From SVM to SMO and Random Feature Kernel Approximation

106033233 資工21 周聖諺

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 α_i, α_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 he dual problem of soft-SVM is

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

We also define the kernel.

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

where ϕ 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 α_1, α_2 from the summation.

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

$$
\begin{aligned}
&= \alpha_1 + \alpha_2 - \frac{1}{2}\alpha_1^2 y_1^2 K_{1,1} - \frac{1}{2}\alpha_2^2 y_2^2 K_{2,2} - \alpha_1\alpha_2 y_1 y_2 K_{1,2} \\
&\quad - \alpha_1 y_1 \sum_{i=3}^{N} \alpha_i y_i K_{i,1} - \alpha_2 y_2 \sum_{i=3}^{N} \alpha_i y_i K_{i,2} + \mathcal{C}\text{onst} \\
&= \alpha_1 + \alpha_2 - \frac{1}{2}\alpha_1^2 K_{1,1} - \frac{1}{2}\alpha_2^2 K_{2,2} - \alpha_1\alpha_2 y_1 y_2 K_{1,2} \\
&\quad - \alpha_1 y_1 \sum_{i=3}^{N} \alpha_i y_i K_{i,1} - \alpha_2 y_2 \sum_{i=3}^{N} \alpha_i y_i K_{i,2} + \mathcal{C}\text{onst}
\end{aligned}
$$

where $\mathcal{C}\text{onst} = \sum_{i=3}^{N} \alpha_i - \frac{1}{2}\sum_{i=3}^{N}\sum_{j=3}^{N} \alpha_i\alpha_j y_i y_j k(x_i, x_j)$. We see it as a constant because it is regardless to α_1, α_2.

2.2.1 The Relation Between The Update Values and The Hyperplane

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

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

We can get

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

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

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

We also denote v_1, v_2 as

$$
\begin{aligned}
v_1 = \sum_{i=3}^{N} \alpha_i y_i K_{i,1} &= \sum_{i=1}^{N} \alpha_i y_i k(x_i, x_1) - \alpha_1^{old} y_1 k(x_1, x_1) - \alpha_2^{old} y_2 k(x_2, x_1) \\
&= f_\phi(x_1) - b - \alpha_1^{old} y_1 K_{1,1} - \alpha_2^{old} y_2 K_{2,1}
\end{aligned}
$$

and v_2 is similar.

$$
\begin{aligned}
v_2 = \sum_{i=3}^{N} \alpha_i y_i K_{i,2} &= \sum_{i=1}^{N} \alpha_i y_i k(x_i, x_2) - \alpha_1^{old} y_1 k(x_1, x_2) - \alpha_2^{old} y_2 k(x_2, x_2) \\
&= f_\phi(x_2) - b - \alpha_1^{old} y_1 K_{1,2} - \alpha_2^{old} y_2 K_{2,2}
\end{aligned}
$$

where α_1^{old} and α_2^{old} are α_1 and α_2 of the previous iteration. Since we see $\alpha_i, i \geq 3$ as constant, α_i shouldn't depends on update variables α_1, α_2.

2.2.2 Rewrite The Complementary Slackness

The constraint can be represented as

$$\sum_{i=1}^{N} \alpha_i y_i = \alpha_1 y_1 + \alpha_2 y_2 + \sum_{i=3}^{N} \alpha_i y_i = 0$$

$$\alpha_1 y_1 + \alpha_2 y_2 = -\sum_{i=3}^{N} \alpha_i y_i = \zeta$$

$$\alpha_1 = \frac{\zeta - \alpha_2 y_2}{y_1}$$

Since y_1 is either 1 or -1, thus

$$\alpha_1 = \zeta y_1 - \alpha_2 y_1 y_2$$

The old ones are the same.

$$\alpha_1^{old} = \zeta y_1 - \alpha_2^{old} y_1 y_2$$

Replace the symbol α_1, v_1, v_2

$$\mathcal{L}_d(\alpha) = (\zeta y_1 - \alpha_2 y_1 y_2) + \alpha_2$$
$$-\frac{1}{2}(\zeta y_1 - \alpha_2 y_1 y_2)^2 K_{1,1} - \frac{1}{2}\alpha_2^2 K_{2,2} - (\zeta y_1 - \alpha_2 y_1 y_2)\alpha_2 y_1 y_2 K_{1,2}$$
$$-(\zeta y_1 - \alpha_2 y_1 y_2) y_1 v_1 - \alpha_2 y_2 v_2$$

$$= (\zeta y_1 - \alpha_2 y_1 y_2) + \alpha_2$$
$$-\frac{1}{2}(\zeta^2 + \alpha_2^2 - 2\zeta\alpha_2 y_2)K_{1,1} - \frac{1}{2}\alpha_2^2 K_{2,2} - (\zeta\alpha_2 y_2 - \alpha_2^2)K_{1,2}$$
$$-(\zeta - \alpha_2 y_2)v_1 - \alpha_2 y_2 v_2$$

2.2.3 Combine the v_1, v_2 and ζ

$$v_1 - v_2 = [\, f_\phi(x_1) - b - \alpha_1^{old} y_1 K_{1,1} - \alpha_2^{old} y_2 K_{2,1}\,] - [\, f_\phi(x_2) - b - \alpha_1^{old} y_1 K_{1,2} - \alpha_2^{old} y_2 K_{2,2}\,]$$
$$= [\, f_\phi(x_1) - b - (\zeta y_1 - \alpha_2^{old} y_1 y_2) y_1 K_{1,1} - \alpha_2^{old} y_2 K_{2,1}\,] - [\, f_\phi(x_2) - b - (\zeta y_1 - \alpha_2^{old} y_1 y_2) y_1 K_{1,2} - \alpha_2^{old} y_2 K_{2,2}\,]$$
$$= [\, f_\phi(x_1) - f_\phi(x_2)\,] + [\, -(\zeta - \alpha_2^{old} y_2)K_{1,1} - \alpha_2^{old} y_2 K_{2,1}\,] - [\, -(\zeta - \alpha_2^{old} y_2)K_{1,2} - \alpha_2^{old} y_2 K_{2,2}\,]$$
$$= [\, f_\phi(x_1) - f_\phi(x_2)\,] + [\, -\zeta K_{1,1} + \alpha_2^{old} y_2 K_{1,1} - \alpha_2^{old} y_2 K_{2,1}\,] - [\, -\zeta K_{1,2} + \alpha_2^{old} y_2 K_{1,2} - \alpha_2^{old} y_2 K_{2,2}\,]$$
$$= f_\phi(x_1) - f_\phi(x_2) - \zeta K_{1,1} + \zeta K_{1,2} + (K_{1,1} + K_{2,2} - 2K_{1,2})\alpha_2^{old} y_2$$

2.2.4 Derive Gradient of α_2

$$\frac{\partial \mathcal{L}_d(\alpha)}{\partial \alpha_2} = -y_1 y_2 + 1 - \frac{1}{2}(2\alpha_2 - 2\zeta y_2)K_{1,1} - \alpha_2 K_{2,2} - (\zeta y_2 - 2\alpha_2)K_{1,2} - (-y_2)v_1 - y_2 v_2$$
$$= (-\alpha_2 K_{1,1} - \alpha_2 K_{2,2} + 2\alpha_2 K_{1,2}) + \zeta y_2 K_{1,1} - \zeta y_2 K_{1,2} - y_1 y_2 + y_2 v_1 - y_2 v_2 + 1$$
$$= -\alpha_2(K_{1,1} + K_{2,2} - 2K_{1,2}) + \zeta y_2 K_{1,1} - \zeta y_2 K_{1,2} - y_1 y_2 + y_2(v_1 - v_2) + 1$$

Replace $v_1 - v_2$ containing old $\alpha_1^{old}, \alpha_2^{old}$ (derived in 2.2.3)

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

Let η and E_i be

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

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

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

Then we can find the relation between new and old α_2 as following

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

To make the notation more clear to identify, we denote α_2^{new} as the new value of the update.

