10-701 Machine Learning - Spring 2012
Problem Set 5

Out: April 11th, 1:30pm
In: April 25th, 1:30pm

TA: Hai-Son Le (hple@cs.cmu.edu)
School Of Computer Science, Carnegie Mellon University

- Homework will be done individually: each student must hand in their own answers. It is acceptable for students to collaborate in figuring out answers and helping each other solve the problems. We will be assuming that, as participants in a graduate course, you will be taking the responsibility to make sure you personally understand the solution to any work arising from such collaboration. You also must indicate on each homework with whom you collaborated.
- Homework is due at the beginning of class on the due date.

1 Hidden Markov Model [60 points]

Hidden Markov Model is an instance of the state space model in which the latent variables are discrete. Let K be the number of hidden states. We use the following notations: $\mathbf{x}$ are the observed variables, $\mathbf{z}$ are the hidden state variables (we use 1-of-K representation: $z_k = 1$, $z_{j\neq k} = 0$ means the hidden state is k). The transition probabilities are given by a $K \times K$ matrix $\mathbf{A}$, where $A_{jk} = p(z_{n,k} = 1|z_{n-1,j} = 1)$ and the initial state variable $\mathbf{z}_1$ are given by a vector of probabilities π: $p(\mathbf{z}_1|\pi) = \prod_{k=1}^{K} \pi_k^{z_{1k}}$. Finally, the emission distribution for a hidden state k is parametrized by ϕ_k: $p(\mathbf{x}_n|\phi_k)$. Let $\mathbf{\Theta} = \{\mathbf{A}, \pi, \phi\}$.

1.1 The full likelihood of a data set

If we have a data set $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$:

1. [**3 points**] What is the full likelihood of observed and latent variables: $p(\mathbf{X}, \mathbf{Z}|\mathbf{\Theta})$? Note $\mathbf{Z} = \{\mathbf{z}_1, \ldots, \mathbf{z}_N\}$ are the hidden states of the corresponding observations.

$$p(\mathbf{X}, \mathbf{Z}|\mathbf{\Theta}) = p(\mathbf{z}_1) \prod_{n=2}^{N} p(\mathbf{z}_n|\mathbf{z}_{n-1}) \prod_{n=1}^{N} p(x_n|z_n) \tag{1}$$

$$= \prod_{k=1}^{K} \pi_k^{z_{1k}} \prod_{n=2}^{N} \prod_{k=1}^{K} \prod_{j=1}^{K} A_{jk}^{z_{nk} z_{n-1,j}} \prod_{n=1}^{N} \prod_{k=1}^{K} p(x_n|\phi_k)^{z_{nk}} \tag{2}$$

2. **[2 points]**What is the likelihood of the data set? (e.g. $p(\mathbf{X}|\mathbf{\Theta})$.

$$p(\mathbf{X}|\mathbf{\Theta}) = \sum_{\mathbf{Z}} p(\mathbf{X}, \mathbf{Z}|\mathbf{\Theta}) \tag{3}$$

1.2 Expectation-Maximization (EM) for Maximum Likelihood Learning

We'd like to derive formulas for estimating $\mathbf{A}$ and ϕ to maximize the likelihood of the data set $p(\mathbf{X}|\mathbf{\Theta})$.

1. **[5 points]** Assume we can compute $p(\mathbf{X}, \mathbf{Z}|\mathbf{\Theta})$ in $O(1)$ time complexity, what is the time complexity of computing $p(\mathbf{X}|\mathbf{\Theta})$?

$O(K^N)$

We use EM algorithm for this task:

- In the E step, we take the current parameter values and compute the posterior distribution of the latent variables $p(\mathbf{Z}|\mathbf{X}, \mathbf{\Theta}^{\text{old}})$.
- In the M step, we find the new parameter values by solving an optimization problem:

$$\Theta^{\text{new}} = \text{argmax}_{\mathbf{\Theta}} Q(\mathbf{\Theta}, \mathbf{\Theta}^{\text{old}}) \tag{4}$$

where

$$Q(\Theta, \Theta^{\text{old}}) = \sum_{\mathbf{Z}} p(\mathbf{Z}|\mathbf{X}, \mathbf{\Theta}^{\text{old}}) \ln p(\mathbf{X}, \mathbf{Z}|\mathbf{\Theta}) \tag{5}$$

