

# From SVM to SMO and Random Feature Kernel Approximation

106033233 資工21 周聖謙

## 1. Abstract

In this article, I will derive SMO algorithm and the Fourier kernel approximation which are well-known algorithm for kernel machine. **SMO** can solve optimization problem of SVM efficiently and the **Fourier kernel approximation** is a kind of kernel approximation that can speed up the computation of the kernel matrix. In the last section, I will apply EDA on the dataset "Women's Clothing E-Commerce Review" and conduct a evaluation of my manual SVM.

## 2. Sequential Minimal Optimization(SMO)

The SMO(Sequential Minimal Optimization) algorithm is proposed from the paper **Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines** in 1998 by J. Platt. In short, SMO picks 2 variables  $\alpha_i, \alpha_j$  for every iteration, regulate them to satisfy KKT condition and, update them. In the following article, I will derive the whole algorithm and provide the evaluation on the simulation and real dataset.

We've known the dual problem of soft-SVM is

$$\begin{aligned} & \sup_{\alpha} \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j k(x_i, x_j) \\ & \text{subject to } 0 \leq \alpha_i \leq C, \sum_{i=1}^N \alpha_i y_i = 0 \end{aligned}$$

We also define the kernel.

$$k(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle$$

where  $\phi$  is an embedding function projecting the data points to a high dimensional space.

However, it's very hard to solve because we need to optimize  $N$  variables. As a result, J. Platt proposed SMO to solve this problem efficiently.

### 2.1 Notation

We denote the target function as  $\mathcal{L}_d(\alpha, C)$

$$\mathcal{L}_d(\alpha) = \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j k(x_i, x_j)$$

We also denote the kernel of  $x_1, x_2$  as  $K_{1,2} = k(x_1, x_2)$ .

### 2.2 Step 1. Update 2 Variable

First, we need to pick 2 variables to update in sequence, so we split the variables  $\alpha_1, \alpha_2$  from the summation.

$$\begin{aligned} \mathcal{L}_d(\alpha) &= \alpha_1 + \alpha_2 - \frac{1}{2} \alpha_1^2 y_1^2 K_{1,1} - \frac{1}{2} \alpha_2^2 y_2^2 K_{2,2} \\ &\quad - \frac{1}{2} \alpha_1 \alpha_2 y_1 y_2 K_{1,2} - \frac{1}{2} \alpha_2 \alpha_1 y_2 y_1 K_{2,1} \\ &\quad - \frac{1}{2} \alpha_1 y_1 \sum_{i=3}^N \alpha_i y_i K_{i,1} - \frac{1}{2} \alpha_1 y_1 \sum_{i=3}^N \alpha_i y_i K_{1,i} \\ &\quad - \frac{1}{2} \alpha_2 y_2 \sum_{i=3}^N \alpha_i y_i K_{i,2} - \frac{1}{2} \alpha_2 y_2 \sum_{i=3}^N \alpha_i y_i K_{2,i} \\ &\quad + \sum_{i=3}^N \alpha_i - \frac{1}{2} \sum_{i=3}^N \sum_{j=3}^N \alpha_i \alpha_j y_i y_j k(x_i, x_j) \\ &= \alpha_1 + \alpha_2 - \frac{1}{2} \alpha_1^2 y_1^2 K_{1,1} - \frac{1}{2} \alpha_2^2 y_2^2 K_{2,2} - \alpha_1 \alpha_2 y_1 y_2 K_{1,2} \\ &\quad - \alpha_1 y_1 \sum_{i=3}^N \alpha_i y_i K_{i,1} - \alpha_2 y_2 \sum_{i=3}^N \alpha_i y_i K_{i,2} + \text{Const} \\ &= \alpha_1 + \alpha_2 - \frac{1}{2} \alpha_1^2 K_{1,1} - \frac{1}{2} \alpha_2^2 K_{2,2} - \alpha_1 \alpha_2 y_1 y_2 K_{1,2} \\ &\quad - \alpha_1 y_1 \sum_{i=3}^N \alpha_i y_i K_{i,1} - \alpha_2 y_2 \sum_{i=3}^N \alpha_i y_i K_{i,2} + \text{Const} \end{aligned}$$

where  $\text{Const} = \sum_{i=3}^N \alpha_i - \frac{1}{2} \sum_{i=3}^N \sum_{j=3}^N \alpha_i \alpha_j y_i y_j k(x_i, x_j)$ . We see it as a constant because it is regardless to  $\alpha_1, \alpha_2$ .

#### 2.2.1 The Relation Between The Update Values and The Hyperplane

We've derive the partial derivative of the dual problem.

$$\frac{\partial \mathcal{L}(w, b, \xi, \alpha, \mu)}{\alpha_{...}} = w - \sum_{i=1}^N \alpha_i y_i x_i = 0$$

We can get

$$w = \sum_{i=1}^N a_i y_i x_i$$

Thus, we can rewrite the hyperplane  $f_\phi(x)$  with kernel.

$$f_\phi(x) = w^\top \phi(x) + b = b + \sum_{i=1}^N a_i y_i k(x_i, x)$$

We also denote  $v_1, v_2$  as

$$\begin{aligned} v_1 &= \sum_{i=3}^N a_i y_i K_{i,1} = \sum_{i=1}^N a_i y_i k(x_i, x_1) - a_1^{\text{old}} y_1 k(x_1, x_1) - a_2^{\text{old}} y_2 k(x_2, x_1) \\ &= f_\phi(x_1) - b - a_1^{\text{old}} y_1 K_{1,1} - a_2^{\text{old}} y_2 K_{2,1} \end{aligned}$$

and  $v_2$  is similar.

$$\begin{aligned} v_2 &= \sum_{i=3}^N a_i y_i K_{i,2} = \sum_{i=1}^N a_i y_i k(x_i, x_2) - a_1^{\text{old}} y_1 k(x_1, x_2) - a_2^{\text{old}} y_2 k(x_2, x_2) \\ &= f_\phi(x_2) - b - a_1^{\text{old}} y_1 K_{1,2} - a_2^{\text{old}} y_2 K_{2,2} \end{aligned}$$

where  $a_1^{\text{old}}$  and  $a_2^{\text{old}}$  are  $a_1$  and  $a_2$  of the previous iteration. Since we see  $a_i, i \geq 3$  as constant,  $a_i$  shouldn't depends on update variables  $a_1, a_2$ .

## 2.2.2 Rewrite The Complementary Slackness

The constraint can be represented as

$$\begin{aligned} \sum_{i=1}^N a_i y_i &= a_1 y_1 + a_2 y_2 + \sum_{i=3}^N a_i y_i = 0 \\ a_1 y_1 + a_2 y_2 &= - \sum_{i=3}^N a_i y_i = \zeta \\ a_1 &= \frac{\zeta - a_2 y_2}{y_1} \end{aligned}$$

Since  $y_1$  is either 1 or -1, thus

$$a_1 = \zeta y_1 - a_2 y_1 y_2$$

The old ones are the same.

$$a_1^{\text{old}} = \zeta y_1 - a_2^{\text{old}} y_1 y_2$$

Replace the symbol  $a_1, v_1, v_2$

$$\begin{aligned} \mathcal{L}_d(a) &= (\zeta y_1 - a_2 y_1 y_2) + a_2 \\ &- \frac{1}{2}(\zeta y_1 - a_2 y_1 y_2)^2 K_{1,1} - \frac{1}{2} a_2^2 K_{2,2} - (\zeta y_1 - a_2 y_1 y_2) a_2 y_1 y_2 K_{1,2} \\ &- (\zeta y_1 - a_2 y_1 y_2) y_1 v_1 - a_2 y_2 v_2 \\ &= (\zeta y_1 - a_2 y_1 y_2) + a_2 \\ &- \frac{1}{2}(\zeta^2 + a_2^2 - 2\zeta a_2 y_2) K_{1,1} - \frac{1}{2} a_2^2 K_{2,2} - (\zeta a_2 y_2 - a_2^2) K_{1,2} \\ &- (\zeta - a_2 y_2) v_1 - a_2 y_2 v_2 \end{aligned}$$

