EECS 545 Homework 3 Solution

Naive Bayes

a.Implementation

Error rate is $0.00125 = 0.125\%$.

b.Proof

First, we want to prove that $\log P(c|x) = \log P(x) + \beta_c^T \phi(x)$. Suppose the bag-of-words feature vector is $x = (x_1, x_2, x_3, \cdots, x_D)$ where $x_d = 0$ indicates the d-th word doesn't appear in corresponding email, π_c is class priors for class c and class-conditional probabilities $\theta_{cd0} = P(x_d = 0|y = c), \theta_{cd1} = P(x_d = 1|y = c)$.

$$
\begin{aligned}
P(c|x) &= \frac{P(x|c)P(c)}{P(x)} \\
P(c|x) &= \frac{1}{P(x)} \pi_c \prod_{d=1}^{D} P(x_d|c) \\
\log P(c|x) &= -\log P(x) + \log \pi_c + \sum_{d=1} D \log P(x_d|c) \\
&= -\log P(x) + \log \pi_c + \log \pi_c + \sum_{d=1}^{D} \log P(X_d = x_d|c) \\
&= -\log P(x) + \log \pi_c + \log \pi_c + \sum_{d=1}^{D} \log(\theta_{cd0}^{1-x_d} \theta_{cd1}^{x_d}) \qquad (1) \\
&= -\log P(x) + 1 \cdot \log \pi_c + \sum_{d=1}^{D} [(1 - x_d) \log(1 - \theta_{cd1}) + x_d \log \theta_{cd1}] \\
&= -\log P(x) + 1 \cdot \log \pi_c + \sum_{d=1}^{D} \left[\log(1 - \theta_{cd1}) + x_d \log \frac{\theta_{cd1}}{1 - \theta_{cd1}} \right] \\
&= -\log P(x) + 1 \cdot (\log \pi_c + \sum_{d}^{D} \log(1 - \theta_{cd1})) + \sum_{d=1}^{D} x_d \frac{\theta_{cd1}}{1 - \theta_{cd1}} \\
&= -\log P(x) + \beta_c^T \phi(x)
\end{aligned}
$$

Let's c represent the target class we choose (spam or ham), $P(c|x) >= P(\bar{c}|x)$ is equivalent to $\beta_c^T \phi(x) >= \beta_{\bar{c}}^T \phi(x) \leftrightarrow (\beta_c - \beta_{\bar{c}})^T \phi(x) >= 0$.

It would be good exercise to extend the result from binary naive Bayes to multi-class naive Bayes.

Kernel

Techniques

Suppose kernels κ_1 and κ_2 define implicit mappings $\phi_1(x)$ and $\phi_2(x)$ such that:

$$\kappa_1(x, x') = \phi_1(x)^T \phi_1(x'), \kappa_2(x, x') = \phi_2(x)^T \phi_2(x')$$

1. Let $\phi(x) = \sqrt{a}\phi_1(x)$, the inner product for $\phi(x)$ and $\phi(x')$ is :

   $$\phi(x)^T \phi(x') = \sqrt{a}\phi_1(x)^T \sqrt{a}\phi_1(x') = a\kappa_1(x, x') = \kappa(x, x')$$

   . Therefore $a\kappa_1(x, x')$ is a valid kernel function.

2. Let $\phi(x) = \begin{bmatrix} \phi_1(x) \\ \phi_2(x) \end{bmatrix}$, the inner product for $\phi(x)$ and $\phi(x')$ is :

   $$\begin{aligned} \phi(x)^T \phi(x') &= \begin{bmatrix} \phi_1(x)^T, \phi_2(x)^T \end{bmatrix} \begin{bmatrix} \phi_1(x') \\ \phi_2(x') \end{bmatrix} \\ &= \phi_1(x)^T \phi_1(x') + \phi_2(x)^T \phi_2(x') = \kappa_1(x, x') + \kappa_2(x, x') \end{aligned}$$

   So $\kappa_1(x, x') + \kappa_2(x, x')$ is a valid kernel function.

