Comparing performance of LDA and TF-IDF on the Five News Groups dataset

Text Mining Project(732A92)

Sridhar Adhikarla
sriad858@student.liu.se
LIU-ID: *sriad858*

May 15, 2020

Abstract

In the current situation, the world is surrounded by news articles. There are hundreds of publications for News articles in America alone, and each publication has a different agenda for publishing news articles. As a part of my Text mining project, I chose to analyse the news articles from some of the American publications, to see if we can successfully identify the publication from the content of the news article. The topic modeling technique, Latent Dirichlet Allocation, and the numerical statistic for textual data, Term Frequency - Inverse Document Frequency, were used to generate features for the classifiers. The performance of these features was evaluated using two classifiers, Multinomial Logistic Regression and Support Vector Machine. The aim of this project was to compare the performance of the feature generating algorithms, Latent Dirichlet Allocation and the Term Frequency - Inverse Document Frequency, on the News Articles dataset. Experimental results show that the Term Frequency - Inverse Document Frequency gives a better overall accuracy with both the classifiers on the News articles dataset.

1 Introduction

In 1800's newspapers became a medium to deliver news, in 1900's the radio came into picture and it became the new medium to deliver news, now in the 2000s news is delivered in various formats through the internet. Today, news articles from various publications are printed, broadcasted over television and also published online. The dataset used in this project [1], was obtained from *Kaggle*. It contains articles from Twenty American news groups, with approximately 150,000 news articles. Due to lack of computational resources, only five of these twenty groups was selected for this project. The dataset for the five news groups contains 53,985 news articles.

Supervised Topic modeling is a commonly used method for extracting features from textual data. The motivation behind majority of the supervised topic models is to find good classification models, with high predictive performance. Latent Dirichlet Allocation

(LDA) [2], is a commonly used topic model for document classification tasks. Jonsson et. al. in their paper [3], suggested the Diagonal Orthant Latent Dirichlet Allocation (DOLDA) algorithm for high dimensional multi-class regression problems. DOLDA is a combination of the LDA and the Diagonal Orthant probit model [4]. Xiea et. al. in their paper [5], combined LDA with Support Vector Machines (SVM) [6] classifier, to solve the problem of multi-class text categorization.

Another frequently used method for extracting features from textual data is Term Frequency - Inverse Document Frequency (TF-IDF) [7], which is a weighting scheme for the words in a document. Hakim et. al. in their paper [8], classified news article in Bahasa Indonesia, with a high prediction accuracy of 98.3%. They got the best results when using SVM with the TF-IDF features.

This project will compare the predictive performance of the LDA and TF-IDF features, when used with the Multinomial Logistic Regression and SVM classifiers, on the News articles dataset. LDA gives a probability distribution over the number of topics, which will be used as a vector of features for an article to train the classifiers. TF-IDF gives a vector of weights, weighing each of the words in the article, which is used as vector of features to train the classifiers. The predictive performance of using these methods on the news articles dataset to predict the publication of the news will be used to evaluate these methods.

2 Data Preprocessing

Each article in the dataset contains a title, content, date on which it was published and the name of publication, which is used as the response variable for this project. The title and the content fields of the article are free unstructured text fields, which require some preprocessing and feature extraction before they can be used as features for a classifier. Since, this project is about comparing the performance of the feature extractors from news articles, LDA [2] and TF-IDF [7], the date field is discarded. Only the textual fields from the article are used to classify the news articles.

Basic preprocessing of the textual fields is carried out using *RegEx*. All the special symbols in the text are removed. Only the words, numbers and alphanumeric from the text are used for feature extraction. The articles are tokenized and the standard English stop words are removed. Each token is also lemmatized to its root form using the *NLTK* library in python.

The dataset is then divided into three parts, *train* with 34,000 articles, *validation* with 8,500 articles and *test* with 10,500 articles. The *train* dataset is used to train the feature extractors and classifier, the *validation* dataset is used to find the best hyper-parameter values and the *test* dataset will be used to evaluate the feature extractors.

3 Theory

This section describes the methods used in this project.

3.1 Term frequency - Inverse Document Frequency

Term Frequency - Inverse Document Frequency (*TF-IDF*) [7], is a numerical statistic that shows how important a word is to a document in a corpus. The *TF-IDF* value increases proportionally with the number of times a word appears in a document and is offset by the number of documents in the corpus that contain the word.

