sum_{n=1}^{N}\sum_{k=1}^{K} a_{nkl}}{\sum_{n=1}^{N}\sum_{k=1}^{K}\sum_{l'=1}^{L} a_{nkl}} = \frac{1}{N}\sum_{n=1}^{N}\sum_{k=1}^{K} a_n kl$$

Boosting

1. The Adaboost algorithm will eventually give zero training error regardless of the type of weak classifier it uses, provided enough iterations are performed.

   (a) True

   (b) False

   Ans: False

2. Boosting algorithm will not select the same weak classifier more than once.

   (a) True

   (b) False

   Ans: False

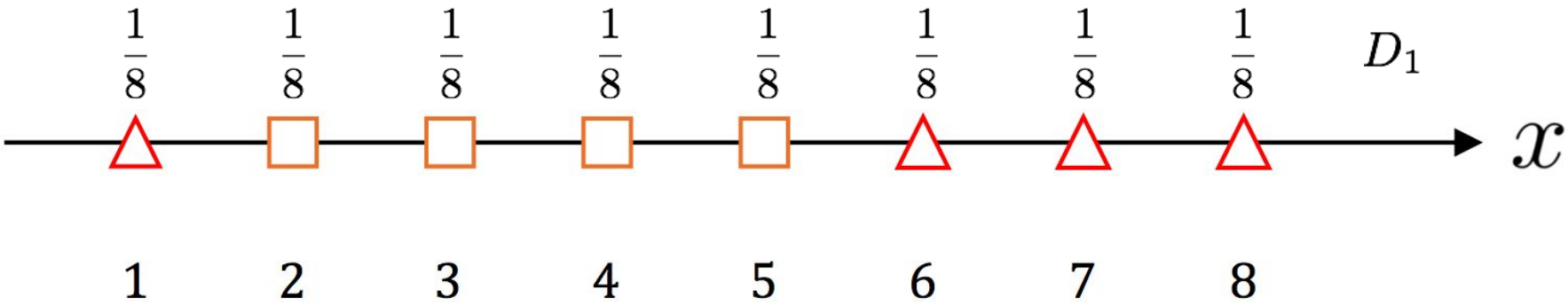
3. In the Adaboost algorithm, weights of the misclassified examples goes up.

   (a) True

   (b) False

   Ans: True

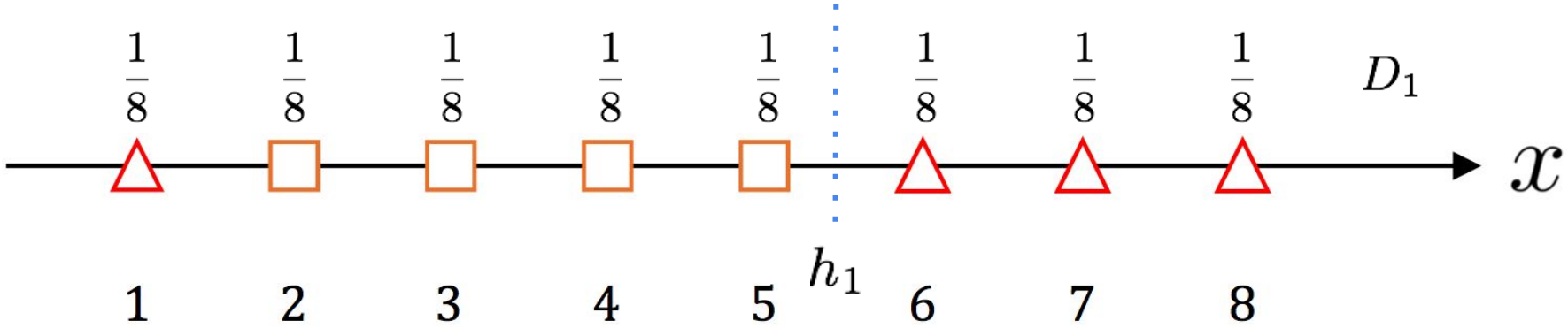
Boosting



16. Please write down the pair (s, b) of the best decision stump h_1, ϵ_1 and the mis-classified data at $t = 1$. If there are multiple equally optimal stump functions, just randomly pick **ONE** of them to be h_1. **(6 points)**

Ans:
$s = -1$ and $b \in [5, 6)$ (any b in this range is fine).
The left-most triangle will be mis-classified.
$\epsilon_1 = \frac{1}{8} = 0.125$

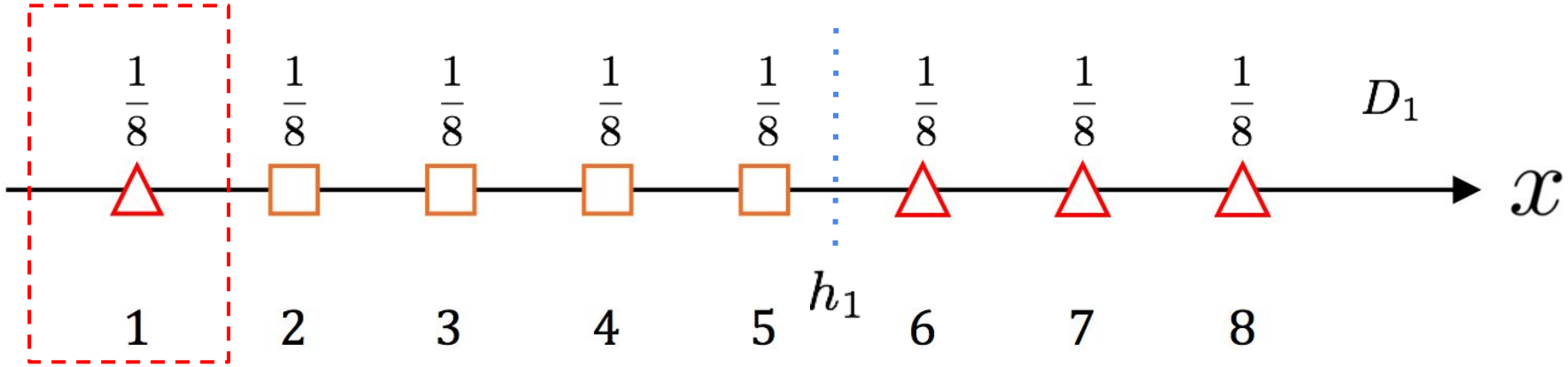
Boosting



16. Please write down the pair (s, b) of the best decision stump h_1, ϵ_1 and the mis-classified data at $t = 1$. If there are multiple equally optimal stump functions, just randomly pick **ONE** of them to be h_1. **(6 points)**

Ans:
$s = -1$ and $b \in [5, 6)$ (any b in this range is fine).
The left-most triangle will be mis-classified.
$\epsilon_1 = \frac{1}{8} = 0.125$

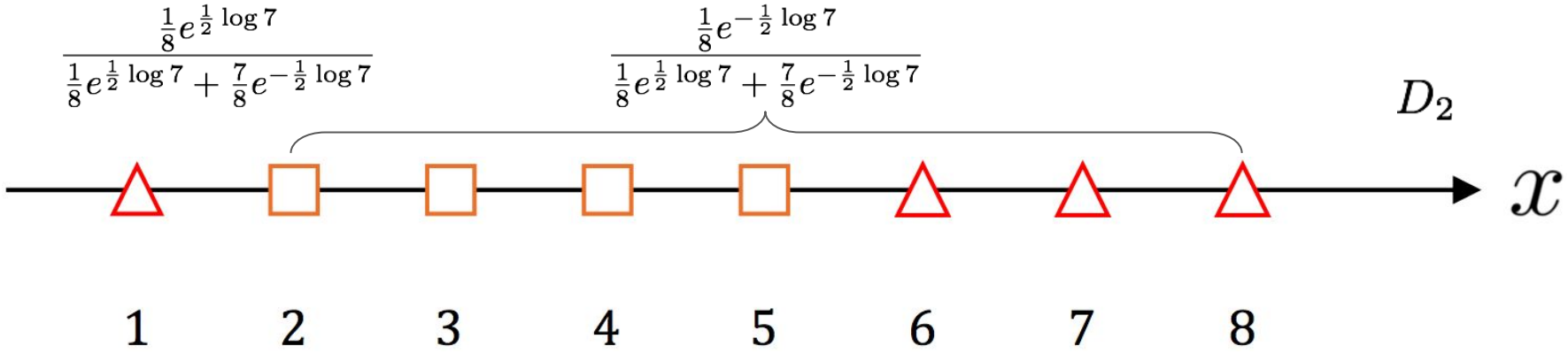
Boosting



16. Please write down the pair (s, b) of the best decision stump h_1, ϵ_1 and the mis-classified data at $t = 1$. If there are multiple equally optimal stump functions, just randomly pick **ONE** of them to be h_1. **(6 points)**

