# Towards Supporting Visual Question and Answering Applications

by

Qiongjie Tian

A Dissertation Presented in Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

Approved June 2017 by the
Graduate Supervisory Committee:

Baoxin Li, Chair
Hanghang Tong
Hasan Davulcu
Yezhou Yang

ARIZONA STATE UNIVERSITY

August 2017

# ABSTRACT

Visual Question Answering (VQA) is a new research area involving technologies ranging from computer vision, natural language processing, to other sub-fields of artificial intelligence such as knowledge representation. The fundamental task is to take as input one image and one question (in text) related to the given image, and to generate a textual answer to the input question. There are two key research problems in VQA: image understanding and the question answering. My research mainly focuses on developing solutions to support solving these two problems.

In image understanding, one important research area is semantic segmentation, which takes images as input and output the label of each pixel. As much manual work is needed to label a useful training set, typical training sets for such supervised approaches are always small. There are also approaches with relaxed labeling requirement, called weakly supervised semantic segmentation, where only image-level labels are needed. With the development of social media, there are more and more user-uploaded images available online. Such user-generated content often comes with labels like tags and may be coarsely labelled by various tools. To use these information for computer vision tasks, I propose a new graphic model by considering the neighborhood information and their interactions to obtain the pixel-level labels of the images with only incomplete image-level labels. The method was evaluated on both synthetic and real images.

In question answering, my research centers on best answer prediction, which addressed two main research topics: feature design and model construction. In the feature design part, most existing work discussed how to design effective features for answer quality / best answer prediction. However, little work mentioned how to design features by considering the relationship between answers of one given question. To fill this research gap, I designed new features to help improve the prediction performance. In the modeling part, to employ the structure of the feature space, I proposed an innovative learning-to-rank model by considering the hierarchical lasso. Experiments with comparison with the state-of-the-art in the best answer prediction literature have confirmed that the proposed methods are effective and suitable for solving the research task.

*I dedicate this dissertation to my wife, Yashu Chen,*

*for her care and love throughout my Ph.D. study.*

# ACKNOWLEDGMENTS

First and foremost, I would like to thank Professor Baoxin Li. Without his help, it would not be possible for me to complete the Ph.D. study. He is a dedicated, outstanding mentor. His kindly encouragement and endless professional support always guide me to become an excellent researcher. He never hesitates sharing his time and energy to help me. Besides above, he is a helpful friend in my personal life. He offers me lots of suggestions on how to enrich my life in United States. With his help, I not only have gained much valuable research experience, but also enjoyed a good life during my time period in Arizona. In addition, I very appreciate it that Prof. Li provided a large number of suggestions on how to successfully obtain internships and full-time jobs.

I would also like to thank Dr. Hanghang Tong, Dr. Hasan Davulcu and Dr. Yezhou Yang for their helpful advice on my research proposal and Ph.D. dissertation, and for sharing their precious time to serve as my dissertation committee members.

Furthermore, I would like to thank Lixue Zhu who was my manager when I worked as a intern at SurveyMonkey Inc. With her guidance, I gained my first piece of industry experience in my life and had a meaningful summer in Palo Alto, CA. Additionally, I would like to thank another team member Akanksha Bhardwaj who assisted me in many of my internship projects.

My colleagues in Professor Baoxin Li's research group inspired me a lot through my entire Ph.D. study. Without their help and project cooperations, I would not be able to finish my dissertation. They are helpful and willing to share their time and energy on my research problems and projects. I learned a lot from them through their important feedback. These are my previous and current team members and my friends: Dr. Qiang Zhang, Nan Li, Dr. Zheshen Wang, Dr. Peng Zhang, Dr. Devi Archana Paladugu, Dr. Zhigang Tu, Dr. Ke Bai, Jing Lu, Hima Bindu, Jun Cao, Parag Chandakkar, Ragav Venkatesan, Yilin Wang, Xu Zhou, Yikang Li, Yuzhen Ding, Jaya Gattupalli, Pak Lun Ding, Jashmi Lagisetty.

Lastly, my heartfelt appreciations go to my wife, Dr. Yashu Chen, and my parents Dianli Tian and Jintian Wang. Their love, care and unconditional support encouraged me to complete my graduate study.