The number of times a word occurs in a document is called *Term Frequency(tf)*. There are many forms of calculating the *tf*, the one used in this project is called Augmented Term Frequency. It is calculated as the raw frequency of a term in the document divided by the raw frequency of the most popular term in the document. This prevents a bias towards longer documents. The *tf* value is calculated as:

$$tf(w,d) = 0.5 + 0.5\frac{f_{t,d}}{max\{f_{t',d} : t' \in d\}}, \tag{1}$$

where $f_{t,d}$ is the frequency of the term in the document and $max\{f_{t',d} : t' \in d\}$ is the frequency of the most popular term in the document.
The *Inverse Document Frequency(idf)* is a measure of how much information the term provides. It is calculated as:

$$idf(w,D) = log\frac{N}{1 + |\{d \in D : w \in d\}|}, \tag{2}$$

where N is the total number of documents in the corpus $N = |D|$ and $|\{d \in D : w \in d\}|$ is the number of documents where the word w appears (also known as document frequency).
The *TF-IDF* is calculated as a product of the *tf* and the *idf*:

$$tfidf(w,d,D) = tf(w,d) \cdot idf(w,D). \tag{3}$$

The words that have high term frequency and low document frequency in the corpus have the highest *tf-idf* value [7]. While calculating the *idf*, as the ratio:

$$\frac{N}{1 + |\{d \in D : w \in d\}|} \geq 1, \tag{4}$$

The value of *idf* and *tf-idf* is always greater than zero.
This project uses an implementation of *TF-IDF* from the python library, `scikit-learn` [9].

3.2 Latent Dirichlet Allocation

Topic modeling is a type of statistical modeling technique in text classification, used for discovering abstract topics that occur in a collection of documents. It is similar to soft clustering on numerical data, where algorithms try to find some natural groups of items. A topic is defined by a distribution over the words and as in soft clustering, each document can be assigned to multiple topics with different probabilities. LDA [2], is an unsupervised topic modeling technique, which uses a set of unobserved groups or topics to explain the observations or documents.

LDA relies on the Dirichlet and the Multinomial distributions to generate the topics from the documents collection [2]. LDA makes the assumption that the documents collection was created from a generative process and tries to reverse engineer the process. Random mixtures over latent topics are used to represent documents, where each of the topics are further categorized by a distribution over the words in the corpus.
Assume a document corpus D which contains M documents, each of length N_i where $i = 1, ..., M$,
Let the total number of unique words in the document corpus, the vocabulary of the document corpus, be of length V,
A generative process with K topics looks like:

1. For each topic $k = 1, ..., K$:
   (a) Simulate a distribution over words $\varphi_k \sim Dirichlet_V(\beta)$
2. For each document $d = 1, ..., M$:
   (a) Simulate topic proportions $\theta_{d|\alpha} \sim Dirichlet_K(\alpha)$
   (b) For $i = 1, ..., N_d$:
      i. Simulate a topic assignment $z_{i,d}|\theta_d \sim Multinomial(\theta_d)$
      ii. Draw a word $w_{i,d}|z_{i,d}, \varphi_{z_{i,d}} \sim Multinomial(\varphi_{z_{i,d}})$

where

- Model Parameters:
  - β is the parameter of the Dirichlet prior on the per-topic word distribution
  - α is the parameter of the Dirichlet prior on the per-document topic distribution
- The unknowns of the model to infer are:
  - φ_k is the vector of word probabilities for each topic k
  - θ_d is the topic proportions for the document d
  - z_d is the topic assignment for the words

A Dirichlet prior is introduced for all the unknown θ_d, z_d and φ_i, to obtain a joint posterior distribution over the variables. As the Dirichlet distribution is conjugate to the Multinomial distribution, the joint posterior distribution follows a Dirichlet distribution:

$$p(\theta_{1:D}, z_{1:D}, \varphi_{1:K}|w_{1:D}) \propto \prod_{i=1}^{K} p(\varphi_i|\beta) \prod_{d=1}^{D} p(\theta_d|\alpha) \left(\prod_{n=1}^{N} p(z_{d,n}|\theta_d) p(w_{d,n}|\varphi_{1:K}, z_{d,n}) \right). \quad (5)$$

The authors of LDA [2], used Gibbs Sampling to obtain $z_{1:D}$, $\theta_{1:D}$ and $\varphi_{1:K}$. An implementation of LDA from the python library `Gensim`, will be used in this project. This LDA implementation in `Gensim` [10], uses Variational Bayesian Sampling to sample from the joint posterior.

