PRML과 부수적인 자료들을 공부하여 간단히 정리하였습니다.

12.1 Principal Component Analysis

- 김성범 교수님의 유튜브영상을 보고 정리했습니다.

변수 선택/추출을 통한 차원 축소

- 변수선택(selection)
  - 장점 : 선택한 변수 해석 용이
  - 단점 : 변수간 상관관계 고려 어려움
- 변수추출(extraction)
  - 장점 : 변수간 상관관계 고려, 일반적으로 변수의 개수를 많이 줄일 수 있음
  - 단점 : 추출된 변수의 해석이 어려움

PCA 개요

- **기존 변수의 선형 결합(linear combination)으로 새로운 변수 생성**
- **원래 data의 분산을 최대한 보존하는 새로운 축을 찾고 그 축에 data를 사영(projection) 시키는 기법**
- 주요 목적
  - 차원 축소
  - 시각화 및 해석

전개 과정

PCA는 기존의 변수를 선형결합하여 새로운 변수를 만든다.

- data X : p * n (변수 p개, data수 n개)
- $\mathbf{a}_i = [a_{i1}, a_{i2}, ..., a_{ip}]^T$: i번째 기저(basis) 또는 계수(loading) , p * 1
- $\mathbf{z}_1, \mathbf{z}_2, ..., \mathbf{z}_p$: 새로 만든 변수 (주성분, Score) , 1 * n
  - 최대 p개의 새로운 변수 생성 가능

$$\mathbf{z}_i = \mathbf{a}_i^T X = a_{11}\mathbf{x}_1 + a_{12}\mathbf{x}_2 + ... + a_{1p}\mathbf{x}_p$$

위처럼 기존 변수의 선형결합을 통해 주성분을 만든다. 이와 같은 선형결합은 선형변환으로도 이해할 수 있다. X를 새로운 축 $\mathbf{a}_i$ 라는 축에 사영시켜서 새로운 변수 $\mathbf{z}_i$를 만든다.

$$Z = \begin{bmatrix} \mathbf{z}_1 \\ \mathbf{z}_2 \\ \cdots \\ \mathbf{z}_p \end{bmatrix} = \begin{bmatrix} \mathbf{a}_1^T X \\ \mathbf{a}_2^T X \\ \cdots \\ \mathbf{a}_p^T X \end{bmatrix} = \begin{bmatrix} \mathbf{a}_1^T \\ \mathbf{a}_2^T \\ \cdots \\ \mathbf{a}_p^T \end{bmatrix} X = A^T X$$

우리의 목표는 기존의 변수 앞에 곱해지는 계수들을 구하는 것이다. 그렇다면 어떤 계수들이 필요한가? PCA에서는 **분산을 최대한 보존하면서 차원을 축소** 하려고 한다. 분산을 최대한 보존하면서 새로운 변수를 만드는 과정을 살펴보자.

- **공분산행렬** : $\Sigma = cov(X) = \frac{1}{n-1} X X^T$
  - 단, X는 centering된 상태
  - 공분산행렬은 p * p의 대칭행렬
  - 공분산행렬의 고유벡터끼리는 서로 직교(orthogonal)

$$max_{\mathbf{a}}[Var(\mathbf{z})] = max_{\mathbf{a}}[\mathbf{a}^T Var(X)\mathbf{a}] = max_{\mathbf{a}}[\mathbf{a}^T \Sigma \mathbf{a}]$$

$\mathbf{a}$가 커지면 분산이 커지기 때문에 제약식을 둔다.

$$\mathbf{a}^T\mathbf{a} = 1$$

이를 Lagrange로 풀면

$$L = \mathbf{a}^T \Sigma \mathbf{a} - \lambda(\mathbf{a}^T \mathbf{a} - 1)$$

$$\frac{\partial L}{\partial \mathbf{a}} = \Sigma \mathbf{a} - \lambda \mathbf{a} = 0$$

$$\Sigma \mathbf{a} = \lambda \mathbf{a}$$

- $\mathbf{a}$는 공분산행렬의 **고유벡터(eigenvector)**
- 이에 매칭되는 λ는 **고유값(eigenvalue)**

따라서 우리가 구하고자 했던 주성분들을 구할 수 있다. 계수를 구했고 이를 통해 주성분을 계산할 수 있다. 그렇다면 이제 각 주성분이 갖고 있는 분산이 얼마나 되는지 알아보자.

$$Var(\mathbf{z}) = \mathbf{a}^T \Sigma \mathbf{a} = \mathbf{a}^T \lambda \mathbf{a} = \lambda \mathbf{a}^T \mathbf{a} = \lambda$$

따라서 **각 주성분의 분산은 이에 해당하는 고유값** 이다. 그래서 주성분을 구성할 때, 고유값이 큰 순서대로 만드는 것이다. 기존의 변수보다 적은 갯수로 비슷한 크기의 분산을 유지할 수 있는 것이다.

