Improved Models In Multimodal Fusion For Inferring User Interests Combined With Feature Selction And Dimension Reduction

Xiong Luo, Long Wang, Yunlong Wang

Abstract

The social network has become an important information platform which has great influence and become an indispensable part of our lives. In social networks, an accurate understanding of user characteristics is especially important. In this paper, our main task is to predict users' interests based on pictures and texts in social networks, which is a classification problem of multimodal fusion in practical. However, when multimodal data is put together, there may occur the dimension disaster problem.

We apply feature selection(FS) and dimension reduction(DR) in feature level both in later fusion and early fusion to solve this problem.

After adding FS/DR to multimodal fusion, not only the dimensions of features are decreased which mean the time of classification will decrease, but the time of the fusion model with some models of FS/DR can be decreased overall. The classification accuracies in different models also obtain improvements in different levels respectively. We also discuss the relation between single modals and multimodal in later fusion.

In our experiments, user interests can be predicted by multimodal modals under FS/DR, and of which with the help the behavior characteristics of users in social networks can be analyze better to help enterprises provide better services and products, and then carry out better network marketing and promotion.

Key Words: multimodal, model fusion, social media, interest prediction, feature selection, diemension reduction

1 Introducton

Social networks have penetrated into every aspect of people's lives. Online shopping has become one of the mainstream shopping methods, and netizens can also use social networks to make friends, play games and interact with each other. Accurate analysis and forecasting of users in social networks can serve political and economic fields, especially for research and applications such as precision business marketing, web-based public opinion detection, web personalized recommendation services, social network product reviews, etc. stand by.

With the rapid development of social networks, the scale of users is constantly expanding [16], and information updates are accelerating, and it becomes more and more difficult for users in social networks to find information of their own interest. How to help social network users filter out information that is not interesting for them and provide personalized recommendation services that meet the needs of users has become a research hotspot of current academic and industrial circles. User interest modeling can provide reliable user information for personalized information services and effectively improve the service quality of personalized information services. It is the basis and core for the effective development of personalized information services.

In this paper, we experiment on dataset crawled from Pinterest. Pinterest is a photo sharing social networking website where users could share and collect pictures. Users download, share and trasnmit pictures(aka pins) in this free website. Users can pin pins to their pinboards and classify pins themselves. However, not all pinboards are classified.

We use multimodal models to classify users and infer interests of users. Considering that fusion models may bring redundent information and the dimension reduction problem. We apply feature selection and dimension reduction to multimodal fusion and get a better performance than the principal models.

2 Related work

There are many studies on inferring user interest based on user information. [21] [24] infer user interest based on text information, and [4] [25] does relevant researches based on information of images. With the success of deep learning in the fields of computer vision and natural language processing, the deep learning research process has begun to involve complex multimodal data, and it is becoming more and more common to use multimodal data to perform complex learning tasks [18]. A lot of related research work has already appeared. For the related work of multimodal learning of images and texts, there are mainly two ways of fusion. One of them is the fusion of the feature level in the early stage, and the other is the fusion of the decision level in the later stage.

Early fusion is a fusion before classification. First, the features in different modes are extracted, and then the fusion is performed by a certain method. The Early fusion fuses the information of different models at the feature level, and connects the features of the image and the text to form a single multi-modal vector, such as [3]. The connected multimodal vector can then be used for classification. Some studies have improved the details of early fusion, such as [7], using pooled layers to combine information from different modalities, and using auxiliary learning tasks to learn common feature spaces. Several variants of pre-integration based on social media content have also been proposed. For example, in sentiment analysis, LSTM (Long Short-Term Memory) is used to combine visual and textual information rather than connecting information in different models [26]; in social event classification, hierarchical classifiers are constructed on cascaded features to classify event types [28]. Early fusion integrates features often by simply concatenating representations from different modals,such as [12].

Later fusions first construct different modalities to obtain separate classification results, and then fuse based on these results [6]. [5] shows that in fact combined processing image and text increases the overall interest classification accuracy compared to uni-modal representations. In a recent study, a variant of late fusion was proposed. The author used KL divergence to consider the distribution of different model results to approximate the performance of different models [27]. According to [18], the performance of late fusion and early fusion depends heavily on the problem. Besides, [14] proposes that compared to early fusion, late fusion tends to be more robust to featuresthat have negative influence. And there also exists hybrid fusion which combines outputs from early fusion and individual unimodal predictors [1].

However, fusion work in such social networks directly use features extracted from extractors, which, as we know, often have high dimensions, which may lead to dimesional disasters [2]. For example, a image feature extracted from final fully-connected layer of Alexnet [13] has 4096 dimensions. With the increase of data dimension, the objective function optimization, parameter estimation and model selection become more and more difficult. Besides, there will be more noises which influence the performance of models. Therefore, to solve this problem, we perform feature selection and feature dimension reduction for multimodal fusion.

3 Model

3.1 Representation

3.1.1 image representations

It's obvious that CNN can perform the state-of-the-art results in image classification. Therefore we complete the feature extraction on the basis of a CNN model known as Alexnet . In particular, we use the model trained on the ImageNet dataset for the image classification. Technically, we feed images to this feature extractor and chose the features of the Fc7 output as features of the image, which are 4096 dimensions.

3.1.2 text representations

Among the problems of natural language processing and text analysis, Word Embedding is a most commonly used model. In this paper, we use Embedding to extract text feature. The word vector model is a model that considers the positional relationship of words [15]. Through the training of a large number of corpora, each word is mapped into a high-dimensional vector. We use the vectors trained on three million words and phrases based on a Google News dataset. Each word will be transormed to vectors of 300 dimensions with these vectors. We calculate the vectors of a phrase by the method proposed in [5].