Using LDA, each document d is summarized as a probability distribution over the K topics. These vectors of probabilities for the documents are used as features for supervised classification models [3].

3.3 Multinomial Logistic Regression

Logistic Regression [11], is a statistical method that is used when the dependent variable is nominal with two levels. For problems where the dependent variable is nominal with more than two levels, *Multinomial Logistic Regression*, which is a generalization of the *logistic regression* to multi-class problems, is used.

Similar to *logistic regression*, the *multinomial logistic regression* uses linear combination of the independent variables to predict the dependent nominal variable [11]. It trains a different linear combination of the independent variables for each of the categories in the dependent variable. The outputs of the linear combinations are combined to get the probabilities for each class, and to make a prediction the class with the highest probability is selected.

This project uses an implementation of *Multinomial Logistic Regression* from the python library `scikit-learn` [9].

3.4 Support Vector Machine

A Support Vector Machine (SVM) [6], is a classifier which learns the features in input that are most useful to distinguish between various possible classes, formally defined by a separating hyper-plane. SVM is a non-probabilistic binary linear classifier, but it can efficiently perform non-linear classification using the kernel trick. The kernel trick implicitly maps the inputs into high dimensional feature spaces.

For a text classification task, since the input features to the SVM will be high dimensional vectors, a linear kernel will be used in this project. This project uses an implementation of *SVM* from the python library `scikit-learn` [9].

4 Results

This section will present the results obtained in this project. The train dataset was used to train the feature extractors and the classifiers, the validation dataset was used to choose the best hyper-parameter values and the test dataset was used to preset the final results of overall accuracy.

4.1 Choice for hyper-parameters

This section will show the results that motivated the choice of the hyper-parameters for the textual feature extraction algorithms, LDA and TF-IDF.

4.1.1 Number of topics for LDA

This section motivates the choice of number of topics for LDA. To find the best value for number of topics, a grid of 10 values in the range 20 to 200 were chosen, and LDA models were trained using the train dataset. Figure 1, shows the prediction accuracy of the Logistic and the SVM classifiers when using the LDA feature vectors with different number of topics.

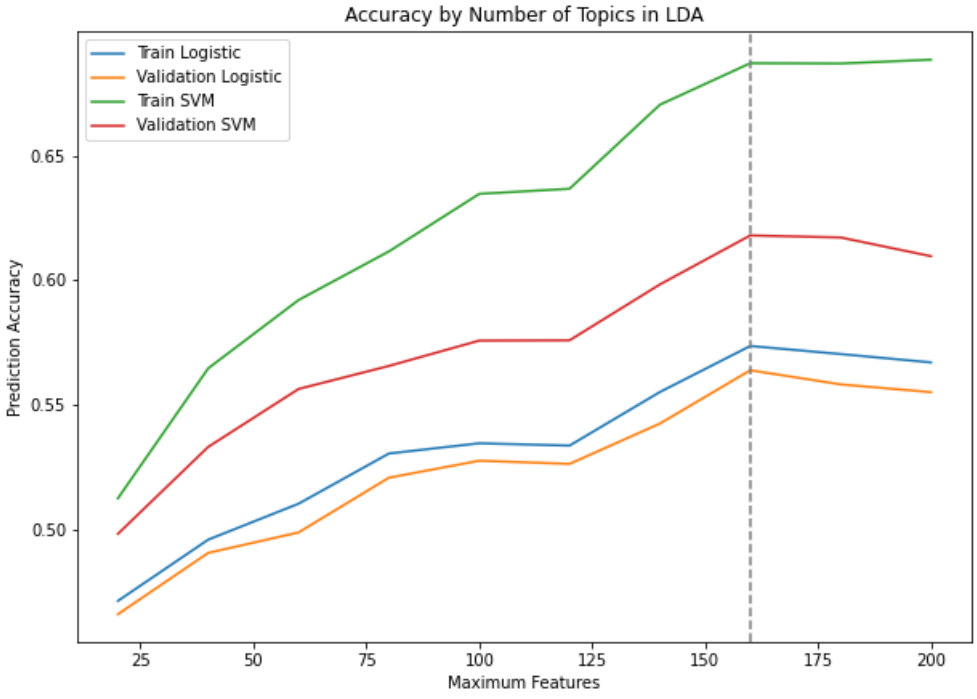

Figure 1: Accuracy of Logistic Regression and SVM for different number of LDA topics.

