

25-2 DSL 정규 세션

Machine Learning 과제

- 본 과제는 학회 정규 세션 「Regression」, 「Classification」, 「Clustering」, 「Dimensionality Reduction」의 내용을 다루며, 개념의 적용과 실제 활용 사례에 대한 이해를 돋기 위해 기획되었습니다. 해당 과제는 평가를 위한 것이 아니므로, 주어진 힌트(💡)를 적극 활용하시고 학회원 간 토론 및 Slack 질의응답을 적극 활용하여 해결해주십시오. 단, 답안 표절이나 LLM의 남용은 금지합니다.
- 서술형 문제는 📜, 코딩 문제는 💻으로 표기가 되어 있습니다. 각 문제에서 요구하는 방법에 맞게 해결하며, 서술형 문제들은 따로 작성하시어 .pdf 파일로, 코딩 문제들은 주어진 .ipynb 파일에 답안을 작성하여 제출해 주십시오.
- 8/20 (수) 23 시 59 분까지 Github에 .pdf 파일과 .ipynb 파일들을 압축하여 하나의 .zip 파일로 끌어 제출해 주십시오. Github에 제출하는 방법을 모른다면 학술부장 혹은 과제 질의응답을 위한 오픈채팅방을 적극적으로 활용해 주십시오.

문제 1 Linear Regression

Regression은 독립변수가 종속변수에 영향을 미치는지 알아보기 위해 실시하는 분석 방법으로, 변수들 간의 관계를 정량적으로 모델링하고 예측하는 데 사용됩니다. 대표적인 유형으로는 Linear Regression(독립변수와 종속변수 간 선형 관계를 가정)과 Logistic Regression(종속변수가 범주형일 때 사용)이 있습니다. 실제 예측 성능을 높이기 위해서는 불필요한 변수를 제거하는 변수 선택(Feature Selection) 및 과적합(overfitting)을 방지하기 위한 정규화(Regularization) 기법이 함께 사용됩니다.

1-1 📜: 관측된 데이터 (x_i, y_i) 가 다음과 같은 선형 회귀 모델을 따른다고 가정합니다:

$$y_i = x_i^T w + \varepsilon_i, \quad \varepsilon_i \sim N(0, \sigma^2)$$

1) OLS를 사용하여 $L(w) = \sum_{i=1}^n (y_i - x_i^T w)^2$ 를 최소화하는 \hat{w} 를 구하세요.

$$\hat{w} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{Y}$$

2) 위 모델에서 y_i 의 조건부 확률분포 $p(y_i | x_i; w)$ 를 바탕으로 w 의 MLE 추정량을 유도하세요.

$$P(y_i | x_i; w) = N(y_i | w^T x_i, \sigma^2)$$

$$\ell(w, x, y) = \sum_{i=1}^n \left[\frac{1}{2} \log \sigma^2 - \frac{1}{2} \log 2\pi + \frac{1}{2\sigma^2} (y_i - w^T x_i)^2 \right]$$

3) 두 접근법의 결과를 비교하고, 그러한 결과가 나타난 이유를 설명하세요.

💡 OLS 와 MLE 모두 convex function을 최소화하는 문제로 변환됩니다.

$$\nabla \ell_{OLS}(w, x, y) = 0 \rightarrow \mathbf{X}^T \mathbf{Y} - \mathbf{X}^T \mathbf{X} w = 0$$

두 결과는 Convex인 이차함수, 1차 항수의
이동값 0을 찾으므로 같은 결과

1-2 📈 : Feature Selection.ipynb 를 참조해주세요 .

💡 Regression 의 변수 선택 기법에는 대표적으로 Forward Selection 과 Backward Selection 이 있습니다. Forward Selection 은 빈 모델(변수 없음)에서 시작하여 설명력이 가장 높은 변수부터 하나씩 추가해나가는 방식입니다. 각 단계에서 가장 유의미한 변수를 추가하고, 모델의 성능이 더 이상 향상되지 않으면 선택을 중단합니다. Backward Selection 은 모든 변수를 포함한 모델에서 시작하여, 가장 덜 유의한 변수부터 하나씩 제거하는 방식입니다. 각 변수의 기여도를 평가하여 성능 향상에 방해가 되는 변수를 제거하며, 지정된 기준을 충족하지 못하면 더 이상 제거하지 않습니다.