12.1.1 Maximum variance formulation

위의 내용과 같다.

12.1.2 Minimum-error formulation

skip

12.1.3 Applications fo PCA

skip

12.2 Probabilistic PCA

- 장점
  - PPCA는 data set의 dominant한 correlation을 잡으면서 Gaussian distribution의 제약된 형식으로 나타낼 수 있다.
  - 몇 개의 고유벡터만을 이용해도 되는 경우, EM algorithm을 사용함으로 computationally 효율적이다. (data covariance를 계산하지 않아도 되므로)
  - Probabilistic model과 EM의 조합으로 data set의 missing value를 다룰수 있다.
  - PPCA는 Bayesian pca의 기반이 된다. Bayesian pca는 주성분의 차원을 자동으로 정해준다.
  - PPCA는 class-conditional density modeling에 사용될 수 있고 이를 통해 classification 문제에 응용될 수 있다.

PPCA는 linear gaussian framework의 한 예시라고 할 수 있다.

- 주성분의 subspace에 해당하는 explicit latent variable $\mathbf{z}$
- prior distribution of $\mathbf{z}$
  - 더 복잡한 gaussian을 가정해도 상관없다.

$$p(\mathbf{z}) = N(\mathbf{0}, \mathbf{I})$$

- gaussian conditional distribution
  - D * M matrix $\mathbf{W}$
  - D차원 vector $\boldsymbol{\mu}$

$$p(\mathbf{x}|\mathbf{z}) = N(\mathbf{W}\mathbf{z} + \boldsymbol{\mu}, \sigma^2 \mathbf{I})$$

PPCA를 generative한 시선으로 살펴보자. latent variable을 정하고 이에 따른 conditional distribution에서 observed value $\mathbf{x}$를 얻을 수 있다.

- mapping from latent space to data space
  - PCA의 반대 과정
  - $\boldsymbol{\epsilon} \sim N(\mathbf{0}, \sigma^2\mathbf{I})$

$$\mathbf{x} = \mathbf{W}\mathbf{z} + \boldsymbol{\mu} + \boldsymbol{\epsilon}$$

그렇다면 우리는 이제 parameter들에 대해 알아야 한다. likelihood를 이용할 것이다. marginal distribution을 표현하면

$$p(\mathbf{x}) = \int p(\mathbf{x}|\mathbf{z})p(\mathbf{z})d\mathbf{z}$$

marginal distribution 또한 gaussian일 것이고

$$E[\mathbf{x}] = E[\mathbf{W}\mathbf{z} + \boldsymbol{\mu} + \boldsymbol{\epsilon}] = \boldsymbol{\mu}$$
$$cov[\mathbf{x}] = E[(\mathbf{W}\mathbf{z} + \boldsymbol{\epsilon})(\mathbf{W}\mathbf{z} + \boldsymbol{\epsilon})^T]$$
$$= E[\mathbf{W}\mathbf{z}\mathbf{z}^T\mathbf{W}^T] + E[\boldsymbol{\epsilon}\boldsymbol{\epsilon}^T] = \mathbf{W}\mathbf{W}^T + \sigma^2\mathbf{I} = \mathbf{C}$$

$$p(\mathbf{x}) = N(\boldsymbol{\mu}, \mathbf{C})$$

따라서 predictive distribution $p(\mathbf{x})$는 parameter $\boldsymbol{\mu}, \mathbf{W}, \sigma^2$으로 이루어져있다. 하지만 여기에 latent space coordinates의 rotation에 해당하는 redundancy가 존재하다.