It can be seen that the validation accuracy for both the classifiers starts to decline after 160 topics. Therefore, 160 topics is the final model chosen to present the results using LDA.

The best validation accuracy obtained was when LDA topics were used as features on the SVM classifier, with the highest validation accuracy of 61.8%.

Figure 2, tries to visualize the LDA topics vectors for the articles. Since, 160 topics gave the best predictive performance, it was chosen for the visualization. The dimensionality of the vectors are reduced to 2 dimensions using an implementation of *TSNE*, from the *sklearn* package in python.

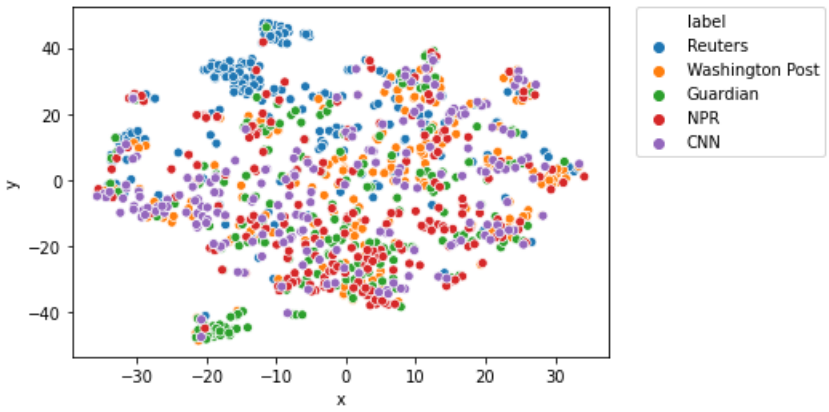

Figure 2: Visualizing the article vectors for 160 LDA topics.

Hundred random articles from each publication in train dataset was chosen for the visualization. No clear separation can be seen from this visualization, the classes *Washington Post*, *Guardian*, *CNN* and *NPR* are all mixed up. The articles from *Reulters* look like they are clustered at the top of the plot.

4.1.2 Number of words for TF-IDF

This section motivates the choice of the max number of words to be used in the TF-IDF feature vectors. The Multinomial Logistic Regression and the SVM classifiers were used to motivate the choice. The Multinomial Logistic Regression was trained on the article TF-IDF feature vectors of 30 different lengths in the range 1000 to 30,000 and the SVM classifier was trained on the article TF-IDF feature vectors of 10 different lengths in the range 1000 to 10,000. Figure 3, shows the train and the validation accuracy for different TF-IDF vector lengths.