2. **[10 points]** Show that

$$Q(\Theta, \Theta^{\text{old}}) = \sum_{k=1}^{K} \gamma(z_{1k}) \ln \pi_k + \sum_{n=2}^{N} \sum_{j=1}^{K} \sum_{k=1}^{K} \xi(z_{n-1,j}, z_{nk}) \ln A_{jk} \tag{6}$$

$$+ \sum_{n=1}^{N} \sum_{k=1}^{K} \gamma(z_{nk}) \ln p(\mathbf{x}_n|\phi_k) \tag{7}$$

where

$$\gamma(z_{nk}) = \mathrm{E}_{p(\mathbf{z}_n|\mathbf{X},\boldsymbol{\Theta}^{\text{old}})}[z_{nk}] \tag{8}$$

$$\xi(z_{n-1,j}, z_{nk}) = \mathrm{E}_{p(\mathbf{z}_{n-1},\mathbf{z}_n|\mathbf{X},\boldsymbol{\Theta}^{\text{old}})}[z_{n-1,j} z_{nk}] \tag{9}$$

Show your derivations.

$$Q(\Theta, \Theta^{\text{old}}) = \sum_{\mathbf{Z}} p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\Theta}^{\text{old}}) \ln p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\Theta}) \tag{10}$$

$$= \sum_{\mathbf{Z}} \Big\{ \sum_{k=1}^{K} z_{1k} \log \pi_k + \sum_{n=2}^{N} \sum_{k=1}^{K} \sum_{j=1}^{K} z_{nk} z_{n-1,j} \log A_{jk} \tag{11}$$

$$+ \sum_{n=1}^{N} \sum_{k=1}^{K} z_{nk} \log p(x_n|\phi_k) \Big\} \tag{12}$$

$$= \sum_{k=1}^{K} \gamma(z_{1k}) \ln \pi_k + \sum_{n=2}^{N} \sum_{j=1}^{K} \sum_{k=1}^{K} \xi(z_{n-1,j}, z_{nk}) \ln A_{jk} \tag{13}$$

$$+ \sum_{n=1}^{N} \sum_{k=1}^{K} \gamma(z_{nk}) \ln p(\mathbf{x}_n|\phi_k) \tag{14}$$

3. **[5 points]** Show that

$$p(\mathbf{X}|\mathbf{z}_{n-1}, \mathbf{z}_n) = p(\mathbf{x}_1, \ldots, \mathbf{x}_{n-1}|\mathbf{z}_{n-1}) p(\mathbf{x}_n|\mathbf{z}_n) p(\mathbf{x}_{n+1}, \ldots \mathbf{x}_N|\mathbf{z}_n) \tag{15}$$

This follows from the following D-separations:

$$p(\mathbf{X}|\mathbf{z}_{n-1}, \mathbf{z}_n) = p(\mathbf{x}_1, \ldots, \mathbf{x}_{n-1}|\mathbf{z}_{n-1}, \mathbf{z}_n) p(\mathbf{x}_n|\mathbf{z}_{n-1}, \mathbf{z}_n) p(\mathbf{x}_{n+1}, \ldots \mathbf{x}_N|\mathbf{z}_{n-1}, \mathbf{z}_n) \tag{16}$$

$$= p(\mathbf{x}_1, \ldots, \mathbf{x}_{n-1}|\mathbf{z}_{n-1}) p(\mathbf{x}_n|\mathbf{z}_n) p(\mathbf{x}_{n+1}, \ldots \mathbf{x}_N|\mathbf{z}_n) \tag{17}$$

4. **[10 points]** In class, we discuss how to compute:

$$\alpha(\mathbf{z}_n) = p(\mathbf{x}_1, \ldots, \mathbf{x}_n, \mathbf{z}_n) \tag{18}$$

$$\beta(\mathbf{z}_n) = p(\mathbf{x}_{n+1}, \ldots, \mathbf{x}_N | \mathbf{z}_n) \tag{19}$$