- $\widetilde{\mathbf{W}} = \mathbf{W}\mathbf{R}$ 라고 정의하자.
  - $\mathbf{R}$는 orthogonal matrix
  - $\mathbf{R}\mathbf{R}^T = \mathbf{I}$

$$\widetilde{\mathbf{W}}\widetilde{\mathbf{W}}^T = \mathbf{W}\mathbf{R}\mathbf{R}^T\mathbf{W} = \mathbf{W}\mathbf{W}$$

$\mathbf{W}, \widetilde{\mathbf{W}}$ 모두 같은 predictive distribution이 나온다. 이런 invariance는 latent space의 rotation 때문이라고 할 수 있다. 나중에 살펴보자.

predictive distribution을 계산하기 위해서는 $\mathbf{C}$의 inverse를 계산해야 한다. 하지만 computationally 간단한 방법이 있다.

$$\mathbf{C}^{-1} = \sigma^{-1}\mathbf{I} - \sigma^{-2}\mathbf{W}\mathbf{M}^{-1}\mathbf{W}^T$$

$$\mathbf{M} = \mathbf{W}^T\mathbf{W} + \sigma^2\mathbf{I}$$

$\mathbf{M}$의 inverse로 간접적으로 구하는게 $O(D^3)$에서 $O(M^3)$로 줄어든다.

posterior distribution은 linear gaussian 공식을 이용하여 구할 수 있다.

$$p(\mathbf{z}|\mathbf{x}) = N(\mathbf{M}^{-1}\mathbf{W}^T(\mathbf{x} - \boldsymbol{\mu}), \sigma^{-2}\mathbf{M})$$

12.2.1 Maximum likelihood PCA

likelihood를 이용하여 model parameter를 추정해보자.

$$\ln p(\mathbf{X}|\boldsymbol{\mu}, \mathbf{W}, \sigma^2) = \sum_{n=1}^{N} \ln p(\mathbf{x}_n|\mathbf{W}, \boldsymbol{\mu}, \sigma^2)$$
$$= -\frac{ND}{2}\ln 2\pi - \frac{N}{2}\ln|\mathbf{C}| - \frac{1}{2}\sum_{n=1}^{N}(\mathbf{x}_n - \boldsymbol{\mu})^T\mathbf{C}^{-1}(\mathbf{x}_n - \boldsymbol{\mu})$$

이를 $\boldsymbol{\mu}$에 대해 미분하여 mle를 구하면 $\boldsymbol{\mu} = \bar{\mathbf{x}}$가 나온다. 이를 log likelihood function에 대입하면

$$\ln p(\mathbf{X}|\boldsymbol{\mu}, \mathbf{W}, \sigma^2) = -\frac{N}{2}\{D\ln 2\pi + \ln|\mathbf{C}| + Tr(\mathbf{C}^{-1}\mathbf{S})\}$$

여기서 $\mathbf{S}$는 data covariance matrix이다. log likelihood function이 $\boldsymbol{\mu}$의 quadratic form이고 unique maximum을 가진다.

$\mathbf{W}, \sigma^2$에 대해 maximize하는 것은 조금 더 복잡하지만 closed-form으로 존재한다.

- $\mathbf{U}_M$: D*M matrix, column들이 covariance matrix 고유벡터의 subset
- $\mathbf{L}_M$: M*M diagonal matrix, 대각성분이 고유값
- $\mathbf{R}$: 임의의 M*M orthogonal matrix

$$\mathbf{W}_{ML} = \mathbf{U}_M(\mathbf{L}_M - \sigma^2\mathbf{I})^{1/2}\mathbf{R}$$

여기서 고유벡터는 당연히 고유벡터의 크기 순서대로 정렬되어 있다. 따라서 위의 식을 통해 $\mathbf{W}$의 column들은 standard PCA의 principal subspace라고 할 수 있다. 다음으로 σ는

$$\sigma^2_{ML} = \frac{1}{D-M}\sum_{i=M+1}^{D}\lambda_i$$

σ^2은 버려진 차원의 variance를 평균낸 것이라고 할 수 있다. (M차원 이외의 것들)

covariance matrix에 대해 조금 더 살펴보자. unit vector $\mathbf{v}$ 방향으로의 predictive distribution의 variance는 $cov[\mathbf{v}^T\mathbf{x}] = \mathbf{v}^T\mathbf{C}\mathbf{v}$ 이다.