3.2 Feature Selection

3.2.1 Binary PSO

Particle swarm optimization(PSO) was presented by Eberhart and Kennedy in 1995 [8]. The particle swarm algorithm mimics the clustering behavior of insects, herds, flocks, and fish groups. These groups search for food in a cooperative way. Each member of the group changes its search mode constantly by learning its own experience and the experience of other members.Here we provide a short description of PSO. Supposing that we search in a d dimension space.For ith particle in pyswarm, we consider p_{best} as its self-optimal position and g_{best} as the optimal position of the particle swarm, and its updation of velocity and position can be expressed as below:

$$v^{'} = wv + c_1 r_1 (p_{best} - x) + c_2 r_2 (g_{best} - x) \tag{1}$$

$$x^{'} = x + v^{'} \tag{2}$$

Where c_1 and c_2 represent are two constant coefficient,while r_1 and r_2 are two random number between zero and one generated for each particle. Given that $v^{'}$ represents velocity of ith particle in time t and x represents position of ith particle in time t, then $v^{'}$ represents velocity in time $t+1$, $x^{'}$ represents position in time $t+1$. If the fitness result of $x^{''}$ is better than x, the position will be updated to $v^{'}$ greedily.

In the binary particle swarm optimization(Bpso) proposed in [11], velocity of ith particle in time t+1 updated in formula (1) of ith particle will be mapped to the range $[0, 1]$:

$$v^{''} = \frac{1}{1 + e^{v'}} \tag{3}$$

What's more, rather than update the position of ith particle in time $t+1$ on the basis of formula (2), it'll be updated according to equation below:

$$x^{'} = \begin{cases} 1 & if\ r3 < sig(v^{''}) \\ 0 & otherwise \end{cases} \tag{4}$$

where r3, which is generated randomly for the ith particle, is a random number in the range of [0,1]. Greedy updation similar to PSO is adopted.

3.2.2 Binary Jaya

Jaya is a simple optimization algorithm proposed by R.Venkata Rao [19] for solving the constrained and unconstrained optimization problems.Here is a short description of Jaya algorithm descripted in [jaya]. Let $f(x)$ is the fitness function to be minimized. Assume that we search in a space whose population size is n and number of design variables is m. For the j_{th} variable, consider the best/worst value of $f(x)$ of all candidate solutions at i_{th} iteration is $x_{best,i}$ / $x_{worst,i}$. At i_{th} iteration, the j_{th} design variable can be updated as following:

$$x^{'}_{k,i,j} = x_{k,i,j} + \rho_1(x_{best,i,j} - |x_{k,i,j}|) - \rho_2(x_{worst,i,j} - |x_{k,i,j}|) \tag{5}$$

where $x^{'}_{k,i,j}$ is the updated value of $x_{k,i,j}$ which is the value of the j_{th} design variable for the k_{th} candidate at the i_{th} iteration. Besides, ρ_1 and ρ_2 are two random number between 0 and 1 for the j_{th} variable at the i_{th} iteratoin. It's obvious that the value of variable is closer to the best solution and further from the worst solution after every iteration.

Binary Jaya algorithm(Bjaya) is the binary version of Jaya algorithm proposed by [17]. In the Bjaya, the value of $x^{'}_{k,i,j}$ from equation (5) is updated based on the following rule:

$$X^{'}_{k,i,j} = \begin{cases} 1 & if\ \rho_3 < f(|X_{k,i,j}|) \\ 0 & otherwise \end{cases} \tag{6}$$

Where ρ_3 represents a random number generated randomly which is between 0 and 1, and $f(|X_{k,i,j}|)$ is expressed as:

$$f(|X_{k,i,j}|) = \frac{e^{\left|2x^{'}_{k,i,j}\right|-1}}{e^{\left|2x^{'}_{k,i,j}\right|+1}} \tag{7}$$

Jaya algorithm and Bjaya algorithm update the value of the design variable greedily similar to PSO.

3.2.3 Variance Feature Selection

VarianceThreshold Feature Selection(VFS) is a simple method of feature selection based on theory proposed in [10]. As mathematically described in equation (8), consider we have N samples of which each has M features, then we're able to calculate the variance of the j_{th} feature δ_j^2 according to the j_{th} feature value of each sample $x_{i,j}$ and their mean value μ_j.

$$\delta_j^2 = \frac{\sum\limits_{i=1}^{N}(x_{i,j} - \mu_j)^2}{N} \tag{8}$$

As is known to all, variance is a measure of the degree of dispersion in a set of data. If the vairance of a feature is close to zero, it refers to that values of this feature are not divergent, which means this feature is basically meaningless for sample distiguishment. Therefore, after obtaining variance of each feature, we keep back features whose variance is bigger than a threshold to choose specific features. The threshold here should be designated specially. In this paper, we search this value with the help of Jaya algorithm mentioned above.

3.2.4 Univariate Feature Selection

Univariate feature selection calculates an indicator for each feature separately, then determine the importance of features based on this indicator and eliminate those unimportant features. KBest Feature Selection(KFS), which select the k features of the k highest scores, is a representative method of univariate feature selection. Fpr Feature Selection(FFS) is another method of univariate feature selection wihch select features whose pvalues of feature scores are below the pre-set alpha. In SelectFpr, we use the same score function as SelectKBest to calculate pvalues. Here we provide a short description of the principal of the score function we use.