💡 변수 선택의 기준으로는 AIC, BIC, R², Adjusted R² 등이 사용됩니다. AIC 와 BIC 는 모두 회귀 모델의 성능과 복잡도를 동시에 고려하며, 값이 작을수록 더 좋은 모델을 의미합니다. Adjusted R²은 변수가 많아질수록 자동으로 R²이 올라가는 현상을 방지하고, 불필요한 변수 추가시 감소하도록 보정된 지표입니다.

💡 VIF 는 다중공선성 문제를 진단하기 위한 지표입니다. VIF > 10 이면 해당 변수는 다른 변수와 강한 상관관계가 있으며 제거를 고려해야 합니다.

💡 QQ-Plot 은 잔차가 정규분포를 따르는지 시각적으로 진단하며, Residual Plot 은 잔차의 등분산성을 진단할 수 있습니다.

1-3 🔔 : 회귀분석에서의 정규화(Regularization)는 다음과 같이 목적 함수 + 제약 조건의 구조로 이해할 수 있습니다.

💡 Ridge 회귀는 다음과 같은 constrained optimization 문제로 표현할 수 있습니다.

$$\min_{\beta} \|y - X\beta\|^2 \quad \text{subject to} \quad \sum_{j=1}^p \beta_j^2 \leq t$$

이는 라그랑주 승수법을 사용하여 penalty term 으로 바꾸면 다음과 같은 형태의 목적함수로 바뀝니다.

$$L(\beta, \lambda) = \|y - X\beta\|^2 + \lambda \sum_{j=1}^p \beta_j^2$$

비슷하게 Lasso 회귀는 다음과 같은 문제로 표현됩니다.

$$\min_{\beta} \|y - X\beta\|^2 \quad \text{subject to} \quad \sum_{j=1}^p |\beta_j| \leq t$$

역시 penalty 항을 이용해 다음과 같은 목적함수로 바꿀 수 있습니다.

$$L(\beta, \lambda) = \|y - X\beta\|^2 + \lambda \sum_{j=1}^p |\beta_j|$$

1) Lasso 회귀는 Ridge 회귀와 어떤 점에서 penalty 항이 다른지, 그리고 이로 인해 기대할 수 있는 효과의 차이를 간략히 설명하세요.

Lasso는 Ridge와 차이점은 Ridge는 회귀성이 폭발적이 위치한 특징↑
즉각적 특성 parameter 값이 0 으로↑, Ridge는 전부 범위가 비슷한 값을 가지도록 유도

2) Lasso 회귀는 Ridge 와 달리 해석해(closed-form solution)가 존재하지 않습니다. 그 이유에 대해 설명하세요.

Lasso는 회귀함수가 폭발적이 . 따라서 예측값을 찾기

문제 2 (Soft-Margin) SVM

SVM은 데이터를 명확하게 구분할 수 있는 마진을 최대화하는 초평면을 찾는 지도 학습 알고리즘입니다.
Slack Variable을 도입해 오차를 허용하는 SVM의 목적 함수는 아래와 같습니다.

$$\min_{\mathbf{w}, b} \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^N \xi_i$$

$$\text{subject to } y^{(i)} (\mathbf{w}^T \mathbf{x}^{(i)} + b) \geq 1 - \xi_i, \quad \forall i = 1, \dots, N,$$

$$\xi_i \geq 0, \quad \forall i = 1, \dots, N.$$

Slack Variable(ξ_i): 오차를 허용할 때 사용하는 변수

C: margin과 training error 간 trade-off를 조절하는 하이퍼 파라미터

2-1 🔥: 아래의 목적 함수가 위의 목적 함수와 같음을 보이세요.

$$\min_{\mathbf{w}, b} L(\mathbf{w}, b) = \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^N \max(0, 1 - y^{(i)} (\mathbf{w}^T \mathbf{x}^{(i)} + b))$$

$$\text{제약: } y_i (\mathbf{w}^T \mathbf{x}_i + b) \geq 1 - \xi_i$$

$$\xi_i \geq 1 - y_i (\mathbf{w}^T \mathbf{x}_i + b)$$

$$\xi_i \geq 0 \text{ 의 제약과 함께 고려}$$

$$\therefore \xi_i \geq \max(0, 1 - y_i (\mathbf{w}^T \mathbf{x}_i + b))$$

$$\min_{\mathbf{w}, b} L(\mathbf{w}, b) = \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^N \max(0, 1 - y_i (\mathbf{w}^T \mathbf{x}_i + b))$$