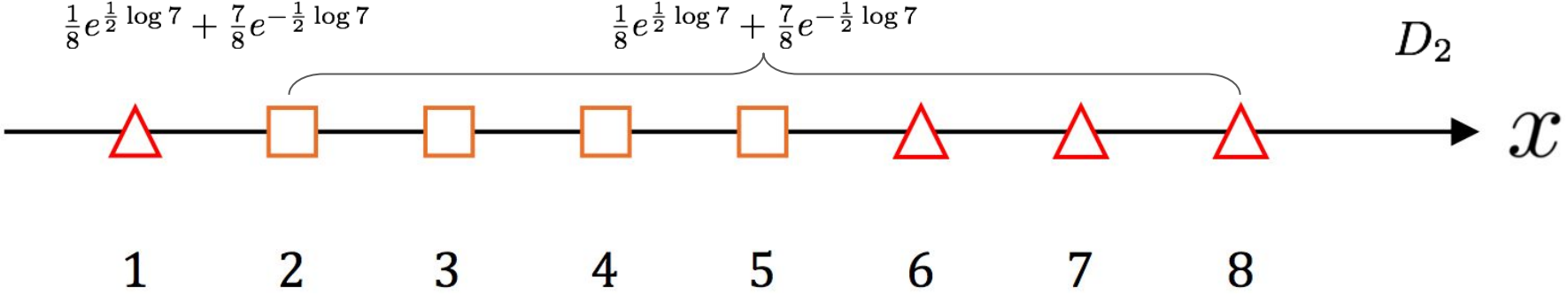
Ans:
$s = -1$ and $b \in [5, 6)$ (any b in this range is fine).
The left-most triangle will be mis-classified.
$\epsilon_1 = \frac{1}{8} = 0.125$

Boosting


$\frac{\frac{1}{8}e^{\frac{1}{2}\log 7}}{\frac{1}{8}e^{\frac{1}{2}\log 7}+\frac{7}{8}e^{-\frac{1}{2}\log 7}}$
$\frac{\frac{1}{8}e^{-\frac{1}{2}\log 7}}{\frac{1}{8}e^{\frac{1}{2}\log 7}+\frac{7}{8}e^{-\frac{1}{2}\log 7}}$
D_2
x
1
2
3
4
5
6
7
8

Boosting

$$\frac{\frac{1}{8}e^{\frac{1}{2}\log 7}}{\frac{1}{8}e^{\frac{1}{2}\log 7}+\frac{7}{8}e^{-\frac{1}{2}\log 7}} \qquad \frac{\frac{1}{8}e^{-\frac{1}{2}\log 7}}{\frac{1}{8}e^{\frac{1}{2}\log 7}+\frac{7}{8}e^{-\frac{1}{2}\log 7}}$$



$$\frac{\frac{1}{8}e^{\frac{1}{2}\log 7}}{\frac{1}{8}e^{\frac{1}{2}\log 7}+\frac{7}{8}e^{-\frac{1}{2}\log 7}}$$

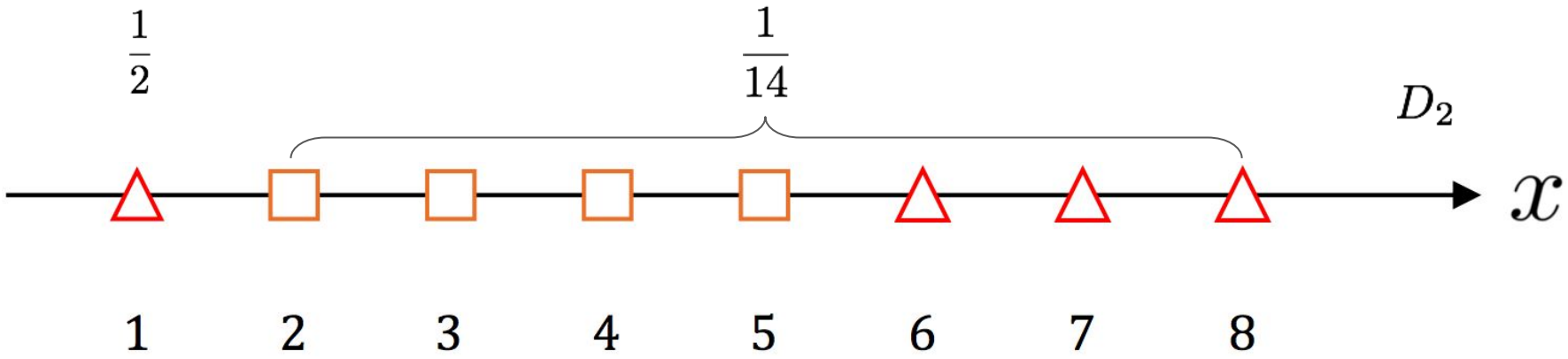
$$= \frac{e^{\frac{1}{2}\log 7}}{e^{\frac{1}{2}\log 7}+7e^{-\frac{1}{2}\log 7}}$$

$$= \frac{e^{\log 7}}{e^{\log 7}+7}$$

$$= \frac{7}{7+7} = \frac{1}{2}$$

Divide the numerator & denominator with $e^{-\frac{1}{2}\log 7}$

Boosting



$$\frac{\frac{1}{8}e^{\frac{1}{2}\log 7}}{\frac{1}{8}e^{\frac{1}{2}\log 7}+\frac{7}{8}e^{-\frac{1}{2}\log 7}}$$

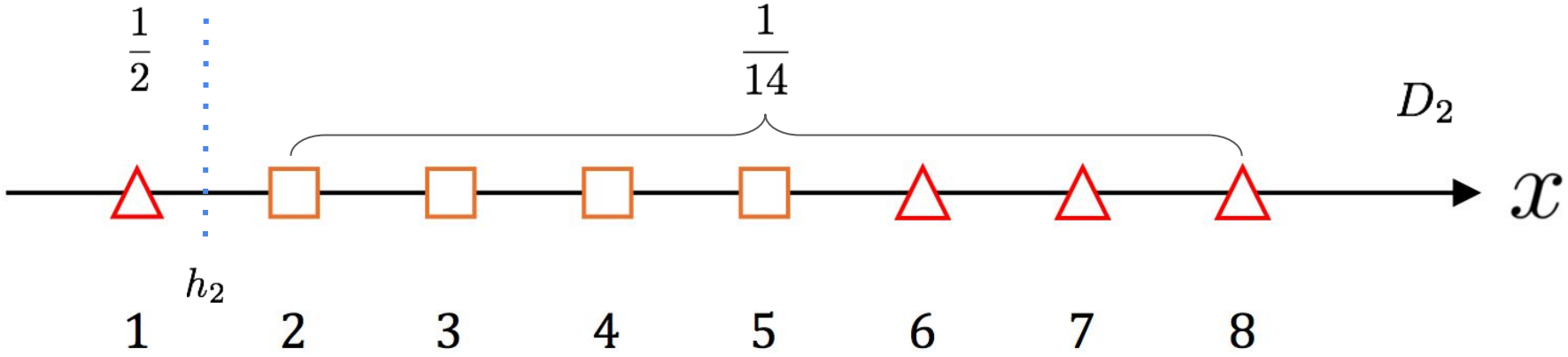
$$= \frac{e^{\frac{1}{2}\log 7}}{e^{\frac{1}{2}\log 7}+7e^{-\frac{1}{2}\log 7}}$$