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

2.3 Step 2. Clip with Bosk Constraint

The new values should satisfy the complementary slackness as

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

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 he 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.

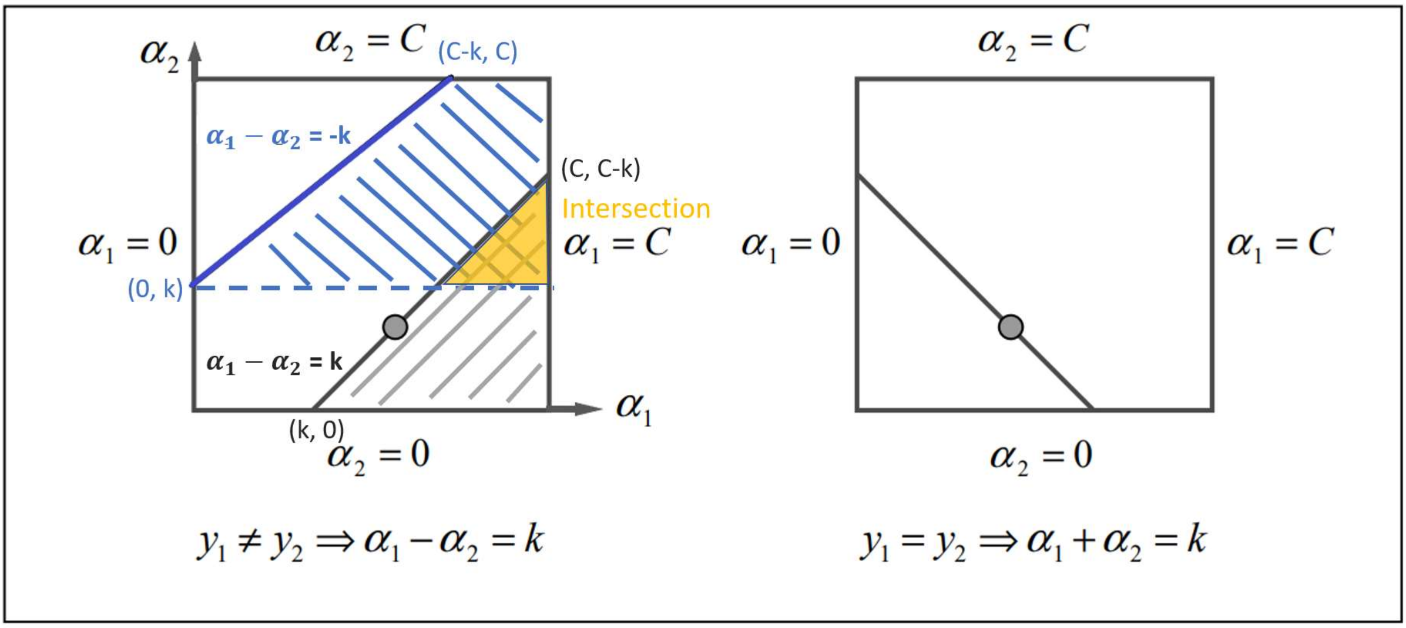


Figure 1

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 bosk constraint. The upper bound of α_2^{new} can be written as

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

and the lower bound is

$$B_L = \max(0, \alpha_2^{old} - \alpha_1^{old})$$

2.3.2 Case 2: Equality

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

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

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

and the lower bound is

$$B_L = \max(0, \alpha_2^{old} + \alpha_1^{old} - C)$$

2.3.3 Clip The Value

According the bound we've derived, we need **clip** the updated variable α_2^{new} to satisfy the constraint. In addition, we denote the new value after clipping as α_2^*.

$$\alpha_2^* = \mathrm{CLIP}\left(\alpha_2^{new}, B_L, B_U\right)$$

2.3.4 Update α_1

We've know the complementary slackness.

$$\alpha_1^* y_1 + \alpha_2^* y_2 = \alpha_1^{old} y_1 + \alpha_2^{old} y_2 = \zeta$$

Move the updated value α_1^* to the left side and we can get

$$\alpha_1^* = \frac{\alpha_1^{old} y_1 + \alpha_2^{old} y_2 - \alpha_2^* y_2}{y_1}$$

$$\alpha_1^* = \alpha_1^{old} + y_1 y_2 (\alpha_2^{old} - \alpha_2^*)$$

2.4 Step 3. Update Bias

The only equation that contains bias b is the function $f_\phi(x) = b + \sum_{i=1}^N \alpha_i y_i k(x_i, x)$. When $0 < \alpha_i^* < C$, 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, $f_\phi^*(x_w) = \sum_{i=3}^N \alpha_i y_i K_{i,w} - \alpha_1^* y_1 K_{1,w} - \alpha_2^* y_2 K_{2,w} + b^* = y_w$ contains updated variables $\alpha_2^*, \alpha_2^*, b^*$.

If $0 < \alpha_1^* < C$, the data point x_1 should right on the margin and $f_\phi^*(x_1) = y_1$. The bias derived from α_1.

$$b_1^* = y_1 - \sum_{i=3}^N \alpha_i y_i K_{i,1} - \alpha_1^* y_1 K_{1,1} - \alpha_2^* y_2 K_{2,1}$$

$$= (y_1 - f_\phi(x_1) + \alpha_1^{old} y_1 K_{1,1} + \alpha_2^{old} y_2 K_{2,1} + b) - \alpha_1^* y_1 K_{1,1} - \alpha_2^* y_2 K_{2,1}$$

$$= -E_1 - y_1 K_{1,1}(\alpha_1^* - \alpha_1^{old}) - y_2 K_{2,1}(\alpha_2^* - \alpha_2^{old}) + b$$

If $0 < \alpha_2^* < C$, the data point x_2 should right on the margin and $f_\phi^*(x_2) = y_2$. The bias derived from α_2.

$$b_2^* = y_2 - \sum_{i=3}^N \alpha_i y_i K_{i,2} - \alpha_1^* y_1 K_{1,2} - \alpha_2^* y_2 K_{2,2}$$

$$= (y_2 - f_\phi(x_2) + \alpha_1^{old} y_1 K_{1,2} + \alpha_2^{old} y_2 K_{2,2} + b) - \alpha_1^* y_1 K_{1,2} - \alpha_2^* y_2 K_{2,2}$$

$$= -E_2 - y_1 K_{1,2}(\alpha_1^* - \alpha_1^{old}) - y_2 K_{2,2}(\alpha_2^* - \alpha_2^{old}) + b$$

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

Given max-iter, otherwise the default value is $\text{max-iter} = 10^3$

For all $\alpha_i = 0, 1 \le i \le N$

$b = 0$

$\text{move} = \infty$

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

- $\alpha_1^* = \alpha_2^* = b^* = \text{move} = 0$
- for(n in $N/2$):
  - Choose the index i, j from 1 to N
  - $E_i = f(x_i) - y_i$
  - $E_j = f(x_j) - y_j$

- $\eta = K_{i,i} + K_{j,j} - 2K_{i,j}$
- $\alpha_j^{new} = \alpha_j + \frac{y_j(E_i - E_j)}{\eta}$

  Bosk Constraint

- if($y_i = y_j$):
  - $B_U = \min(C, \alpha_j + \alpha_i)$
  - $B_L = \max(0, \alpha_j + \alpha_i - C)$
- else:
  - $B_U = \min(C, C + \alpha_j - \alpha_i)$
  - $B_L = \max(0, \alpha_j - \alpha_i)$
- $\alpha_j^* = \mathrm{CLIP}\left(\alpha_j^{new}, B_L, B_U\right)$
- $\alpha_i^* = \alpha_i + y_i y_j (\alpha_j - \alpha_j^*)$