3. Suppose $\phi_1(x) \in \mathbb{R}^n$ and $\phi_2(x) \in \mathbb{R}^m$, we can construct a n*m dimensional vector $\phi(x)$ that:

   $$\phi(x) = [\phi_1(x)^{(1)}\phi_2(x)^{(1)}, \phi_1(x)^{(1)}\phi_2(x)^{(2)} \cdot \phi_1(x)^{(1)}\phi_2(x)^{(m)}, \phi_1(x)^{(2)}\phi_2(x)^{(1)} \cdot \phi_1(x)^{(n)}\phi_2(x)^{(1)}]$$

   By this definition, the inner product is:

   $$\begin{aligned} \phi(x)^T \phi(x')^T &= \sum_{i=1}^{n} \sum_{j=1}^{m} \phi_1(x)^{(i)} \phi_2(x)^{(i)} \phi_1(x')^{(j)} \phi_2(x')^{(j)} \\ &= \sum_{i=1}^{n} \sum_{j=1}^{m} [\phi_1(x)^{(1)} \phi_1(x')^{(j)}] \cdot [\phi_2(x)^{(i)} \phi_2(x')^{(j)}] \\ &= [\sum_{i=1}^{n} \phi_1(x)^{(i)} \phi_1(x')^{(i)}] \cdot [\sum_{j=1}^{m} \phi_2(x)^{(j)} \phi_2(x')^{(j)}] \\ &= \kappa_1(x, x')\kappa_2(x, x') \end{aligned}$$

   This result can be extended to countable infinite dimensions. Using Mercer's theorem to prove is recommended, but for simplicity we only give the proof by constructing an implicit feature mapping. Please be careful not to confuse the kernel function $\kappa(\cdot, \cdot)$ with the Gram matrix $\mathcal{K}$ over a set of vectors, they are totally different.

4. Let $\phi(x) = [f(x)]$, and $\kappa(x, x') = f(x)f(x')$.

5. We do mathematical induction on $d \in \mathbb{N}$.

   - for $d = 1$, $\kappa(x, x') = \kappa_1(x, x')$ is a valid kernel function.

- for $d = k + 1$, because $\kappa_1(x, x')^k$ is a valid kernel by induction, use the third technique we proved above, $\kappa(x, x')^d = \kappa_1(x, x')^k \kappa(x, x')$ is a valid kernel.

6. We do mathematical induction on $d \in \mathbb{N}$ where d is the degree of the polynomial function p.
   - for $d = 0$, $\kappa(x, x') = a$ is a valid kernel function (a > 0 because coefficients is/are positive).
   - for $d = k+1$, we can write p into $p(x) = ax^d + p'(x)$ where $p'(x)$'s degree is k. Therefore $p'(\kappa_1(x, x'))$ is a valid kernel function. From the first and the fifth techniques we proved above, $a\kappa_1(x, x')^d$ is a valid kernel. Therefore $p(\kappa_1(x, x'))$ is a valid kernel.

Infinite Features

We will first prove Gaussian kernel is a valid kernel function, and therefore proves it has infinite.

valid kernel

We can simplify the Gaussian kernel at first:

$$\begin{aligned}\exp(-\frac{\|x - x'\|^2}{2\sigma^2}) &= \exp(-\frac{x \cdot x - 2x \cdot x' + x' \cdot x'}{2\sigma^2}) \\ &= \exp(-\frac{x \cdot x}{2\sigma^2}) \exp(-\frac{x' \cdot x'}{2\sigma^2}) \exp(\frac{x \cdot x'}{\sigma^2})\end{aligned}$$

