Course: Machine Learning by Dr. Seyyed Salehi

Homework: HW5

Name: Mohammad Mohammadi

Student ID: 402208592

Question 1

Consider the following three data points in a 2-dimensional space.

(1,1),(0,0),(−1,−1)

a) Find the first principal component (write down the vector).

b) If we project the data onto a one-dimensional space using the first principal component, what are the projected data points? Calculate the variance of the projected data.

c) If we reconstruct the data from the one-dimensional space back to the two-dimensional space, what is the reconstruction error?

Answer

Part (a)

Compute the Mean Vector:

$$\mu = \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i = \frac{1}{3}\begin{pmatrix} 1+0+(-1) \\ 1+0+(-1) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

Compute the Covariance Matrix

$$\mathbf{X} = \begin{pmatrix} 1 & 1 \\ 0 & 0 \\ -1 & -1 \end{pmatrix}$$

$$\mathbf{\Sigma} = \frac{1}{n-1}\mathbf{X}^T\mathbf{X} = \frac{1}{2}\begin{pmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \end{pmatrix}\begin{pmatrix} 1 & 1 \\ 0 & 0 \\ -1 & -1 \end{pmatrix} = \frac{1}{2}\begin{pmatrix} 2 & 2 \\ 2 & 2 \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$$

Compute the Eigenvalues and Eigenvectors

$$\det(\mathbf{\Sigma} - \lambda\mathbf{I}) = \begin{vmatrix} 1-\lambda & 1 \\ 1 & 1-\lambda \end{vmatrix} = (1-\lambda)^2 - 1 = \lambda^2 - 2\lambda = 0$$

$$\lambda_1 = 2, \quad \lambda_2 = 0$$

For $\lambda_1 = 2$:

$$(\mathbf{\Sigma} - 2\mathbf{I})\,\mathbf{v} = 0 \Rightarrow \begin{pmatrix} -1 & 1 \\ 1 & -1 \end{pmatrix}\begin{pmatrix} v_1 \\ v_2 \end{pmatrix} = 0$$

Solving the above equation, we get:

$$v_1 = v_2$$

So, the eigenvector corresponding to $\lambda_1 = 2$ is:

$$\mathbf{v}_1 = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

PArt (b)

To project the data onto the first principal component:

$$\mathbf{y} = \mathbf{X}\mathbf{v}_1 = \begin{pmatrix} 1 & 1 \\ 0 & 0 \\ -1 & -1 \end{pmatrix}\frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ 1 \end{pmatrix} = \frac{1}{\sqrt{2}}\begin{pmatrix} 2 \\ 0 \\ -2 \end{pmatrix} = \begin{pmatrix} \sqrt{2} \\ 0 \\ -\sqrt{2} \end{pmatrix}$$

The variance of the projected data is:

$$\mathrm{Var}(\mathbf{y}) = \frac{1}{n}\sum_{i=1}^{n} y_i^2 = \frac{1}{3}\left((\sqrt{2})^2 + 0^2 + (-\sqrt{2})^2\right) = \frac{1}{3}(2+0+2) = \frac{4}{3}$$

Part (c)

To reconstruct the data from the one-dimensional space back to the two-dimensional space:

$$\mathbf{X}_{\text{reconstructed}} = \mathbf{y}\mathbf{v}_1^T = \begin{pmatrix} \sqrt{2} \\ 0 \\ -\sqrt{2} \end{pmatrix} \left(\frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \end{pmatrix} \right) = \begin{pmatrix} 1 & 1 \\ 0 & 0 \\ -1 & -1 \end{pmatrix}$$

The reconstruction error is the sum of the squared differences between the original and reconstructed data points:

$$\text{Reconstruction Error} = \sum_{i=1}^{n} \|\mathbf{x}_i - \mathbf{x}_{\text{reconstructed},i}\|^2 = \sum_{i=1}^{n} \|\mathbf{x}_i - \mathbf{x}_i\|^2 = 0$$

Thus, the reconstruction error is zero.

Question 2

For the data points on the left side of Figure 1, draw the direction of the first principal component (PCA) without considering the labels of the data points (PCA does not consider data labels).

