CSCI567 Machine Learning (Spring 2021)

Sirisha Rambhatla

University of Southern California

March 19, 2021

Outline

1. Review of last lecture
2. Gaussian mixture models

Outline

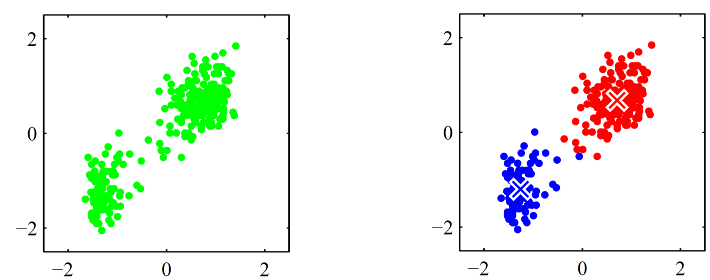
1. Review of last lecture
2. Gaussian mixture models

Clustering: formal definition

Given: data points $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N \in \mathbb{R}^{\mathsf{D}}$ and $\#$clusters K we want

Output: group the data into K clusters, which means

- find assignment $\gamma_{nk} \in \{0, 1\}$ for each data point $n \in [N]$ and $k \in [K]$ s.t. $\sum_{k \in [K]} \gamma_{nk} = 1$ for any fixed n
- find the cluster centers $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K \in \mathbb{R}^{\mathsf{D}}$



Alternating minimization

Instead, use a heuristic that alternatingly minimizes over $\{\gamma_{nk}\}$ and $\{\boldsymbol{\mu}_k\}$:

Initialize $\{\boldsymbol{\mu}_k^{(1)}\}$

For $t = 1, 2, \ldots$

- find
$$\{\gamma_{nk}^{(t+1)}\} = \operatorname*{argmin}_{\{\gamma_{nk}\}} F\left(\{\gamma_{nk}\}, \{\boldsymbol{\mu}_k^{(t)}\}\right)$$
- find
$$\{\boldsymbol{\mu}_k^{(t+1)}\} = \operatorname*{argmin}_{\{\boldsymbol{\mu}_k\}} F\left(\{\gamma_{nk}^{(t+1)}\}, \{\boldsymbol{\mu}_k\}\right)$$

The K-means algorithm

Step 0 Initialize $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K$

Step 1 Fix the centers $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K$, assign each point to the closest center:
$$\gamma_{nk} = \mathbb{I}\left[k == \operatorname*{argmin}_c \|\boldsymbol{x}_n - \boldsymbol{\mu}_c\|_2^2\right]$$

Step 2 Fix the assignment $\{\gamma_{nk}\}$, update the centers
$$\boldsymbol{\mu}_k = \frac{\sum_n \gamma_{nk}\boldsymbol{x}_n}{\sum_n \gamma_{nk}}$$

Step 3 Return to Step 1 if not converged

Outline

Gaussian mixture models

Gaussian mixture models (GMMs) are a probabilistic approach for clustering

- more explanatory than minimizing the K-means objective
- can be seen as a soft version of K-means

To solve GMM, we will introduce a powerful method for learning probabilistic model: **Expectation–Maximization (EM) algorithm**

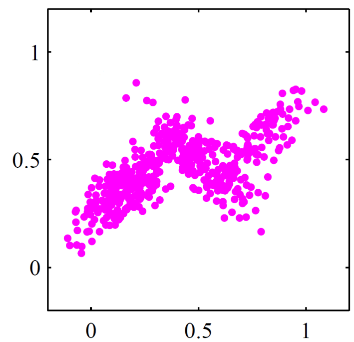
A generative model

For classification, we discussed the sigmoid model to "explain" how the labels are generated.

Similarly, for clustering, we want to come up with a probabilistic model p to **"explain" how the data is generated**.

That is, each point is an independent sample of $\boldsymbol{x} \sim p$.

What probabilistic model generates data like this?

