PRML chapter7

1. Maximum Margin classifiers
   in chapter 6 we think about the kernel from all of the train data but in this chapter, we consider the portion of the train data
   - consider the classify problem using $y(\mathbf{x}) = \mathbf{w}^{\mathrm{T}}\phi(\mathbf{x}) + b$
   - margin is the minimmum distance between the point and the classification boundary (let margin be $y = 1, -1$)
   - support vector is a near data from the boundary (sometimes nearest)
   - solve $\arg\max\left\{\frac{1}{\|\mathbf{w}\|}\min_{\mathbf{n}}\left[t_n\left(\mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}\left(\mathbf{x}_n\right)+b\right)\right]\right\}$ in simple shape
   - doing scale transformation using $t_n\left(\mathbf{w}^T\phi\left(\mathbf{x}_n\right)+b\right) = 1$, we can make above formula

     $$\arg\min\frac{1}{2}\|\mathbf{w}\|^2$$

   - Optimization problem of Lagrangian function

     $$L(\mathbf{w}, b, \mathbf{a}) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{n=1}^{N} a_n\left\{t_n\left(\mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}\left(\mathbf{x}_n\right)+b\right)-1\right\}$$

   - dedicate it with respect to b and $\mathbf{w}$, we gain maximize problem

     $$\tilde{L}(\mathbf{a}) = \sum_{n=1}^{N} a_n - \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N} a_n a_m t_n t_m k\left(\mathbf{x}_n, \mathbf{x}_m\right)$$

     with respect to a
   - The first equation is written as this shape

     $$y(\mathbf{x}) = \sum_{n=1}^{N} a_n t_n k\left(\mathbf{x}, \mathbf{x}_n\right) + b$$

   - from KKT,

     $$a_n 0 t_n y\left(\mathbf{x}_n\right) - 10 a_n\left\{t_n y\left(\mathbf{x}_n\right) - 1\right\} = 0$$

   - when $a_n = 0$, since $\{t_n y\left(\mathbf{x}_n\right) - 1\} \neq 0$, which means it does not affect the prediction
   - when a point satisfies $a_n \neq 0$, it is called support vector
   - After solving the quadratic programming problem and calculate $\mathbf{a}$, we gain

     $$b = \frac{1}{N_S}\sum_{n\in S}\left(t_n - \sum_{m\in\mathcal{S}} a_m t_m k\left(\mathbf{x}_n, \mathbf{x}_m\right)\right)$$

     using it

- The boundary made from SVM is only depend on the support vector

(a) overlapping class distributions

- permit some misclasses (In the bove discussion, we consider only the data that is linearly separable in the feature space)
- define discrimination function as $t_n y(\mathbf{x}_n) \geq 1 - \xi_n$
- minimize $C \sum_{n=1}^{N} \xi_n + \frac{1}{2}\|\mathbf{w}\|^2$, under C is a penalty
- the lagrangian function is

$$L(\mathbf{w}, b, \xi, \mathbf{a}, \mu) = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{n=1}^{N} \xi_n - \sum_{n=1}^{N} a_n \{t_n y(\mathbf{x}_n) - 1 + \xi_n\} - \sum_{n=1}^{N} \mu_n \xi_n,$$

  then consider KKT conditions

- same flow as above. first calculate lagrangian function, then using KKT, we gain lagrangian function under dual representations. Then we solve quadratic programming problem
- SVM can do stochastic prediction (Platt, 2000)

(b) relation to logistic regression

- define objective function as

$$\sum_{n=1}^{N} E_{\mathrm{SV}}(y_n t_n) + \lambda\|\mathbf{w}\|^2$$

$$(E_{\mathrm{SV}}(y_n t_n) = [1 - y_n t_n]_+)$$

  , and compare with logistic regression

(c) Multiclass SVMs

- SVM can append to multiclass classification problem
- In this book some algorithms are introduced qualitatively

(d) SVMs for regression

- in order to get the sparse solution, we replace square error function with ϵ-insensitive error function, which is expressed as $E_e(y(\mathbf{x}) - t) = \begin{cases} 0 \\ |y(\mathbf{x}) - t| - \epsilon \end{cases}$
- As before we re-express the optimization problem by introducing slack variables
- regularizes erroe function is

$$C\sum_{n=1}^{N} E_\epsilon(y(\mathbf{x}_n) - t_n) + \frac{1}{2}\|\mathbf{w}\|^2$$

  and re-express as

$$C\sum_{n=1}^{N}\left(\xi_n + \widehat{\xi}_n\right) + \frac{1}{2}\|\mathbf{w}\|^2$$