For the data points on the right side of Figure 1, draw the direction of the Fisher Linear Discriminant (FLD) for reducing the linear error (considering circle points as the positive class and square points as the negative class).

Answer

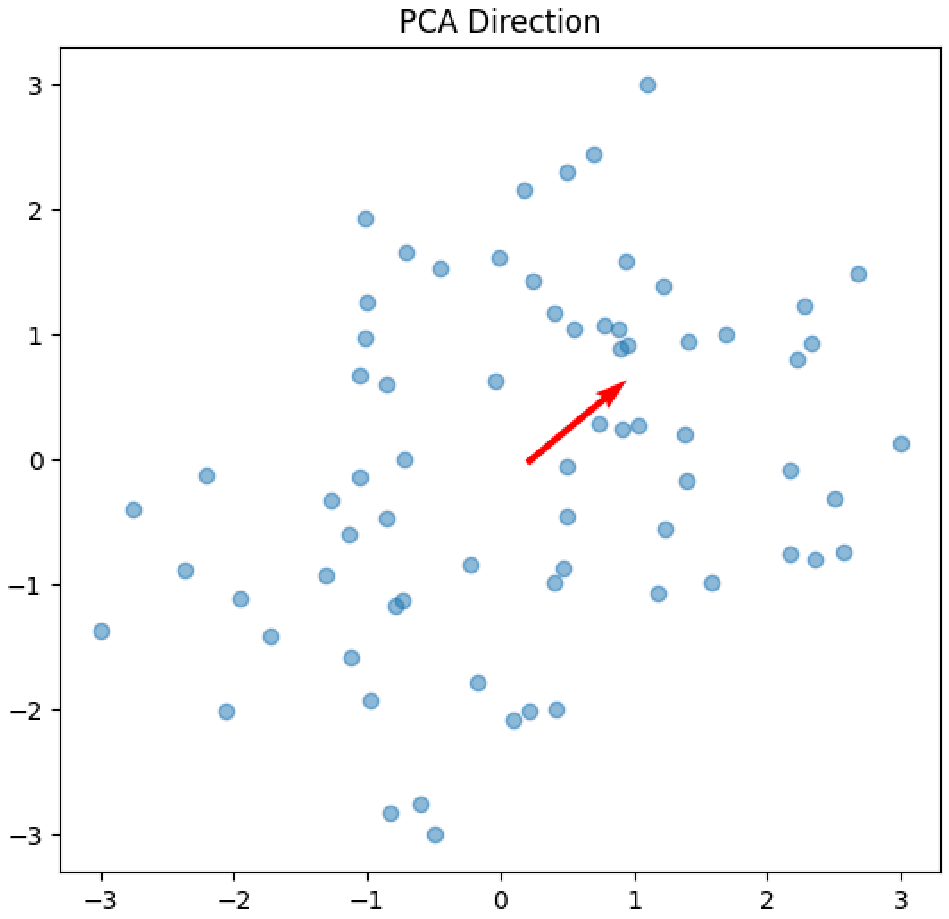
I used the tool in the link below in order to replicate the data in the plot of the assignment for the sake of good visualization.