Let mean variances between feature/label is V_b whose degree of freedom is d_b and mean variances within feature/label is V_w whose degree of freedom is d_w. Suppose V_b and V_w obey the chi-square distribution. Then we can obtain a statistics F(aka F distribution):

$$F = \frac{V_b/d_b}{V_w/d_w} \tag{9}$$

The probability density function of the F distribution is:

$$\epsilon(x; d_b, d_w) = \begin{cases} \frac{\chi(\frac{d_b+d_w}{2})}{\chi(\frac{d_b}{2})\chi(\frac{d_w}{2})}(\frac{d_b}{d_w})^{\frac{d_b}{2}}(1+\frac{d_b}{d_w}x)^{\frac{d_b+d_w}{2}}x^{\frac{d_b}{2}-1} & x \geq 0 \\ \\ 0 & x < 0 \end{cases} \tag{10}$$

In SelectKBest, we calculate F-value of a feature in equation (9) between values of this feature and corresponding label as a score and select features according to scores. Specifically, we select features according to the k higheset scores. Each F-value corresponds a unique p value is the probability that a more extreme result will occur when the null hypothesis is true than the resulting sample observation. Besides, the higher the F-value is, the lower the p value is. For more detailed content, we recommend [22]. If the p value is less than the significance level alpha, the null hypothesis should be rejected. After setting alpha, we use the p values to decide whether to reject the null hypothesis or not. When the p-value is less than alpha in a hypothetical test, the null hypothesis is rejected, which will result in a false positive if the null hypothesis is a invalid hypothesis. The number of false positive results determines the calculation result of FPR(False Positive Rate). In other words, we set the alpha value to filter the features based on Fpr in feature selection using FFS.

3.2.5 L1-based Feature Selecion

Linear model using the L1 norm as a penalty will result in sparse solutions [9]. The L1 norm is descripted as below:

$$||z||_1 = \sum_{l=1}^{n} |z_l| \tag{11}$$

Where z is a vector whose dimension is n. With the help of these models, we can reduce dimension of features which we'll later take as input of fusion models. Frequently used linear models are Lasso, LogisticRegression and Linear Support

Vector Classifier(LinearSVC). In this paper, we use LinearSVC model penalized with the L1 norm to select features and feed selected features to our multimodal fusion models. We call this Model Feature Selection(MFS).

3.3 Dimension Reduction

3.3.1 Local Linear Embeddings

Local Linear Embeddings(LLE) assume that data is linear in local parts and a certain data can be represented by several samples adjacent to it. For sample z_i, we asume that it can be linearly represented by its nearest k samples:

$$z_i = \sum_{j=1}^{k} \psi_{i,j} z_j \tag{12}$$

Where $\psi_{i,j}$ refers to linear relation between sample z_i and sample z_j. To represent z_i, firstly we obtain k nearest samples of sample z_i, then we need to find linear relation between z_i and these samples. To represent all samples we need to find a weight matrix. This is equivalent to a regression problem with certain qualifications as descried in equation(13). We can use mean variance as its loss function and calculate the weight matrix via minimizing the loss function:

$$\min_{\psi_1,\psi_2,\ldots,\psi_n} \quad \sum_{i=1}^{n} \left\| z_i - \sum_{j=1}^{k} \psi_{i,j} z_j \right\|_2^2 \tag{13}$$

$$s.t. \quad \sum_{j=1}^{k} \psi_{i,j} = 1 \tag{14}$$

Where $z_i = (z_{i1}, z_{i,2}, ..., z_{i,d})$ represents the i_{th} sample which has d dimensions. Consider $r_i = (r_{i1}, r_{i,2}, ..., r_{i,d'})$ is the coordinate of low dimensional space coordinating to z_i with $d^{'}$ dimensions while $\psi_i = (\psi_{i,j}, \psi_{i,2}, ..., \psi_{i,k})$ is the weight vector of i_{th} sample with its k nearest neighboors. For samples in low dimensional space maintaining the same relationship as in high dimensionl space, let $R = (r_1, r_2, ..., r_n)$ are coordinates of samples in low dimensional space, with the weight matrix $\Psi = (\psi_1, \psi_2, ..., \psi_n)$ obtained from equation (12) and equation (13) previously then we can obtain R [30]:

$$\min_{r_1,r_2,\ldots,r_n} \quad \sum_{i=1}^{n} \left\| r_i - \sum_{j=1}^{k} \psi_{i,j} r_j \right\|_2^2 \tag{15}$$

Let $S = (I - \Psi)^{\mathrm{T}}(I - \Psi)$, the optimization problem of equation (12) and eqation (13) can be simplified as below:

$$\min_{R} \quad tr(RSR^{\mathrm{T}}) \tag{16}$$

$$s.t. \quad R^{\mathrm{T}} R = I \tag{17}$$

Where tr refers to trace function. Since R is what we want, We can solve this problem by applying eigen composition to S. And the matrix cosisting of eigenvectors corresponding to the smallest $d^{'}$ eigenvalues of S is the coordinates of samples in low dimensional space.

3.3.2 PCA

PCA is proposed in [23]. In dimension reduction algorithm based on PCA, the data is converted from the original coordiante system to the new coordinate system which is determined by the data itself. The nature of PCA algorithm is to find some directions in which the data set is able to obtain the maximum variance. These directions are orthogonal to each other. The bigger the features of covariance matrix of the intial data set are, the bigger the corresponding variances are, and the greater the amount of information projected in corresponding feature vector is. Then we can achieve our goals of dimension reduction by delete data in the directions corresponding to low variances.

In genearl, PCA could convert linearly related high dimensional variables into linearly independent low dimensional variables. Specifically, the data set is converted from the initial coordinate system into a new coordinate system which is determined by the data set itself. The first coordinate (the first principal component) choose the direction with the largest variance. The second coordinate choose the direction which is orthogoanl to the first one with the largest variance. This process is repeated for d times. If we want to reduce the data set to d' dimensions, we keep back the first d' coordinates. The coordiante system determined in this way is able to represent all samples properly as far as possible. Finally, we project the original data set into the new coordinate system and obtain dimensionalized data.