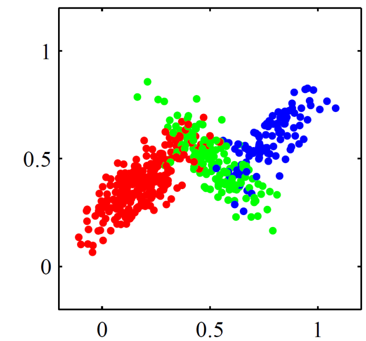


GMM: intuition

GMM is a natural model to explain such data

Assume there are 3 ground-truth Gaussian models. To generate a point, we

- first **randomly pick one of the Gaussian models**,
- then **draw a point according this Gaussian**.



Hence the name "**Gaussian mixture model**".

GMM: formal definition

A GMM has the following density function:

$$p(\boldsymbol{x}) = \sum_{k=1}^{K} \omega_k N(\boldsymbol{x} \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

where

- K: the number of Gaussian components (same as #clusters we want)
- $\omega_1, \ldots, \omega_K$: mixture weights, a distribution over K components
- $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$: mean and covariance matrix of the k-th Gaussian
- N: the density function for a Gaussian

Another view

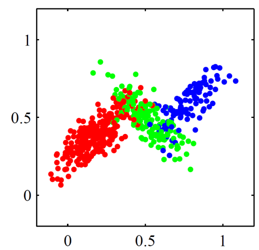
By introducing a **latent variable** $z \in [K]$, which indicates cluster membership, we can see p as a **marginal distribution**

$$p(\boldsymbol{x}) = \sum_{k=1}^{K} p(\boldsymbol{x}, z = k) = \sum_{k=1}^{K} p(z = k) p(\boldsymbol{x} | z = k) = \sum_{k=1}^{K} \omega_k N(\boldsymbol{x} \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

$\boldsymbol{x}$ and z are both random variables drawn from the model

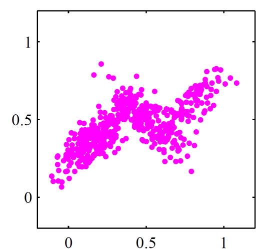
- $\boldsymbol{x}$ is observed
- z is unobserved/latent

An example



The conditional distributions are

$$p(\boldsymbol{x} \mid z = \text{red}) = N(\boldsymbol{x} \mid \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)$$
$$p(\boldsymbol{x} \mid z = \text{blue}) = N(\boldsymbol{x} \mid \boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)$$
$$p(\boldsymbol{x} \mid z = \text{green}) = N(\boldsymbol{x} \mid \boldsymbol{\mu}_3, \boldsymbol{\Sigma}_3)$$



The marginal distribution is

$$p(\boldsymbol{x}) = p(\text{red})N(\boldsymbol{x} \mid \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1) + p(\text{blue})N(\boldsymbol{x} \mid \boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2) + p(\text{green})N(\boldsymbol{x} \mid \boldsymbol{\mu}_3, \boldsymbol{\Sigma}_3)$$

Learning GMMs

Learning a GMM means finding all the parameters $\boldsymbol{\theta} = \{\omega_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1}^K$.

In the process, we will learn the latent variable z_n as well:

$$p(z_n = k \mid \boldsymbol{x}_n) \triangleq \gamma_{nk} \in [0, 1]$$

i.e. "soft assignment" of each point to each cluster, as opposed to "hard assignment" by K-means.

GMM is more explanatory than K-means

- both learn the cluster centers $\boldsymbol{\mu}_k$'s
- in addition, GMM learns cluster weight ω_k and covariance $\boldsymbol{\Sigma}_k$, thus
  - we can *predict probability of seeing a new point*
  - we can *generate synthetic data*

How to learn these parameters?

An obvious attempt is **maximum-likelihood estimation (MLE)**: find

$$\operatorname*{argmax}_{\boldsymbol{\theta}} \ln \prod_{n=1}^{N} p(\boldsymbol{x}_n ; \boldsymbol{\theta}) = \operatorname*{argmax}_{\boldsymbol{\theta}} \sum_{n=1}^{N} \ln p(\boldsymbol{x}_n ; \boldsymbol{\theta}) \triangleq \operatorname*{argmax}_{\boldsymbol{\theta}} P(\boldsymbol{\theta})$$

This is called incomplete log-likelihood (since z_n's are unobserved), and is *intractable in general* (non-concave problem).