2-2 🔥: 우리는 SGD(Stochastic Gradient Descent) 방식으로 SVM을 train시키고자 합니다.
먼저 SGD에 대해서 Loss Function은 다음과 같이 정의됩니다.

$$L(\mathbf{w}, b) = \frac{1}{2} \|\mathbf{w}\|^2 + C \cdot \max(0, 1 - y_i (\mathbf{w}^T \mathbf{x}_i + b))$$

Gradient Descent 방법이기에, 손실 함수의 가중치에 대한 기울기($\frac{\partial L}{\partial w}$)와 손실 함수의 편향에 대한 기울기($\frac{\partial L}{\partial b}$)를 구해야 합니다. 이는 다음과 같습니다.

$$\frac{\partial L}{\partial \mathbf{w}} = \begin{cases} \mathbf{w} & \text{if } y_i (\mathbf{w}^T \mathbf{x}_i + b) \geq 1 \\ \mathbf{w} - C \cdot y_i \mathbf{x}_i & \text{if } y_i (\mathbf{w}^T \mathbf{x}_i + b) < 1 \end{cases}$$

$$\frac{\partial L}{\partial b} = \begin{cases} 0 & \text{if } y_i (\mathbf{w}^T \mathbf{x}_i + b) \geq 1 \\ C \cdot y_i & \text{if } y_i (\mathbf{w}^T \mathbf{x}_i + b) < 1 \end{cases}$$

위의 기울기 식들이 성립함을 증명하세요.

💡 Loss Function과 if 조건문의 관련성을 보세요!

If $y_i (\mathbf{w}^T \mathbf{x}_i + b) \geq 1$ 이면 Max 값이 0 이므로 $L(\mathbf{w}, b) = \frac{1}{2} \|\mathbf{w}\|^2$

" " " ≤ 1 이면 Max 값이 $1 - y_i (\mathbf{w}^T \mathbf{x}_i + b)$ 이므로 $L(\mathbf{w}, b) = \frac{1}{2} \|\mathbf{w}\|^2 + C(1 - y_i (\mathbf{w}^T \mathbf{x}_i + b))$

그리고 기울기($L(\mathbf{w}, b)$) 하면 위의 값이 나옵니다

2-3 📈 : SVM.ipynb 를 참조하세요.

💡 2-2 결과와 아래 알고리즘을 참고하면 도움이 될겁니다!

Algorithm 1 SVM Training with Stochastic Gradient Descent

```
Initialize weights  $\mathbf{w} \leftarrow 0$  and bias  $b \leftarrow 0$ 
for iteration = 1 to  $n\_iters$  do
    for each sample  $(\mathbf{x}_i, y_i)$  in dataset do
        Compute condition
        if condition is True then
            Update  $\mathbf{w} \leftarrow \mathbf{w} - \text{learning\_rate} \cdot \frac{\partial L}{\partial \mathbf{w}}$ 
        else
            Update  $\mathbf{w} \leftarrow \mathbf{w} - \text{learning\_rate} \cdot \frac{\partial L}{\partial \mathbf{w}}$ 
            Update  $b \leftarrow b - \text{learning\_rate} \cdot \frac{\partial L}{\partial b}$ 
        end if
    end for
end for
return weights  $\mathbf{w}$  and bias  $b$ 
```

문제 3 EM algorithm

이번 문제에서는 숨겨진 변수(latent variable)가 포함된 확률 모델에서 최대우도추정(MLE)을 수행하는 대표적 기법인 Expectation-Maximization(EM) 알고리즘을 다루고자 합니다. EM 알고리즘은 관측되지 않은 변수로 인해 직접 최적화가 어려운 상황에서, 반복적인 기대-최대화 단계를 통해 수렴 가능한 근사 최적해를 찾는 과정을 제시합니다. 본 문제를 통해 EM 알고리즘이 어떻게 유도되며, 구체적으로 Gaussian Mixture Model(GMM)에 어떻게 적용되는지를 수학적으로 해석하고 분석해보는 것이 목적입니다.