## TABLE OF CONTENTS

Page

APPENDIX

# LIST OF TABLES

Table Page

# LIST OF FIGURES

Figure Page

## Chapter 1

# INTRODUCTION

Vision is one important function we have to access our world, however there are a lot of people who are visually impaired. According the statistics from National Health Interview Survey (NHIS) 2015, in U.S., there are 23.7 million adult reported vision loss, while in the world, there are about 285 million people who are visually impaired. To help them live independently, there is a lot of existing research in this regard. Baker *et al.* (2016) proposes a new tactile system which substitutes QR codes for text and can help blind students who are not familiar with Braille. In Fusco and Morash (2015), Giovanni Fusco *et al.* proposes one computer-vision based approach to help individuals with visual impairments to read the tactile graphics by tracking their fingers. Nevertheless most of these research on the tactile graphics cannot give the semantic information to the individuals with visual impairment directly. For example, given one image, it is difficult to get the semantic information from the transformed tactile graphics for the blind. To deal with this problem, visual question answering becomes a new and promising research topic. In Bigham *et al.* (2010), authors propose one system (VizWiz) to answer the questions related with given image. The system works as follows: one user takes one picture using his or her phone and then asks one question about this image; then the remote workers receiving this particular question will send the answer back. But these research requires lots of manual work. In order to automate the entire system, another new field came into being, which is visual question answering.

As shown in Antol *et al.* (2015), visual question answering (VQA) is a combination of Computer Vision, Natural Language Processing and Reasoning. The input of VQA is one image and one content related natural language question while the output is an text-based answer to the given question. The input question can be very simple with *yes* or *no* answers.

It can also be one complex question requiring reasoning and detailed content information from the input image, for example, “How many balls are there?” and “what kind of store is this?”

So far there are two main strands of research on VQA. One is charactered by the fact that output answers are open-ended. A demo is shown in Figure 1.1 which is from VQA Dataset [1] . The other one takes more information as multiple answer choices as input and outputs which one is correct (see Fig. 1.2).

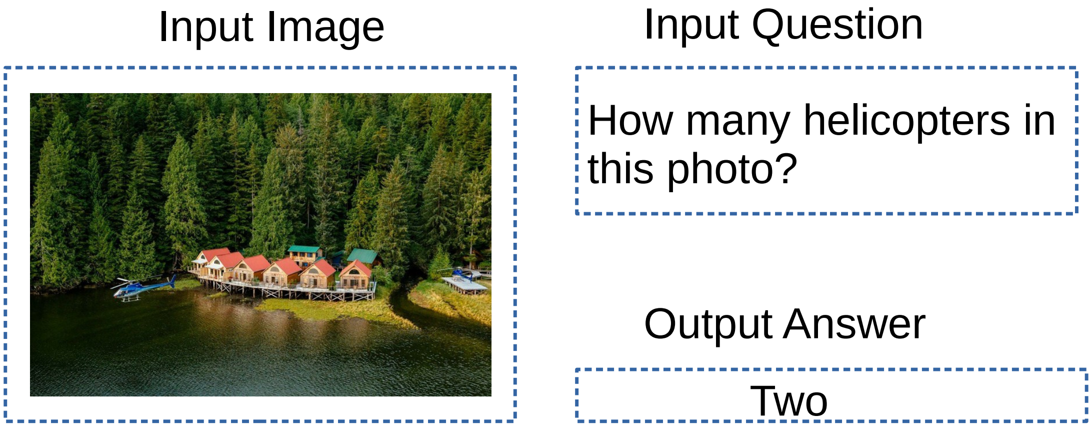
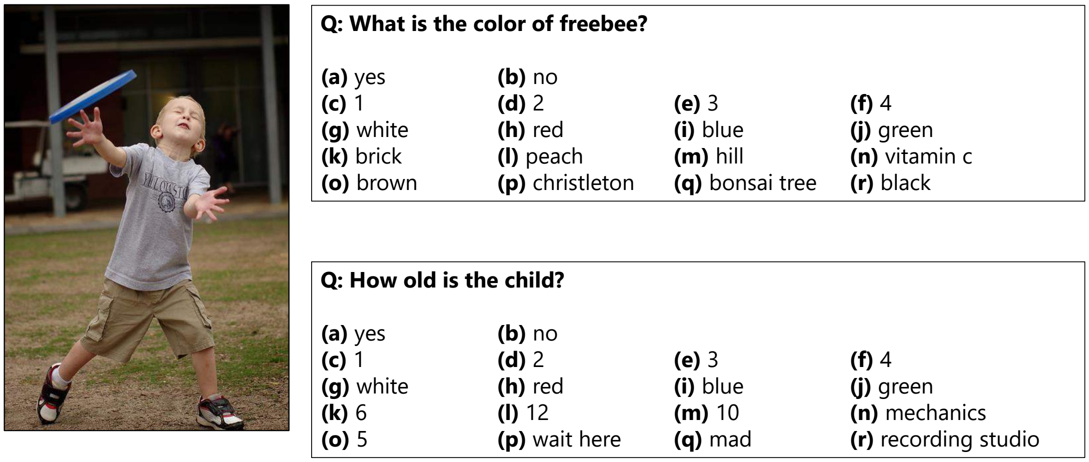


Figure 1.1: This figure shows how visual QA works where the output is open ended.