One solution is to still apply GD/SGD, but a much more effective approach is the **Expectation–Maximization (EM) algorithm**.

Preview of EM for learning GMMs

Step 0 Initialize $\omega_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k$ for each $k \in [K]$

Step 1 (E-Step) **update the "soft assignment"** (fixing parameters)

$$\gamma_{nk} = p(z_n = k \mid \boldsymbol{x}_n) \propto \omega_k N(\boldsymbol{x}_n \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

Step 2 (M-Step) **update the model parameter** (fixing assignments)

$$\omega_k = \frac{\sum_n \gamma_{nk}}{N} \qquad \boldsymbol{\mu}_k = \frac{\sum_n \gamma_{nk} \boldsymbol{x}_n}{\sum_n \gamma_{nk}}$$

$$\boldsymbol{\Sigma}_k = \frac{1}{\sum_n \gamma_{nk}} \sum_n \gamma_{nk} (\boldsymbol{x}_n - \boldsymbol{\mu}_k)(\boldsymbol{x}_n - \boldsymbol{\mu}_k)^{\mathrm{T}}$$

Step 3 return to Step 1 if not converged

We will see how this is a special case of EM.

Demo

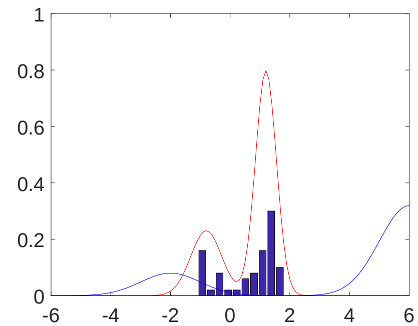
Generate 50 data points from a mixture of 2 Gaussians with

- $\omega_1 = 0.3, \mu_1 = -0.8, \Sigma_1 = 0.52$
- $\omega_2 = 0.7, \mu_2 = 1.2, \Sigma_2 = 0.35$

histogram represents the data

red curve represents the ground-truth density $p(\boldsymbol{x}) = \sum_{k=1}^{K} \omega_k N(\boldsymbol{x} \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$

blue curve represents the learned density for a specific round



EM_demo.pdf shows how the blue curve moves towards red curve quickly via EM

EM algorithm

In general EM is **a heuristic to solve MLE with latent variables** (not just GMM), i.e. find the maximizer of

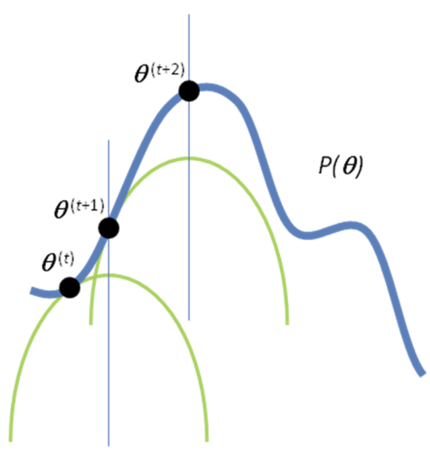
$$P(\boldsymbol{\theta}) = \sum_{n=1}^{N} \ln p(\boldsymbol{x}_n \,; \boldsymbol{\theta}) = \sum_{n=1}^{N} \ln \int_{z_n} p(\boldsymbol{x}_n, z_n \,; \boldsymbol{\theta}) dz_n$$

- $\boldsymbol{\theta}$ is the parameters for a general probabilistic model
- $\boldsymbol{x}_n$'s are observed random variables
- z_n's are latent variables

Again, directly solving the objective is intractable.