$$= \frac{e^{\log 7}}{e^{\log 7}+7}$$

$$= \frac{7}{7+7} = \frac{1}{2}$$

Divide the numerator & denominator with $e^{-\frac{1}{2}\log 7}$

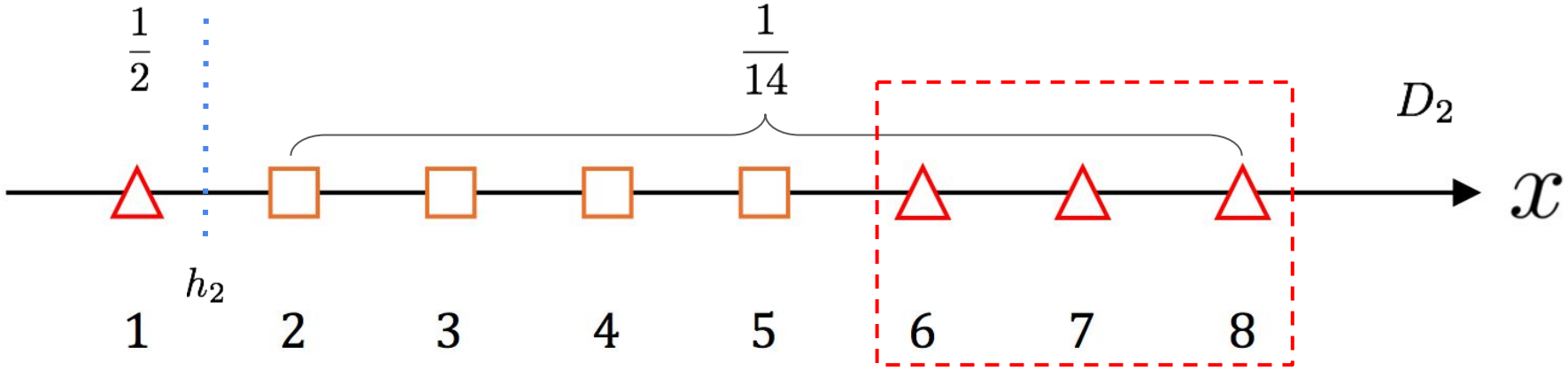
Boosting



17. Please write down the pair (s, b) of the best decision stump h_2, ϵ_2 and the mis-classified data at $t = 2$. If there are multiple equally best stump functions, just randomly pick **ONE** of them to be h_2. **(6 points)**

$s = 1$ and $b \in [1, 2)$ (any b is this range is fine).
The rightmost 3 triangles will be misclassified.
$\epsilon_2 = \frac{3}{14}$

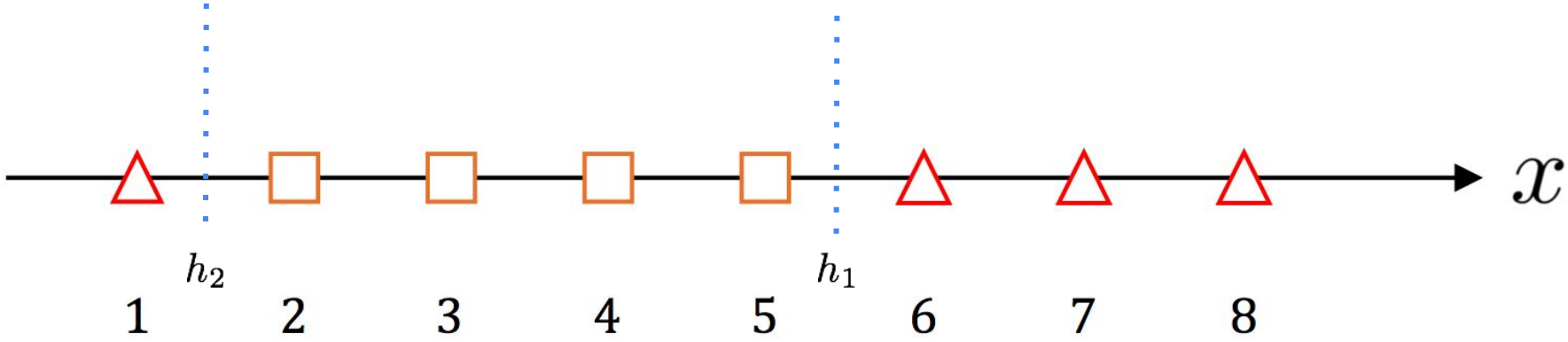
Boosting



17. Please write down the pair (s, b) of the best decision stump h_2, ϵ_2 and the mis-classified data at $t = 2$. If there are multiple equally best stump functions, just randomly pick **ONE** of them to be h_2. **(6 points)**

$s = 1$ and $b \in [1, 2)$ (any b is this range is fine).
The rightmost 3 triangles will be misclassified.
$\epsilon_2 = \frac{3}{14}$

Boosting

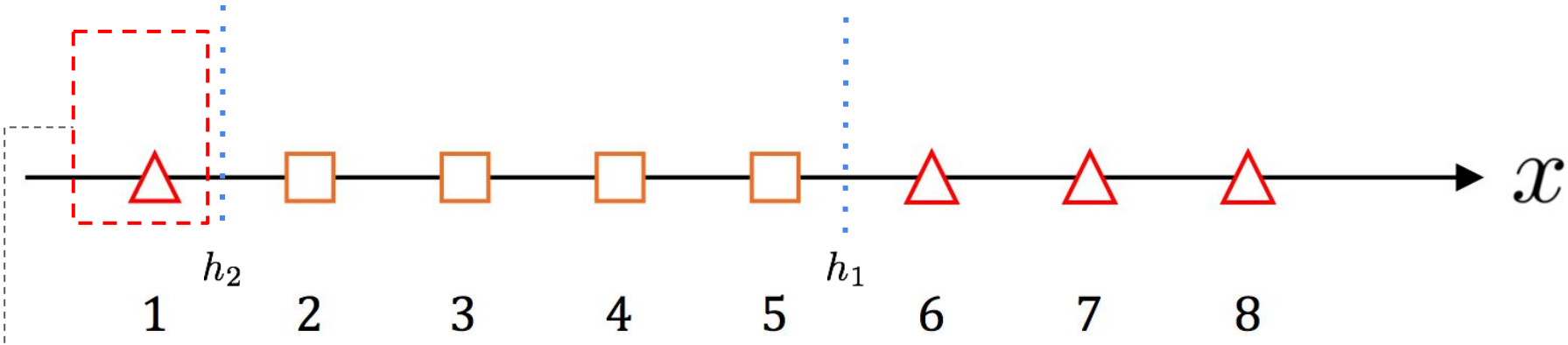


$$H(\boldsymbol{x}) = \beta_1 h_1 + \beta_2 h_2$$

$$\beta_1 = \frac{1}{2}\log 7$$

$$\beta_2 = \frac{1}{2}\log \frac{11}{3}$$

Boosting



$$H(\boldsymbol{x}) = \beta_1 h_1 + \beta_2 h_2$$

$$\beta_1 = \frac{1}{2}\log 7 \qquad \beta_2 = \frac{1}{2}\log\frac{11}{3}$$

18. Suppose we run AdaBoost for two rounds and observe that β_1 and β_2 are both positive but not equal. Will the training accuracy of the final classifier H after these two rounds (see Line 2) be 1? Explain why or why not. **(3 points)**

Ans:
No.

Data at 1 will always be misclassified.

SVM

4. Which of the following is not a true statement about Lagrangian duality?

   (a) The lagrangian dual function is convex.

   (b) They can be solved with convex optimization

   (c) Duality lets us formulate optimality conditions for constrained optimization problems.

   (d) They can be optimized in the dual space.