- 첫번째로 $\mathbf{v}$가 principal component에 orthogonal하다고 가정하자. 즉, 버려진 고유벡터의 linear combination인 것이다.
  - 그러면 $\mathbf{v}^T\mathbf{U} = \mathbf{0}$이고
  - $\mathbf{v}^T\mathbf{C}\mathbf{v} = \sigma^2$ 이다. (지금까지 나온 공식을 이용)
  - 따라서 model은 principal subspace에 orthogonal한 noise variance(=average of the discarded eigenvector)를 predict 한다.
- 다음으로 $\mathbf{v} = \mathbf{u}_i$ 라고 가정하자.
  - $\mathbf{u}_i$는 principal subspace에 들어간 eigenvector
  - $\mathbf{v}^T\mathbf{C}\mathbf{v} = (\lambda_i - \sigma^2) + \sigma^2 = \lambda_i$
  - 즉, 이 model은 그 principal axes에 해당하는 variance(고유값)를 잘 잡았다고 할 수 있다.

만약에 $M = D$라고 한다면, 즉 차원축소를 하지 않았을때 marginal distribution의 covariance는 sample covariance와 같아진다.

- $\mathbf{U}_M = \mathbf{U}, \mathbf{L}_M = \mathbf{L}$
- orthgonality를 이용하면 $\mathbf{U}\mathbf{U}^T = \mathbf{I}, \mathbf{R}\mathbf{R}^T = \mathbf{I}$

$$\mathbf{C} = \mathbf{U}(\mathbf{L} - \sigma^2\mathbf{I})^{1/2}\mathbf{R}\mathbf{R}^T(\mathbf{L} - \sigma^2\mathbf{I})^{1/2}\mathbf{U}^T + \sigma^2\mathbf{I} = \mathbf{U}\mathbf{L}\mathbf{U}^T = \mathbf{S}$$

PPCA를 reverse해보자.

$$E[\mathbf{z}|\mathbf{x}] = \mathbf{M}^{-1}\mathbf{W}_{ML}^T(\mathbf{x} - \bar{\mathbf{x}})$$

이를 이용하여 point $\mathbf{x}$를 구해보면

$$\mathbf{W}E[\mathbf{z}|\mathbf{x}] + \boldsymbol{\mu}$$

형태가 linear regression과 비슷하다.

만약에 $\sigma^2 \to 0$ 으로 만들면 posterior mean은

$$(\mathbf{W}_{ML}^T\mathbf{W}_{ML})^{-1}\mathbf{W}_{ML}^T(\mathbf{x} - \bar{\mathbf{x}})$$

이는 standard PCA에서와 똑같은 결과이다. data point를 latent space로 orthogonal projection한 것이다.

12.2.2 EM algorithm for PCA

위에서 closed-form으로 parameter들에 대해 구하였다. 하지만 고차원에서는 EM algorithm이 더 효율적이다. 계산과정은 생략하고 수식에 대해서만 살펴보자.

- complete-data log likelihood

$$\ln p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\mu}, \mathbf{W}, \sigma^2) = \sum_{n=1}^{N}\{\ln p(\mathbf{x}_n|\mathbf{z}_n) + \ln p(\mathbf{z}_n)\}$$

- E-step

$$\begin{aligned} E[\ln p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\mu}, \mathbf{W}, \sigma^2)] = -\sum_{n=1}^{N}\{\frac{D}{2}\ln(2\pi\sigma^2) + \frac{1}{2}Tr(E[\mathbf{z}_n\mathbf{z}_n^T]) \\ + \frac{1}{2\sigma^2}\|\mathbf{x}_n - \boldsymbol{\mu}\|^2 - \frac{1}{\sigma^2}E[\mathbf{z}_n]^T\mathbf{W}^T(\mathbf{x}_n - \boldsymbol{\mu}) \\ + \frac{1}{2\sigma^2}Tr(E[\mathbf{z}_n\mathbf{z}_n^T])\mathbf{W}^T\mathbf{W})\} \end{aligned}$$

- M-step

$$\mathbf{W}_{new} = [\sum_{n=1}^{N}(\mathbf{x}_n - \bar{\mathbf{x}})E[\mathbf{z}_n]^T][\sum_{n=1}^{N}E[\mathbf{z}_n\mathbf{z}_n^T]]^{-1}$$

$$\begin{aligned} \sigma^2_{new} = \frac{1}{ND}\sum_{n=1}^{N}\{\|\mathbf{x}_n - \bar{\mathbf{x}}\|^2 - 2E[\mathbf{z}_n]^T\mathbf{W}_{new}^T(\mathbf{x}_n - \bar{\mathbf{x}}) \\ + Tr(E[\mathbf{z}_n\mathbf{z}_n^T]\mathbf{W}_{new}^T\mathbf{W}_{new})\} \end{aligned}$$

- N와 D 가 클수록 EM이 computationally 효과적이다.
- missing data를 다룰 수 있다. (MAR)
- $\sigma^2 \rightarrow 0$ 으로 standard PCA에서도 EM을 이용할 수 있다.