## 2.2.3 Combine the $v_1, v_2$ and $\zeta$

$$\begin{aligned} v_1 - v_2 &= [f_\phi(x_1) - b - a_1^{\text{old}} y_1 K_{1,1} - a_2^{\text{old}} y_2 K_{2,1}] - [f_\phi(x_2) - b - a_1^{\text{old}} y_1 K_{1,2} - a_2^{\text{old}} y_2 K_{2,2}] \\ &= [f_\phi(x_1) - b - (\zeta y_1 - a_2^{\text{old}} y_1 y_2) y_1 K_{1,1} - a_2^{\text{old}} y_2 K_{2,1}] - [f_\phi(x_2) - b - (\zeta y_1 - a_2^{\text{old}} y_1 y_2) y_1 K_{1,2} - a_2^{\text{old}} y_2 K_{2,2}] \\ &= [f_\phi(x_1) - f_\phi(x_2)] + [-(\zeta - a_2^{\text{old}} y_2) K_{1,1} - a_2^{\text{old}} y_2 K_{2,1}] - [-(\zeta - a_2^{\text{old}} y_2) K_{1,2} - a_2^{\text{old}} y_2 K_{2,2}] \\ &= [f_\phi(x_1) - f_\phi(x_2)] + [-\zeta K_{1,1} + a_2^{\text{old}} y_2 K_{1,1} - a_2^{\text{old}} y_2 K_{2,1}] - [-\zeta K_{1,2} + a_2^{\text{old}} y_2 K_{1,2} - a_2^{\text{old}} y_2 K_{2,2}] \\ &= f_\phi(x_1) - f_\phi(x_2) - \zeta K_{1,1} + \zeta K_{1,2} + (K_{1,1} + K_{2,2} - 2K_{1,2}) a_2^{\text{old}} y_2 \end{aligned}$$

## 2.2.4 Derive Gradient of $a_2$

$$\begin{aligned} \frac{\partial \mathcal{L}_d(a)}{\partial a_2} &= -y_1 y_2 + 1 - \frac{1}{2}(2a_2 - 2\zeta y_2) K_{1,1} - a_2 K_{2,2} - (\zeta y_2 - 2a_2) K_{1,2} - (-y_2) v_1 - y_2 v_2 \\ &= (-a_2 K_{1,1} - a_2 K_{2,2} + 2a_2 K_{1,2}) + \zeta y_2 K_{1,1} - \zeta y_2 K_{1,2} - y_1 y_2 + y_2 v_1 - y_2 v_2 + 1 \\ &= -a_2 (K_{1,1} + K_{2,2} - 2K_{1,2}) + \zeta y_2 K_{1,1} - \zeta y_2 K_{1,2} - y_1 y_2 + y_2 (v_1 - v_2) + 1 \end{aligned}$$

Replace  $v_1 - v_2$  containing old  $a_1^{\text{old}}, a_2^{\text{old}}$  (derived in 2.2.3)

$$\begin{aligned}\frac{\partial \mathcal{L}_d(\alpha)}{\partial \alpha_2} &= -\alpha_2(K_{1,1} + K_{2,2} - 2K_{1,2}) + \zeta y_2 K_{1,1} - \zeta y_2 K_{1,2} - y_1 y_2 + y_2 [f_\phi(x_1) - f_\phi(x_2) - \zeta K_{1,1} + \zeta K_{1,2} + (K_{1,1} + K_{2,2} - 2K_{1,2})\alpha_2^{\text{old}} y_2] + 1 \\ &= -(K_{1,1} + K_{2,2} - 2K_{1,2})\alpha_2 + (K_{1,1} + K_{2,2} - 2K_{1,2})\alpha_2^{\text{old}} + y_2(f_\phi(x_1) - f_\phi(x_2)) + y_2 - y_1\end{aligned}$$

Let  $\eta$  and  $E_i$  be

$$\begin{aligned}\eta &= K_{1,1} + K_{2,2} - 2K_{1,2}, \quad E_i = f_\phi(x_i) - y_i \\ \frac{\partial \mathcal{L}_d(\alpha)}{\partial \alpha_2} &= -\eta \alpha_2 + \eta \alpha_2^{\text{old}} + y_2(E_1 - E_2)\end{aligned}$$

Since we want to minimize the gradient, let the gradient be 0.

$$-\eta \alpha_2 + \eta \alpha_2^{\text{old}} + y_2(E_1 - E_2) = 0$$

Then we can find the relation between new and old  $\alpha_2$  as following

$$\alpha_2 = \alpha_2^{\text{old}} + \frac{y_2(E_1 - E_2)}{\eta}$$

To make the notation more clear to identify, we denote  $\alpha_2^{\text{new}}$  as the new value of the update.

$$\alpha_2^{\text{new}} = \alpha_2^{\text{old}} + \frac{y_2(E_1 - E_2)}{\eta}$$

## 2.3 Step 2. Clip with Box Constraint

The new values should satisfy the complementary slackness as

$$\alpha_1 y_1 + \alpha_2 y_2 = \zeta, \quad 0 \leq \alpha_i \leq C$$

Since  $y_1, y_2$  may have different labels, thus we consider 2 cases. The first case is  $y_1 \neq y_2$  as the left part of the figure 1 and another case is  $y_1 = y_2$  which corresponds to the right part of the figure.

Note that there is another line in quadrant 3 in the case 2 but it doesn't show in the figure due to the limit of the size.