3-1 💡 : 잠재 변수 Z 가 있는 경우, Likelihood 함수는 $L(\theta; X, Z) = p(X, Z|\theta)$ 로 정의됩니다. θ 에 대한 MLE, 관측한 데이터 X 가 가장 잘 설명되도록 하는 파라미터 θ 를 찾는 방법은 X 에 대한 Marginal Likelihood $p(X|\theta)$ 를 최대화하는 것입니다. 이것이 EM 알고리즘의 다음 두 단계를 반복하며 구해질 수 있음을 보이세요.

$$E-step : Q(\theta|\theta^{(t)}) = E_{Z \sim p(Z|X,\theta^{(t)})} [\log p(X, Z|\theta)]$$
$$M-step : \theta^{(t+1)} = \underset{\theta}{\operatorname{argmax}} Q(\theta|\theta^{(t)})$$

($\theta^{(t)}$ 는 t 번째 iteration에서의 parameter 값)

💡 $\log p(X|\theta) = \log \frac{p(X,Z|\theta)}{p(Z|X,\theta)}$

💡 $: Q(\theta|\theta^{(t)}) = E_{Z \sim p(Z|X,\theta^{(t)})} [\log p(X, Z|\theta)] = \sum_Z p(Z|X, \theta^{(t)}) \log p(X, Z|\theta)$

3-1

$$\begin{aligned} \log p(x|\theta) &= \log \sum_z p(x,z|\theta) = \log \sum_z g^{(z)} \frac{p(x,z|\theta)}{g^{(z)}} \\ &\downarrow \\ \log E_{g^{(z)}} \left[\frac{p(x,z|\theta)}{g^{(z)}} \right] &\geq E_{g^{(z)}} \left[\log \frac{p(x,z|\theta)}{g^{(z)}} \right] \\ &\downarrow \\ \sum_z g^{(z)} \log \frac{p(x,z|\theta)}{g^{(z)}} &\quad (\text{ELBO}) \\ &\downarrow \\ \ell(g, \theta) &= \sum_z g^{(z)} \log p(x,z|\theta) - \sum_z g^{(z)} \log g^{(z)} \\ &\quad \text{이후 분자 대화} \quad \text{연도 3회 (상수)} \\ &\downarrow \\ Q(\theta|\theta^{(t)}) &= \sum_z p(z|x, \theta^{(t)}) \log p(x,z|\theta) \\ &= E_{z|x, \theta^{(t)}} [\log p(x,z|\theta)] \\ &\quad \text{각각 } \theta^{(t+1)} = \underset{\theta}{\operatorname{argmax}} Q(\theta|\theta^{(t)}) \end{aligned}$$



3-2 🔞: GMM 의 파라미터를 EM 알고리즘을 통해 추정할 때, $Q(\theta | \theta^{(t)})$ 는 다음과 같이 표현할 수 있습니다.

$$\begin{aligned} Q(\theta | \theta^{(t)}) &= E_{Z|X, \theta^{(t)}}[\log f_\theta(Z, X)] \\ &= \sum_{i=1}^n E_{Z_i|X_i, \theta^{(t)}}[\log f_\theta(Z_i, X_i)] \\ &= \sum_{i=1}^n \sum_{k=1}^K \gamma_{k,i,\theta^{(t)}} \log P(Z_i = k, X_i | \theta) \\ &= \sum_{i=1}^n \sum_{k=1}^K \gamma_{k,i,\theta^{(t)}} \log [w_k \mathcal{N}(X_i | \mu_k, \Sigma_k)] \\ \gamma_{k,i,\theta^{(t)}} &= P(Z_i = k | X_i, \theta^{(t)}) \\ &= \frac{P(Z_i = k | \theta^{(t)}) \cdot P(X_i | Z_i = k, \theta^{(t)})}{\sum_{j=1}^K P(Z_i = j | \theta^{(t)}) \cdot P(X_i | Z_i = j, \theta^{(t)})} \\ &= \frac{w_k^{(t)} \mathcal{N}(X_i | \mu_k^{(t)}, \Sigma_k^{(t)})}{\sum_{j=1}^K w_j^{(t)} \mathcal{N}(X_i | \mu_j^{(t)}, \Sigma_j^{(t)})} \end{aligned}$$