12.2.3 Bayesian PCA

PCA를 하기 위해서는 적절한 M을 직접 찾아야한다. 하지만 Bayesian 접근법을 통해 그 과정을 없앨 수 있다. 이전에 공부했듯이 parameter를 marginalize out하면 된다. 다양한 방법이 있는데 여기서는 *evidence approximation* 으로 해보자. 일단 $\mathbf{W}$에 대해서만 고려해보자. 다른 parameter도 prior를 통해 완전한 Bayesian 접근을 할 수 있다.

$$p(\mathbf{W}|\boldsymbol{\alpha}) = \prod_{i=1}^{M}(\frac{\alpha_i}{2\pi})^{D/2}exp\{-\frac{1}{2}\alpha_i\mathbf{w}_i^T\mathbf{w}_i\}$$

$$p(\mathbf{X}|\boldsymbol{\alpha}, \boldsymbol{\mu}, \sigma^2) = \int p(\mathbf{X}|\mathbf{W}, \boldsymbol{\mu}, \sigma^2)p(\mathbf{W}|\boldsymbol{\alpha})d\mathbf{W}$$

근데 integration이 intractable하기에 Laplace approximation을 이용한다. 만약 posterior가 peaked되어 있고 data set이 크다면 marginal likelihood를 α_i에 대해 최대화하면

$$\alpha_i^{new} = \frac{D}{\mathbf{w}_i^T\mathbf{w}_i}$$

이를 통해 EM으로 구하면 where $\mathbf{A} = diag(\alpha_i)$

$$\mathbf{W}_{new} = [\sum_{n=1}^{N}(\mathbf{x}_n - \bar{\mathbf{x}})E[\mathbf{z}_n]^T][\sum_{n=1}^{N}E[\mathbf{z}_n\mathbf{z}_n^T] + \sigma^2\mathbf{A}]^{-1}$$

12.2.4 Factor analysis

linear-Gaussian latent variable model이라는 점에서 PPCA와 비슷하다. 다른 점은 conditional distribution에서 분산의 형태이다. PPCA에서는 isotropic covariance였는데 Factor analysis에서는 diagonal covariance를 고려한다.

- $\mathbf{\Psi}$: D * D diagonal covariance

$$p(\mathbf{x}|\mathbf{z}) = N(\mathbf{x}|\mathbf{W}\mathbf{z} + \boldsymbol{\mu}, \mathbf{\Psi})$$

Factor analysis model은

- explaining the observed covariance structure of the data
  - by representing the independent variance associated with each coordinate in the matrix $\mathbf{\Psi}$
  - and capturing the covariance between variables in the matrix $\mathbf{W}$

Factor analysis에서

- *factor loadings* : columns of $\mathbf{W}$ (which capture the correlations between observed variables)
- *uniquenesses* : diagonal elements of $\mathbf{\Psi}$ (which represent the independent noise variances for each of the variables)

이제 parameter들에 대해 구해보자. $\boldsymbol{\mu}$는 똑같이 sample mean이다. 나머지 parameter들은 closed form이 존재하지 않는다. 따라서 EM algorithm을 이용한다.

- E-step
  - $\mathbf{G} = (\mathbf{I} + \mathbf{W}^T\mathbf{\Psi}^{-1}\mathbf{W})^{-1}$

$$E[\mathbf{z}_n] = \mathbf{G}\mathbf{W}^T\mathbf{\Psi}^{-1}(\mathbf{x}_n - \bar{\mathbf{x}})$$
$$E[\mathbf{z}_n\mathbf{z}_n^T] = \mathbf{G} + E[\mathbf{z}_n]E[\mathbf{z}_n]^T$$