https://drawdata.xyz/

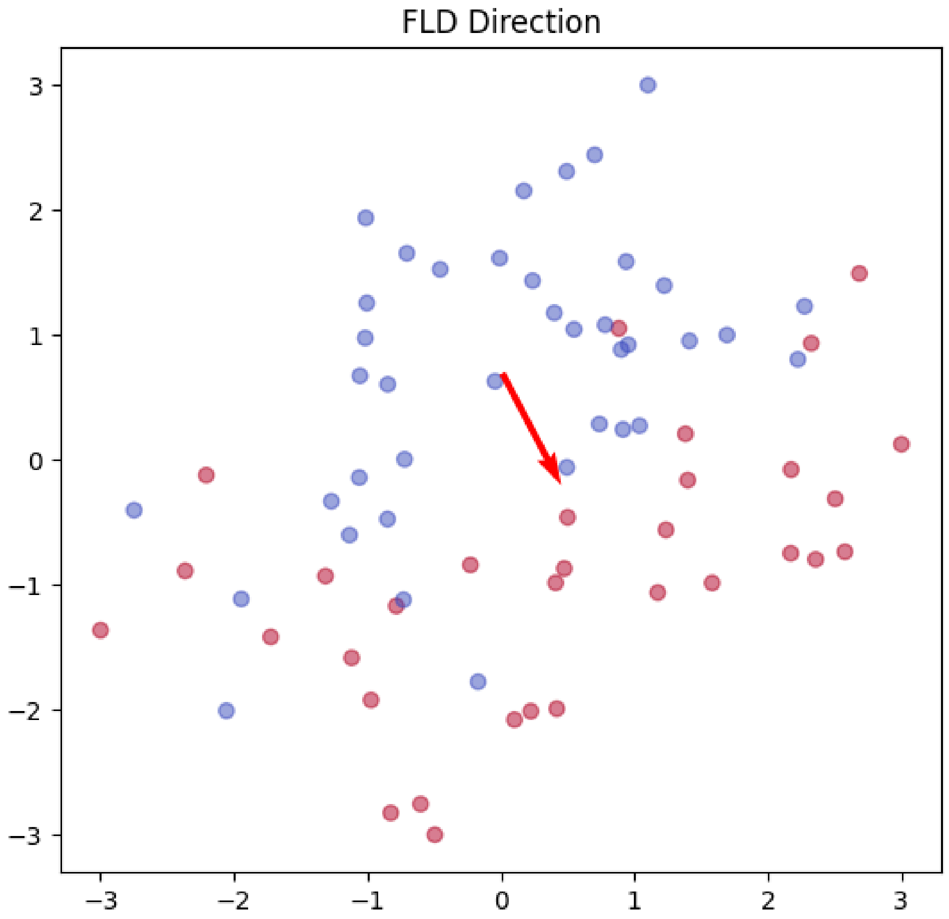
In [ ]:

```
import numpy as np
import matplotlib.pyplot as plt

points_json = [{"x":-35.56583350156126,"y":31.59222003891938,"color":"b"},{"x":-9.196719754251372,"y":72.34559066483416,"color"

points = np.array([[point["x"], point["y"]] for point in points_json])
labels = np.array([0 if point["color"] == "a" else 1 for point in points_json])

# Normalize points between -3 and +3 as the assignment plots
min_val = np.min(points, axis=0)
max_val = np.max(points, axis=0)
```

```python
def normalize(points, min_val, max_val):
    return 6 * (points - min_val) / (max_val - min_val) - 3

normalized_points = normalize(points, min_val, max_val)

left_data = normalized_points
right_data = normalized_points

mean_vec = np.mean(left_data, axis=0)
cov_mat = np.cov(left_data.T)
eigenvalues, eigenvectors = np.linalg.eig(cov_mat)
pca_direction = eigenvectors[:, np.argmax(eigenvalues)]

class_0 = right_data[labels == 0]
class_1 = right_data[labels == 1]
mean_0 = np.mean(class_0, axis=0)
mean_1 = np.mean(class_1, axis=0)
within_class_scatter = np.cov(class_0.T) + np.cov(class_1.T)
between_class_scatter = np.outer((mean_1 - mean_0), (mean_1 - mean_0))
eigenvalues_fld, eigenvectors_fld = np.linalg.eig(np.linalg.inv(within_class_scatter).dot(between_class_scatter))
fld_direction = eigenvectors_fld[:, np.argmax(eigenvalues_fld)]

fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(14, 6))

ax1.scatter(left_data[:, 0], left_data[:, 1], alpha=0.5)
ax1.quiver(mean_vec[0], mean_vec[1], pca_direction[0], pca_direction[1], angles='xy', scale_units='xy', scale=1, color='r')
ax1.set_title('PCA Direction')

ax2.scatter(right_data[:, 0], right_data[:, 1], c=labels, cmap='coolwarm', alpha=0.5)
ax2.quiver(mean_0[0], mean_0[1], fld_direction[0], fld_direction[1], angles='xy', scale_units='xy', scale=1, color='r')
ax2.set_title('FLD Direction')

plt.show()
```



PCA Direction



FLD Direction

Question 3

In this problem, we want to examine the K-means clustering algorithm. Assume $X = \{x_1, x_2, \ldots, x_n\}$ are the data points and γ is an indicator matrix such that $\gamma_{ij} = 1$ if x_i belongs to cluster j, and 0 otherwise. Suppose the cluster means are $\mu_1, \mu_2, \ldots, \mu_k$. The objective function for the data is calculated as follows:

$$J(\gamma, \mu_1, \ldots, \mu_k) = n \sum_{j=1}^{k} \gamma_{ij} \|x_i - \mu_j\|^2$$

Also, consider $C = 1, \ldots, k$ as your set of clusters.