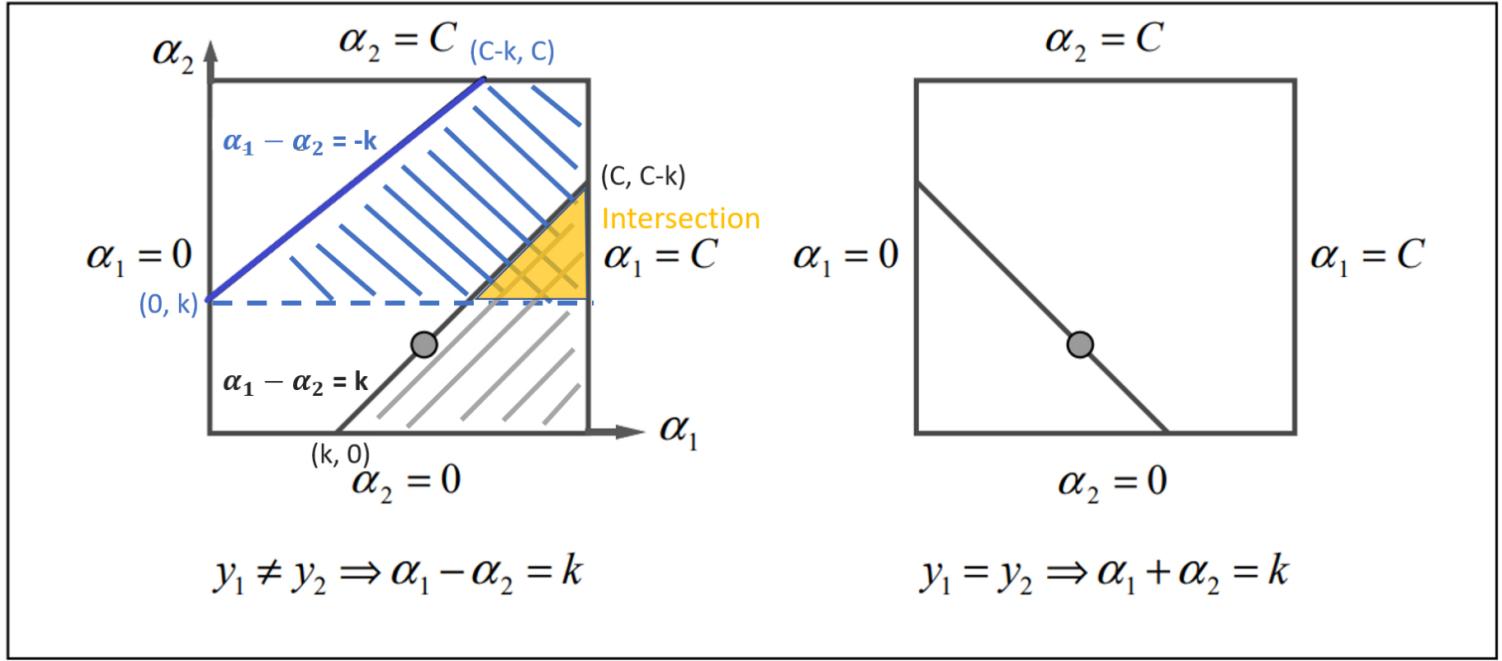


Figure 1

### 2.3.1 Case 1: Inequality

When  $y_1 \neq y_2$ , the equation is either  $\alpha_1 - \alpha_2 = k$  or  $\alpha_1 - \alpha_2 = -k$  where  $k = |\zeta|$  is a positive constant.

First, we consider the blue area  $\alpha_1 - \alpha_2 = -k$ . We can see  $\alpha_1 \in [C, k] = [C, \alpha_2 - \alpha_1]$ . The upper bound should be  $C$  and the lower bound should be  $\alpha_2 - \alpha_1$ .

$$B_U = C, B_L = \alpha_2 - \alpha_1$$

Next, we consider the grey area  $\alpha_1 - \alpha_2 = k$ . We can see  $\alpha_1 \in [0, C-k] = [0, C + \alpha_2 - \alpha_1]$ . The upper bound should be  $C + \alpha_2 - \alpha_1$  and the lower bound should be 0.

$$B_U = C + \alpha_2 - \alpha_1, B_L = 0$$

Combine 2 cases, both new and old values should satisfy the box constraint. The upper bound of  $\alpha_2^{\text{new}}$  can be written as

$$B_U = \min(C, C + \alpha_2^{\text{old}} - \alpha_1^{\text{old}})$$

and the lower bound is

$$B_L = \max(0, a_2^{\text{old}} - a_1^{\text{old}})$$

### 2.3.2 Case 2: Equality

When  $y_1 = y_2$ , the equation is either  $a_1 + a_2 = k$  or  $a_1 + a_2 = -k$  where  $k$  is a positive constant.

In similar way, we can derive the case of equality. The upper bound can be written as

$$B_U = \min(C, a_2^{\text{old}} + a_1^{\text{old}})$$

and the lower bound is

$$B_L = \max(0, a_2^{\text{old}} + a_1^{\text{old}} - C)$$

### 2.3.3 Clip The Value

According the bound we've derived, we need clip the updated variable  $a_2^{\text{new}}$  to satisfy the constraint. In addition, we denote the new value after clipping as  $a_2^*$ .

$$a_2^* = \text{CLIP}(a_2^{\text{new}}, B_L, B_U)$$

### 2.3.4 Update $\alpha_1$

We've know the complementary slackness.

$$a_1^* y_1 + a_2^* y_2 = a_1^{\text{old}} y_1 + a_2^{\text{old}} y_2 = \zeta$$

Move the updated value  $a_1^*$  to the left side and we can get

$$\begin{aligned} a_1^* &= \frac{a_1^{\text{old}} y_1 + a_2^{\text{old}} y_2 - a_2^* y_2}{y_1} \\ a_1^* &= a_1^{\text{old}} + y_1 y_2 (a_2^{\text{old}} - a_2^*) \end{aligned}$$

## 2.4 Step 3. Update Bias

The only equation that contains bias  $b$  is the function  $f_\phi(x) = b + \sum_{i=1}^N a_i y_i k(x_i, x)$ . When  $0 < a_i^* < C$ , it means that the data point  $x_i$  is right on the margin such that  $f_\phi(x) = y_i$ ,  $f_\phi^*(x_i) = y_i$  and the bias  $b_1^*, b_2^*$  can be derived directly. Note that for convenience,  $f_\phi^*(x_w) = \sum_{i=3}^N a_i y_i K_{i,w} - a_1^* y_1 K_{1,w} - a_2^* y_2 K_{2,w} + b^* = y_w$  contains updated variables  $a_2^*, a_2^*, b^*$ .

If  $0 < a_1^* < C$ , the data point  $x_1$  should right on the margin and  $f_\phi^*(x_1) = y_1$ . The bias derived from  $a_1$ .

$$\begin{aligned} b_1^* &= y_1 - \sum_{i=3}^N a_i y_i K_{i,1} - a_1^* y_1 K_{1,1} - a_2^* y_2 K_{2,1} \\ &= (y_1 - f_\phi(x_1) + a_1^{\text{old}} y_1 K_{1,1} + a_2^{\text{old}} y_2 K_{2,1} + b) - a_1^* y_1 K_{1,1} - a_2^* y_2 K_{2,1} \\ &= -E_1 - y_1 K_{1,1} (a_1^* - a_1^{\text{old}}) - y_2 K_{2,1} (a_2^* - a_2^{\text{old}}) + b \end{aligned}$$

If  $0 < a_2^* < C$ , the data point  $x_2$  should right on the margin and  $f_\phi^*(x_2) = y_2$ . The bias derived from  $a_2$ .

$$\begin{aligned} b_2^* &= y_2 - \sum_{i=3}^N a_i y_i K_{i,2} - a_1^* y_1 K_{1,2} - a_2^* y_2 K_{2,2} \\ &= (y_2 - f_\phi(x_2) + a_1^{\text{old}} y_1 K_{1,2} + a_2^{\text{old}} y_2 K_{2,2} + b) - a_1^* y_1 K_{1,2} - a_2^* y_2 K_{2,2} \\ &= -E_2 - y_1 K_{1,2} (a_1^* - a_1^{\text{old}}) - y_2 K_{2,2} (a_2^* - a_2^{\text{old}}) + b \end{aligned}$$

When the data point  $x_i, x_j$  are both not on the margin, we choose the average of  $b_1^*, b_2^*$  as the updated value.

$$b^* = \frac{b_1^* + b_2^*}{2}$$

For more detail, please see the pseudo code.

## 2.5 Pseudo Code

---

Given  $C$ , otherwise the default value is  $C = 5$

Given  $\epsilon$ , otherwise the default value is  $\epsilon = 10^{-6}$

Given max-iter, otherwise the default value is max-iter =  $10^3$

For all  $a_i = 0, 1 \leq i \leq N$

$b = 0$

move =  $\infty$

...  
...

while( $\text{move} > \epsilon$  and  $\text{iter} \leq \text{max-iter}$ ):

- $a_1^* = a_2^* = b^* = \text{move} = 0$
- for( $n \in N/2$ ):
  - Choose the index  $i, j$  from 1 to  $N$

$$\circ E_i = f(x_i) - y_i$$

$$\circ E_j = f(x_j) - y_j$$

$$\circ \eta = K_{i,i} + K_{j,j} - 2K_{i,j}$$

$$\circ a_j^{\text{new}} = a_j + \frac{y_i(E_i - E_j)}{\eta}$$

#### Bosk Constraint

- if( $y_i = y_j$ ):
  - $B_U = \min(C, a_j + a_i)$
  - $B_L = \max(0, a_j + a_i - C)$

◦ else:

- $B_U = \min(C, C + a_j - a_i)$
- $B_L = \max(0, a_j - a_i)$

$$\circ a_j^* = \text{CLIP}(a_j^{\text{new}}, B_L, B_U)$$

$$\circ a_i^* = a_i + y_i y_j (a_j - a_j^*)$$

#### Update Bias

$$\circ b_i^* = -E_i - y_i K_{i,i} (a_i^* - a_i) - y_j K_{j,i} (a_j^* - a_j) + b$$

$$\circ b_j^* = -E_j - y_i K_{i,j} (a_i^* - a_i) - y_j K_{j,j} (a_j^* - a_j) + b$$

◦ if( $0 \leq a_i \leq C$ ):

$$\quad ■ b^* = b_i^*$$

◦ else if( $0 \leq a_j \leq C$ ):

$$\quad ■ b^* = b_j^*$$

◦ else:

$$\quad ■ b^* = \frac{b_i^* + b_j^*}{2}$$

$$\circ \text{move} = \text{move} + |a_1^* - a_1| + |a_2^* - a_2| + |b^* - b|$$

$$\circ a_i = a_i^*, \quad a_j = a_j^*, \quad b = b^*$$

- iter = iter + 1

## 3. Fourier Kernel Approximation

The Fourier kernel approximation is proposed from the paper **Random Features for Large-Scale Kernel Machines** on NIPS'07. It's a widely-used approximation to accelerate the kernel computing especially for the high dimensional dataset. For a dataset with dimension  $D$  and data points  $N$ , the time complexity of computing the exact kernel is  $\mathcal{O}(DN^2)$  and the Fourier kernel approximation is  $\mathcal{O}(SN^3)$  with  $S$  samples. While the dimension goes up, the approximation remains the same computing time because it is regardless to the dimension of the dataset.

### 3.1 Bochner's Theorem

If  $\phi : \mathbb{R}^n \rightarrow \mathbb{C}$  is a positive definite, continuous, and satisfies  $\phi(0) = 1$ , then there is some Borel probability measure  $\mu \in \mathbb{R}^n$  such that  $\phi = \hat{\mu}$

Thus, we can extend the Bochner's theorem to kernel.

### 3.2 Theorem 1

According to Bochner's theorem, a continuous kernel  $k(x, y) = k(x - y) \in \mathbb{R}^d$  is positive definite if and only if  $k(\delta)$  is the Fourier transform of a non-negative measure.

If a shift-invariant kernel  $k(\delta)$  is a properly scaled, Bochner's theorem guarantees that its Fourier transform  $p(\omega)$  is a proper probability distribution. Defining  $\zeta_\omega(x) = e^{i\omega'x}$ , we have

$$k(x - y) = \int_{\omega} p(\omega) e^{i\omega'(x-y)} d\omega = E_{\omega}[\zeta_{\omega}(x)\zeta_{\omega}(y)]$$

where  $\zeta_{\omega}(x)\zeta_{\omega}(y)$  is an unbiased estimate of  $k(x, y)$  when  $\omega$  is drawn from  $p(\omega)$ .

With Monte-Carlo simulation, we can approximate the integral with the summation over the probability  $p(\omega)$ .

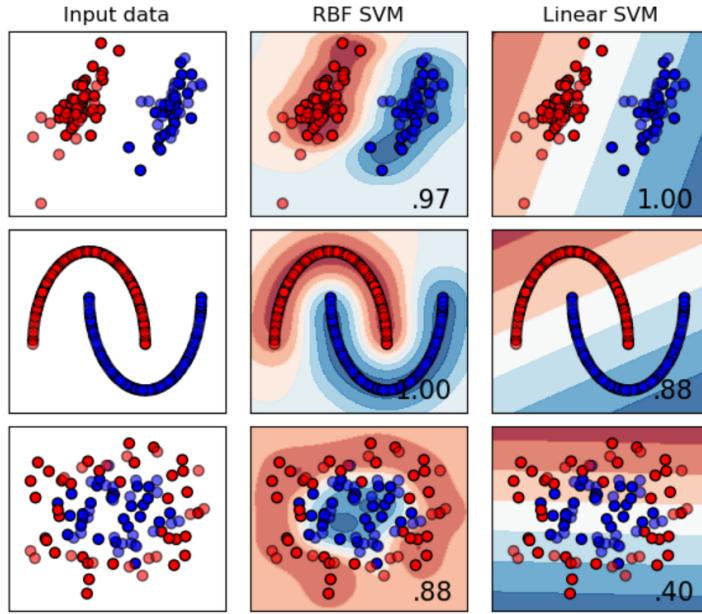
$$z(x)'z(y) = \frac{1}{D} \sum_{j=1}^D z_{w_j}(x)z_{w_j}(y)$$

$$z_{\omega}(x) = \sqrt{2}\cos(\omega x + b) \text{ where } \omega \sim p(\omega)$$

In order to approximate the RBF kernel  $k(x, y) = e^{-\frac{\|x-y\|^2}{2}}$ , we draw  $\omega$  from Fourier transformed distribution  $p(\omega) = \mathcal{N}(0, 1)$ .

## 4. Experiments

### 4.1 Simulation With Exact Kernel



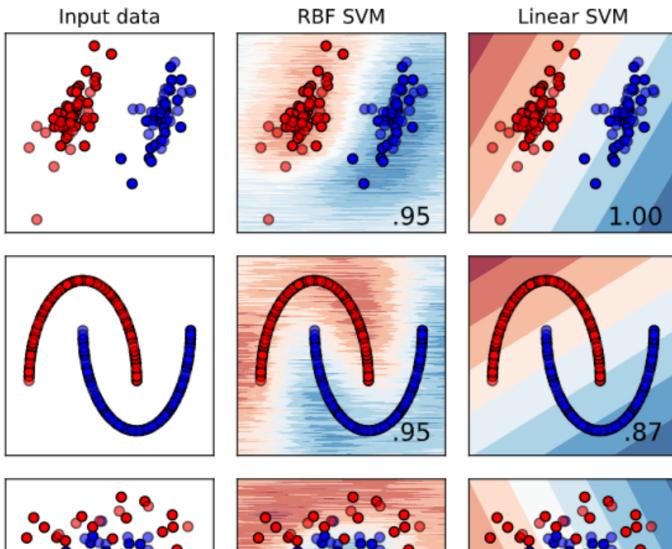
The parameters of SVM:

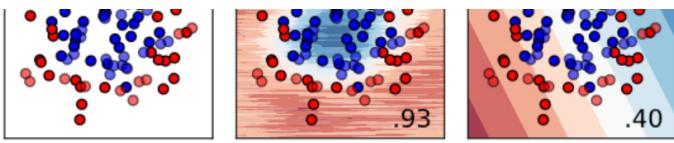
- C: 0.6
- $\gamma$  of RBF: 2

Here we generate 3 kinds of data. The first row is generated by a Gaussian mixture model. The second row is like a moon generated by Scikit-Learn package. The third one is also generated by Scikit-Learn package and the package generates 2 circles, one is in the inner side and the other one is in the outer side.

The SMO and kernel seem work properly even under noise and nonlinear dataset.