- M-step

$$\mathbf{W}_{new} = [\sum_{n=1}^{N}(\mathbf{x}_n - \bar{\mathbf{x}})E[\mathbf{z}_n]^T][\sum_{n=1}^{N}E[\mathbf{z}_n\mathbf{z}_n^T]]^{-1}$$
$$\mathbf{\Psi}_{new} = diag\{\mathbf{S} - \mathbf{W}_{new}\frac{1}{N}\sum_{n=1}^{N}E[\mathbf{z}_n](\mathbf{x}_n - \bar{\mathbf{x}})^T\}$$

12.3 Kernel PCA

이전에 SVM을 공부했을 때의 kernel trick과 같은 맥락이다. nonlinear transformation $\phi(\mathbf{x})$으로 M차원의 새로운 feature space에서 PCA를 하는 것이다. 이를 통해 nonlinear한 차원 특성을 파악할 수 있다. (구글에 예시그림을 찾아보면 이해가 더 쉬울 것이다.) 이제 그 과정을 간단히 살펴보자. $\sum_n \phi(\mathbf{X}_n) = \mathbf{0}$라고 하자(zero mean).

- M * M covariance matrix

$$\mathbf{C} = \frac{1}{N}\sum_{n=1}^{N}\phi(\mathbf{x}_n)\phi(\mathbf{x}_n)^T$$

$$\mathbf{C}\mathbf{v}_i = \lambda_i\mathbf{v}_i$$

우리의 목표는 eigenvalue problem을 explicit하게 구하지 않고 kernel을 이용하는 것이다. PCA에서의 과정과 동일하지만 kernel을 이용하는 방향으로 접근한다. (어쨌든 eigenvector $\mathbf{v}_i$를 구하면 된다)

$$\frac{1}{N}\sum_{n=1}^{N}\phi(\mathbf{x}_n)\{\phi(\mathbf{x}_n)^T\mathbf{v}_i\} = \lambda_i\mathbf{v}_i$$

위 식에서 $\phi(\mathbf{x}_n)^T\mathbf{v}_i$가 스칼라가 되므로 vector $\mathbf{v}_i$는 아래와 같은 linear combination으로 나타낼 수 있다.

$$\mathbf{v}_i = \sum_{n=1}^{N} a_{in}\phi(\mathbf{x}_n)$$

이를 위의 식에 대입하면

$$\frac{1}{N}\sum_{n=1}^{N}\phi(\mathbf{x}_n)\phi(\mathbf{x}_n)^T\sum_{m=1}^{N}a_{im}\phi(\mathbf{x}_m) = \lambda_i\sum_{n=1}^{N}a_{in}\phi(\mathbf{x}_n)$$

kernel의 형태를 만들기 위해 양변에 $\phi(\mathbf{x}_l)^T$을 곱하자.

- $k(\mathbf{x}_n, \mathbf{x}_m) = \phi(\mathbf{x}_n)^T\phi(\mathbf{x}_m)$: kernel function

$$\frac{1}{N}\sum_{n=1}^{N}k(\mathbf{x}_l, \mathbf{x}_n)\sum_{m=1}^{N}a_{im}k(\mathbf{x}_l, \mathbf{x}_m) = \lambda_i\sum_{n=1}^{N}a_{in}k(\mathbf{x}_l, \mathbf{x}_n)$$

이를 matrix notation으로 나타내면

$$\mathbf{K}^2\mathbf{a}_i = \lambda_i N\mathbf{K}\mathbf{a}_i$$

아래와 같이 eigenvalue problem을 풀어서 $\mathbf{a}_i$ 값을 구할 수 있다. 이를 통해 우리가 구하고자 했던 $\mathbf{v}_i$를 kernel function으로 접근할 수 있게 된다.

$$\mathbf{K}\mathbf{a}_i = \lambda_i N\mathbf{a}_i$$

추가로 제약도 고려해야한다. eigenvector의 길이가 1인 제약식이 있다.

$$1 = \mathbf{v}_i^T\mathbf{v}_i = \mathbf{a}_i^T\mathbf{K}\mathbf{a}_i$$

이를 통해 우리는 최종적으로 point $\mathbf{x}$가 eigenvector i에 projection하는 값을 구할 수 있게 된다. 그것도 kernel function을 이용해서! 놀랍다.

$$y_i(\mathbf{x}) = \phi(\mathbf{x})^T\mathbf{v}_i = \sum_{n=1}^{N}a_{in}\phi(\mathbf{x})^T\phi(\mathbf{x}) = \sum_{n=1}^{N}a_{in}k(\mathbf{x}, \mathbf{x}_n)$$