Based on the K-means algorithm that was taught in class, answer the following questions.

a) Show that the algorithm terminates after a finite number of steps. (How many different values can γ take?)

b) Suppose $\hat{x}$ is the sample mean of the data points. Consider the following values.

$$T(X) = \frac{1}{n}\sum_{i=1}^{n} \|x_i - \hat{x}\|^2$$

$$W_j(X) = \frac{\sum_{i=1}^{n} \gamma_{ij}\|x_i - \mu_j\|^2}{\sum_{i=1}^{n} \gamma_{ij}}$$

$$B(X) = \sum_{j=1}^{k} \frac{\sum_{i=1}^{n} \gamma_{ij}}{n}\|\mu_j - \hat{x}\|^2$$

Here, $T(X)$ represents the total variance, $W_j(X)$ represents the within-cluster variance, and $B(X)$ represents the between-cluster variance.

Show the relationship between these three values.

Show that K-means can be interpreted as a method for minimizing the weighted sum of within-cluster variances and approximately maximizing the between-cluster variances.

c) Show that the minimum of J is a non-increasing function with respect to k, the number of clusters. Therefore, why does choosing the number of clusters by minimizing J does not work well?

Answer

Part (a)

The K-means algorithm terminates after a finite number of steps because the objective function $J(\gamma, \mu_1, \ldots, \mu_k)$ either decreases or remains the same in each iteration, and there are only a finite number of possible cluster assignments.

Finite Number of Possible Values for γ

For n data points and k clusters, each data point can be assigned to one of the k clusters. Therefore, there are k^n possible ways to assign the n data points to the k clusters.

Objective Function $J(\gamma, \mu_1, \ldots, \mu_k)$

The K-means algorithm aims to minimize the objective function:

$$J(\gamma, \mu_1, \dots, \mu_k) = n \sum_{j=1}^{k} \gamma_{ij} \|x_i - \mu_j\|^2$$

In each iteration, the assignments and the cluster means are updated in such a way that the objective function does not increase. Since there are a finite number of possible assignments and the objective function decreases or remains constant in each iteration, the algorithm must eventually reach a point where no further decrease is possible. At this point, the algorithm terminates.

Part (b)

Total Variance $T(X)$

$$T(X) = \frac{1}{n} \sum_{i=1}^{n} \|x_i - \hat{x}\|^2$$

where $\hat{x}$ is the mean of all data points:

$$\hat{x} = \frac{1}{n} \sum_{i=1}^{n} x_i$$

Within-Cluster Variance $W_j(X)$

$$W_j(X) = \frac{\sum_{i=1}^{n} \gamma_{ij} \|x_i - \mu_j\|^2}{\sum_{i=1}^{n} \gamma_{ij}}$$

where μ_j is the mean of the data points in cluster j:

$$\mu_j = \frac{\sum_{i=1}^{n} \gamma_{ij} x_i}{\sum_{i=1}^{n} \gamma_{ij}}$$

Between-Cluster Variance $B(X)$

$$B(X) = \sum_{j=1}^{k} \frac{\sum_{i=1}^{n} \gamma_{ij}}{n} \|\mu_j - \hat{x}\|^2$$

Relationship Between $T(X)$, $W_j(X)$, and $B(X)$

The total variance $T(X)$ can be decomposed into the sum of the within-cluster variances and the between-cluster variance:

$$T(X) = \frac{1}{n}\sum_{i=1}^{n}\|x_i - \hat{x}\|^2 = \sum_{j=1}^{k}\frac{\sum_{i=1}^{n}\gamma_{ij}}{n}W_j(X) + B(X)$$

Hence, this relationship shows that the total variance is the sum of the weighted within-cluster variances and the between-cluster variance.