High level idea

Keep maximizing **a lower bound of P that is more manageable**



Derivation of EM

Finding the lower bound of P:

$$\begin{aligned}
\ln p(\boldsymbol{x} \,; \boldsymbol{\theta}) &= \ln \int_z p(\boldsymbol{x}, z \,; \boldsymbol{\theta}) dz \\
&= \ln \int_z q(z) \frac{p(\boldsymbol{x}, z \,; \boldsymbol{\theta})}{q(z)} dz && \text{(true for any dist. } q\text{)} \\
&= \ln \mathbb{E}_{z\sim q}\left[\frac{p(\boldsymbol{x}, z \,; \boldsymbol{\theta})}{q(z)}\right] \\
&\geq \mathbb{E}_{z\sim q}\left[\ln \frac{p(\boldsymbol{x}, z \,; \boldsymbol{\theta})}{q(z)}\right] && \text{(Jensen's inequality)} \\
&= \mathbb{E}_{z\sim q}\left[\ln p(\boldsymbol{x}, z \,; \boldsymbol{\theta})\right] + H(q)
\end{aligned}$$

where, $H(q) = -\mathbb{E}_{z\sim q}[\ln q(z)]$ is the Entropy. Therefore, for an observation $\boldsymbol{x}$ we have

$$\ln p(\boldsymbol{x} \,; \boldsymbol{\theta}) \geq \mathbb{E}_{z\sim q}\left[\ln p(\boldsymbol{x}, z \,; \boldsymbol{\theta})\right] + H(q)$$

Alternatively maximize the lower bound

Therefore, we obtain a lower bound for the log-likelihood function

$$\begin{aligned} P(\boldsymbol{\theta}) &= \sum_{n=1}^{N} \ln p(\boldsymbol{x}_n\,; \boldsymbol{\theta}) \\ &\geq \sum_{n=1}^{N} \left(\mathbb{E}_{z_n \sim q_n} \left[\ln p(\boldsymbol{x}_n, z_n\,; \boldsymbol{\theta}) \right] + H(q_n) \right) = F(\boldsymbol{\theta}, \{q_n\}) \end{aligned}$$

This holds for *any* $\{q_n\}$, so how do we choose? Naturally, *the one that maximizes the lower bound* (i.e. the tightest lower bound)!

Equivalently, this is the same as alternatingly maximizing F over $\{q_n\}$ and $\boldsymbol{\theta}$ (similar to K-means).

Maximizing over $\{q_n\}$

Fix $\boldsymbol{\theta}^{(t)}$, the solution to

$$\operatorname*{argmax}_{q_n} \mathbb{E}_{z_n \sim q_n} \left[\ln p(\boldsymbol{x}_n, z_n\,; \boldsymbol{\theta}^{(t)}) \right] + H(q_n)$$

is $q_n^{(t)}$ s.t.

$$q_n^{(t)}(z_n) = p(z_n \mid \boldsymbol{x}_n\,; \boldsymbol{\theta}^{(t)}) \propto p(\boldsymbol{x}_n, z_n\,; \boldsymbol{\theta}^{(t)})$$

i.e., the *posterior distribution of z_n* given $\boldsymbol{x}_n$ and $\boldsymbol{\theta}^{(t)}$. (See MLaPP 11.4.7)

So at $\boldsymbol{\theta}^{(t)}$, we found the tightest lower bound $F\left(\boldsymbol{\theta}, \{q_n^{(t)}\}\right)$:

- $F\left(\boldsymbol{\theta}, \{q_n^{(t)}\}\right) \leq P(\boldsymbol{\theta})$ for all $\boldsymbol{\theta}$.
- $F\left(\boldsymbol{\theta}^{(t)}, \{q_n^{(t)}\}\right) = P(\boldsymbol{\theta}^{(t)})$ (verify using Slide 20 and MLaPP 11.4.7)