Show that

$$\xi(\mathbf{z}_{n-1}, \mathbf{z}_n) = p(\mathbf{z}_{n-1}, \mathbf{z}_n | \mathbf{X}) \tag{20}$$

$$= \frac{\alpha(\mathbf{z}_{n-1}) p(\mathbf{x}_n | \mathbf{z}_n) p(\mathbf{z}_n | \mathbf{z}_{n-1}) \beta(\mathbf{z}_n)}{p(\mathbf{X})} \tag{21}$$

$$\xi(\mathbf{z}_{n-1}, \mathbf{z}_n) = p(\mathbf{z}_{n-1}, \mathbf{z}_n | \mathbf{X}) \tag{22}$$

$$= \frac{p(\mathbf{X} | \mathbf{z}_{n-1}, \mathbf{z}_n) p(\mathbf{z}_{n-1}, \mathbf{z}_n)}{p(\mathbf{X})} \tag{23}$$

$$= \frac{p(\mathbf{x}_1, \ldots, \mathbf{x}_{n-1} | \mathbf{z}_{n-1}) p(\mathbf{x}_n | \mathbf{z}_n) p(\mathbf{x}_{n+1}, \ldots, \mathbf{x}_N | \mathbf{z}_n) p(\mathbf{z}_n | \mathbf{z}_{n-1}) p(\mathbf{z}_{n-1})}{p(\mathbf{X})} \tag{24}$$

$$= \frac{\alpha(\mathbf{z}_{n-1}) p(\mathbf{x}_n | \mathbf{z}_n) p(\mathbf{z}_n | \mathbf{z}_{n-1}) \beta(\mathbf{z}_n)}{p(\mathbf{X})} \tag{25}$$

How would you compute $p(\mathbf{X})$?

$$p(\mathbf{X}) = \sum_{\mathbf{z}_N} \alpha(\mathbf{z}_N) \tag{26}$$

5. **[5 points]** Show how to compute $\gamma(z_{nk})$ and $\xi(z_{n-1,j}, z_{nk})$ using $\alpha(\mathbf{z}_n)$, $\beta(\mathbf{z}_n)$ and $\xi(\mathbf{z}_{n-1}, \mathbf{z}_n)$.

$$\gamma(z_{nk}) = \mathrm{E}_{p(\mathbf{z}_n | \mathbf{X}, \boldsymbol{\Theta}^{\text{old}})}[z_{nk}] \tag{27}$$

$$= p(z_{nk} = 1 | \mathbf{X}, \boldsymbol{\Theta}^{\text{old}}) \tag{28}$$

$$= \frac{\alpha(\mathbf{z} : z_{nk} = 1) \beta(\mathbf{z} : z_{nk} = 1)}{p(\mathbf{X})} \tag{29}$$

$$\xi(z_{n-1,j}, z_{nk}) = \mathrm{E}_{p(\mathbf{z}_{n-1}, \mathbf{z}_n | \mathbf{X}, \boldsymbol{\Theta}^{\text{old}})}[z_{n-1,j} z_{nk}] \tag{30}$$

$$= p(z_{n-1,j} = 1, z_{nk} = 1 | \mathbf{X}, \boldsymbol{\Theta}^{\text{old}}) \tag{31}$$

$$= \xi(\mathbf{z}_{n-1}, \mathbf{z}_n : z_{n-1,j} = 1, z_{nk} = 1) \tag{32}$$

6. [**5 points**] Show that if any elements of the parameters π or $\mathbf{A}$ for a hidden Markov model are initially set to 0, then those elements will remain zero in all subsequent updates of the EM algorithm.

The update rules for π and $\mathbf{A}$:

$$\pi_k = \frac{\gamma(z_{1k})}{\sum_{j=1}^{K} \gamma(z_{1j})} \tag{33}$$

$$A_{jk} = \frac{\sum_{n=1}^{N} \xi(z_{n-1,j} z_{nk})}{\sum_{l=1}^{K} \sum_{n=2}^{N} \xi(z_{n-1,j} z_{nl})} \tag{34}$$

Considering the equation (25), if $A_{jk} = 0$ in the current iteration, $\xi(z_{n-1,j} z_{nk})$ is also zero (the term $p(\mathbf{z}_n|\mathbf{z}_{n-1})$ is zero). Therefore, all the future updates always set this to zero.