Ans: A

Review - Duality

Given a general minization problem,

$$\min_{x \in \mathbb{R}^n} f(x)$$

$$s.t. h_i(x) \leq 0, i = 1, ..m$$

$$l_i(x) = 0, j = 1, ...r$$

These do not need to be convex functions. We can write the Lagrangian:

$$L(x, u, v) = f(x) + \sum u_i h_i(x) + \sum v_i l_i(x)$$

Contrain u_i to be non-negative. One property of the Lagrangian is:

$$f(x) \geq L(x, u, v)$$

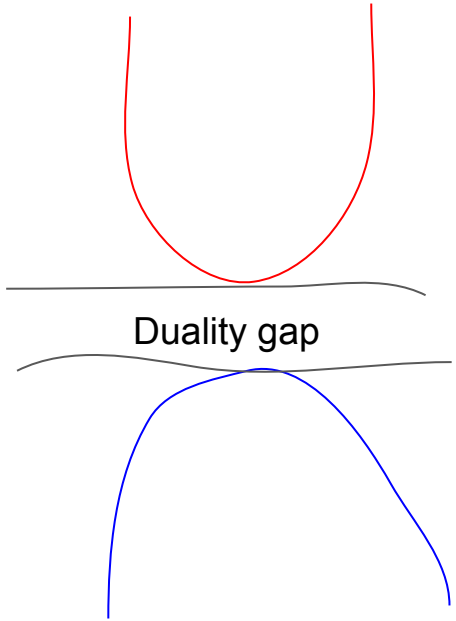
where x is feasible. So it's a lower bound of $f(x)$ over the feasible set. Minimize the both sides, we have

$$f^* \geq \min_{x \in C} L(x, u, v) \geq \min_{x \in \mathbb{R}^n} L(x, u, v) = g(u, v)$$

Here $g(u, v)$ is called the dual funciton, which provides a lower bound of the primal optimal value f^*. To get the best lower bound, g is maximized over u, v, yielding the dual problem

$$\max_{v \in \mathbb{R}^m, v \in \mathbb{R}^r} g(u, v)$$

$$s.t. u \geq 0$$

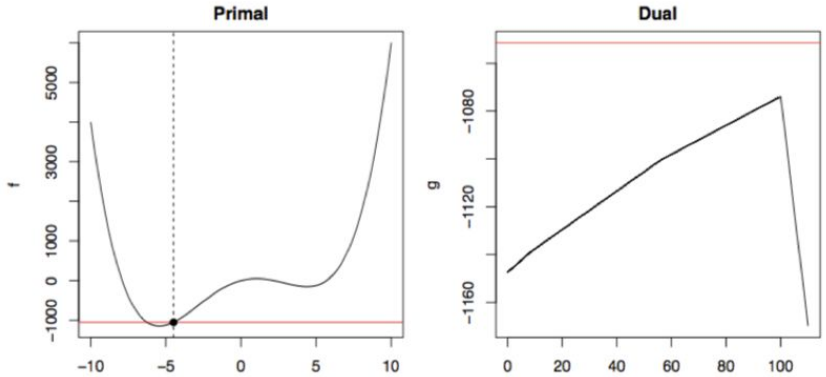


Duality Example

Example:

$$\min f(x) = x^4 - 50x^2 + 100x, s.t. x \geq -4.5$$

Minimizing the Lagrangian over x involves the differential of f, which is a cubic function. It has a closed-form solution of the roots.



Minimizing a convex function and maxmizing a concave function over a convex set are both convex problems

SVM

5. In a soft margin SVM, what is the behavior of the width of the margin $\left(\frac{1}{\|w\|}\right)$ as $C \to \infty$

   (a) Behaves like hard margin.

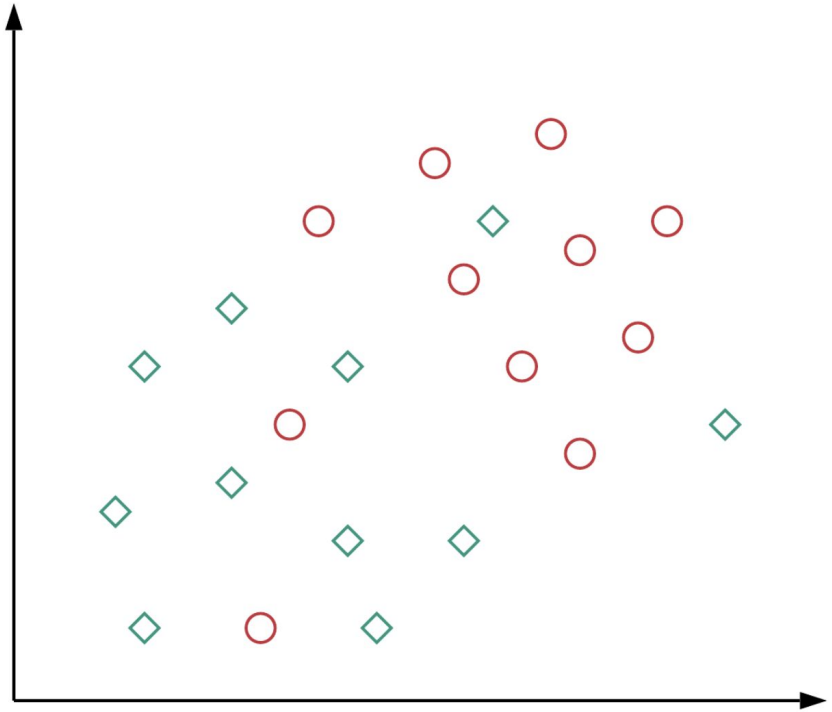
   (b) Goes to zero

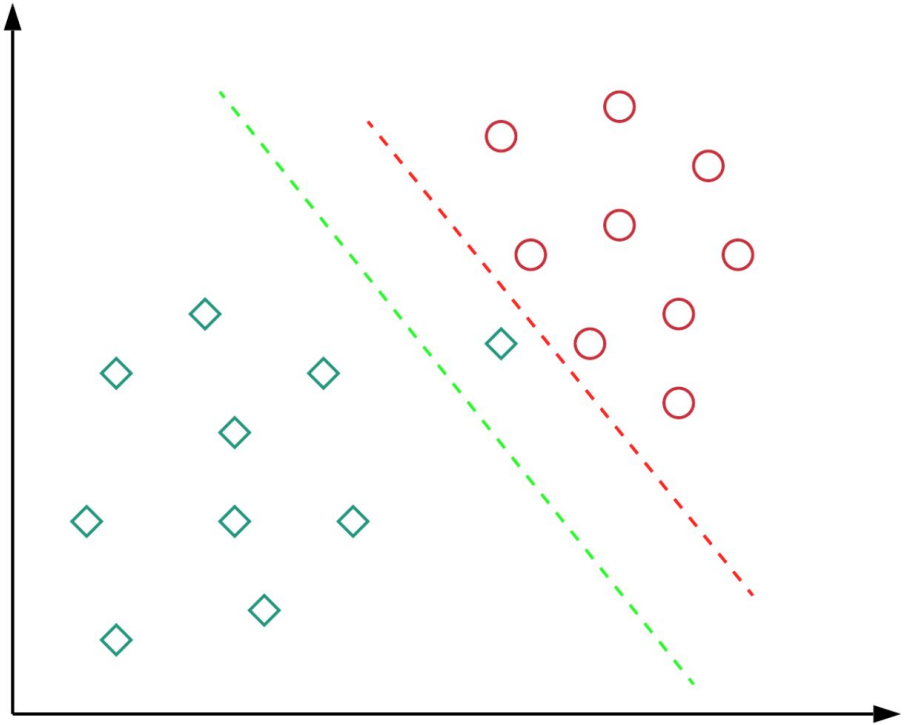
   (c) Goes to infinity.

   (d) None of the above.

Ans: A

SVM - Motivation of Soft Margin





Soft Margin SVM

The objective function becomes

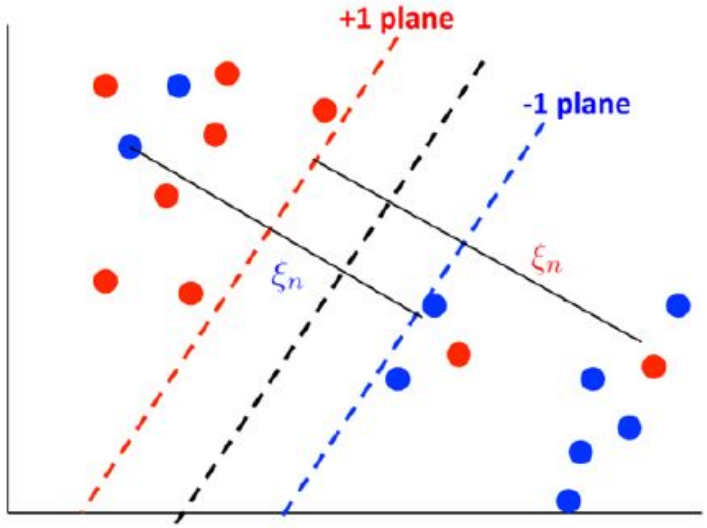
$$\min_{\boldsymbol{w},b,\{\xi_n\}} \frac{1}{2}\|\boldsymbol{w}\|_2^2 + C\sum_n \xi_n$$

subject to

$$y_n(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}_n) + b) \geq 1 - \xi_n, \quad \forall n$$
$$\xi_n \geq 0, \quad \forall\, n$$

where C is a new hyperparameter.