Maximizing over $\boldsymbol{\theta}$

Fix $\{q_n^{(t)}\}$, maximize over $\boldsymbol{\theta}$:

$$\begin{aligned} &\operatorname*{argmax}_{\boldsymbol{\theta}} F\left(\boldsymbol{\theta}, \{q_n^{(t)}\}\right) \\ &= \operatorname*{argmax}_{\boldsymbol{\theta}} \sum_{n=1}^{N} \mathbb{E}_{z_n \sim q_n^{(t)}} \left[\ln p(\boldsymbol{x}_n, z_n\,; \boldsymbol{\theta}) \right] \qquad (H(q_n^{(t)}) \text{ is independent of } \boldsymbol{\theta}) \\ &\triangleq \operatorname*{argmax}_{\boldsymbol{\theta}}\ Q(\boldsymbol{\theta}\,; \boldsymbol{\theta}^{(t)}) \qquad (\{q_n^{(t)}\} \text{ are computed via } \boldsymbol{\theta}^{(t)}) \end{aligned}$$

Q is the (expected) **complete likelihood** and is usually more tractable.

General EM algorithm

Step 0 Initialize $\boldsymbol{\theta}^{(1)}$, $t = 1$

Step 1 (E-Step) **update the posterior of latent variables**

$$q_n^{(t)}(\cdot) = p(\cdot \mid \boldsymbol{x}_n\,; \boldsymbol{\theta}^{(t)})$$

and obtain **Expectation** of complete likelihood

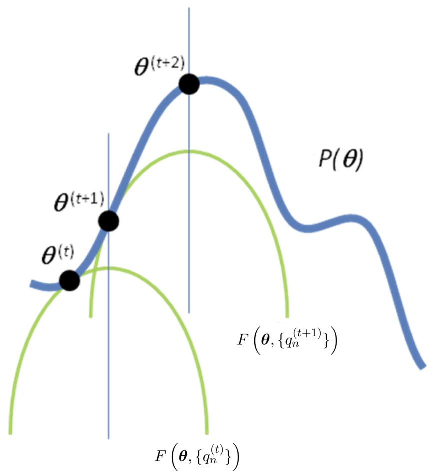
$$Q(\boldsymbol{\theta}\,; \boldsymbol{\theta}^{(t)}) = \sum_{n=1}^{N} \mathbb{E}_{z_n \sim q_n^{(t)}} \left[\ln p(\boldsymbol{x}_n, z_n\,; \boldsymbol{\theta}) \right]$$

Step 2 (M-Step) **update the model parameter** via **Maximization**

$$\boldsymbol{\theta}^{(t+1)} \leftarrow \operatorname*{argmax}_{\boldsymbol{\theta}} Q(\boldsymbol{\theta}\,; \boldsymbol{\theta}^{(t)})$$

Step 3 $t \leftarrow t + 1$ and return to Step 1 if not converged

Pictorial explanation



$P(\boldsymbol{\theta})$ is non-concave, but $Q(\boldsymbol{\theta}; \boldsymbol{\theta}^{(t)})$ often is concave and easy to maximize.

$$\begin{aligned} P(\boldsymbol{\theta}^{(t+1)}) &\geq F\left(\boldsymbol{\theta}^{(t+1)} ; \{q_n^{(t)}\}\right) \\ &\geq F\left(\boldsymbol{\theta}^{(t)} ; \{q_n^{(t)}\}\right) \\ &= P(\boldsymbol{\theta}^{(t)}) \end{aligned}$$

So EM always increases the objective value and will converge to some local maximum (similar to K-means).

Apply EM to learn GMMs

E-Step:

$$\begin{aligned} q_n^{(t)}(z_n = k) &= p\left(z_n = k \mid \boldsymbol{x}_n ; \boldsymbol{\theta}^{(t)}\right) \\ &\propto p\left(\boldsymbol{x}_n, z_n = k ; \boldsymbol{\theta}^{(t)}\right) \\ &= p\left(z_n = k ; \boldsymbol{\theta}^{(t)}\right) p(\boldsymbol{x}_n \mid z_n = k ; \boldsymbol{\theta}^{(t)}) \\ &= \omega_k^{(t)} N\left(\boldsymbol{x}_n \mid \boldsymbol{\mu}_k^{(t)}, \boldsymbol{\Sigma}_k^{(t)}\right) \end{aligned}$$

This computes the "soft assignment" $\gamma_{nk} = q_n^{(t)}(z_n = k)$, i.e. conditional probability of $\boldsymbol{x}_n$ belonging to cluster k.

