Matlab Problem of the Second Practice Day

Gaussian Classifier

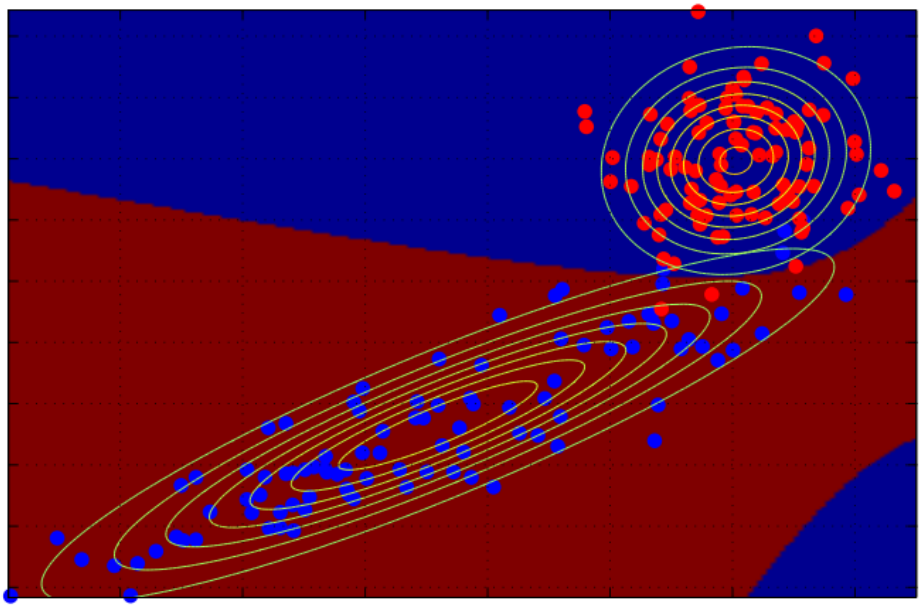
Gaussian Distribution은 다음과 같다.

$$x \sim N(\mu, \Sigma)$$

$$p(x) = \frac{1}{\sqrt{(2\pi)^D \,|\Sigma|}} \exp(-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu))$$

$$x \in \mathbb{R}^D, \mu \in \mathbb{R}^D, \Sigma \in \mathbb{R}^{D \times D}$$

D는 데이터 x의 차원이다. x나 μ가 벡터라는 점에, 그리고 Σ가 행렬이라는 점에 유의하자. 이 때, μ는 평균 벡터이며, Σ는 공분산 행렬이다. (공분산, covariance란 무엇인가? 검색해보자.)



Gaussian Classifier는 데이터 집합 $\{x_i\}$과 이에 상응하는 class $\{c_i\}$에 대하여, 각 class에 해당하는 데이터의 분포가 Gaussian Distribution을 따른다고 가정한다.

$$p(x \mid c = k) = \frac{1}{\sqrt{(2\pi)^D \,|\Sigma_k|}} \exp(-\frac{1}{2}(x-\mu_k)^T \Sigma_k^{-1}(x-\mu_k))$$

$$p(c=k) \propto N_k$$

$$\begin{aligned} c_i^* &= \arg\max_k p(c_i = k \mid x_i) \\ &= \arg\max_k p(x_i \mid c_i = k)\, p(c_i = k) \\ &= \arg\max_k p(x_i \mid c_i = k; \mu_{1:K}, \Sigma_{1:k})\, p(c_i = k; N_{1:k}) \end{aligned}$$

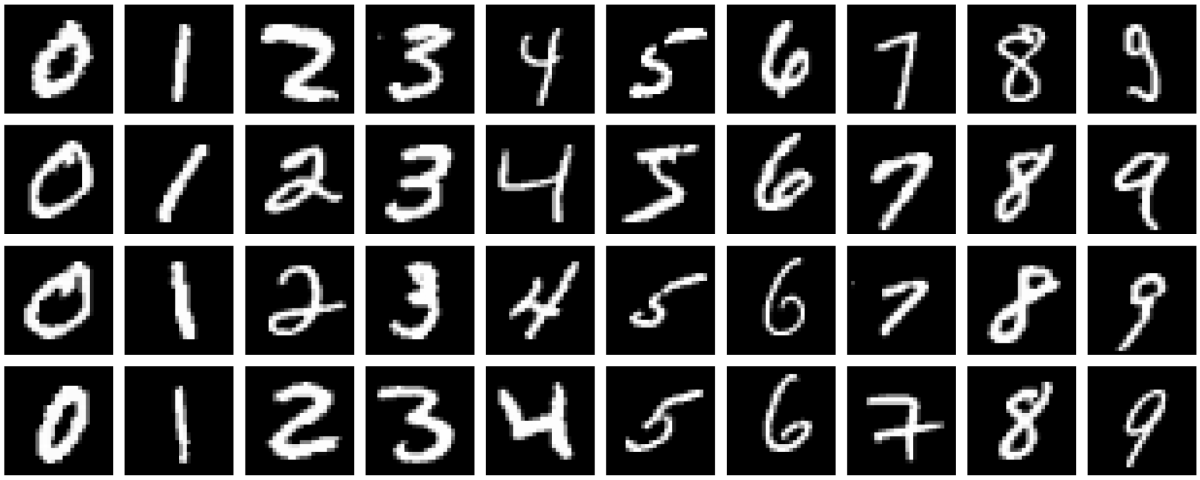
이 때, μ_k 는 class가 k인 x 의 집합의 평균 벡터로, Σ_k 는 class가 k인 x 집합의 공분산 행렬로, N_k 를 class가 k인 x 의 개수로 가정할 수 있다.