To support the applications in VQA, my dissertation focuses on two main topics. The first is to understand the content of a given image, which involves techniques from image understanding (e.g., semantic segmentation). The second is to answer given questions.

In the area of image understanding, one important research area is semantic segmentation, which takes images as input and outputs the label of each pixel. In the existing literature, many research works concentrate on the supervised semantic segmentation. These works assume that training images have all pixel-level labels. For existing research related with weakly supervised semantic segmentation, it requires dataset containing images with complete image-level labels, which needs much manual work to generate training dataset.

[1] VQA data: `http://www.visualqa.org/download.html`



**Q: What is the color of freebee?**

| | | | |
|---|---|---|---|
| **(a)** yes | **(b)** no | | |
| **(c)** 1 | **(d)** 2 | **(e)** 3 | **(f)** 4 |
| **(g)** white | **(h)** red | **(i)** blue | **(j)** green |
| **(k)** brick | **(l)** peach | **(m)** hill | **(n)** vitamin c |
| **(o)** brown | **(p)** christleton | **(q)** bonsai tree | **(r)** black |

**Q: How old is the child?**

| | | | |
|---|---|---|---|
| **(a)** yes | **(b)** no | | |
| **(c)** 1 | **(d)** 2 | **(e)** 3 | **(f)** 4 |
| **(g)** white | **(h)** red | **(i)** blue | **(j)** green |
| **(k)** 6 | **(l)** 12 | **(m)** 10 | **(n)** mechanics |
| **(o)** 5 | **(p)** wait here | **(q)** mad | **(r)** recording studio |

Figure 1.2: One demo demonstrates the multiple-choice questions for VQA. It consists of one image, two questions and answer candidates.

As social media develops, there are more and more user-uploaded images available on-line (e.g., Flickr). However, as one kind of user-generated content, it is difficult to get the pixel-level labels, even the complete image-level labels. It is inefficient to label these images manually, but the incomplete image-level labels are easy to obtain. To generate pixel-level labels of images with only incomplete image-level labels, I propose a new graphic model by utilizing the pixel neighborhood information. Several experiments are conducted on different commonly used datasets to demonstrate the performance of the proposed algorithm.

In question answering, my research centered on best answer prediction. At the feature design part, most existing work discussed how to design effective features for answer quality / best answer prediction from different aspects. However, little research mentions how to design features by considering the relationship between possible answers of one given question. To fill this research blank, I designed new features to help improve the model performance for the research problem of best answer prediction. Experiments show the effectiveness of proposed features. Some research on Twitter also shows how to design effective features for data from community sites. In the data modeling part, I propose a new learning-to-rank model by considering the hierarchical lasso showing the influence of the structure of feature space. In this model, I assume that there exists one hierarchical structure in the feature space. Comparison with the state of the art in the best answer prediction confirms my assumption and demonstrates that the proposed learning-to-rank technique is suitable for solving the research problem.

### 1.1 Weakly Semantic Segmentation via Generalized Conditional Random Field

Semantic segmentation, by which an image is decomposed into regions with their respective semantic labels, is often the first step towards image understanding (Figure 1.3 shows clearly what semantic segmentation is [2]). Existing research in this regard is mainly performed under two conditions: the fully-supervised setting that relies on a set of images with pixel-level labels and the weakly-supervised one that uses image-level labels. In both cases, the labeling task is time-consuming and laborious, and thus training data are always limited. In practice, there are voluminous on-line images, which unfortunately often have

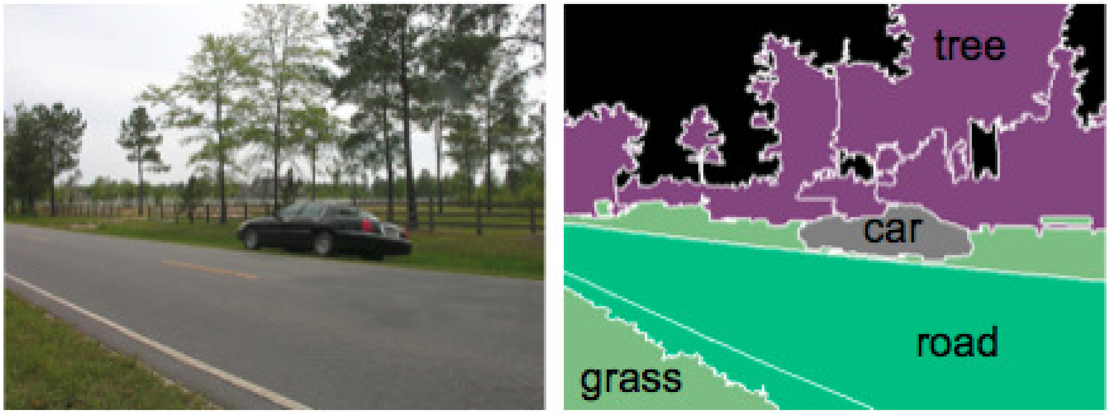


Figure 1.3: Illustration of semantic segmentation. The left one is the input image, while the right one is the output which has the label of each pixel (this figure is from CVPR 2013 tutorial).