Consider $Z = (z_1, z_2, ..., z_m)$ where $z_i = (z_{i,1}, z_{i,2}, ..., z_{i,d})^{\mathrm{T}}$ represents the i_{th} sample, is the original data set with d dimensions. Let $U = (u_1, u_2, ..., u_d)$ is the new coordiante system we're looking for. The variance after projecting is represented as $\sum_i U^{\mathrm{T}} z_i z_i^{\mathrm{T}} U$. Then our optimization function seeking for the new coordinate system can be represented as below [30]:

$$\max_{U} \quad tr(U^{\mathrm{T}} Z Z^{\mathrm{T}} U) \tag{18}$$

$$s.t. \quad U^{\mathrm{T}} U = I \tag{19}$$

Consider λ_i is the lagrange multiplier correponding to u_i,with the help of Lagrange Multiplier method, we get:

$$ZZ^T u_i = \lambda_i u_i \tag{20}$$

We apply eigenvalue composition to ZZ^{T} and take eigenvectors corresponding to the top d' largest eigenvalues to form $U^{'} = (u_1^{'}, u_2^{'}, ..., u_{d'}^{'})$. $U^{'}$ is the new coordinate system we want.

$$Y = U^{'\mathrm{T}} Z \tag{21}$$

Then we can obtain the dimensionalized data Y projected into the new coordinate system according to equation (20).

3.3.3 Kernel PCA

In Kernel PCA(KPCA), the original data is converted by means of linear mapping into a high dimensional space from where the converted data is projected via PCA into another low dimensional space [20]. Assume that we project the converted data into the space determined by $U = (u_1, u_2, ..., u_d)$ and y_i is the coordinate in high dimensional space of z_i. The optimization process can be described as below [30]:

$$\left(\sum_{i=1}^{m} y_i y_i^{\mathrm{T}}\right) u_j = \lambda_j u_j \tag{22}$$

$$\begin{aligned} u_j &= \frac{1}{\lambda_j}\left(\sum_{i=1}^{m} y_i y_i^{\mathrm{T}}\right) u_j \\ &= \sum_{i=1}^{m} y_i \frac{y_i^{\mathrm{T}} u_j}{\lambda_j} \\ &= \sum_{i=1}^{m} y_i \beta_i^j \end{aligned} \tag{23}$$

Here β_i^j represents the j_{th} value of β_i which owns d values in total. Suppose that function φ map z_i from original space into high dimensional space and get y_i, we can transform equation (21) and equation (22) and get:

$$\left(\sum_{i=1}^{m} \varphi(x_i)\varphi(x_i)^{\mathrm{T}}\right) e_j = \lambda_j e_j \tag{24}$$

$$u_j = \sum_{i=1}^{m} \varphi(x_i)\varphi_i^j \tag{25}$$

We can express φ which is ambiguous in most situations with the help of kernel function γ:

$$\gamma(x_i, x_j) = \varphi(x_i)^{\mathrm{T}}\varphi(x_j) \tag{26}$$

Simplify equation (25) via equation (23) and equation (24), we can get:

$$\Gamma\beta^j = \lambda_j \beta^j \tag{27}$$

Where $\Gamma_{i,j} = \gamma(x_i, x_j)$ is the matrix of γ. Similarly, we apply eigenvalue composition to Γ and take the eigenvectors corresponding to the top $d^{'}$ largest eigenvalues of Γ and get a set of $\beta^j(j = 1, 2, ..., d^{'})$. The j_{th} projected coordinate $z_j^{'}(j = 1, 2, ..., d^{'})$ in low dimensional space can be calculated as below:

$$
\begin{aligned}
z_j^{'} &= e_j^{\mathrm{T}} y \\
&= e_j^{\mathrm{T}} \varphi(x) \\
&= \sum_{i=1}^{m} \beta_i^j \varphi(x_i)^{\mathrm{T}} \varphi(x) \\
&= \sum_{i=1}^{m} \beta_i^j \varphi(x_i, x)
\end{aligned} \tag{28}
$$

3.3.4 Latent Semantic Index

The Latent Semantic Index(LSI), a dimension reduction method which is common in text classification [29], obtain the subjects of texts based on the singular value composition(SVD). Here we give a brief description of SVD and the application in this paper. Consider Z is the samples as mentioned before, we apply SVD to Z and get P, Σ, and Q^{T}. P is a d by d matrix in which vectors are often called left singular vectors, and Q is a m by m matrix in which vectors are often called right singular vectors while Σ is a d by m matrix whose elements on the diagonal are called singular values and non-diagonal elements are 0. The mathmatical representation is below:

$$Z = P \Sigma Q^{\mathrm{T}} \tag{29}$$

If we arrange singular elements in Σ from small to large, we're able to find that these arranged elements reduce sharply. Most important semantic information of Z can be reserved with the help of only the largest $d^{'}$ singular values:

$$Z^{'} = P_{d'} \Sigma_{d'} Q_{d'}^{\mathrm{T}} \tag{30}$$

Where $\Sigma = diag(\theta_1, \theta_2, ..., \theta_d^{'})$ is a diagonal matrix consisting of d' diagonal elements correponding to the largest $d^{'}$ singular values in Σ. $P_{d'}$ and $Q_{d'}^{\mathrm{T}}$ consist of left singular vectors and right singular vectors responding to these $d^{'}$ singular elements. Then original data Z can be converted to $Z_{d'}$ which has $d^{'}$ dimensions with the help of $P_{d'}$:

$$Z_{d'} = P_{d'}^{T} Z \tag{31}$$

$Z_{d'}$, which is a $d^{'}$ by m matrix in which each vector has $d^{'}$ dimensions, is what we want.