$$\mu_k = \frac{1}{N_k} \sum_{c_i=k} x_i$$

$$\Sigma_k = \frac{1}{N_k} \sum_{c_i=k} (x_i - \mu_k)(x_i - \mu_k)^T$$

분류를 수행하기 위해 위 문서에서 선택한 가정에 대해 어떻게 생각하는가? Parameter에 대한 가정이 적절한가? 어떻게 그 가정이 정당화될 수 있을까? PRML 책 p.200~202 에서 그 가정에 대해 다루고 있다. 여기서는, Maximum Likelihood Learning을 통하여 Parameter를 학습한다. 핵심 아이디어는 likelihood값에 parameter에 대한 미분을 취하고, 그 미분 값이 0이 된다고 가정하는 방정식을 세운 뒤, 이를 analytic하게 풀어 정해진 해를 얻는 것이다.

수업 시간에 잠시 소개된 MNIST data는 사람들이 손으로 쓴 숫자 필기체와 그 숫자의 class로 구성된 데이터이다. 학습 데이터는 60000개, 테스트 데이터는 10000개이다. http://yann.lecun.com/exdb/mnist/ 에서 데이터와 현재까지의 이 데이터에 대한 분류 성능 보고를 확인할 수 있다.



우리는 그 중에 학습 데이터 1000개를 사용하여 학습하고, 테스트 데이터 100개를 사용하여 학습 결과를 검증할 것이다.

10개의 Gaussian Distribution으로 MNIST 숫자 데이터를 학습해보자. 아래의 코드는 Gaussian Distribution을 각 Class에 대해서 학습하고 이를 바탕으로 첫 번째 테스트 데이터의 확률을 구한 것이다. (여기서 학습은 무엇인가? 학습 대상이 되는 Parameter는 무엇인가? 학습 방법은 ML에 해당하는가? MAP에 해당하는가? Bayesian Learning에 해당하는가?)

아래의 코드에서는 Gaussian Classifier를 학습하고, 이를 바탕으로 첫 번째 테스트 데이터의 분포를 추정해 보았다.

```
p_class = zeros(1,10);
for j = [1:10]
    train_subx{j} = train_x(train_c==mod(j,10),:);
    mu{j} = mean(train_subx{j});
    sigma{j} = cov(train_subx{j})+0.1*eye(size(train_x,2));
    p_class(j) = size(train_subx{j},1)/size(train_x,1);
end


for j = [1:10]
    p_test_x1(j) = p_class(j)*mvnpdf(test_x(1,:),mu{j},sigma{j});
end
```

Matlab Code: Calculate probability of Gaussian Distribution - 실습 때 학생들에게 주어지는 코드

(위 코드에서 eye(size(train_x,2)) 의 역할은 무엇인가?)

주어진 코드를 바탕으로 Gaussian Classifier를 구현, Confusion Matrix를 만들자. 테스트 데이터

에 대한 Gaussian Classifier의 Accuracy는 얼마인가?

Matlab Code: Gaussian Classifier

<< APPENDIX >>

1. Confusion Matrix

Confusion Matrix는 데이터의 실제 class와 classifier가 분류한 class의 관계를 나타내는 행렬이다. 아래의 예에서 분류기는 13개의 Rabbit data 중 11개를 실제 Rabbit이라고 분류하였다. 반면, Dog라고 분류된 8개의 데이터 중 단지 3개만이 실제 Dog data이다.

		Predicted class		
		Cat	Dog	Rabbit
Actual class	Cat	5	3	0
	Dog	2	3	1
	Rabbit	0	2	11