### 4.2 Simulation With Approximated Kernel



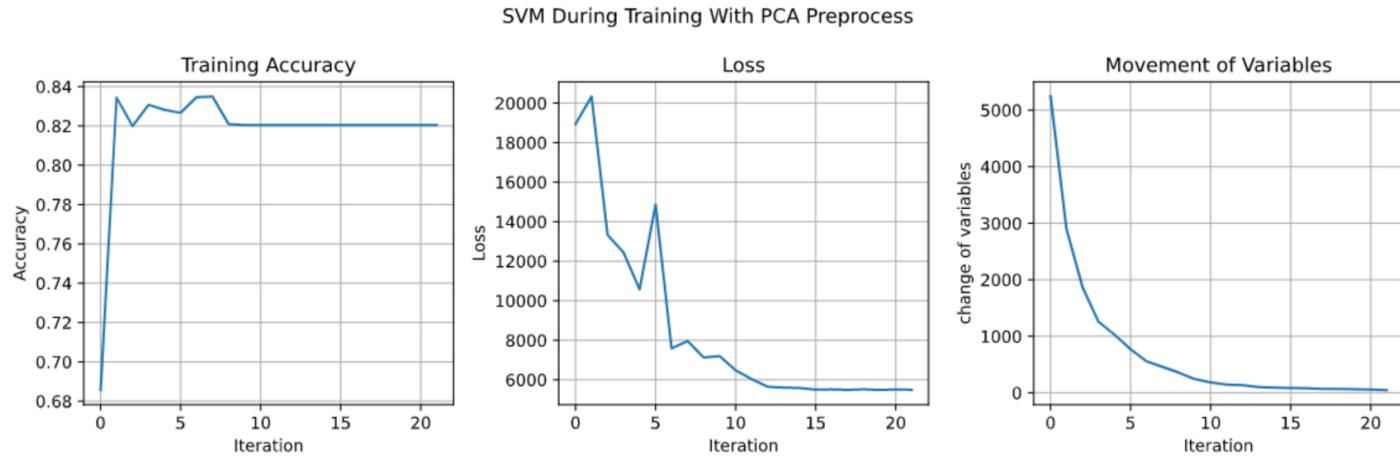


We draw 200 samples from  $p(\omega)$  to approximate the RBF kernel. As we can see, the testing accuracies are close to the ones of exact kernels in most of cases.

### 4.3 Real Dataset

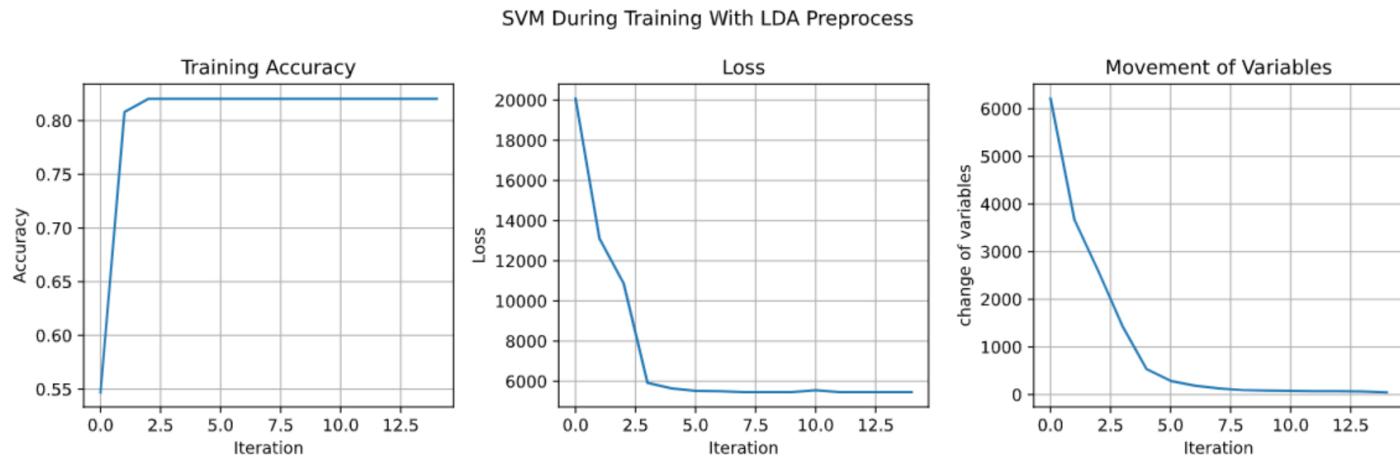
#### 4.3.1 PCA Preprocess

Apply SVM on the "Women's Clothing E-Commerce Review Dataset" with  $C = 0.6$  and  $\gamma$  of RBF kernel = 2, the training accuracy is 82.03% and the testing accuracy is 81.54%. The accuracy, loss and, the movement of variables are showed in the following graph.



As we can see, the movement of variable gets smaller during training and converge around 50 and the accuracy remains about 82%.

#### 4.3.2 LDA Preprocess



The training accuracy is also 82.03% and the testing accuracy is 81.54%, but the curves are smoother than the ones of PCA.

## 5. Data Analysis

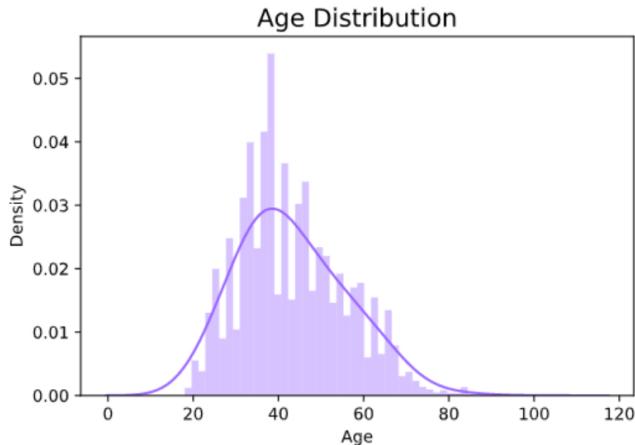
### 5.1 Overview

The dataset is called "Women's Clothing E-Commerce Review" which contains reviews written by customers for a online clothing shop. It has 9 features and each feature represents the meaning as the following table.

Features	Description
Clothing ID	Integer Categorical variable that refers to the specific piece being reviewed.
Age	Positive Integer variable of the reviewers age.
Title	String variable for the title of the review.
Review	String variable for the review body.

<b>Rating</b>	Positive Ordinal Integer variable for the product score granted by the customer from 1 Worst, to 5 Best.
<b>Recommended IND</b>	Binary variable stating where the customer recommends the product where 1 is recommended, 0 is not recommended.
<b>Positive Feedback Count</b>	Positive Integer documenting the number of other customers who found this review positive.
<b>Division Name</b>	Categorical name of the product high level division.
<b>Department Name</b>	Categorical name of the product department name.
<b>Class Name</b>	Categorical name of the product class name.

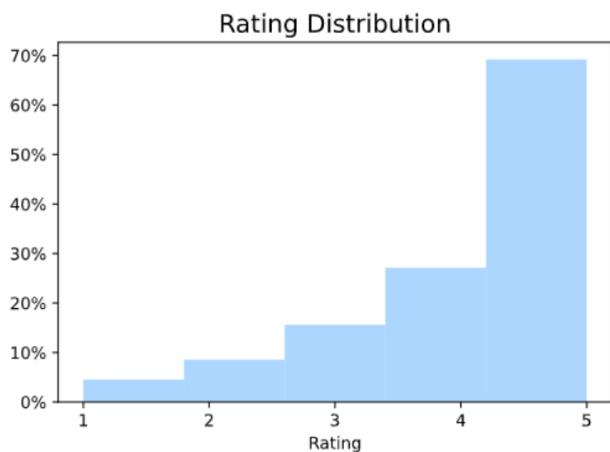
#### Age Distribution



As we can see, the peak of the age distribution is about 40. The population below 40 years old is a half of total users.

The average age of the customers buying "casual bottoms" is 26 which is much lower the average age of total customers 42.

#### Rating Distribution



According to the graph, most of the users(more than 50%) gives 5 points in their comments.

The average rating of all goods is 4.2 but the class "Trend" has only 3.8.