3.4 Multimodal Fuson

3.4.1 Later Fusion

Assume y_{i1} and y_{i2} are predicting results of image and text separately for sample i. Then the fusion predicting result y_i of i_{th} sample is represented as below:

$$y_i = w y_{i1} + (1 - w) y_{i2} + b \tag{32}$$

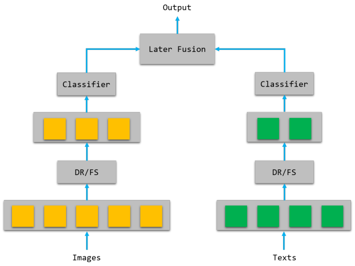


Figure 1: Later Fusion

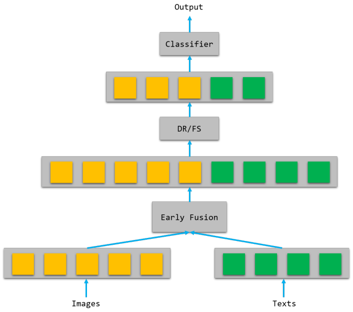


Figure 2: Early Fusion

Where w is the weight coefficient balance the predictions of images and texts and b is a bias term in this linear fusion model.

As is shown in Fugure 1, we take images and texts as input and then extract features of input. Then we apply feature selection and dimension reduction into extracted features, on the basis of which we accomplish classification independently. Then the results of classification are combined in our later fusion model.

3.4.2 Early Fusion

We give a short description of the later fusion model used in our experiments. Consider h_{i1} which has d_1 dimensions and h_{i2} which have d_2 dimensions are features extacted from two single modals. The early fusion accomplish fusion at the feature level and the fusional feature of sample i can be expressed in mathematical form as below:

$$h_i = (h_{i1}; h_{i2}) \tag{33}$$

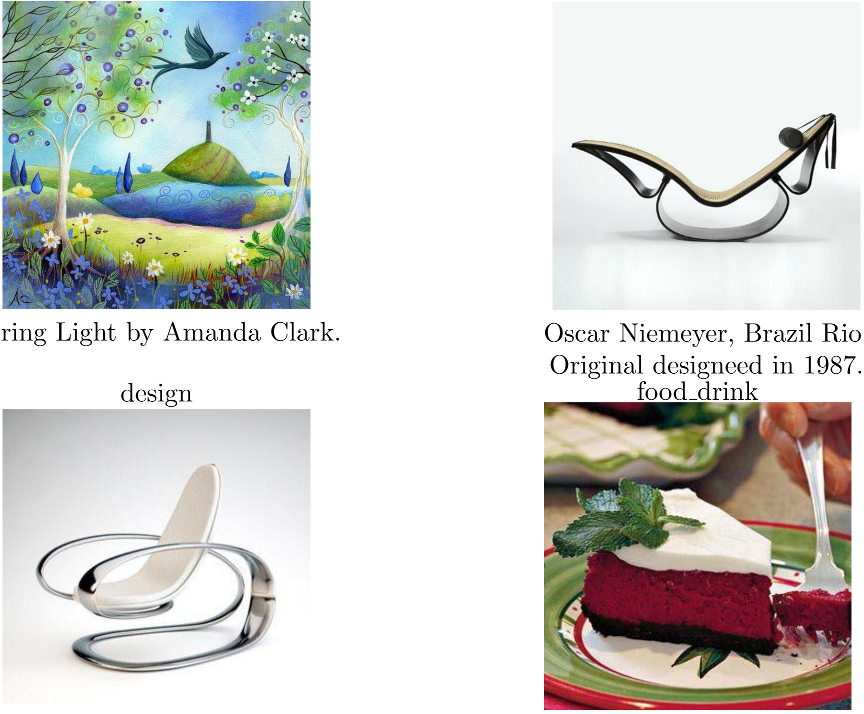
Where h_i have $d_1 + d_2$ dimensions.

In early fusion, we firstly accomplish early fusion with features extracted from different modals by concatenating them as shown in figure 2, then we apply feature selection and diemnsion reduction to the output of the early fusion model. After that we classify the output using classifications.

4 Experiments

4.1 Dataset

We crawled over 40,000 pictures from 32 classes according to the data list in [15]. We pick 5 classes according to number of pictures from all data to avoid the imbalance of data. Specifically, the 5 classes are art, design, diy_crafts, film_music_books, food_drink. In each class we randomly select 1000 pictures and its correponding texts to train and test in our experiments. For image



art

Spring Light by Amanda Clark.

design

Oscar Niemeyer, Brazil Rio. Original designeed in 1987.

design

Phillip Grass.

food_drink

Red Velvet Cheesecake.

Figure 3: Here are four examples in our dataset. For each example, the first line represent the category marked by users and we these categories as labels in Experiments. The second line is the picture while the third line is corresponding texts which may be comments or descriptions.

representation, we split the pictures of every class into three parts: training set, validaton set and test set to ensure the generalization performance of the cnn model. The ratio between the training set, the validaton set and the test set is 7:2:1. After the work of feature representation, the image feature dataset and the text feature dataset are splited into two parts with 90% of the dataset as training set and 10% of the dataset as test set in each category. Regarding the hyperparameters, we use five fold cross validation on the trainging dataset and determine models. Finally, we test on the test set based on trained models.

For the labels, we'll use categories indicated by users as labels and encode them as numbers from 0 to 4. For the pictures, we'll resize them to the same length and width, based on which we extract features using cnn. For the texts, as is shown in Figure (3), each picture of dataset used in our experiments has a corresponding textual description. Among the texts there are some punctuation which we will ignore in the lab. Besides, uppercase letters are all converted to lowercase letters. Then we use pre-trained word vectors to vectorize them into features.

4.2 Classification

4.2.1 Result

Table 1 Dimension of features for different models.