  Update Bias

- $b_i^* = -E_i - y_i K_{i,i}(\alpha_i^* - \alpha_i) - y_j K_{j,i}(\alpha_j^* - \alpha_j) + b$
- $b_j^* = -E_j - y_i K_{i,j}(\alpha_i^* - \alpha_i) - y_j K_{j,j}(\alpha_j^* - \alpha_j) + b$
- if($0 \leq \alpha_i \leq C$):
  - $b^* = b_i^*$
- else if($0 \leq \alpha_j \leq C$):
  - $b^* = b_j^*$
- else:
  - $b^* = \frac{b_i^* + b_j^*}{2}$
- $\mathrm{move} = \mathrm{move} + |\alpha_1^* - \alpha_1| + |\alpha_2^* - \alpha_2| + |b^* - b|$
- $\alpha_i = \alpha_i^*, \quad \alpha_j = \alpha_j^*, \quad b = b^*$

- $\mathrm{iter} = \mathrm{iter} + 1$

---

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 $\phi : \mathbb{R}^n \to \mathbb{C}$ 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 $\zeta_\omega(x) = e^{j\omega' x}$, we have

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

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

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

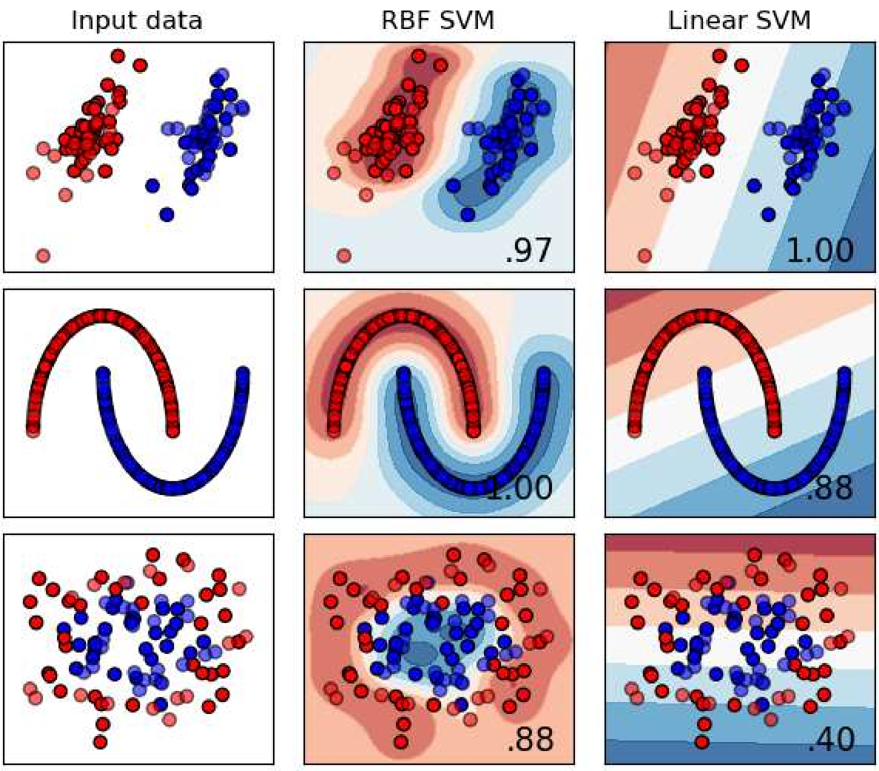
$$z(x)'z(y) = \frac{1}{D}\sum_{j=1}^{D} z_{w_j}(x)z_{w_j}(y)$$

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

In order to approximate the RBF kernel $k(k, y) = e^{-\frac{||x-y||_2^2}{2}}$, we draw ω from Fourier transformed distribution $p(\omega) = \mathcal{N}(0, 1)$.

4. Experiments

4.1 Simulation With Exact Kernel



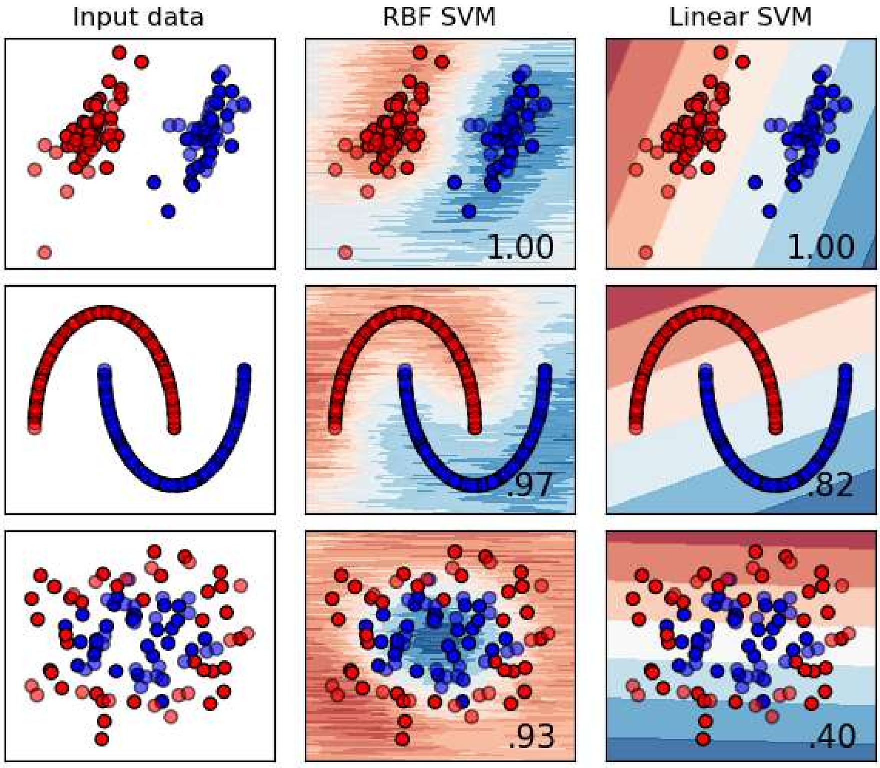
The parameters of SVM:

- C: 0.6
- γ 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 generate 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

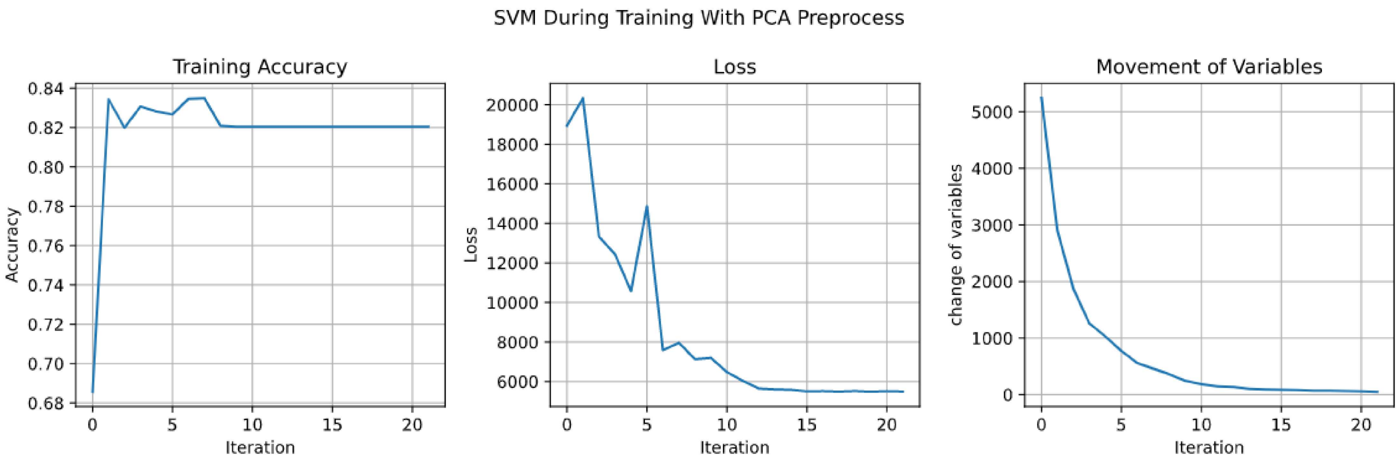


We draw 200 samples from $p(\omega)$ to approximate the RBF kernel. As we can see, the testing accuracies are close to the ones of exact kernels in most of cases.