- lagrangian function is

$$L = C\sum_{n=1}^{N}\left(\xi_n + \widehat{\xi}_n\right) + \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{n=1}^{N}\left(\mu_n \xi_n + \hat{\mu}_n \hat{\xi}_n\right) - \sum_{n=1}^{N} a_n(\epsilon + \xi_n + y_n - t_n) - \sum_{n=1}^{N} \widehat{a}_n\left(\epsilon + \widehat{\xi}_n - y_n + t_n\right)$$

- from above, we gain

$$\widetilde{L}(\mathbf{a},\widehat{\mathbf{a}})=-\frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N}\left(a_n-\widehat{a}_n\right)\left(a_m-\widehat{a}_m\right)k\left(\mathbf{x}_n,\mathbf{x}_m\right)-\epsilon\sum_{n=1}^{N}\left(a_n+\widehat{a}_n\right)+\sum_{n=1}^{N}\left(a_n-\widehat{a}_n\right)t_n$$

  and optimize it with respect to a_n and $\widehat{a}_n$
- $y(\mathbf{x})=\sum_{n=1}^{N}\left(a_n-\widehat{a}_n\right)k\left(\mathbf{x},\mathbf{x}_n\right)+b$ represents the predicted value
- the points out of the ϵ tube are the support vectors

(e) Computational learning theory

- PAC is a learning framework that tells us how much data we need for learning and calculate the time for learning

2. Relevance vector machines

- revise SVM using the bayesian technique

(a) RVM for regression

- RVM model is

$$y(\mathbf{x})=\sum_{n=1}^{N}w_n k\left(\mathbf{x},\mathbf{x}_n\right)+b$$

- likelihood function is

$$p(\mathbf{t}|\mathbf{X},\mathbf{w},\beta)=\prod_{n=1}^{N}p\left(t_n|\mathbf{x}_n,\mathbf{w},\beta\right)$$

- weight prior takes the form of

$$p(\mathbf{w}|\boldsymbol{\alpha})=\prod_{i=1}^{M}\mathcal{N}\left(w_i|0,\alpha_i^{-1}\right)$$

  , which enables most of the weight parameters to be zero. We could gain sparse model
- posterior distribution for the weights

$$p(\mathbf{w}|\mathbf{t},\mathbf{X},\boldsymbol{\alpha},\beta)=\mathcal{N}(\mathbf{w}|\mathbf{m},\boldsymbol{\Sigma})$$
$$\mathbf{m}=\beta\boldsymbol{\Sigma}\Phi^{\mathrm{T}}\mathbf{t}$$
$$\boldsymbol{\Sigma}=\left(\mathbf{A}+\beta\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi}\right)^{-1}$$

- the values of α and β are determined evidence approximation. We maximize with respect to α and β

$$p(\mathbf{t}|\mathbf{X},\boldsymbol{\alpha},\beta)=\int p(\mathbf{t}|\mathbf{X},\mathbf{w},\beta)p(\mathbf{w}|\boldsymbol{\alpha})\mathrm{d}\mathbf{w}$$

- After we gain hyper parameter a^{*},β^{*}, the predictive distribution is

$$p(t|\mathbf{x},\mathbf{X},\mathbf{t},\mathbf{a}^{\star},\beta^{\star})=\int p(t|\mathbf{x},\mathbf{w},\beta^{\star})p(\mathbf{w}|\mathbf{X},\mathbf{t},\boldsymbol{\alpha}^{\star},\beta^{\star})\mathrm{d}\mathbf{w}=\mathcal{N}(t|\mathbf{m}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}),\sigma^2(\mathbf{x}))$$
$$\sigma^2(\mathbf{x})=\left(\beta^{*}\right)^{-1}+\phi(\mathbf{x})^{\mathrm{T}}\boldsymbol{\Sigma}\phi(\mathbf{x})$$

- RVM takes more time to learn than SVM

(b) Analysis of sparsity

- examine the reason why we could gain sparse solution in RVM

(c) RVM for regression