Fsize Modal Model	*image*	*word*	*later*	*early*
Initial	4096	300	4396	4396
Bpso	2031	163	2194	2166
Bjaya	2000	152	2152	2170
VFS	2318	290	2608	1175
KFS	2038	273	2311	1014
MFS	3141	299	3440	2102
FFS	3146	273	3419	989
LLE	3272	289	3561	3598
PCA	660	200	860	150
KPCA	647	200	847	151
LSI	889	230	1119	179

The table (1), where Fsize refers to size of features and Model represents the modeled used in fusion while Modal is representative of single modal or muti-modal, displays classification accuracies in our dataset. Each line is responding to a model mentioned before. The table shows the dimensions to which we have reduced the feature after using corresponding models.

Table (2) displas accuracies of classification of dataset under different modals using different models. The first line is the classification accuracy when no feature selection or dimensionality reduction is used. The second to seventh lines are the results of using feature selection, and the eighth to eleventh lines are the results after using dimensionality reduction.

Table 2 Accuracy of classification for different models.

Acc\Modal Algorithm	*image*	*text*	*later*	*early*
Initial	0.6	0.632	0.718	0.708
Bpso	0.63	0.658	**0.748**	**0.74**
Bjaya	0.638	0.66	0.73	0.742
VFS	0.604	0.642	0.726	0.718
KFS	0.624	0.648	0.728	0.722
MFS	0.616	0.638	0.728	0.718
FFS	0.61	0.648	0.726	0.722
LLE	0.618	0.662	**0.742**	**0.744**
PCA	0.618	0.648	0.726	0.73
KPCA	0.616	0.648	0.728	0.73
LSI	0.616	0.652	0.726	0.726

It's obvious that we have improved the accuracies of the classification using these techniques. Among the models of feature selection, Bpso can get good results, achieving an accuracy of 0.748 in the later fusion and an accuracy of 0.74 in early fusion. In the dimensionality reduction models, using LLE can achieve an accuracy of 0.742 in later fusion and an accuracy of 0.744 in early fusion.

Table 3 Time of later fusion.

time/ms		Feature Selection						min
		MFS	FFS	VFS	KFS	Bpso	Bjaya	
mutimodal	later	32.3	28.4	17.5	28.7	21.7	21.6	17.5
	early	32.2	27.7	26.5	26.2	17.8	16.0	16.0

Table 4 Time of early fusion.

time/ms		Dimension Reduction					min
		Initial	LLE	LSI	PCA	KPCA	
mutimodal	later	24.0	5099.1	19.6	26.6	121.3	19.6
	early	12.1	3978.5	22.2	25.0	115.0	12.1

We compare the value of time improved in later and early fusion and present the result in Table (3) and Table (4). The starting point of time for each model is set to the time point when DF/FS starts, and the end point is set to the time when the classification ends. To better present the results, we normalize all the time divided by the time of the initial classification both in later fusion and early fusion. For example, in later fusion, the time of initial model is 2s and the time of VFS is 4s then the normalized time of VFS is $4/2 = 2$s. From table (3) and table (4), we know that For feature fusion, some models increase the processing time of the feature fusion model and some reduce these times whether in Feature Selection or Dimension Reduction. In particular, for the late fusion model, both feature selection and dimensionality reduction increase its time compared with the original model. Besides, from a time perspective, the performance of LLE is poor relatively.

4.2.2 Analysis

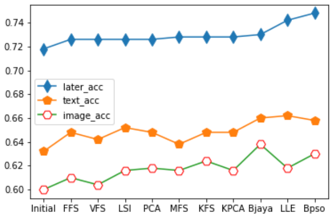


Figure 4: Accuracy of Later Fusion and corresponding single modals

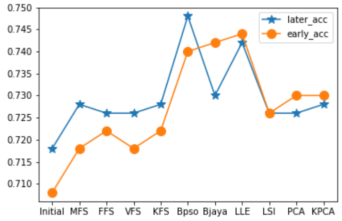


Figure 5: Accuracy of early and later Fusion for different models

Figure (4) shows the classification results of image single mode and text single mode and later fusion. As the accuracy of later fusion continues to increase, the accuracies of corresponding single modals all improve overall. However, it doesn't mean that the higher the accuracies of single modals are, the higher the higher accuracy the later fusion obtain.

Figure (5) shows classification accuracies obtained when using different methods of feature selection or dimension reduction. The leftest column is the pricipal fusion result, and the second to the seventh column are corresponding to different models of feature selection while the eighth to the last are corresponding to models of dimension reduction. Consider the feature selected data as a new dataset different from the original one, we can draw a conclusion that later fusion model and early fusion model we used have different performances on different datasets while the later fusion performances better in most datasets. If we consider these fusion models using different methods of feature selection and dimension reduction as different models for the same dataset, we can draw a conclusion that it's not sure which one is better between later fusion and early fusion because the performances of different models for the same dataset are different.

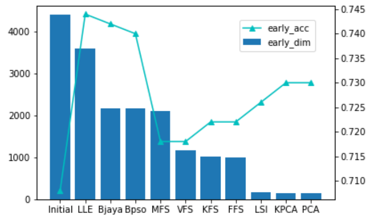


Figure 6: Accuracy and dimension of Early Fusion

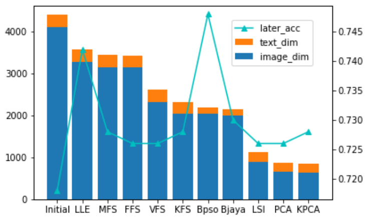


Figure 7: Accuracy and dimension of later fusion

To examine the relation between accuracies and dimensions of classification of early fusion, we plot the accuracies and corresponding dimensions in figure(6). From figure (6) we can see that as the dimensions continues to decline, the accuracies do not decline gradually. In other words, there is no linear relationship between dimemsions and accuracies. We can draw the similar conclusion from figure(7) that there is no direct general relationships between dimensions and accuracies of later fusion for all models.