4.3 Real Dataset

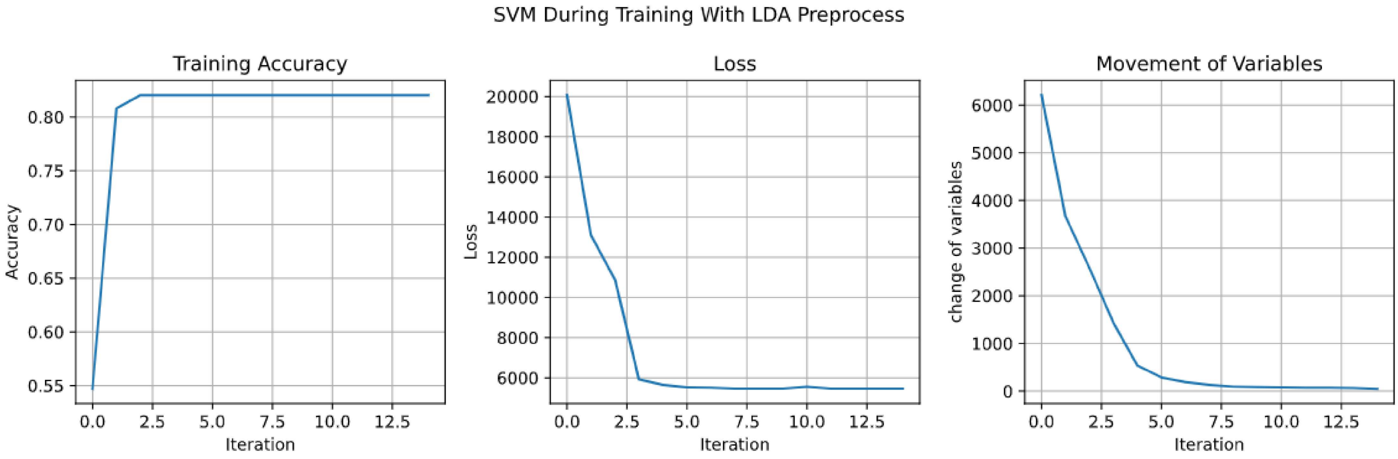
4.3.1 PCA Preprocess

Apply SVM on the "Women's Clothing E-Commerce Review Dataset" with C = 0.6 and γ 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.

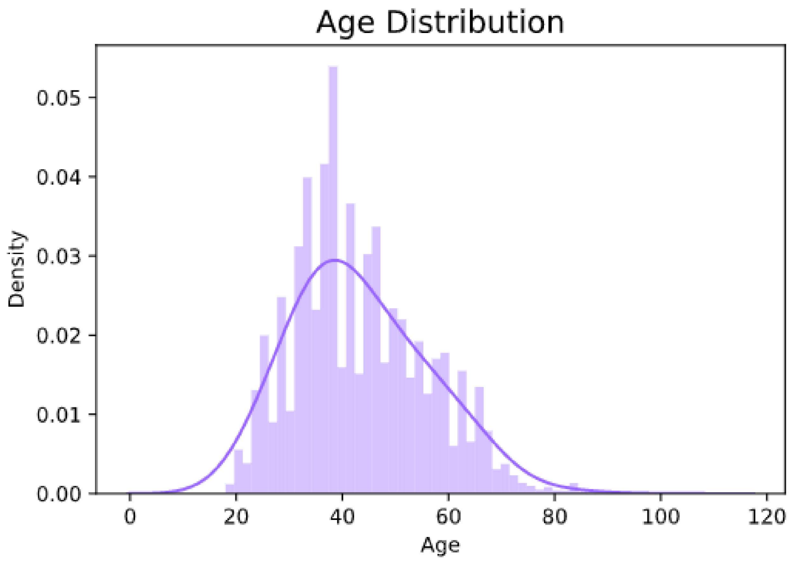
5. Data Analysis

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

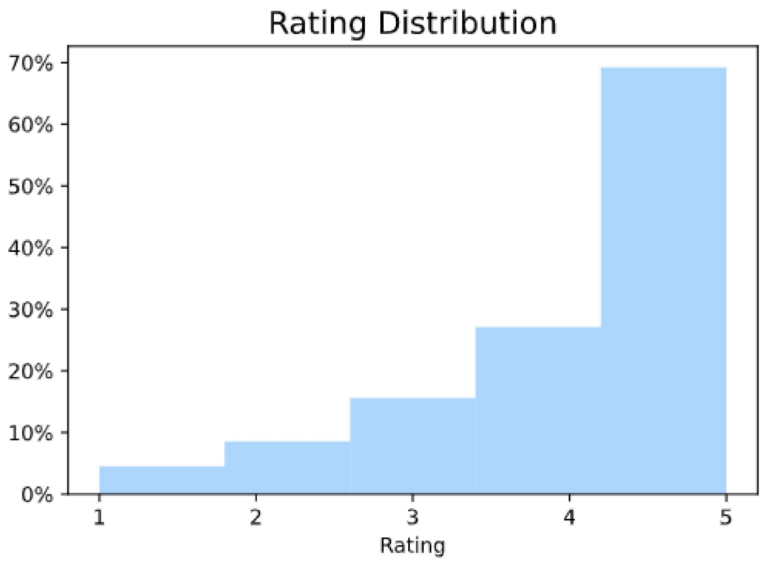
Age Distribution



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.