1.3 A coin game [15 points]

Two students X and Y from Cranberry Lemon University play a stochastic game with a fair coin. X and Y take turn with X going first. All the coin flips are recorded and the game finishes when a sequence of THT first appears. The player who last flips the coin is the winner. Two players can flip the coin many times as follows. At his turn, each time X flips the original coin, he also flips an extra biased coin ($p(H) = 0.3$.) He stops only if the extra coin lands head, otherwise he repeats flipping the original and extra coins, ... (The flips of this extra coin are not recorded.) On the other hand, at his turn, Y flips the coin until T appears (All of his flips are recorded).

You are given a sequence of recorded coin flips, you would like to infer the winner and as well as the flips of each player.

1. [**10 points**] Describe a HMM to model this game.

Figure 1. Note that for a valid HMM, the output of the current state is independent of the next state, given the current state ($\mathbf{x_n} \perp \mathbf{z}_{n+1}|\mathbf{z}_n$). Many answers violate this constraint.

2. [**5 points**]How would you use this HMM model to infer the (most probable) winner and the (most probable) flips of each player?

It is straightforward to use Viterbi's algorithm to infer the most probably values of hidden variables. From this sequence of values, we can infer the the (most probable) winner and the (most probable) flips of each player.

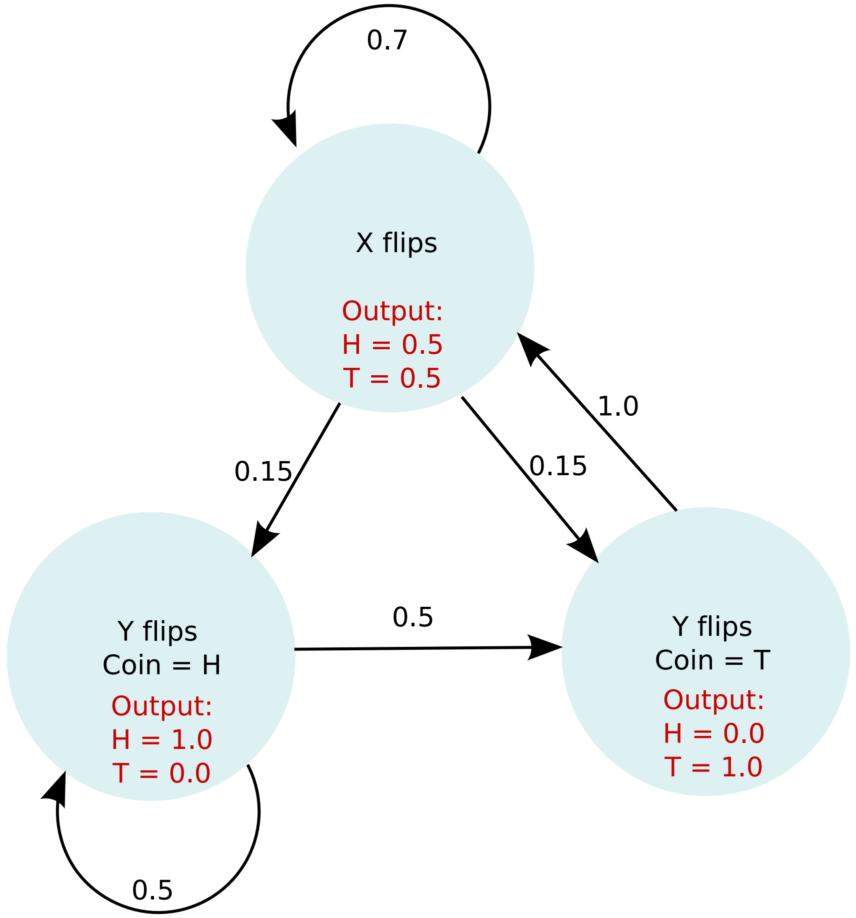

Figure 1: A HMM for the coin game

2 Dimensionality Reduction [20 points]

2.1 Singular value decomposition

In linear algebra, the singular value decomposition (SVD) is a factorization of a real matrix $\mathbf{X}$ as:

$$\mathbf{X} = \mathbf{U}\mathbf{S}\mathbf{V}^T \tag{35}$$

If the dimension of $\mathbf{X}$ is $m \times n$, where without loss of generality $m \geq n$, $\mathbf{U}$ is an $m \times n$ matrix, $\mathbf{S}$ is an $n \times n$ diagonal matrix and $\mathbf{V}^T$ is also an $n \times n$ matrix. Furthermore, $\mathbf{U}$ and $\mathbf{V}$ are orthonormal matrices: $\mathbf{U}\mathbf{U}^T = \mathbf{I}$ and $\mathbf{V}\mathbf{V}^T = \mathbf{I}$.