#### Topics

#### Topics in LDA model





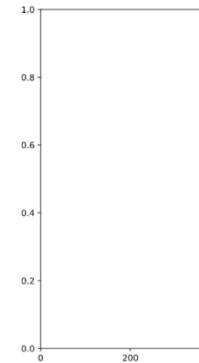
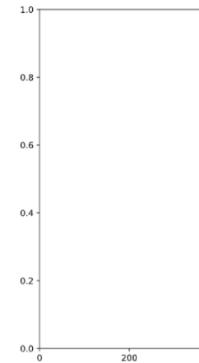
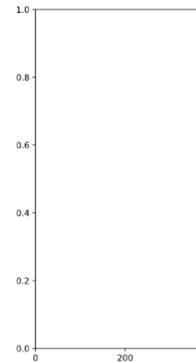
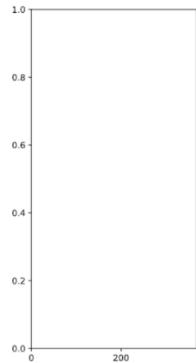
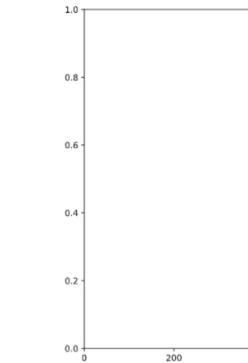
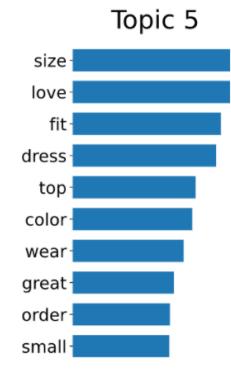
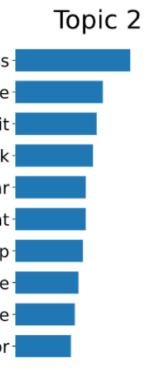
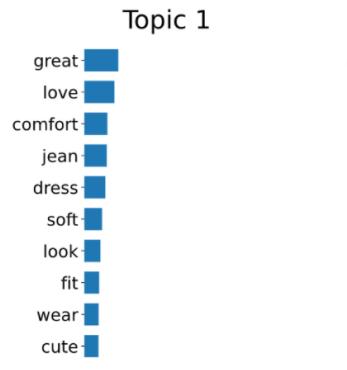
The graph is the result of LDA with 5 topics. Topics 3 and 4 seem interesting. Topic 3 seems related to boys and their girlfriend, since words like girlfriend, gift and, birthday appear in the top 10 words. We can infer that most purchases in this topic are the gifts for girlfriends by the boyfriends. Topic 4 seems also related to gift but not between lovers. The comments are mainly about receivers' compliment.

## 5.2 Rating

Rating Score 5



Topics in LDA model

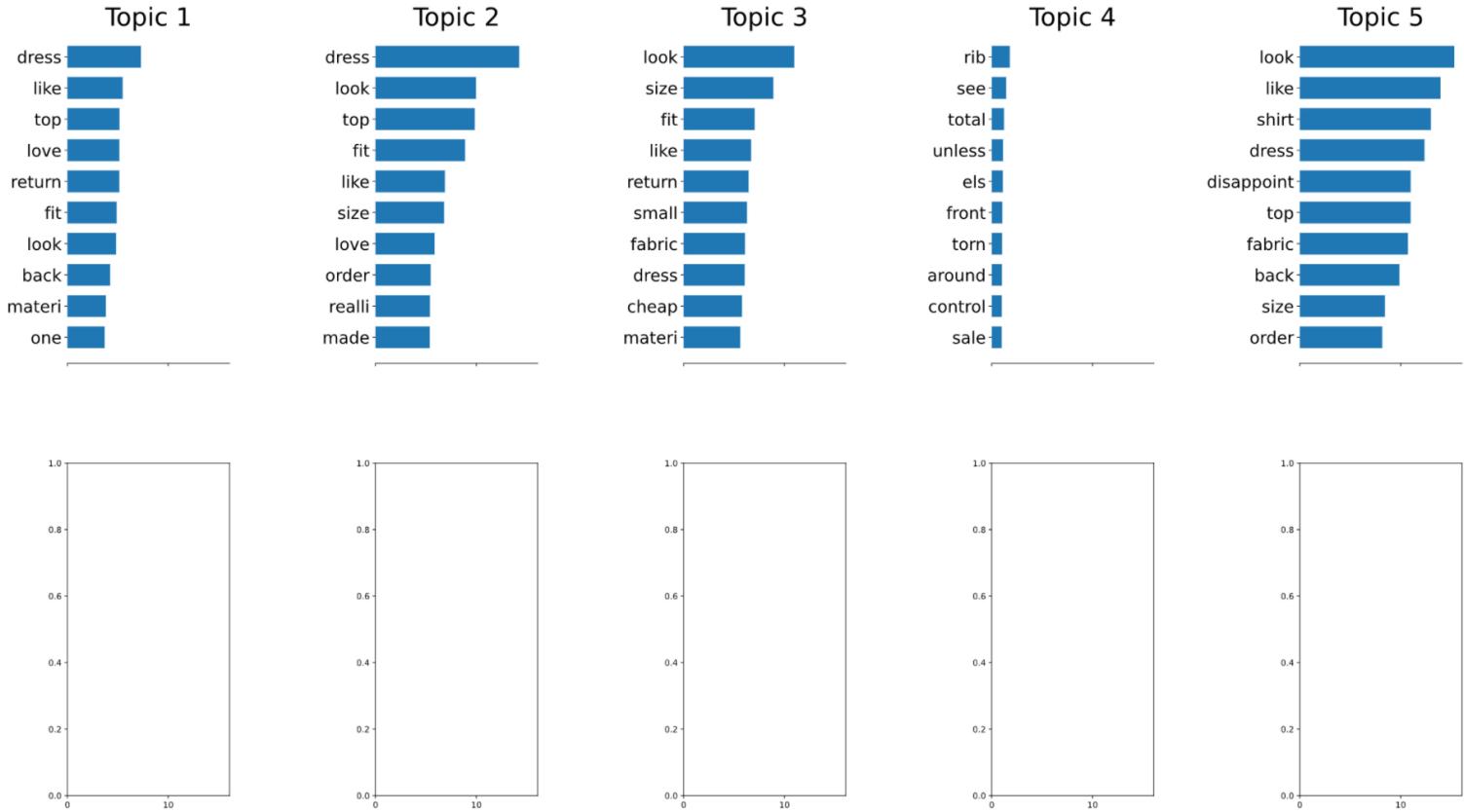


There are many positive words in the word cloud like perfect, comfort, style...

Rating Score 1



Topics in LDA model



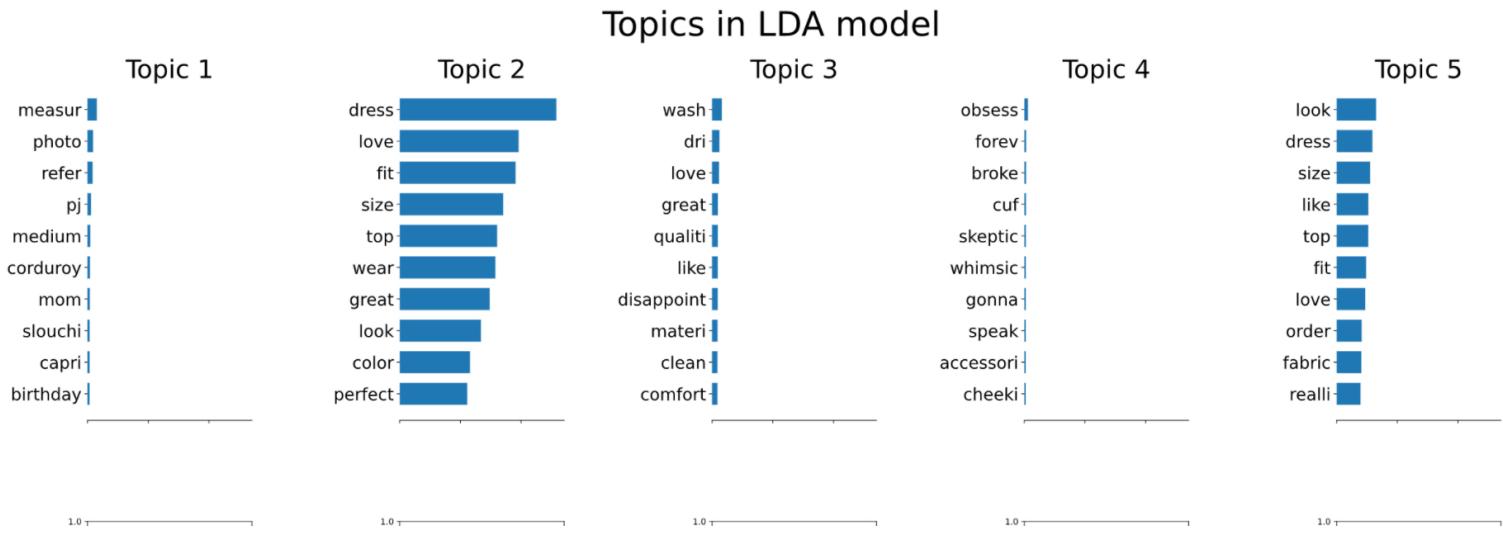
An interesting thing is that "cute" seems neutral since it appears in both side. It may be caused by cultural difference between western and eastern societies.