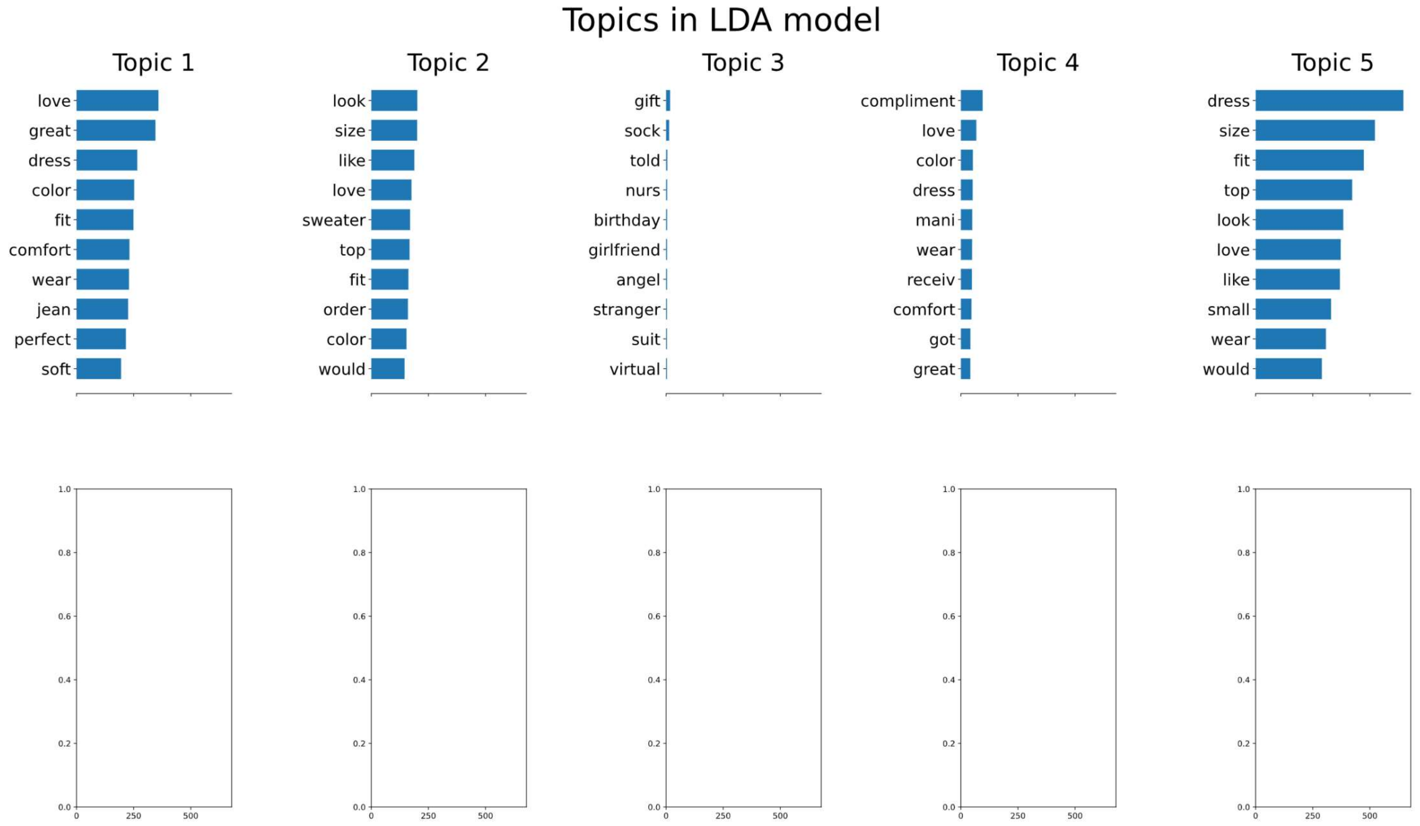
Rating Distribution



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.

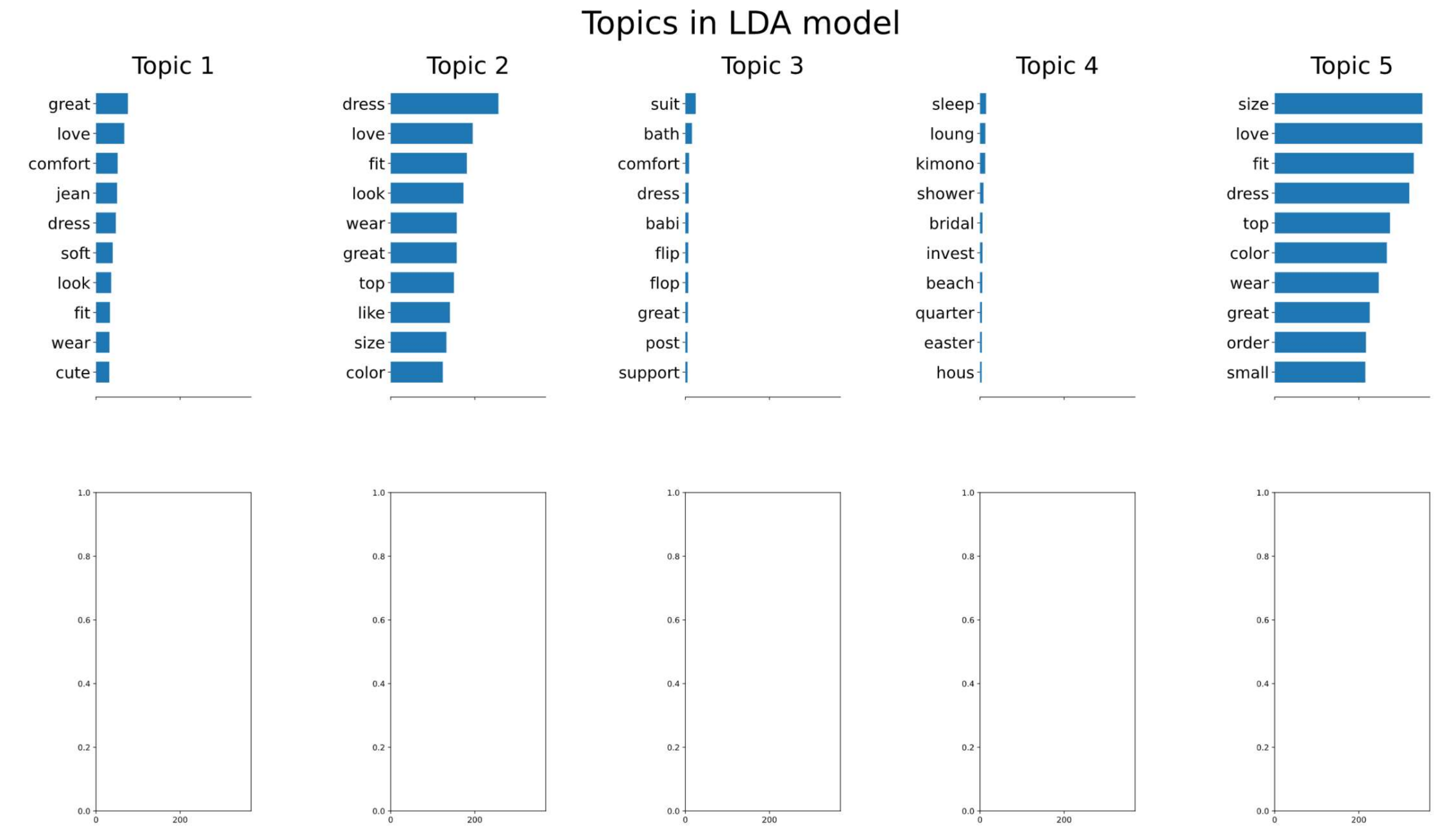
Topics



The graph is the result of LDA with 5 topics. Ttopics are 3 and 4 seems interesting. The 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 of purchases in this topic are the gifts for girlfriends by the boyfriends. The topic 4 seems also related to gift but not between lovers. The comment are mainly about receivers' compliment.