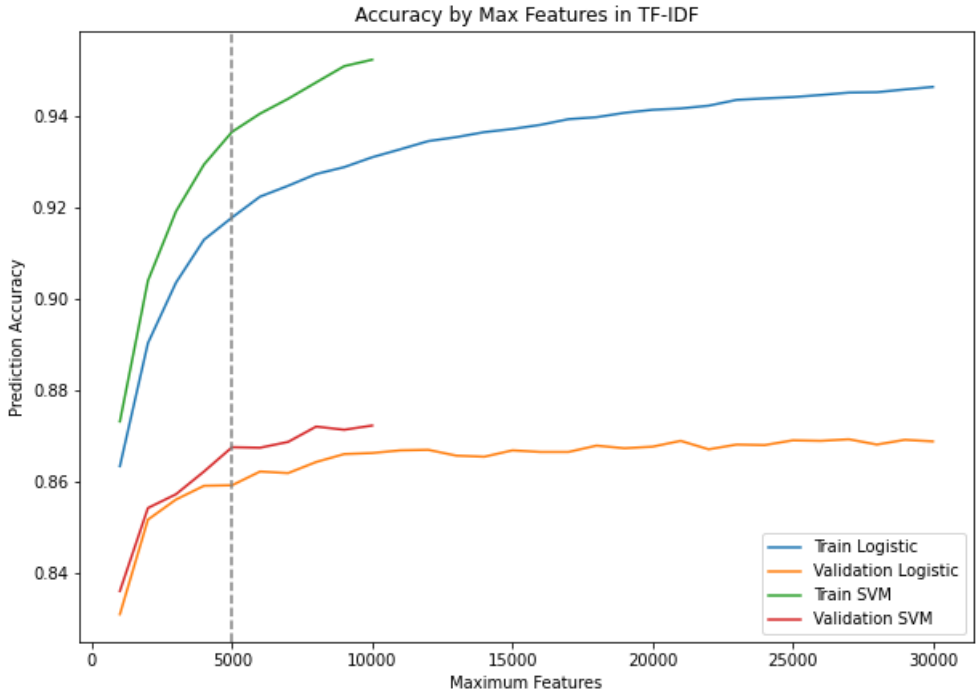

Figure 3: Accuracy of Logistic Regression and SVM for different number of TF-IDF features.

It can be seen that the overall accuracy on the validation data, for both the classifiers, does not increase much after 5000. Therefore, 5000 is chosen as the best value for the max number of words to be used from TF-IDF. The SVM classifier, performs better than the Logistic classifier in the range it was trained, with the best accuracy of 87.2% on the validation dataset.

4.2 Overall accuracy

This section will show the final results after using the best hyper-parameter values obtained from the previous section. The Table 1 shows the overall accuracy obtained on the test dataset:

Nr.	Classifier	Prediction Accuracy	
		LDA 160 topics	TF-IDF 5000 features
1	Multinomial Logistic Regression	0.5676	0.8713
2	Support Vector Machine	0.6213	0.8778

Table 1: Overall accuracy for different classifiers with different features as input.

It can be clearly seen that the TF-IDF features perform much better on the News articles dataset for both the classifiers.

5 Discussion

This section will discuss the results obtained in further detail.

5.1 Number of topics for LDA

The semantic relation in textual data is usually ignored. The LDA model learns the semantic relation from the data and gives a probability distribution over number of topics for each article. The authors of DOLDA [3] found 40 and 100 topics to give the best results when working with the bug reports dataset. The authors of [5], found a similar increase in prediction accuracy until 100 topics and then a decline. This is the reason, this project tries to explore different number of topics for LDA and their predictive performance using the News articles data. A grid of values in the range 20 - 200 were used in the experiment and best predictive performance on the validation dataset was observed for 160 topics.

5.2 Number of words for TF-IDF

The train dataset contains approximately 150,000 unique words. If we don't select limit to the maximum number of words to be used with TF-IDF, we would get feature vectors of length 150,000. Not all the words contain valuable information, and it would require a lot of memory and computational power to train a classifier with such large feature vectors. This is the reason a limit to the maximum number of words is selected when using TF-IDF. The words with the highest TF-IDF values are selected, and the weights given to these words for each article is used as a feature vector for the article.

Figure 3, shows that after 5000, the validation accuracy does not increase much. The train time for the classifiers keep increasing linearly, but the validation accuracy curve is starting to flatten. This is the reason 5000 features was chosen as the best hyper-parameter for TF-IDF.

5.3 Comparison to previous work

A lot of studies have been done on the 20 News Groups dataset. Some of the best results obtained are summarized in the Table 2.

Nr.	Method	Prediction Accuracy
1	SGC	88.5
2	NABoE-full	88.1
3	RMDL	87.91
4	Graph Star	86.9

Table 2: Accuracy obtained by other studies on 20 News Groups dataset.

Simplifying Graph Convolutional Networks (SGC) [12], is a variant of the Graph Convolutional Networks (GCN), which tries to reduce excess complexity in the GCN models. It

gets the best predictive performance of 88,5% on the News dataset.

Neural Attentive Bag-of-Entities (NABoE) [13], proposes a neural network model that performs text classification using entities in a knowledge base. Entities help capture the semantics in the text. This model combines entity detection with a novel neural attention mechanism, which helps it achieve the state of the art result on the dataset. This model performs close to the SGC model, with an accuracy of 88.1%.

Random Multimodel Deep Learning (RMDL) [14], uses an ensemble deep learning approach to solve the problem and Graph Star Net [15], uses a novel and unified graph neural net architecture.