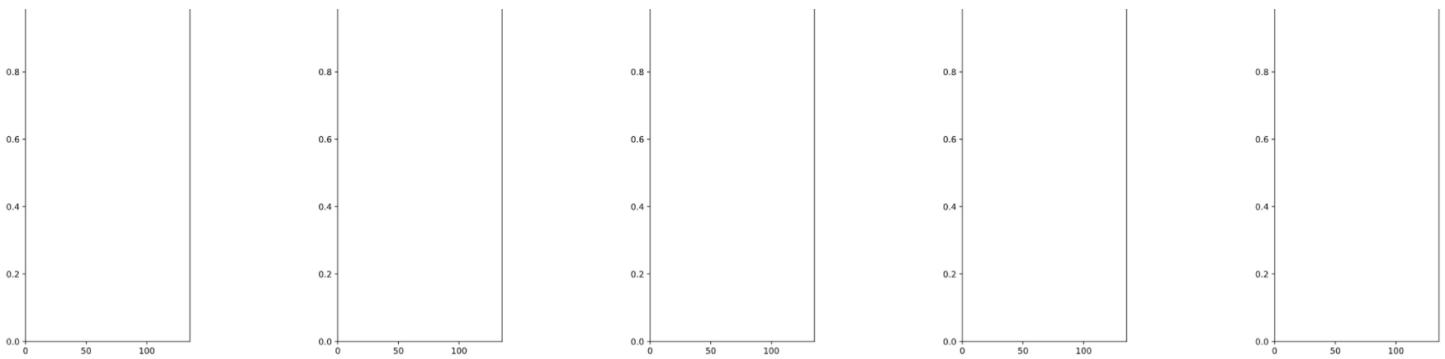
### 5.3 Ages

#### LDA

Love and cute are everywhere. Also, some positive words and common nouns appear in every ages. It's boring for me to focus on that common phenomenon. Here I just mention something interesting from the data.

#### Less Than 30

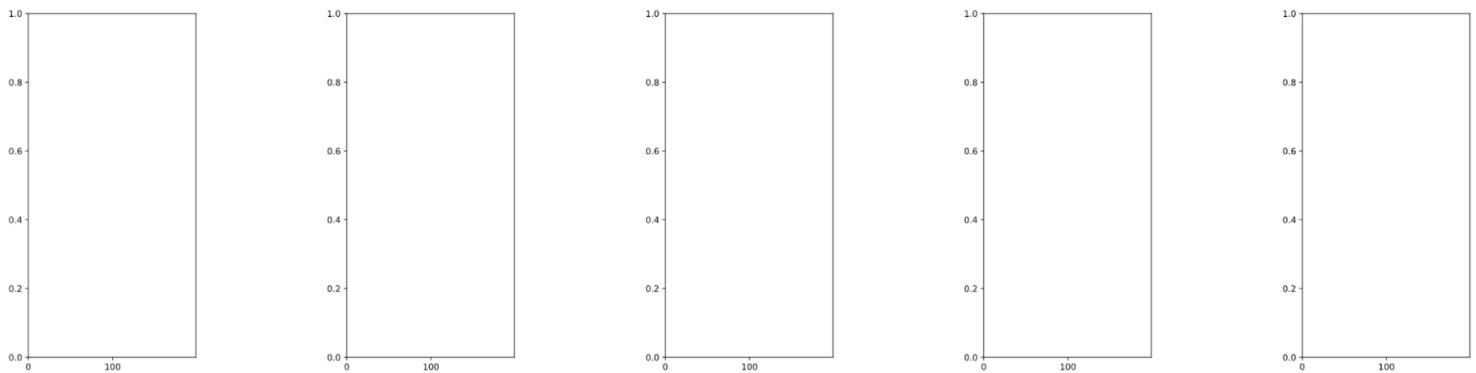
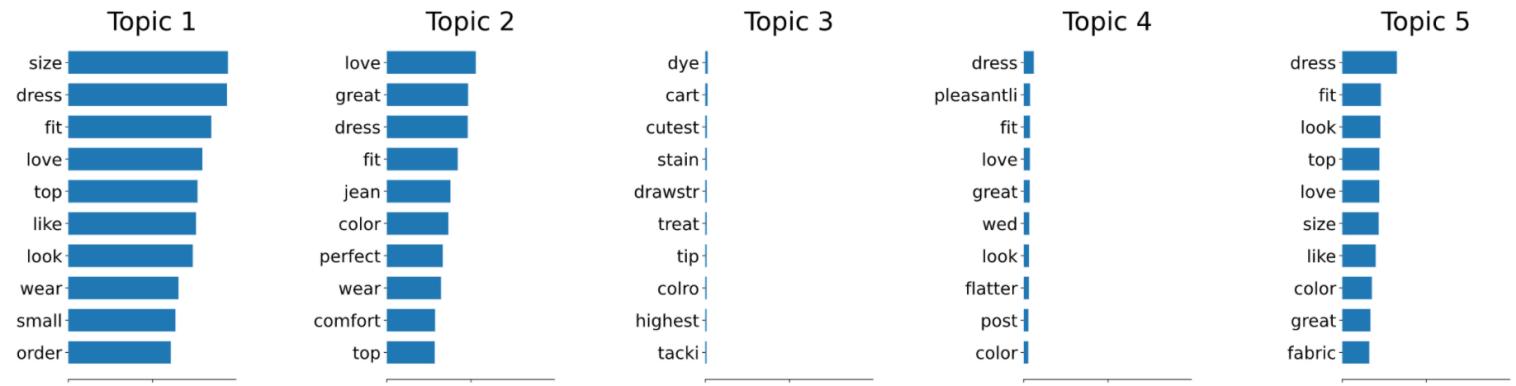




An interesting thing is that the topic 4 contains broke, cheeky, obsess and, forever. It seems that topic 4 is about the relationship. In addition topic 4 only appears under 30 years old.