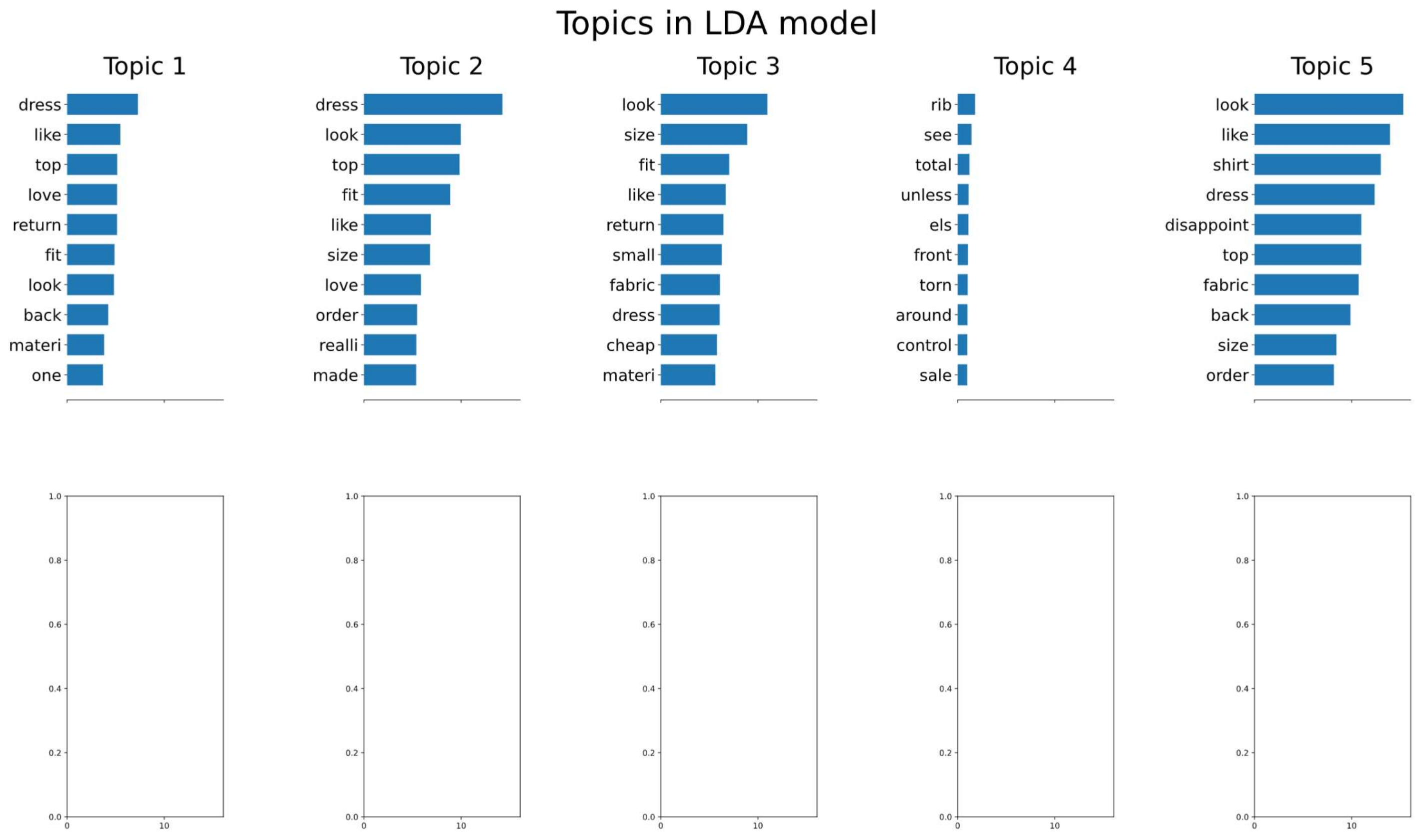
5.2 Rating

Rating Score 5



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

Rating Score 1



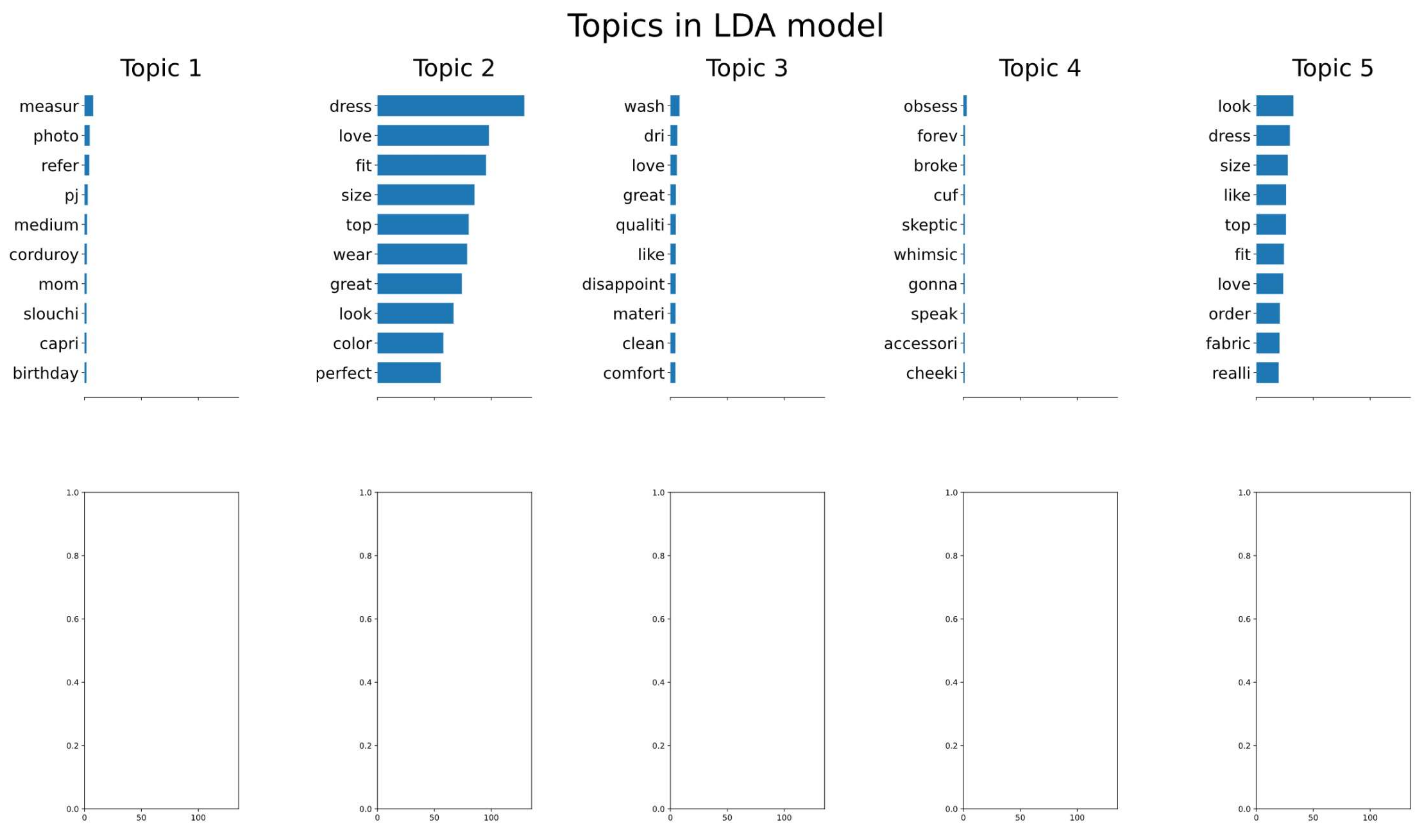
An interesting thing is that "cute" seems neutral since it appears in both side. It may be caused by cultural difference between western and eastern societies.