Apply EM to learn GMMs

M-Step:

$$\begin{aligned} \operatorname*{argmax}_{\boldsymbol{\theta}} Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(t)}) &= \operatorname*{argmax}_{\boldsymbol{\theta}} \sum_{n=1}^{N} \mathbb{E}_{z_n \sim q_n^{(t)}} \left[\ln p(\boldsymbol{x}_n, z_n ; \boldsymbol{\theta})\right] \\ &= \operatorname*{argmax}_{\boldsymbol{\theta}} \sum_{n=1}^{N} \mathbb{E}_{z_n \sim q_n^{(t)}} \left[\ln p(z_n ; \boldsymbol{\theta}) + \ln p(\boldsymbol{x}_n | z_n ; \boldsymbol{\theta})\right] \\ &= \operatorname*{argmax}_{\{\omega_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}} \sum_{n=1}^{N} \sum_{k=1}^{K} \gamma_{nk} \left(\ln \omega_k + \ln N(\boldsymbol{x}_n \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)\right) \end{aligned}$$

To find $\omega_1, \ldots, \omega_K$, solve

$$\operatorname*{argmax}_{\boldsymbol{\omega}} \sum_{n=1}^{N} \sum_{k=1}^{K} \gamma_{nk} \ln \omega_k$$

To find each $\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k$, solve

$$\operatorname*{argmax}_{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k} \sum_{n=1}^{N} \gamma_{nk} \ln N(\boldsymbol{x}_n \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

M-Step (continued)

Solutions to previous two problems are very natural, for each k

$$\omega_k = \frac{\sum_n \gamma_{nk}}{N}$$

i.e. (weighted) fraction of examples belonging to cluster k

$$\boldsymbol{\mu}_k = \frac{\sum_n \gamma_{nk} \boldsymbol{x}_n}{\sum_n \gamma_{nk}}$$

i.e. (weighted) average of examples belonging to cluster k

$$\boldsymbol{\Sigma}_k = \frac{1}{\sum_n \gamma_{nk}} \sum_n \gamma_{nk} (\boldsymbol{x}_n - \boldsymbol{\mu}_k)(\boldsymbol{x}_n - \boldsymbol{\mu}_k)^{\mathrm{T}}$$

i.e (weighted) covariance of examples belonging to cluster k

Putting it together

EM for learning GMMs:

Step 0 Initialize $\omega_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k$ for each $k \in [K]$

Step 1 (E-Step) **update the "soft assignment"** (fixing parameters)

$$\gamma_{nk} = p(z_n = k \mid \boldsymbol{x}_n) \propto \omega_k N(\boldsymbol{x}_n \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

Step 2 (M-Step) **update the model parameter** (fixing assignments)

$$\omega_k = \frac{\sum_n \gamma_{nk}}{N} \qquad \boldsymbol{\mu}_k = \frac{\sum_n \gamma_{nk} \boldsymbol{x}_n}{\sum_n \gamma_{nk}}$$

$$\boldsymbol{\Sigma}_k = \frac{1}{\sum_n \gamma_{nk}} \sum_n \gamma_{nk} (\boldsymbol{x}_n - \boldsymbol{\mu}_k)(\boldsymbol{x}_n - \boldsymbol{\mu}_k)^{\mathrm{T}}$$

Step 3 return to Step 1 if not converged

Connection to K-means

K-means is in fact a special case of EM for (a simplified) GMM:

- assume $\boldsymbol{\Sigma}_k = \sigma^2 \boldsymbol{I}$ for some fixed σ so only ω_k and $\boldsymbol{\mu}_k$ are parameters
- when $\sigma \to 0$, EM becomes K-means

GMM is a soft version of K-means and it provides a probabilistic interpretation of the data, which means we can predict and generate data after learning.