This formulation is called the soft-margin SVM.



https://towardsdatascience.com/support-vector-machines-soft-margin-formulation-and-kernel-trick-4c9729dc8efe

SVM

6. Which of the following statements is not true about SVM?

   (a) For two dimensional data points, the separating hyperplane learned by a linear SVM will be a straight line.

   (b) In theory, a Gaussian kernel SVM can model any complex separating hyperplane.

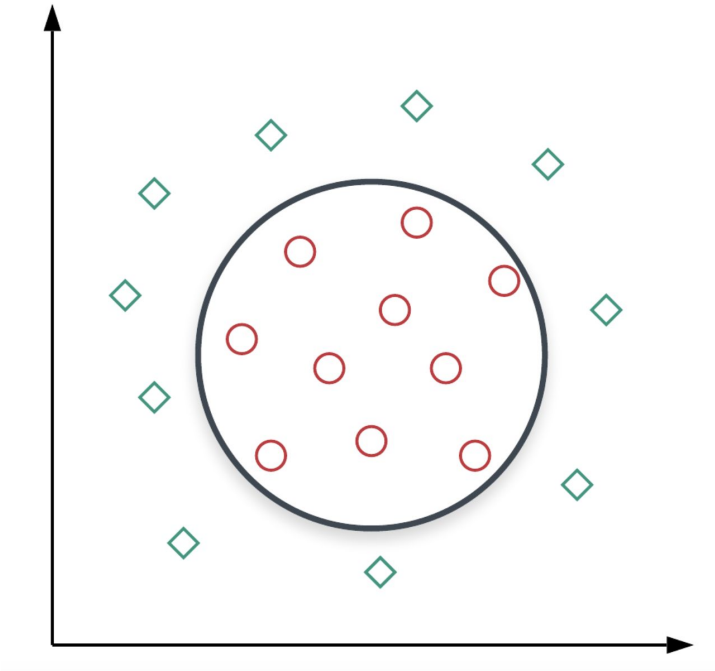
   (c) The support vectors are expected to remain the same between linear kernels and higher?order polynomial kernels.

   (d) Overfitting in an SVM is a function of number of support vectors.

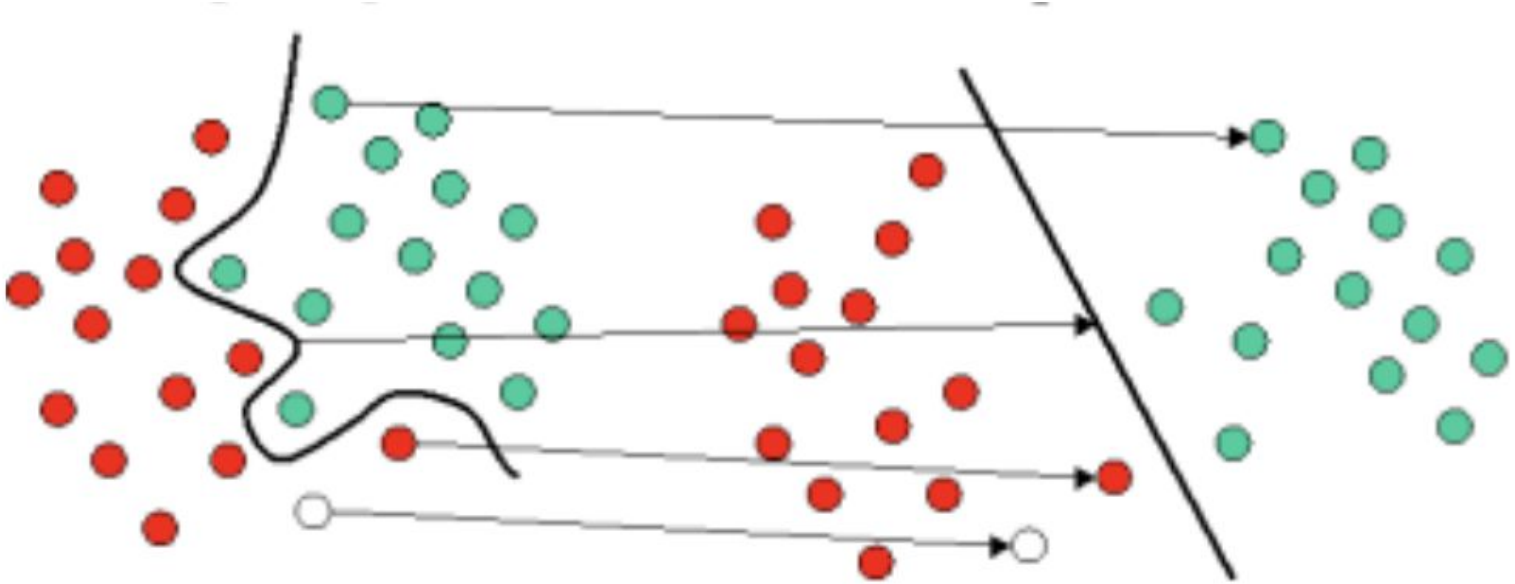
   Ans: C

SVM - Kernel Trick

“Kernel Trick” to tackle the problem of linear inseparability.



SVM - Overfitting



SVM

Consider a max-margin linear SVM that solves the following optimization problem.

$$\begin{aligned} \min_{\mathbf{w},b} \quad & \frac{1}{2}\|\mathbf{w}\|^2 \\ \text{s.t.} \quad & y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1, \quad i = 1, \ldots, n \end{aligned}$$

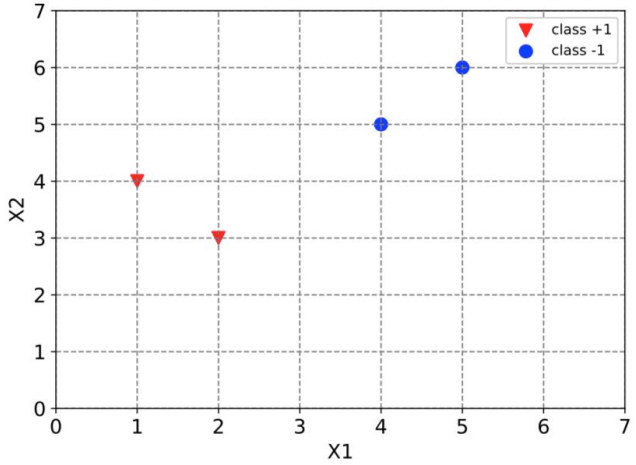
where the optimization parameters are $\mathbf{w}, b$, and the training set consists of points $\{(\mathbf{x}, y)\}_{i=1}^n$. $\mathbf{x}$ is a vector of real values, and $y \in \{-1, 1\}$ is the class label. In this section, you will derive the dual optimization problems.

12. Write down the Lagrangian $\mathcal{L}$ of the above SVM optimization problem. Express your answer as the Lagrangian $\mathcal{L}$ in terms of $\mathbf{w}, b, \alpha_i$, where α_i is the Lagrange multiplier for the inequality constraint at each data instance i. (5 points)

Ans: $\mathcal{L} = \frac{1}{2}\|\mathbf{w}\|^2 + \sum_{i=1}^n \alpha_i \left(1 - y_i(\mathbf{w}^T\mathbf{x}_i + b)\right)$, where $a_i \geq 0$

SVM

13. Support vector machines learn a decision boundary leading to the largest margin from both classes. You are training SVM on a tiny dataset with 4 points shown in the below Figure. This dataset consists of two examples with class label -1 (denoted with plus), and two examples with class label +1 (denoted with triangles). Find the weight vector **w** and bias b. What's the equation corresponding to the decision boundary?