5.3 Ages

LDA

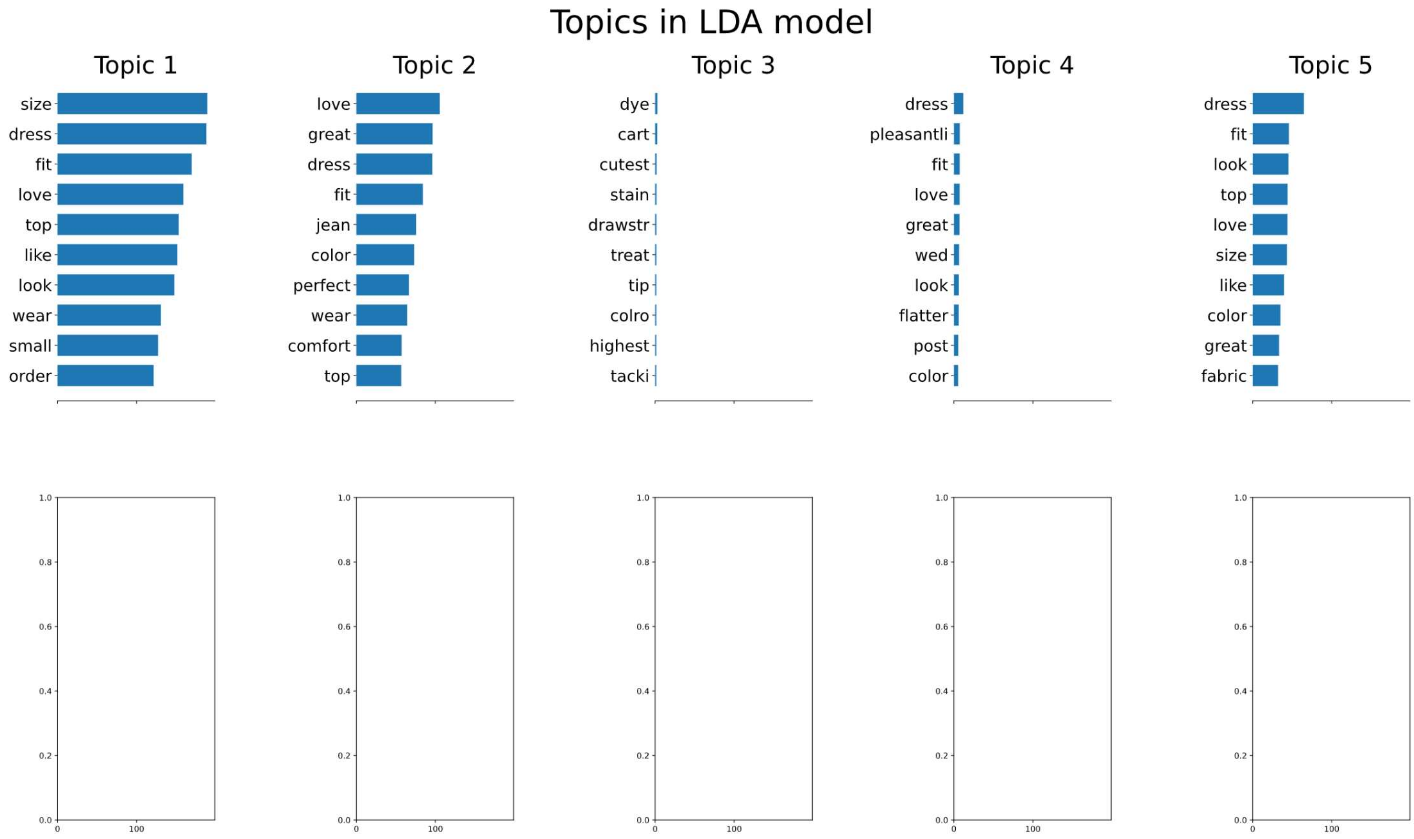
Love and cute are everywhere. Also, some positive words and common nouns appear in every ages. It's boring for me to focus on that common phenomenon. Here I just mention something interesting from the data.

Less Than 30



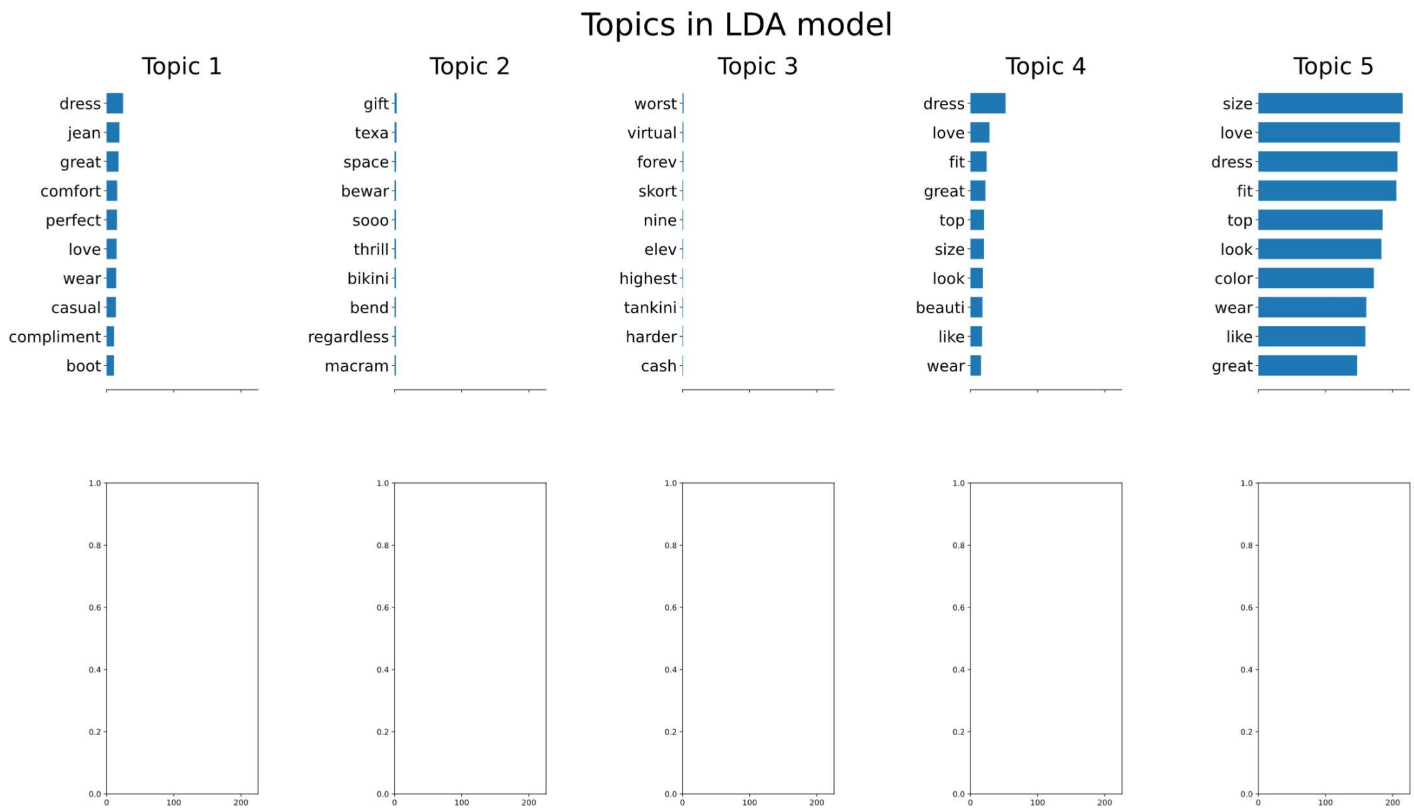
An interesting thing is that the topic 4 contains broke, cheeky, obsess and, forever. It seems that topic 4 is about the relationship. In addition topic 4 only appears under 30 years old.