2.2 PCA and SVD

Consider a dataset of observations $\{\mathbf{x}_n\}$ where $n = 1, \ldots, N$. We assume that the examples are zero-centered such that $\bar{\mathbf{x}} = \sum_{n=1}^{N} \mathbf{x}_n = 0$. PCA algorithm computes the covariance matrix:

$$\mathbf{S} = \frac{1}{N} \sum_{n=1}^{N} \mathbf{x}_n \mathbf{x}_n^T \tag{36}$$

The principal components $\{\mathbf{u}_i\}$ are eigenvectors of $\mathbf{S}$.

Let $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_N]$, a $D \times N$ matrix where each column is one example $\mathbf{x}_n$. If $\mathbf{U}\mathbf{S}'\mathbf{V}^T$ is a SVD of $\mathbf{X}$,

1. [**10 points**] Show that the principal components $\{\mathbf{u}_i\}$ are columns of $\mathbf{U}$. This shows the relationship between PCA and SVD.

$$\mathbf{S} = \mathbf{X}\mathbf{X}^T \tag{37}$$

The principal components $\{\mathbf{u}_i\}$ are eigenvectors of $\mathbf{S}$, which are vectors such that: $\mathbf{S}\mathbf{u}_i = \lambda_i \mathbf{u}_i$.
$\mathbf{X} = \mathbf{U}\mathbf{S}'\mathbf{V}^T$ so,

$$\begin{aligned}
\mathbf{S} &= \mathbf{X}\mathbf{X}^T && (38)\\
&= \mathbf{U}\mathbf{S}'\mathbf{V}^T\mathbf{V}\mathbf{S}'\mathbf{U}^T && (39)\\
&= \mathbf{U}\mathbf{S}'^2\mathbf{U}^T && (40)
\end{aligned}$$

Therefore, the columns of $\mathbf{U}$ are eigenvectors of S.

2. [**3 points**]When the number of dimensions is much larger than the number of datapoints ($D \gg N$), is it better to do PCA by using the covariance matrix or using SVD?

Assume that the complexity of SVD is $O(ND\min(N,D))$ and the complexity of solving eigenvector problem is $O(D^3)$, we should use SVD.

3. [**7 points**] Consider the following data set:

$$\mathbf{D} = \begin{pmatrix} 1 & 1 & 1 \\ \epsilon & 0 & 0 \\ 0 & \epsilon & 0 \\ 0 & 0 & \epsilon \end{pmatrix} \tag{41}$$

where ϵ is a tiny number. Each column is one example. First zero-center the data set and then do PCA using two techniques: 1) by using the covariance matrix and 2) by using SVD. What do you observe?

First zero-center the data,

$$\bar{\mathbf{D}} = \begin{pmatrix} 0 & 0 & 0 \\ \frac{2\epsilon}{3} & -\frac{1\epsilon}{3} & -\frac{1\epsilon}{3} \\ -\frac{1\epsilon}{3} & \frac{2\epsilon}{3} & -\frac{1\epsilon}{3} \\ -\frac{1\epsilon}{3} & -\frac{1\epsilon}{3} & \frac{2\epsilon}{3} \end{pmatrix} \tag{42}$$

The covariance matrix is,

$$\mathbf{S} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & \frac{2\epsilon^2}{3} & -\frac{1\epsilon^2}{3} & -\frac{1\epsilon^2}{3} \\ 0 & -\frac{1\epsilon^2}{3} & \frac{2\epsilon^2}{3} & -\frac{1\epsilon^2}{3} \\ 0 & -\frac{1\epsilon^2}{3} & -\frac{1\epsilon^2}{3} & \frac{2\epsilon^2}{3} \end{pmatrix} \tag{43}$$

This matrix is singular so it does not have an eigen decompostion. In fact, when $\epsilon \to 0$, the condition number of this matrix, which is defined as the ratio betwen the maximum and the minimum eigenvalue $\frac{\lambda_{max}}{\lambda_{min}} \to 0$ (because $\lambda_{min} \to 0$), so the numerical methods to solve for eigenvalues are unstable. However, we still can do SVD of X.