From the fourth technique, we know $\exp(-\frac{x \cdot x}{2\sigma^2}) \cdot \exp(-\frac{x' \cdot x'}{2\sigma^2})$ is a valid kernel function. By the Taylor theorem, $\exp(\frac{x \cdot x'}{\sigma^2})$ can be written as:

$$\begin{aligned}\exp(\frac{x \cdot x'}{\sigma^2}) &= 1 + (\frac{x \cdot x'}{\sigma^2}) + \frac{1}{2!}(\frac{x \cdot x'}{\sigma^2})^2 + \cdots + \frac{1}{n!}(\frac{x \cdot x'}{\sigma^2})^n + \cdots \\ &= \sum_{k=0}^{\infty} \frac{1}{k!}(\frac{x \cdot x'}{\sigma^2})^k\end{aligned}$$

Because $k!$ increases rapidly, so the radius of convergence is $(-\infty, \infty)$ in terms of $x \cdot x'$, we can use the polynomial function $f_n(\frac{x \cdot x'}{\sigma^2}) = \sum_{k=0}^{n} \frac{1}{k!}(\frac{x \cdot x'}{\sigma^2})^k$ to approach. Therefore this function is a valid kernel function.

Infinity Features

We can use different function to generate features, but features may not be independent. Imagine we define an implicit mapping as $\begin{bmatrix} 1 \\ x \\ 7x + 3 \end{bmatrix}$, though the dimension of the vector is three, actually we only have two independent features cause $7x + 3$ can be represented as a linear combination of 1 and x. A mapping with infinite features actually refers to a mapping with infinite independent features.

It easy to proved that $1, x_i, x_i^2, x_i x_j, x_i^3, \cdots, x_i^n, x_i x_j^{(n-1)}, \cdots,$ are many independent features. According to previous derivation, we can represent the kernel $\exp(\frac{x \cdot x'}{\sigma^2})$ as $\sum_{k=0}^{\infty} \frac{1}{k!}(\frac{x \cdot x'}{\sigma^2})^k$. For the kernel function $(x \cdot x')^k$, the implicit feature vector would be like:

$$\phi(x) = [x_1 x_1 x_1 \cdots x_1, x_1 x_1 x_1 \cdots x_2, \ldots, x_1 x_1 x_2 \cdots, x_1 \ldots x_n x_n x_n, \cdots x_n]$$

each $x_1 x_1 x_1 \cdots x_1$ are of degree k. By this definition, we have:

$$\begin{aligned}
\phi(x)^T \phi(x') &= \sum_{d_1 d_2 d_3 \cdots d_k} x_{d_1} x_{d_2} x_{d_3} \cdots x_{d_k} \cdot x_{d_1} x_{d_2} x_{d_3} \cdots x_{d_k} \\
&= \sum_{d_1 d_2 d_3 \cdots d_k} \prod_{i=1}^{k} x_{d_i} x'_{d_i} \\
&= \prod_{i=1}^{k} (\sum_{j=1}^{s} x_j x'_j) \\
&= \prod_{i=1}^{k} (x \cdot x') \\
&= (x \cdot x')^k
\end{aligned}$$

Let's denote the implicit feature mapping defined by $(x \cdot x')^k$ as $\phi_k(x)$, we can generate a feature mapping $\phi(x)$ by concatenating those finite feature mappings $\phi_k(x)$ like:

$$\phi(x) = \sqrt{\frac{1}{0!\sigma^0}} \phi_0(x) \bigotimes \sqrt{\frac{1}{1!\sigma^2}} \phi_1(x) \bigotimes \sqrt{\frac{1}{2!\sigma^4}} \phi_2(x) \bigotimes \cdots$$

where $\bigotimes$ stands for the concatenate operation. In the end we need to put up with $\exp(-\frac{x \cdot x}{2\sigma^2})$, and get the final feature mapping for Gaussian kernels $\psi(x) = \exp(-\frac{x \cdot x}{2\sigma^2})\phi(x)$. Therefore we can conclude that an infinite subset of these features is independent, this feature mapping is an infinite feature mapping.