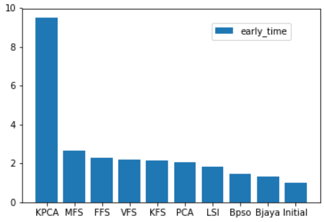


Figure 8: Time of Early Fusion

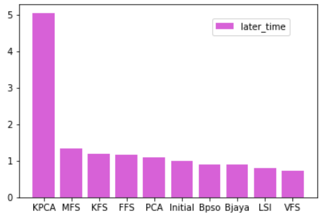


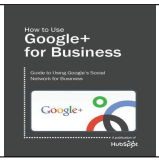
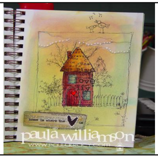
Figure 9: Time of later fusoin

As we all know, the time of classification will decrease as the dimension decreases[LogisticRegression time]. But can the joinging of (FS/DR) decrease the time of fusion model overall? In figure (8) and figure (9), we display the time of processing after FS/DR(included) of later fusion and early fusion as shown in figure (1) and figure(2) for test datasets. To better compare the time differences between different models, we divide the time of all models by the time of the original fusion models, so the values of the ordinate in figure (8) and figure (9) is the ratio between time and time. It's worth noting that inthese two figures we do not add the information about LLE for the time of fusion under LLE is too long, whichis obvious in table (3). From these two figures we know that the time of all methods in early fusion is longer than the original one. And ffter using Bpso, Bjaya, LSI and VFS methods in the later fusion, time of fusion model is lower than the original one. In all methods, the time of models using Bpso and Bjaya is not much different from the original time of no feature selection and dimension reduction. This shows that the difference in time of models under different methods os feature selection or dimension reduction is quite obvious.

4.3 Prediction

Based on the results of previous experiments, after considering the time and accuracy factors of models synthetically, we select Pso model and apply it to fusion model to predict user interests. As is shown in Figure (10), for the same pins we obtain different results of prediction from different models. For the first three examples, classifiers can classify accurately whether Pso is used for feature selection or not. In the last sample, the result of early fusion, later fusion and early fusion with feature selectoin based on Pso all get wrong predictions except for later fusion with Pso. In the fourth example, "looks good to me" is used to describe this food. However, the prediction of Early fusion is "diy_cafts" while the result of early fusion with the help of Pso correct this error. Meanwhile, we can see that the prediction of later fusion obtain a correct prediction while the answer of later fusion based on Pso is wrong. We can draw a conclusion that not all pins using Pso for feature selection have better results than before in later fusion.

Table 5 Prediction of several images of test dataset.

Image					
Text	great for applique	book love	altered art	looks good to me	pattern
Label	diy_crafts	film_music_books	art	food_drink	design
Early	diy_crafts	film_music_books	art	diy_crafts	diy_crafts
PsoEarly	diy_crafts	film_music_books	art	food_drink	diy_crafts
Later	diy_crafts	film_music_books	art	food_drink	diy_crafts
PsoLater	diy_crafts	film_music_books	art	film_music_books	design

We randomly select 20 images and corresponding texts from test dataset and use them to analog users. We use later fusion with feature selection based on Pso and use LogisticRegression to classifier them. We use the distribution of user data under each classes show them in figure (10). In figure (10), the abscissa is enclosed in a circle, and the five discrete points represent five different categories. The ordinate represents the probability distribution of user data for each user under the current category. We can infer the interests of each user broadly from figure (10). It's obvious that user1 is more interested in "diy_crafts" and "film_music_books" while user2 perfers "design". User3 likes "art" best and user4's favorite category is "food_drink". After predicting the users interest,

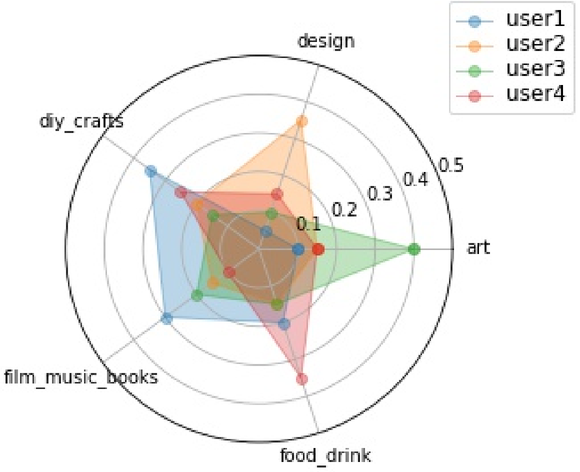


Figure 10: Interests distribution of Users

the website can make recommendations based on this information respectively. According to the distribution of these classification results, the user's interest can be well predicted. At the same time, these classification results can also lay a good foundation for user recommendation based on recommendation algorithm. Better classification results frequently refer to more accurate predictions, which shows that our research has great practical significance.

5 Conclusion

In this article, we apply six methods of feature selection and four methods of dimensionality reduction to later fusion and early fusion of images and texts. These methods can improve the classification results of multimodal fusion more or less. We analyze the relationship between single-modal and multi-modal fusion classification results, and discuss the relationship between dimensions and accuracy. At the same time, we compare the accuracy of different models with the corresponding time. Finally, we predict the user interest based on the multimodal fusion classification with feature selection based on Pso.

In the future work, We will try to use better methods to extract features of images and texts, and apply feature selection and dimensionality reduction techniques to other early fusion methods. In addition, we will further explore the relatioship between the results of single modals and multimodal in early fusion.

References

[1] Baltrušaitis, T., Ahuja, C., and Morency, L.-P. Multimodal machine learning: A survey and taxonomy. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2018).

[2] Berchtold, S., Böhm, C., and Kriegal, H.-P. The pyramid-technique: towards breaking the curse of dimensionality. In *ACM SIGMOD Record* (1998), vol. 27, ACM, pp. 142–153.