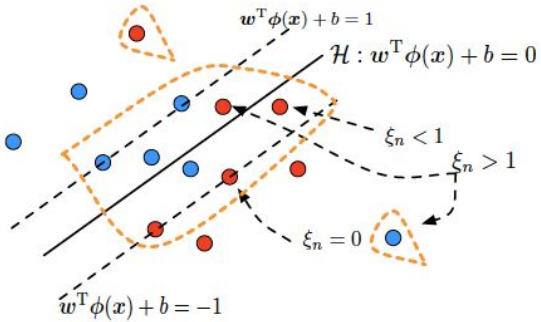
Geometric interpretation of support vectors

A support vector satisfies $\alpha_n^* \neq 0$ and

$$1 - \xi_n^* \leq y_n(\boldsymbol{w}^{*\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}_n) + b^*)$$

When

- $\xi_n^* = 0$, $y_n(\boldsymbol{w}^{*\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}_n) + b^*) = 1$ and thus the point is $1/\|\boldsymbol{w}^*\|_2$ away from the hyperplane.
- $\xi_n^* < 1$, the point is classified correctly but does not satisfy the large margin constraint.
- $\xi_n^* > 1$, the point is misclassified.



Support vectors (circled with the orange line) are *the only points that matter!*

SVM

Ans: SVM tries to maximize the margin between two classes. Therefore, the optimal decision boundary is diagonal and it crosses the point (3,4). It is perpendicular to the line between support vectors (4,5) and (2,3), hence it is slope is $m = -1$. Thus the line equation is $(x_2 - 4) = -1(x_1 - 3) = x_1 + x_2 = 7$. From this equation, we can deduce that the weight vector has to be of the form (w_1, w_2), where $w_1 = w_2$. It also has to satisfy the following equations:

$2w_1 + 3w_2 + b = 1$ and

$4w_1 + 5w_2 + b = -1$

Hence, $w_1 = w_2 = -1/2$ and $b = 7/2$

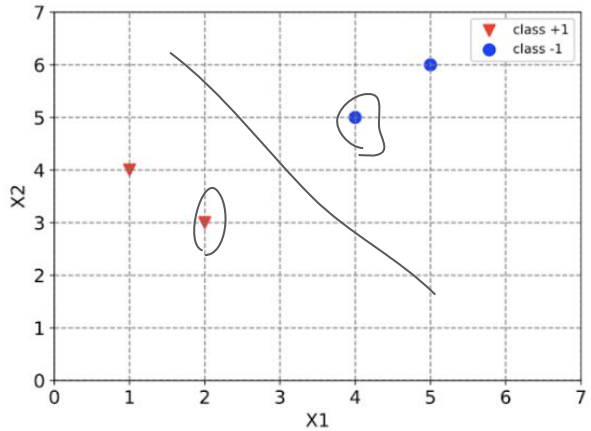
$$\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}) + b = 1$$

$$\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}) + b = 0$$

We have three unknown variables (w1, w2, b) and their equations, thus we can solve them easily.

SVM

14. Circle the support vectors and draw the decision boundary.



Definition

A Markov chain is a stochastic process with the **Markov property**: a sequence of random variables $X_1, X_2, \cdots, X_T$ s.t.

$$P(X_{t+1}|X_1, X_2, \cdots, X_t) = P(X_{t+1}|X_t)$$

i.e. *the current state only depends on the most recent state*.

We denote the transition and initial probabilities as

$$a_{s,s'} = P(X_{t+1} = s'|X_t = s), \quad \pi_s = P(X_1 = s)$$

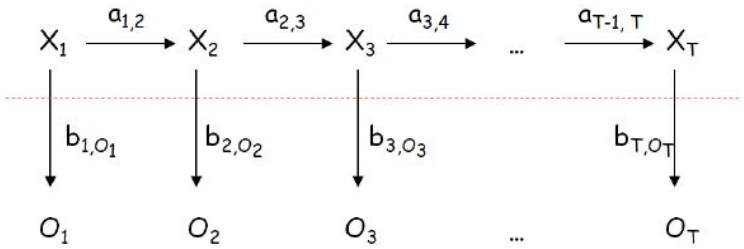
Each state $X_t \in 1, 2, \ldots, S$ also "emits" some **outcome** O_t based on the following model

$$P(O_t \mid X_t = s) = b_{s,O_t} \qquad \textbf{(emission probability)}$$

independent of anything else.

The model parameters are $(\{\pi_s\}, \{a_{s,s'}\}, \{b_{s,O_t}\}) = (\boldsymbol{\pi}, \boldsymbol{A}, \boldsymbol{B})$.

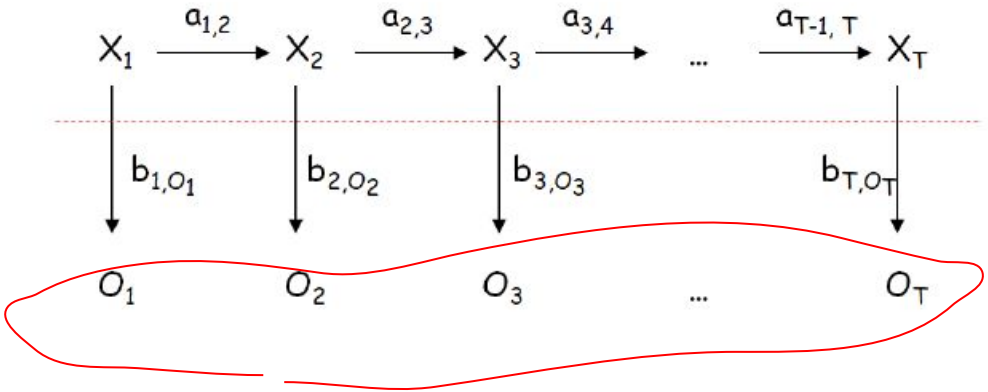
A generic hidden Markov model is illustrated in this picture:



- **Problem 1**: Scoring and evaluation

Given an observation sequence $O_1, O_2, \ldots, O_T$ and a model $(\boldsymbol{\pi}, \boldsymbol{A}, \boldsymbol{B})$, how to compute efficiently the probability of $P(O_1, O_2, \ldots, O_T)$?

A generic hidden Markov model is illustrated in this picture:

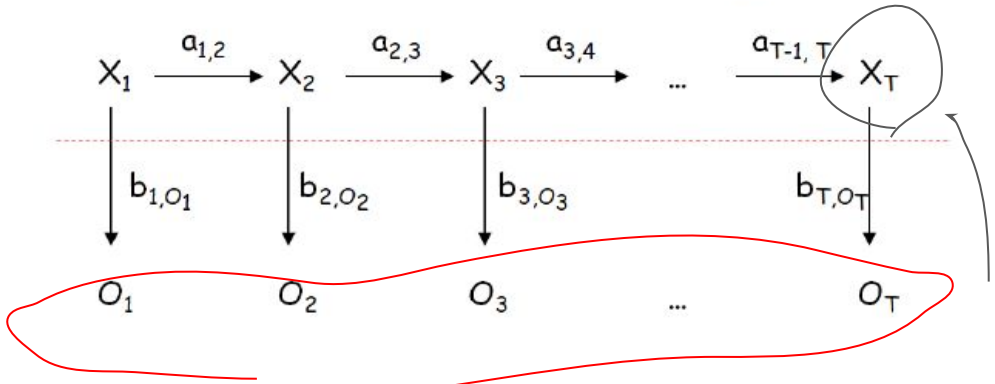


- **forward messages**: for each s and t

$$\alpha_s(t) = P(X_t = s, O_{1:t} = o_{1:t})$$

The intuition is, if we observe up to time t, what is the likelihood of the Markov chain in state s?

A generic hidden Markov model is illustrated in this picture:

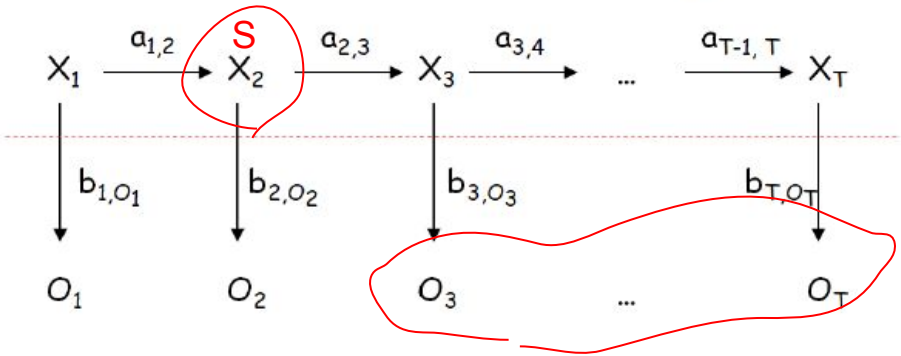


- **backward messages**: for each s and t

$$\beta_s(t) = P(O_{t+1:T} = o_{t+1:T} \mid X_t = s)$$

The interpretation is: if we are told that the Markov chain at time t is in the state s , then what are the likelihood of observing future observations from $t+1$ to T?