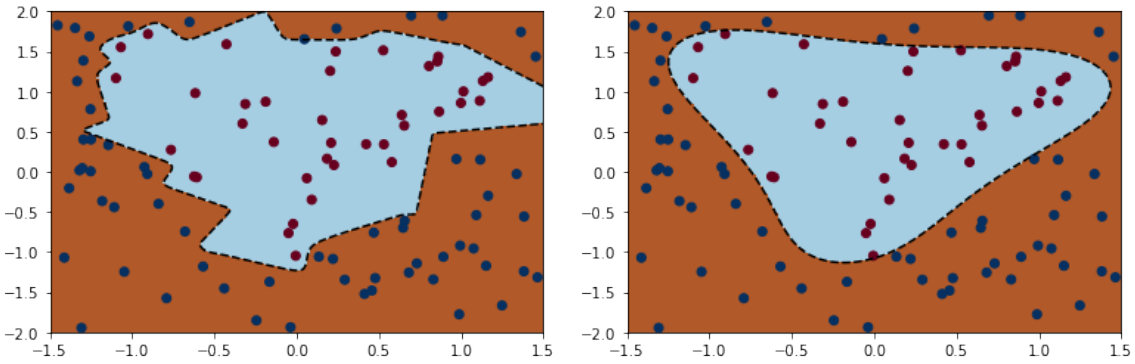
Kernel Percepetron

Description

We represent weight vector w as $\sum \alpha_i y_i x_i$. Therefore, the predictor $\hat{y} = sgn(w^T x) = sgn\left(\sum \alpha_i y_i \kappa(x_i, x)\right)$.

Data: Sample vectors $x_1 \cdots x_n$ and labels $y_1 \cdots y_n$
Result: coefficients $\alpha_1, \cdots \alpha_n$ that represent weight vector w
Initialize $\alpha_i = 0$;
for $d = 1$ **to** *max epoch* **do**
  for $i = 1$ **to** n **do**
    $\hat{y_i} = sgn\left(\sum_{j=1}^{n} \alpha_j y_j \kappa(x_i, x_j)\right)$;
    if $\hat{y_i} \cdot y_i <= 0$ **then**
      $\alpha_i \leftarrow \alpha_i + 1$;
    end
  end
end

Result

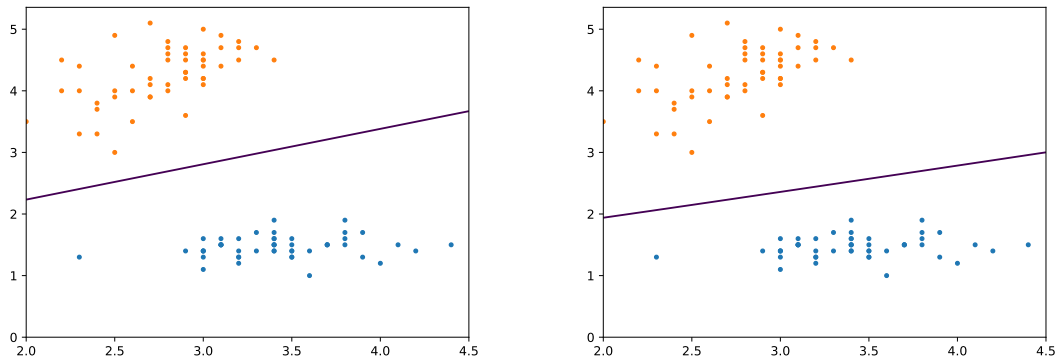


(a) $\sigma = 0.1$ (b) $\sigma = 1.0$

When σ^2 increase, the decision curve becomes smooth, because the covariance between two fixed points increases.

Another important attribute is, the Guassian kernel is very strong since its implicit mapping has infinite features. If there aren't two same points in two different class, Gaussian kernel perceptron is always guaranteed to find a curve to separate these two sets.

QDA and LDA



(a) *LDA* (b) *QDA*