K-means Interpretation

K-means aims to minimize the objective function J, which is the sum of the within-cluster variances:

$$J(\gamma, \mu_1, \ldots, \mu_k) = \sum_{j=1}^{k}\sum_{i=1}^{n}\gamma_{ij}\|x_i - \mu_j\|^2 = \sum_{j=1}^{k}W_j(X)\sum_{i=1}^{n}\gamma_{ij}$$

By minimizing the within-cluster variances, K-means indirectly maximizes the between-cluster variance $B(X)$.

Part (c)

Objective Function J

The objective function $J(\gamma, \mu_1, \ldots, \mu_k)$ is minimized for a given number of clusters k. As k increases, the value of J generally decreases because data points are assigned to more clusters, reducing the within-cluster variance.

Non-Increasing Function

The minimum of J is a non-increasing function of k. This is because as the number of clusters increases, each data point can be assigned to its own cluster, eventually leading to a situation where J is zero when $k = n$.

Choosing the Number of Clusters

Minimizing J does not work well for choosing the number of clusters because it would always suggest using as many clusters as there are data points (i.e., $k = n$), leading to overfitting. Instead, methods like the "elbow method," silhouette score, or cross-validation should be used to find an optimal balance between the number of clusters and the objective function.

Question 4

Your friend has two coins: one red coin and one blue coin, with biases p_r and p_b, respectively (the red coin has probability p_r and the blue coin has probability p_b of landing heads). He also has a preference π for selecting the red coin. He does m flips. Each time, he flips a coin, he first selects the red coin with probability π or the blue coin with probability $1 - \pi$. This process is independent for each flip, and we observe only the outcomes (heads or tails) without knowing which coin was selected. For each flip i, a random variable X_i is defined as follows:

$$X_i = \begin{cases} 1 & \text{if the } i\text{-th flip is head,} \\ 0 & \text{otherwise.} \end{cases}$$

Given the observations for $X = \{x_1, \ldots, x_m\}$ as a sequence of random variables, we want to estimate $\theta = (\pi, p_r, p_b)$. To assist with this, define an additional unobserved random variable Z_i for each flip as follows:

$$Z_i = \begin{cases} 1 & \text{if the red coin was used for the } i\text{-th flip,} \\ 0 & \text{otherwise.} \end{cases}$$

(a) Write down an expression for the joint probability distribution of X and Z given θ:

$$p(x, z; \theta) = \ldots$$

(b) Write down the complete-data log-likelihood $\mathcal{L}_c(\theta)$:

$$\mathcal{L}_c(\theta) = \sum_{i=1}^{m} \ln p(x_i, z_i; \theta)$$

(c) Suppose you know the values of z_i. Derive the maximum likelihood estimates for the parameter θ and write expressions for $\hat{\pi}, \hat{p_r}, \hat{p_b}$ with respect to x_i and z_i.

(d) Without knowing the values of z_i we use EM algorithm to estimate parameters. The algorithm starts with the initial parameters of θ^0 and θ^t shows the parameters in the beginning of the t-th iteration. In the E-step the algorithm needs to calculate the value of $P(Z_i = 1 | X_i = x_i; \theta^t)$. Calculate this value.

(e) For each flip i, the calculated value in the previous section is shown as γ_i^t. Therefore, the expected complete-data log-likelihood will be:

$$\sum_{i=1}^{m} (\gamma_i^{(t)} \ln p(x_i, 1; \theta) + (1 - \gamma_i^{(t)}) \ln p(x_i, 0; \theta))$$

In the M-step, we need to estimate θ^{t+1} in a way that we maximize the expected complete-data log-likelihood. Show the updated parameters of θ^{t+1}. (Write an expression for π^{t+1}, p_r^{t+1}, p_b^{t+1} with respect to x_i and γ_i^t.)