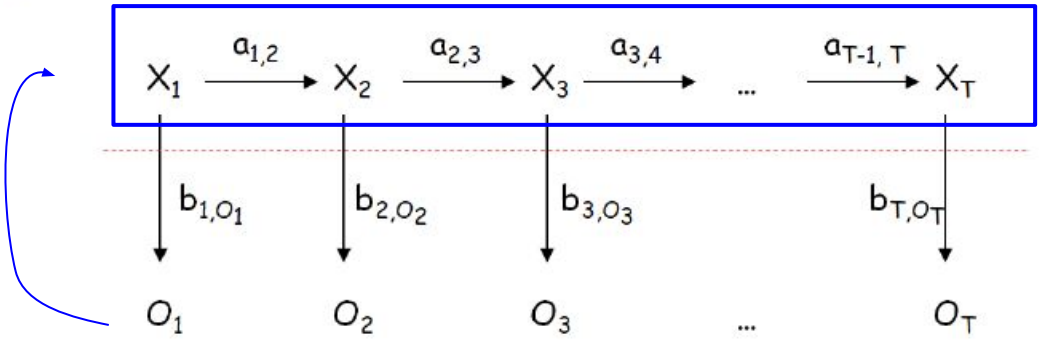
A generic hidden Markov model is illustrated in this picture:



- **Problem 2**: Decoding (Viterbi algorithm)

Given an observation sequence $O_1, O_2, \ldots, O_T$ and a model $(\boldsymbol{\pi}, \boldsymbol{A}, \boldsymbol{B})$, how do we determine the optimal corresponding state sequence $X_1, X_2, \ldots, X_T$ that best explains how the observations were generated?

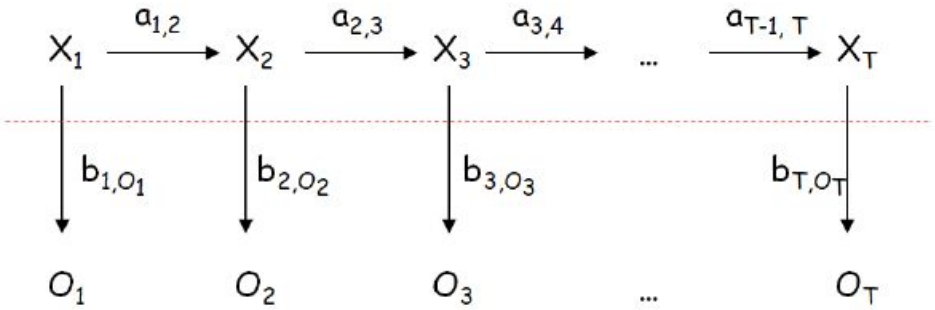
A generic hidden Markov model is illustrated in this picture:



- **Problem 3**: Training

Given an observation sequence $O_1, O_2, \ldots, O_T$, how to adjust the parameters $(\boldsymbol{\pi}, \boldsymbol{A}, \boldsymbol{B})$ to maximize the probability of $P(O_1, O_2, \ldots, O_T)$? In the other words, find a model to best fit the observed data. we will solve this by the Baum–Welch algorithm.

A generic hidden Markov model is illustrated in this picture:



HMM

7. Which of the following statements of hidden Markov model (HMM) is true?

   (a) We can infer the backward message at time t from the backward message at time t + 1 using the backward algorithm.

   The definition of backward message

   (b) Given a sequence of observations and a learned HMM, we can infer the real corresponding path of hidden states.

   Not real, instead, the most possible

   (c) We can learn a HMM using the forward algorithm.

   Using the Baum-Welsh (forward-backward) algorithm

   (d) None of the above.

   With forward algorithm, we can do likelihood computation.

Ans: A

Computing backward messages

Again establish a recursive formula

$$\beta_s(t) = P(O_{t+1:T} \mid X_t = s) = P(O_{t+1:T}, X_t = s)/P(X_t = s) =$$

$$= \sum_{s'} P(O_{t+1:T}, X_{t+1} = s', X_t = s)/P(X_t = s) \qquad \text{(marginalizing)}$$

$$= \sum_{s'} P(O_{t+1:T} \mid X_{t+1} = s', X_t = s) P(X_{t+1} = s' \mid X_t = s)$$

$$= \sum_{s'} a_{s,s'} P(O_{t+1:T} \mid X_{t+1} = s') = \sum_{s'} a_{s,s'} P(O_{t+1}, O_{t+2:T} \mid X_{t+1} = s')$$

$$= \sum_{s'} a_{s,s'} P(O_{t+1} \mid O_{t+2:T}, X_{t+1} = s') P(O_{t+2:T} \mid X_{t+1} = s')$$

$$= \sum_{s'} a_{s,s'} b_{s',o_{t+1}} \beta_{s'}(t+1)$$

Base case: $\beta_s(T) = 1$ (prove it!)

HMM

8. Both GMM and HMM can be learned by applying the EM algorithm.

   (a) True

   (b) False

Ans: A

HMM

9. Which is not true about the Baum-Welsh algorithm?

   (a) It is used to find unknown parameters of a hidden markov model.

   (b) It uses a forward-backward algorithm to maximize the probability of an observation.

   (c) It computes the most likely sequence of hidden states given an observation sequence.

   (d) It is a special case of the EM algorithm.

Viterbi decoding

Ans: C

HMM

Recall a hidden Markov model is parameterized by

- initial state distribution $P(X_1 = s) = \pi_s$
- transition distribution $P(X_{t+1} = s' | X_t = s) = a_{s,s'}$
- emission distribution $P(O_t = o | X_t = s) = b_{s,o}$

1. Given a sequence of observations $O_{1:t}$, compute the previous Viterbi path probability from the current time step, which is $\max_{x_{1:t-1}} P(X_t = s, X_{1:t-1}, O_{1:t})$ in terms of forward message, backward message, transition probabilities, emission probabilities as needed. **(4 points)**

$$\alpha_s(t) = P(X_t = s, O_{1:t} = o_{1:t})$$

$$\beta_s(t) = P(O_{t+1:T} = o_{t+1:T} | X_t = s)$$

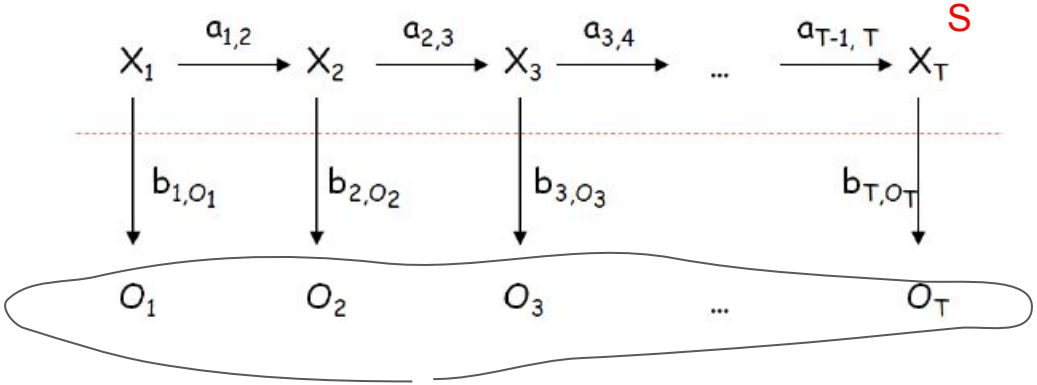
Actually, this is the delta we calculated when doing viterbi decoding.

We define DP subproblems in the following way – the highest probable state sequence that ends at $X_t = s$ given observations $O_1, O_2, \ldots, O_t$

$$\delta_s(t) = \max_{X_{1:t-1}} P(X_{1:t-1}, X_t = s, O_{1:t})$$

In the next slide we compute $\delta_s(t)$ recursively.