(n 은 관측 데이터의 개수, K 는 cluster의 개수, $\gamma_{k,i,\theta^{(t)}}$ 는 책임값(Responsibility)로 X_i 가 k 번째 gaussian 분포에 속할 확률, $w_k^{(t)}$ 는 데이터 Z_i 가 cluster k 에 속할 확률, $\mu_k^{(t)}$ 와 $\Sigma_k^{(t)}$ 는 각각 k 번째 gaussian 분포의 평균과 공분산)

이때 EM 알고리즘으로 다음 파라미터 $w_k^{(t+1)}$ 이 아래와 같이 추정됨을 보이세요

$$w_k^{(t)} = \frac{1}{n} \sum_{i=1}^n \gamma_{k,i,\theta^{(t)}}$$



w_k 는 확률값이므로 다음 제약 조건을 만족해야합니다. (Lagrange Multiplier 를 활용해보세요.)

$$\sum_{k=1}^K w_k = 1 \text{ and } w_k \geq 0$$



동일한 논리로 모든 k 에 대한 책임값 $\gamma_{k,i,\theta^{(t)}}$ 들의 합 또한 1입니다.

3-3 🔞 (Optional) : 또한 $\mu_k^{(t+1)}$ 과 $\Sigma_k^{(t+1)}$ 이 아래와 같이 추정됨을 보이세요.

$$\begin{aligned} \mu_k^{(t+1)} &= \frac{\sum_{i=1}^n \gamma_{k,i,\theta^{(t)}} X_i}{\sum_{i=1}^n \gamma_{k,i,\theta^{(t)}}} \\ \Sigma_k^{(t+1)} &= \frac{1}{\sum_{i=1}^n \gamma_{k,i,\theta^{(t)}}} \sum_{i=1}^n \gamma_{k,i,\theta^{(t)}} (X_i - \mu_k^{(t+1)}) (X_i - \mu_k^{(t+1)})^T \end{aligned}$$

3-4 GMM.ipynb 는 EM 알고리즘을 활용한 GMM 기반 클러스터링 예시들을 확인해보는 코드를 다루고 있습니다. 코드 내에서 EM 알고리즘으로 추정한 parameter 들이 위 문제에서 유도한 수식과 일치하는지 확인해보세요. 실험 결과와 이론을 대조해보며 알고리즘과 그 활용에 충분한 고민을 해보기 바랍니다.

문제 4 PCA

PCA 는 고차원 데이터의 구조를 요약하는 동시에, 주요 분산 성분만을 보존하고 잡음(노이즈) 성분은 제거하는 효과가 있어, 특징 추출, 전처리, 시각화, 노이즈 제거 등의 다양한 작업에 널리 활용됩니다. 본 문제에서는 이미지 데이터를 입력으로 받아 PCA 를 직접 구현하고, 주성분 수에 따른 압축·복원 결과를 시각적으로 비교합니다. 본 과제의 구현 및 실습은 PCA.ipynb 파일에서 확인할 수 있습니다.

4-1 : PCA 와 PCA 를 이용한 이미지 압축·복원 코드를 완성하고, 결과를 확인하세요.

PCA의 Component, 즉 PC는 많이 사용했을 때 설명해줘

4-2 : PCA 의 주성분 개수를 조절하며 노이즈 제거 실험을 해보고, 실험 결과를 분석하여 서술하세요.

0부터 256 까지 Components를 50 씩 늘린 결과,

전체 설명내용은 볼수있다

4-3 (Optional): PCA.ipynb 를 수행하면서 느낀 PCA 의 단점에 대해 언급해 보세요. 이 단점을 보완할 수 있는 방법이 있을까요? 생각나는 대로 자유롭게 적어보세요.

PCA 의 시간 복잡도는 어떻게 될까요?

https://en.wikipedia.org/wiki/Power_iteration

Reference

- 24-2 Regression + SVM + 비지도학습 (11 기 김현진, 김정우)
- 25-1 Machine Learning (12 기 김은희, 이정우)
- 25-1 Unsupervised Learning #2 (12 기 복지민)
- Statistical Machine Learning, Prof. Lee, Spring 2024, Yonsei Univ.
- Deep Learning, Prof. Park, Spring 2025, Yonsei Univ
- Regression Analysis, Prof. Jeon, Fall 2023, Yonsei Univ.
- Bishop, C. M. (2006). Pattern recognition and machine learning. Springer.

Data Science Lab

담당자: 13 기 강승우, 한연주
ksw030721@yonsei.ac.kr
amiee1510@gmail.com