30s

### Topics in LDA model

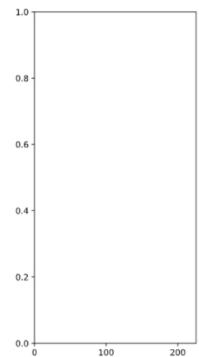
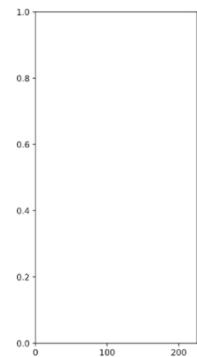
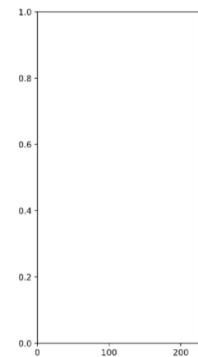
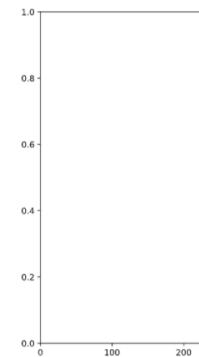
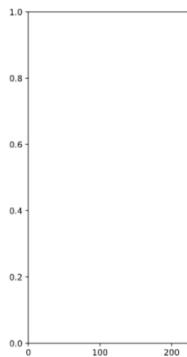


The topic 4 contains wed, love and, pleasant. It perhaps is about wedding.

40s

### Topics in LDA model

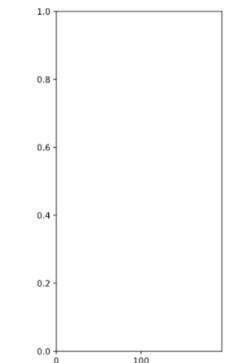
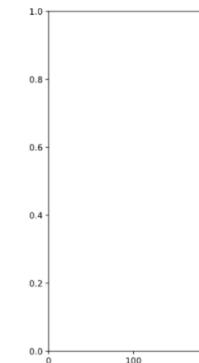
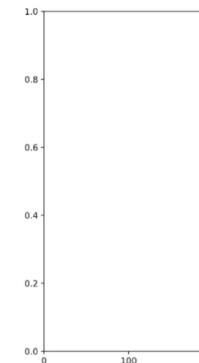
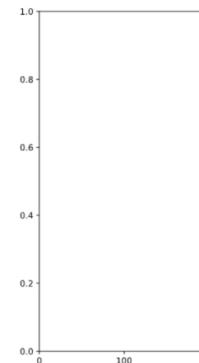
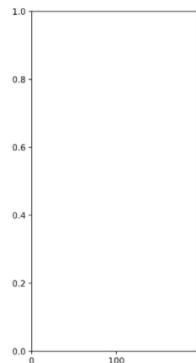
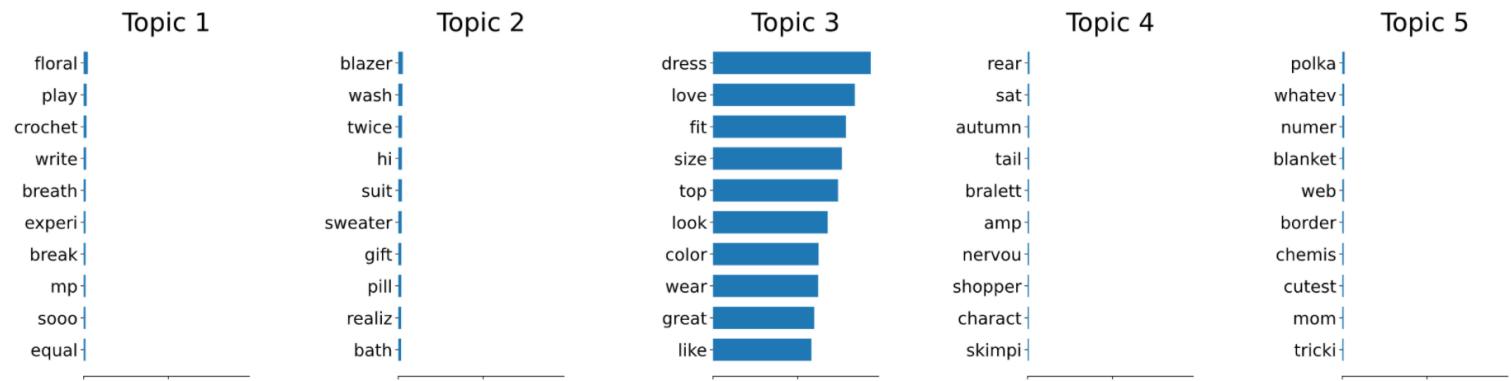




It's a little bit weird that one topic is about the "bikini" and the other one is about "tankini" which is a kind of swimsuit covering the whole body.

50s

### Topics in LDA model

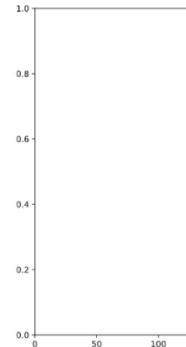
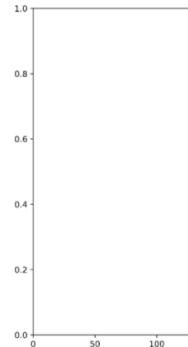
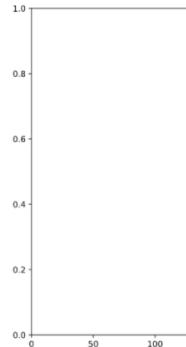
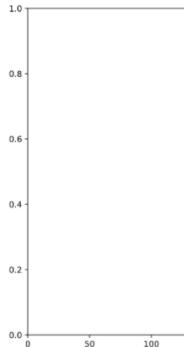
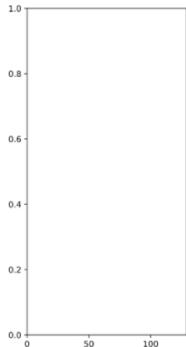


I don't know why there is a topic about "bralett" which means "sexy intimates". Does it mean the shirley valentine for the customers?

More Than 60

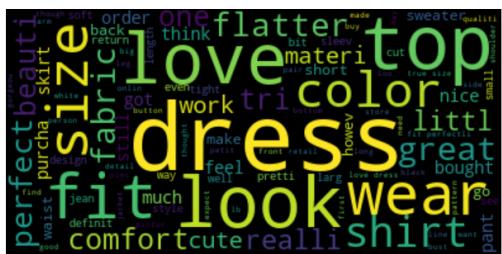
### Topics in LDA model





#### Word Cloud

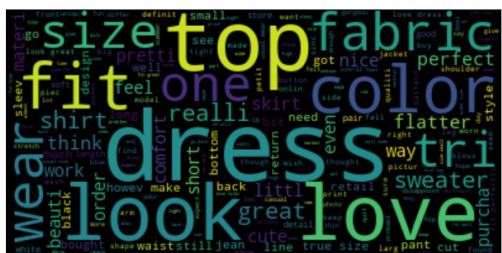
##### Less Than 30



30s



40s



50s



##### More Than 60



## 7. Conclusion

In this article, we've seen detailed derivation of SMO algorithm and the implementation of SVM. We've also conducted the evaluation on the simulated dataset and real dataset. On the other hand, we've reviewed the Fourier kernel approximation briefly and compared the approximation kernel with the exact kernel. Finally, we've also seen an EDA on the Women's E-Commerce Clothing Reviews dataset and had some interesting insights.

## 7. Reference

### SMO

- [Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines](#)
- [現代啟示錄 - Karush-Kuhn-Tucker \(KKT\) 條件](#)
- [現代啟示錄 - Lagrange 乘數法](#)
- [之乎 - 机器学习算法实践-SVM中的SMO算法](#)
- [之乎 - Python · SVM \( 四 \) · SMO 算法](#)
- [Machine Learning Techniques \(機器學習技法\)](#)

### Kernel Approximation

- [NIPS'07 - Random Features for Large-Scale Kernel Machines](#)
- [論文閱讀: Random Features for Large-Scale Kernel Machines](#)

### Dataset

- [Movie Review Data \(Binary Sentimental Analytics\)](#)
- [Kaggle - Text Classification using SpaCy \(with Amazon fine food reviews dataset: Binary Sentimental Analytics\)](#)
- [Examples of Data Sets for Text Analysis](#)
- [Kaggle - Text Classification Dataset](#)
- [Kaggle - Women's E-Commerce Clothing Reviews](#)