[3] Bruni, E., Tran, G. B., and Baroni, M. Distributional semantics from text and images. In *Proceedings of the GEMS 2011 workshop on geometrical models of natural language semantics* (2011), Association for Computational Linguistics, pp. 22–32.

[4] Cheng, A.-J., Chen, Y.-Y., Huang, Y.-T., Hsu, W. H., and Liao, H.-Y. M. Personalized travel recommendation by mining people attributes from community-contributed photos. In *Proceedings of the 19th ACM international conference on Multimedia* (2011), ACM, pp. 83–92.

[5] Cinar, Y. G., Zoghbi, S., and Moens, M.-F. Inferring user interests on social media from text and images. In *Data Mining Workshop (ICDMW), 2015 IEEE International Conference on* (2015), IEEE, pp. 1342–1347.

[6] Cui, B., Tung, A. K., Zhang, C., and Zhao, Z. Multiple feature fusion for social media applications. In *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data* (2010), ACM, pp. 435–446.

[7] Duong, C. T., Lebret, R., and Aberer, K. Multimodal classification for analysing social media. *arXiv preprint arXiv:1708.02099* (2017).

[8] Eberhart, R., and Kennedy, J. A new optimizer using particle swarm theory. In *Micro Machine and Human Science, 1995. MHS'95., Proceedings of the Sixth International Symposium on* (1995), IEEE, pp. 39–43.

[9] Friedman, J., Hastie, T., and Tibshirani, R. *The elements of statistical learning*, vol. 1. Springer series in statistics New York, NY, USA:, 2001.

[10] Guyon, I., and Elisseeff, A. An introduction to variable and feature selection. *Journal of machine learning research 3*, Mar (2003), 1157–1182.

[11] Kennedy, J., and Eberhart, R. C. A discrete binary version of the particle swarm algorithm. In *Systems, Man, and Cybernetics, 1997. Computational Cybernetics and Simulation., 1997 IEEE International Conference on* (1997), vol. 5, IEEE, pp. 4104–4108.

[12] Kiela, D., and Bottou, L. Learning image embeddings using convolutional neural networks for improved multi-modal semantics. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (2014), pp. 36–45.

[13] Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems* (2012), pp. 1097–1105.

[14] Lan, Z.-Z., Bao, L., Yu, S.-I., Liu, W., and Hauptmann, A. G. Multimedia classification and event detection using double fusion. *Multimedia tools and applications 71*, 1 (2014), 333–347.

[15] Mikolov, T., Chen, K., Corrado, G., and Dean, J. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* (2013).

[16] Nguyen, T. T., Duong, C. T., Weidlich, M., Yin, H., and Nguyen, Q. V. H. Retaining data from streams of social platforms with minimal regret. In *Twenty-sixth International Joint Conference on Artificial Intelligence* (2017), no. EPFL-CONF-227978.

[17] Prakash, T., Singh, V., Singh, S., and Mohanty, S. Binary jaya algorithm based optimal placement of phasor measurement units for power system observability. *Energy Conversion and Management 140* (2017), 34–35.

[18] Ramachandram, D., and Taylor, G. W. Deep multimodal learning: A survey on recent advances and trends. *IEEE Signal Processing Magazine 34*, 6 (2017), 96–108.

[19] Rao, R. Jaya: A simple and new optimization algorithm for solving constrained and unconstrained optimization problems. *International Journal of Industrial Engineering Computations 7*, 1 (2016), 19–34.

[20] Schölkopf, B., Smola, A., and Müller, K.-R. Kernel principal component analysis. In *International Conference on Artificial Neural Networks* (1997), Springer, pp. 583–588.

[21] Shen, W., Wang, J., Luo, P., and Wang, M. Linking named entities in tweets with knowledge base via user interest modeling. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining* (2013), ACM, pp. 68–76.

[22] Stevens, J. P. Analysis of covariance. In *Applied Multivariate Statistics for the Social Sciences, Fifth Edition.* Routledge, 2012, pp. 299–326.

[23] Wold, S., Esbensen, K., and Geladi, P. Principal component analysis. *Chemometrics and intelligent laboratory systems 2*, 1-3 (1987), 37–52.

[24] Yin, D., Guo, S., Chidlovskii, B., Davison, B. D., Archambeau, C., and Bouchard, G. Connecting comments and tags: improved modeling of social tagging systems. In *Proceedings of the sixth ACM international conference on Web search and data mining* (2013), ACM, pp. 547–556.

[25] You, Q., Bhatia, S., and Luo, J. A picture tells a thousand wordsabout you! user interest profiling from user generated visual content. *Signal Processing 124* (2016), 45–53.

[26] You, Q., Cao, L., Jin, H., and Luo, J. Robust visual-textual sentiment analysis: When attention meets tree-structured recursive neural networks. In *Proceedings of the 2016 ACM on Multimedia Conference* (2016), ACM, pp. 1008–1017.

[27] You, Q., Luo, J., Jin, H., and Yang, J. Cross-modality consistent regression for joint visual-textual sentiment analysis of social multimedia. In *Proceedings of the Ninth ACM international conference on Web search and data mining* (2016), ACM, pp. 13–22.

[28] Zeppelzauer, M., and Schopfhauser, D. Multimodal classification of events in social media. *Image and Vision Computing 53* (2016), 45–56.

[29] Zheng, W., An, L., and Xu, Z. Dimensionality reduction by combining category information and latent semantic index for text categorization. *JOURNAL OF INFORMATION &COMPUTATIONAL SCIENCE 10*, 8 (2013), 2463–2469.

[30] Zhou, Z.-H. *Machine Learning*. Tsinghua University Press, Beijing:, 2016.