3 Markov Decision Process [20 points]

1. [**10 points**] A standard MDP is described by a set of states S, a set of actions A, a transition function T, and a reward function R. Where $T(s,a,s')$ gives the probability of transitioning to s' after taking action a in state s, and $R(s)$ gives the immediate reward of being in state s. A k-order MDP is described in the same way with one exception. The transition function T depends on the current state s and also the previous

k-1 states. That is, $T(s_{k-1}, \ldots s_1, s, a, s') = p(s', a, s, s_1, \ldots s_{k-1})$ gives the probability of transitioning to state s' given that action a was taken in state s and the previous $k-1$ states were $(s_{k-1}, \ldots, s_1)$.

Given a k-order MDP $M = (S; A; T; R)$ describe how to construct a standard (first-order) MDP $M' = (S', A', T', R')$ that is equivalent to M. Here equivalent means that a solution to M' can be easily converted into a solution to M. Be sure to describe S', A', T', and R'. Give a brief justification your construction.

The state space for M' is $S' = S^k$, so that each state in M' is a k-tuple of states in S. That is, each state in S' is of the form $(s, s_1, \ldots, s_{k-1})$ where each component is a state in S. The actions of M' are the same as those of M, i.e. $A' = A$. Intuitively each state $(s, s_1, \ldots, s_{k-1})$ of M' encodes the state s at the current time and the previous $k-1$ states. This is all the information needed to determine the distribution over next states. The reward function of M' is defined as $R'((s, s_1, \ldots, s_{k-1})(= R(s)$, which means that the reward in the new MDP only depends on the current state s of M. Finally the transition function of M' is defined as:

$$T'(s, s_1, \ldots, s_{k-1}), a, \mathbf{s}) = Pr(s'|a, s, s_1, \ldots, s_{k-1}), \text{ if } \mathbf{s} = s, s_1, \ldots, s_{k-1}) \quad (44)$$

$$= 0 \text{ otherwise} \quad (45)$$

This definition of the transition function enforces that the history is maintained correctly after state transitions and that the new state s' has probability given by the k-th order model. In particular, there is zero transition probability of moving to a state that does not update the history correctly, which simply involves shifting the history in the current state by one step. It is easy to verify that there is a one-to-one correspondence between sequences of states in M and M' that have non-zero probability of being generated by some policy. Further, the probability of those sequences under any policy is equal.

2. [**10 points**] The Q-learning update rule for deterministic MDPs is as follows:

$$Q(s, a) \leftarrow R(s, a) + \gamma \max_{a'} Q(s', a') \quad (46)$$

where $s' = f(s, a)$ is the action to be taken.

Prove that Q-learning converges in deterministic MDPs.

This question is a giveaway. The proof is the theorem 13.1 from Mitchell's book.

Let

$$\Delta_n = \|\hat{Q}_n - Q\|_\infty = \max_{s,a} |\hat{Q}_n(s,a) - Q(s,a)| \tag{47}$$

We can bound the iterative update as:

$$\begin{aligned}
&|\hat{Q}_{n+1}(s_n, a_n) - Q(s_n, a_n)| && (48)\\
=&|R(s,a) + \gamma \max_{a'} \hat{Q}_n(s_{n+1}, a') - R(s,a) - \gamma \max_{a'} Q(s_{n+1}, a')| && (49)\\
=&\gamma | \max_{a'} \hat{Q}_n(s_{n+1}, a') - \max_{a'} Q(s_{n+1}, a')| && (50)\\
\leq&\gamma \max |\hat{Q}_n(s_{n+1}, a') - Q(s_{n+1}, a')| && (51)\\
\leq&\gamma \Delta_n && (52)
\end{aligned}$$

Consider now some interval $[n_1, n_2]$ over which all state-action pairs (s,a) appear at least once. Using the above relation and simple induction, it follows that $\Delta_{n_2} \leq \gamma\Delta_{n_1}$. Since $\gamma < 1$ and since there is an infinite number of such intervals by assumption, it follows that $\Delta_n \to 0$.