30s



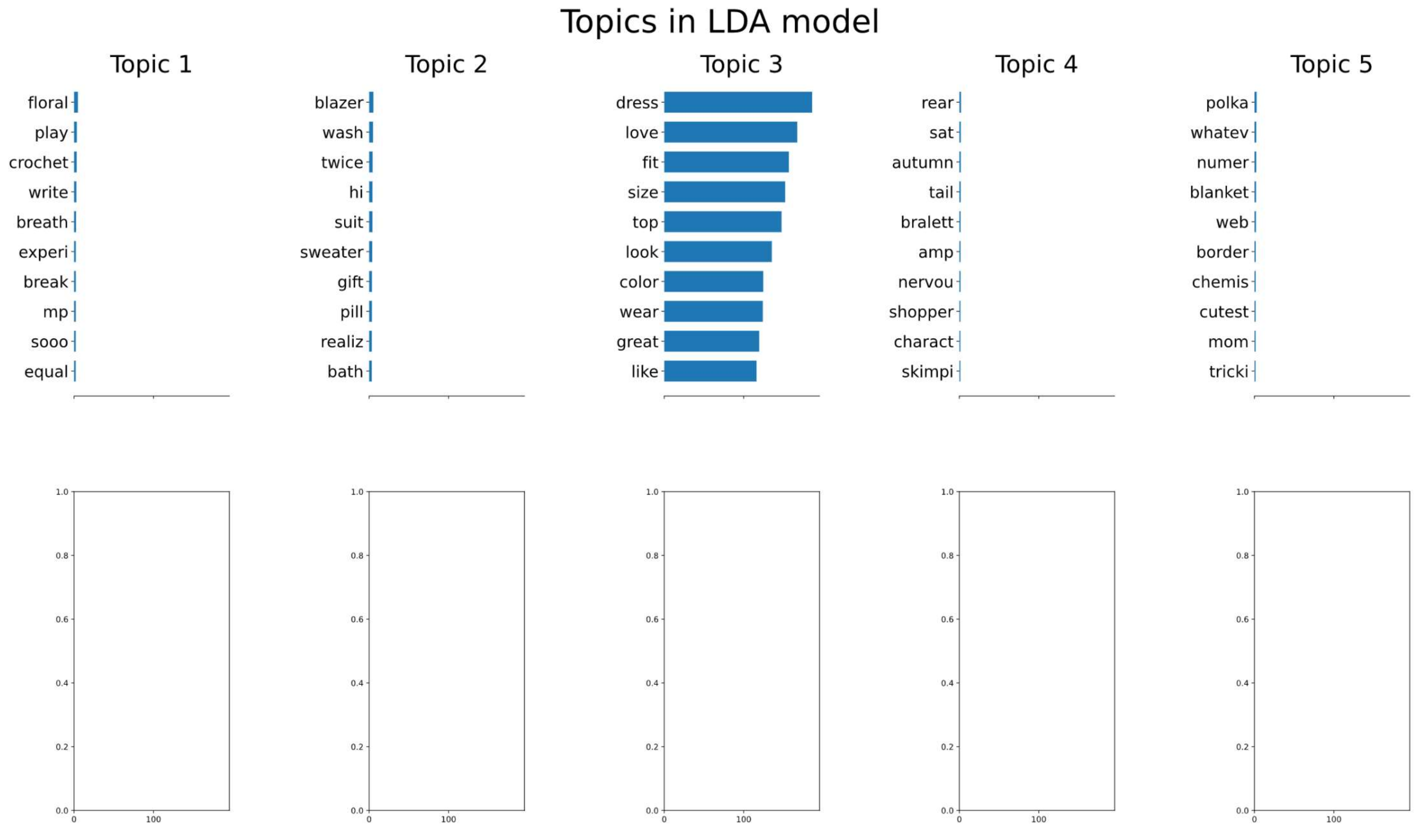
The topic 4 contains wed, love and, pleasant. It perhaps is about wedding.

40s



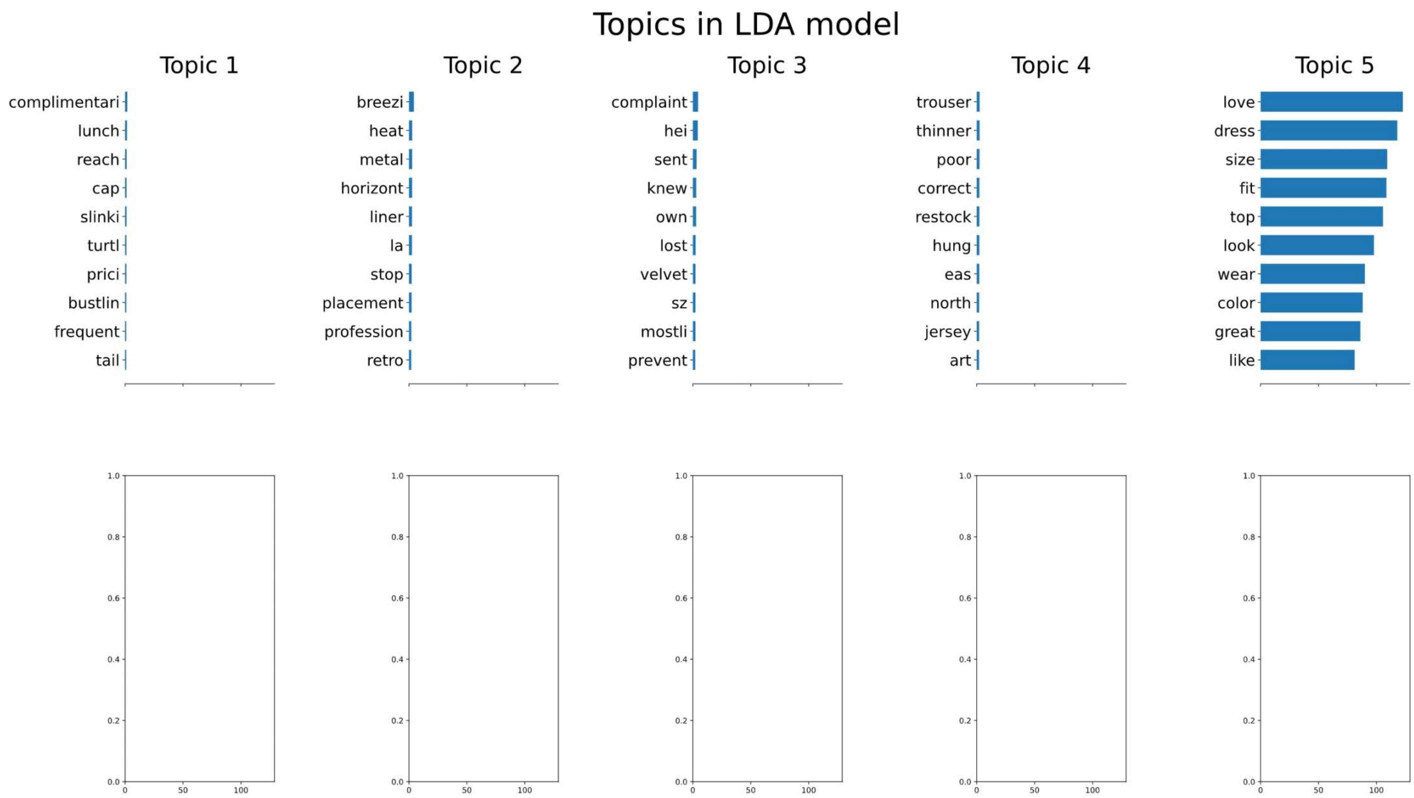
It's a little bit weird that one topic is about the "bikini" and the other one is about "tankini" which is a kind of swimsuit covering the whole body.

50s



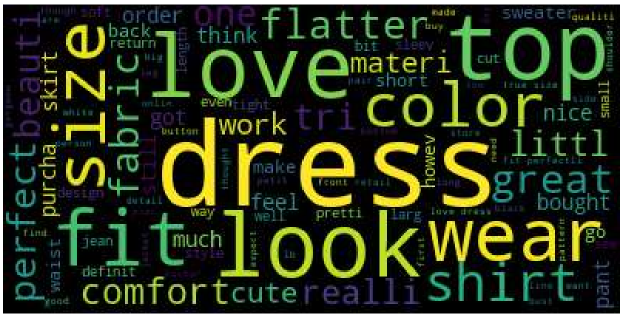
I don't know why there is a topic about "bralett" which means "sexy intimates". Does it mean the shirley valentine for the customers?

More Than 60



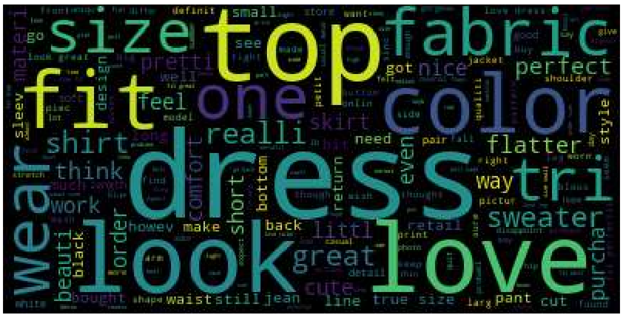
Word Cloud

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