The results of this project are not directly comparable to these studies, as the project uses just a subset of the dataset with 5 News groups. The prediction accuracy on the test dataset obtained in this project 87.7%, which is close to these studies, but it is just for 5 of the 20 News groups.

The main aim for this study was to compare the predictive performance of the TF-IDF and LDA methods for feature extraction from the textual fields of the News articles dataset. The semantic feature of the text are learned using the LDA method, but that did not improve the predictive performance for this dataset. TF-IDF helps identify important differentiating keywords or pairs of words, which helps improve the predictive performance of the classifiers.

6 Problems Faced

The TF-IDF vectors for the news articles gave a good predictive performance with the classifiers, but since the TF-IDF vectors are high dimensional, the SVM classifier took a long time to train. The training time for the SVM classifier, for every 1000 extra TF-IDF features, increased by 1 hour. The computational complexity due to the large dataset was the only problem faced during this project.

7 Conclusion

The best accuracy obtained in this project of predicting the publication of news articles was 87.78%, using 5000 TF-IDF features with the SVM classifier. This is very promising results, given the simplicity of the models used in this project. The results can be further improved finding optimal hyper-parameter values for the classifiers as well, like trying different kernels for the SVM classifier. This project was trying to compare the predictive performance of the TF-IDF and the LDA methods on the news articles data, and tried to find the best parameters for these feature extractors. The results obtained clearly show that the TF-IDF features perform much better than the LDA topics for the News articles dataset.

References

[1] Andrew Thompson. "All the news." In: (2017). URL: https://www.kaggle.com/snapcrack/all-the-news.

[2] David M. Blei, Andrew Y. Ng., Michael I. Jordan. "Latent Dirichlet Allocation." In: *Journal of Machine Learning Research, Issue 3, Pages 993-1022* (2003).

[3] M. Magunsson, L. Jonsson, M. Villani. "DOLDA - A regularized supervised topic model for high-dimensional multi-class regression." In: (2016).

[4] James E. Johndrow, Kristian Lum, David B. Dunson. "Diagonal Orthant Multinomial Probit Models." In: (2013).

[5] Kunlun Lia, Jing Xiea, Xue Sunb, Yinghui Ma, Hui Bai. "Multi-class text categorization based on LDA and SVM." In: (2011).

[6] Cortes, Corinna and Vapnik, Vladimir. "Support-vector networks." In: *Machine learning* (1995).

[7] Sammut C., Webb G.I. "TF–IDF." In: *Encyclopedia of Machine Learning. Springer, Boston, MA* (2011).

[8] A. A. Hakim, A. Erwin, K. I. Eng, M. Galinium and W. Muliady. "Automated document classification for news article in Bahasa Indonesia based on term frequency inverse document frequency (TF-IDF) approach." In: *6th International Conference on Information Technology and Electrical Engineering (ICITEE)* (2014).

[9] Pedregosa, F. and Varoquaux, G. and Gramfort, A. and Michel, V. and Thirion, B. and Grisel, O. and Blondel, M. and Prettenhofer, P. and Weiss, R. and Dubourg, V. and Vanderplas, J. and Passos, A. and Cournapeau, D. and Brucher, M. and Perrot, M. and Duchesnay, E. "Scikit-learn: Machine Learning in Python." In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[10] RadimŘehůřek and Petr Sojka. "Gensim: Software Framework for Topic Modelling with Large Corpora." In: *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks* (2010).

[11] C.M. Bishop. "Pattern recognition and machine learning." In: *12th ed. Springer, pp.32–33, 225-284, 197–209, 685–691* (2006).

[12] Felix Wu, Tianyi Zhang, Amauri Holanda de Souza Jr., Christopher Fifty, Tao Yu, Kilian Q. Weinberger. "Simplifying Graph Convolutional Networks." In: (2019).

[13] Ikuya Yamada, Hiroyuki Shindo. "Neural Attentive Bag-of-Entities Model for Text Classification." In: (2019).

[14] Kamran Kowsari, Mojtaba Heidarysafa, Donald E. Brown, Kiana Jafari Meimandi, Laura E. Barnes. "RMDL: Random Multimodel Deep Learning for Classification." In: (2018).

[15] Lu Haonan, Seth H. Huang, Tian Ye, Guo Xiuyan. "Graph Star Net for Generalized Multi-Task Learning." In: (2019).