A generic hidden Markov model is illustrated in this picture:



HMM

Review Page 20/43, Lecture 11

$$\delta_s(t) = \max_{x_{1:t-1}} P(X_t = s, X_{1:t-1}, O_{1:t})$$

$$= \max_{x_{1:t-1}} P(X_t = s, O_t, X_{1:t-1}, O_{1:t-1})$$

$$= \max_{s'} \max_{x_{1:t-2}} P(X_t = s, O_t | X_{1:t-1}, O_{1:t-1}) P(X_{t-1} = s', X_{1:t-2}, O_{1:t-1})$$

$$= \max_{s'} \delta_{s'}(t-1) P(X_t = s, O_t | X_{1:t-1}, O_{1:t-1})$$

$$= \max_{s'} \delta_{s'}(t-1) P(O_t, X_t | X_{1:t-1})$$

Bayes rule

$$= \max_{s'} \delta_{s'}(t-1) P(O_t | X_t, X_{1:t-1}) P(X_t | X_{1:t-1})$$

$$= \max_{s'} \delta_{s'}(t-1) P(O_t | X_t) P(X_t | X_{t-1})$$

$$= b_{s,o} \max_{s'} a_{s',s} \delta_{s'}(t-1)$$

HMM

2. Suppose we have a HMM model, each O_t takes a value in $\{A, C, G, T\}$ and each X_t takes one of the two possible states $\{s_1, s_2\}$. This HMM has the following parameters $\mathbf{\Theta} = \{\pi_i, a_{ij}, b_{ik}\}$ for $i \in \{1, 2\}$, $j \in \{1, 2\}$, and $k \in \{A, C, G, T\}$:

$$\pi_1 = P(X_1 = s_1) = 0.7; \quad \pi_2 = P(X_1 = s_2) = 0.3$$

$$a_{11} = 0.8, \quad a_{12} = 0.2, \quad a_{21} = 0.4, \quad a_{22} = 0.6$$

$$b_{1A} = 0.4, \quad b_{1C} = 0.1, \quad b_{1G} = 0.4, \quad b_{1T} = 0.1$$

$$b_{2A} = 0.2, \quad b_{2C} = 0.3, \quad b_{2G} = 0.2, \quad b_{2T} = 0.3$$

$$\delta_s(t) = \max_{x_{1:t-1}} P(X_t = s, X_{1:t-1}, O_{1:t}) = b_{s,o_t} \max_{s'} a_{s',s} \delta_{s'}(t-1)$$

$$\Delta_s(t) = \operatorname*{argmax}_{x_{t-1}} \max_{x_{1:t-2}} P(X_t = s, X_{1:t-1} | O_{1:t}) = \operatorname*{argmax}_{\boldsymbol{s'}} a_{s',s} \delta_{s'}(t-1)$$

Now we have observed the output sequence [A G], what is the most likely sequence of states that produce this? **(7 points)**

Viterbi Algorithm

For each $s \in [N]$, compute $\delta_s(1) = \pi_s b_{s,o_1}$.

For each $t = 2, \ldots, T$,

- for each $s \in [N]$, compute

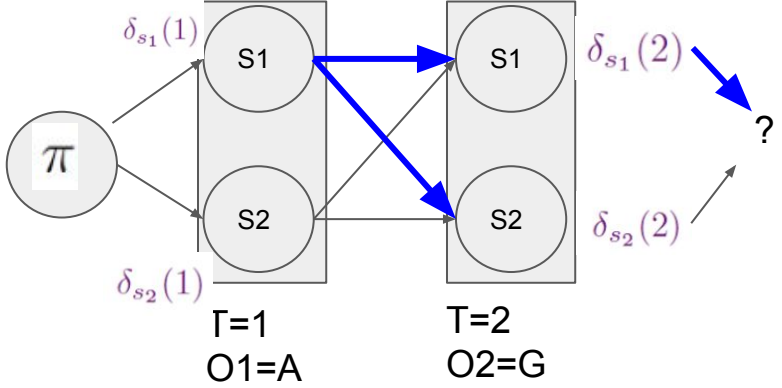
$$\delta_s(t) = b_{s,o_t} \max_{s'} a_{s',s} \delta_{s'}(t-1)$$

$$\Delta_s(t) = \operatorname*{argmax}_{s'} a_{s',s} \delta_{s'}(t-1)$$

Backtracking: let $o_T^* = \operatorname{argmax}_s \delta_s(T)$.
For each $t = T, \ldots, 2$: set $o_{t-1}^* = \Delta_{o_t^*}(t)$.

Output the most likely path $o_1^*, \ldots, o_T^*$.

HMM - Viterbi



Ans:

$$\delta_{s_1}(1) = \pi_{s_1} b_{s_1,o_1} = 0.7 * 0.4 = 0.28$$

$$\delta_{s_2}(1) = \pi_{s_2} b_{s_2,o_1} = 0.3 * 0.2 = 0.06 \textbf{(2 points)}$$

$$\delta_{s_1}(2) = b_{s_1,o_2} \max_{s'} a_{s',s} \delta_{s'}(1) = 0.4 * \max\{0.8 * 0.28, 0.4 * 0.06\} = 0.4 * 0.8 * 0.28 = 0.0896$$

$$\delta_{s_2}(2) = b_{s_2,o_2} \max_{s'} a_{s',s} \delta_{s'}(1) = 0.2 * \max\{0.2 * 0.28, 0.6 * 0.06\} = 0.2 * 0.2 * 0.28 = 0.0112 \textbf{(2 points)}$$

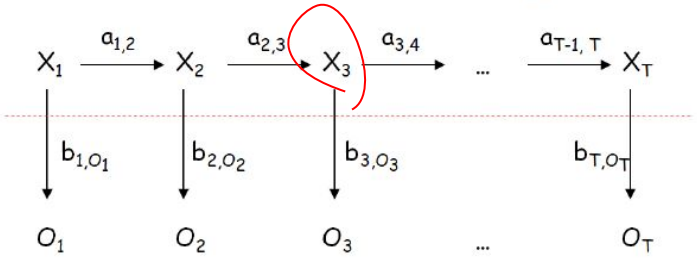
$$\Delta_{s_1}(2) = \operatorname{argmax}_{s'} a_{s',s} \delta_{s'}(1) = s_1$$

$$\Delta_{s_2}(2) = \operatorname{argmax}_{s'} a_{s',s} \delta_{s'}(1) = s_1 \textbf{(1 points)}$$

since $\delta_{s_1}(2) > \delta_{s_2}(2)$, so we choose $x_2^* = s_1$, using trace-back table, then $x_1^* = s_1$ **(2 points)**

HMM

A generic hidden Markov model is illustrated in this picture:



3. For an arbitrary $T_0 < T$, it is possible to compute the most likely hidden state path $X_1^*, \ldots, X_{T_0}^*$ given the entire observations $O_1, \ldots, O_T$. Your task is to compute ONLY the last state $X_{T_0}^*$. Feel free to use δ_s, forward and backward messages as needed. **(4 points)**

Ans:

$$x_{T_0}^* = \underset{s}{\text{argmax}} \max_{x_{1:T_0-1}} P\left(X_{T_0} = s, X_{1:T_0-1} = x_{1:T_0-1}, O_{1:T} = o_{1:T}\right) \textbf{(3 points)}$$

$$= \underset{s}{\text{argmax}} \max_{x_{1:T_0-1}} P\left(X_{T_0} = s, X_{1:T_0-1} = x_{1:T_0-1}, O_{1:T_0} = o_{1:T_0}\right) \times$$

$$P\left(O_{T_0+1:T} = o_{T_0+1:T} | X_{T_0} = s, X_{1:T_0-1} = x_{1:T_0-1}, O_{1:T_0} = o_{1:T_0}\right) \textbf{(2 points)}$$

$$= \underset{s}{\text{argmax}} \left(\max_{x_{1:T_0-1}} P\left(X_{T_0} = s, X_{1:T_0-1} = x_{1:T_0-1}, O_{1:T_0} = o_{1:T_0}\right) \right) P\left(O_{T_0+1:T} = o_{T_0+1:T} | X_{T_0} = s\right) \textbf{(1 points}$$

$$= \underset{s}{\text{argmax}} \delta_s\left(T_0\right) \beta_s\left(T_0\right) \textbf{(1 points)}$$

P(AB)=P(A)*P(B|A)

Markov property

HMM

One more suggestion:

Review lecture 10 and 11.