Answer

Part (a)

The joint probability distribution of X and Z given $\theta = (\pi, p_r, p_b)$ can be written as:

$$p(x, z; \theta) = p(X = x, Z = z; \theta)$$

Since each flip is independent, we can write the joint probability as the product of the probabilities for each flip i:

$$p(x, z; \theta) = \prod_{i=1}^{m} p(x_i, z_i; \theta)$$

For each flip i:

$$p(x_i, z_i; \theta) = p(Z_i = z_i; \pi) p(X_i = x_i | Z_i = z_i; p_r, p_b)$$

where

$$p(Z_i = 1; \pi) = \pi \quad \text{and} \quad p(Z_i = 0; \pi) = 1 - \pi$$

and

$$p(X_i = x_i | Z_i = 1; p_r) = \begin{cases} p_r & \text{if } x_i = 1 \\ 1 - p_r & \text{if } x_i = 0 \end{cases}$$

$$p(X_i = x_i | Z_i = 0; p_b) = \begin{cases} p_b & \text{if } x_i = 1 \\ 1 - p_b & \text{if } x_i = 0 \end{cases}$$

Thus, the joint probability for each flip i can be written as:

$$p(x_i, z_i; \theta) = \begin{cases} \pi p_r & \text{if } z_i = 1 \text{ and } x_i = 1 \\ \pi(1 - p_r) & \text{if } z_i = 1 \text{ and } x_i = 0 \\ (1 - \pi) p_b & \text{if } z_i = 0 \text{ and } x_i = 1 \\ (1 - \pi)(1 - p_b) & \text{if } z_i = 0 \text{ and } x_i = 0 \end{cases}$$

Part (b)

The complete-data log-likelihood $\mathcal{L}_c(\theta)$ is given by:

$$\mathcal{L}_c(\theta) = \sum_{i=1}^{m} \ln p(x_i, z_i; \theta)$$

Using the joint probabilities derived in part (a):

$$\mathcal{L}_c(\theta) = \sum_{i=1}^{m} \left[z_i \left(x_i \ln(\pi p_r) + (1 - x_i) \ln(\pi(1 - p_r))\right) + (1 - z_i) \left(x_i \ln((1 - \pi)p_b) + (1 - x_i) \ln((1 - \pi)(1 - p_b))\right)\right]$$

Part (c)

If we know the values of z_i, we can derive the maximum likelihood estimates for θ:

Estimate of π

$$\hat{\pi} = \frac{1}{m} \sum_{i=1}^{m} z_i$$

Estimate of p_r

$$\hat{p_r} = \frac{\sum_{i=1}^{m} z_i x_i}{\sum_{i=1}^{m} z_i}$$

Estimate of p_b

$$\hat{p_b} = \frac{\sum_{i=1}^{m} (1 - z_i) x_i}{\sum_{i=1}^{m} (1 - z_i)}$$

Part (d)

In the E-step, we need to calculate the value of $P(Z_i = 1 | X_i = x_i; \theta^t)$:

$$\gamma_i^t = P(Z_i = 1 | X_i = x_i; \theta^t) = \frac{P(Z_i = 1; \pi^t) P(X_i = x_i | Z_i = 1; p_r^t)}{P(X_i = x_i; \theta^t)}$$

where

$$P(X_i = x_i; \theta^t) = P(Z_i = 1; \pi^t) P(X_i = x_i | Z_i = 1; p_r^t) + P(Z_i = 0; (1 - \pi^t)) P(X_i = x_i | Z_i = 0; p_b^t)$$

So,

$$\gamma_i^t = \frac{\pi^t (p_r^t)^{x_i} (1 - p_r^t)^{1-x_i}}{\pi^t (p_r^t)^{x_i} (1 - p_r^t)^{1-x_i} + (1 - \pi^t)(p_b^t)^{x_i} (1 - p_b^t)^{1-x_i}}$$

Part (e)

In the M-step, we maximize the expected complete-data log-likelihood:

$$\sum_{i=1}^{m} (\gamma_i^{(t)} \ln p(x_i, 1; \theta) + (1 - \gamma_i^{(t)}) \ln p(x_i, 0; \theta))$$

Updating parameters:

Update π

$$\pi^{t+1} = \frac{1}{m} \sum_{i=1}^{m} \gamma_i^t$$

Update p_r

$$p_r^{t+1} = \frac{\sum_{i=1}^{m} \gamma_i^t x_i}{\sum_{i=1}^{m} \gamma_i^t}$$

Update p_b

$$p_b^{t+1} = \frac{\sum_{i=1}^{m} (1 - \gamma_i^t) x_i}{\sum_{i=1}^{m} (1 - \gamma_i^t)}$$