[2] http://cvn.ecp.fr/tutorials/cvpr2013/

only incomplete image-level labels (tags) but would otherwise be potentially useful for a learning-based algorithm. Only limited efforts have been attempted on using such coarsely and incompletely labelled data for semantic segmentation. For this piece of work, I propose a new approach to semantic segmentation of a set of partially-labelled images, using a formulation considering information from multiple visual similar images. Details are shown in Chapter 3.

## 1.2 New Feature Design Method for Best Answer Prediction

Community-based question-answering (CQA) services contribute to solving many difficult questions we have. For each question in such services, one best answer can be designated, among all answers, often by the asker. However, many questions on typical CQA sites are left without a best answer even if when good candidates are available. In this part, we attempt to address the problem of predicting if an answer may be selected as the best answer, based on learning from labeled data. The key tasks include designing features measuring important aspects of an answer and identifying the most importance features. Experiments with a Stack Overflow dataset show that the contextual information among the answers should be the most important factor to consider. Details are shown in Chapter 4.

## 1.3 New Learning-to-rank Approach to Best Answer Prediction

In community question and answering sites, pairs of questions and their high-quality answers (like best answers selected by askers) can be valuable knowledge available to others. However lots of questions receive multiple answers but askers do not label either one as the accepted or best one even when some replies answer their questions. To solve this problem, high-quality answer prediction or best answer prediction has been one of important topics in social media. These user-generated answers often consist of multiple "views", each capturing different (albeit related) information (e.g., expertise of the asker, length of the answer, etc.). Such views interact with each other in complex manners that should carry a lot of information for distinguishing a potential best answer from others. Little existing work has explored such interactions for better prediction. To explicitly model these information, we propose a new learning-to-rank method, ranking support vector machine (RankSVM) with weakly hierarchical lasso in this section. The evaluation of the approach was done using data from Stack Overflow. Experimental results demonstrate that the proposed approach has superior performance compared with approaches in state-of-the-art. Details are shown in Chapter 5.

# Chapter 2

# FOUNDATIONS AND PRELIMINARIES

In this chapter, some preliminary knowledge is introduced including graphical models and learning to rank techniques.

## 2.1 Graphical Model

There are two important and popular graphical models which are commonly used in the computer vision area: one is Markov Random Field and the other one is Conditional Random Field. One of the main assumptions that underlie these random field models is that the input variables are not totally independent to each other but there are some structural interactions between them.

### *2.1.1 Markov Random Field*

This model is used in low-level image processing tasks for example image de-noising. Let us assume that there are several variables which are annotated as $X = \{x_i, i \in \{1, 2, ..., n\}\}$. These variables are not independent and identically distributed. One variable's value is dependent on other variables. Let us denote that one variable $x_i$ is related with variables in one subset $N_i$ but is independent with the others except $N_i$. Here each variable is usually named as one *site* while the corresponding $N_i$ is named as *neighorhood*. Then we can have this probability equation for Markov Random Field models (see Eqn.2.1)

$$P(X) = \Pi P(x_i | N_i) \tag{2.1}$$

We can draw these variables as nodes in a undirected graph. If two nodes are neighbors to each other, then we draw one edge between them. Then we obtain the graph representation of a given Markov Random Field. In order to reduce the computation cost, it is common to consider first order and second order neighborhood without high-order information. According to Hammersley Clifford Theorem Besag (1975), the cost function has been split into two parts: unary potentials and the pairwise potentials. Unary potentials are cost incurred by the first order neighborhood only. This cost measures the mismatch between the groundtruth and predicted values. In Markov Random Field models, one important constraint is that variables insides one neighborhood are likely to have same values (labels). This constraint is shown as pairwise potentials in the cost function. For example, in the Ising model in image restoration Geman and Geman (1984), we can see that the pairwise potentials are measured as follows: if two pixels have the same values, pairwise potentials are zeros, while if they have different values, then the potentials are set to be large values. The cost function for Markov Random Field with the first order and the second order neighborhood are shows as follows:

$$f(X) = \sum_{i=1}^{n} \phi(x_i) + \sum_{i}^{n} \sum_{j \in N_i} \psi(x_i, x_j) \tag{2.2}$$

### 2.1.2 Conditional Random Field

Compared with Markov Random Field, conditional random field model Lafferty *et al.* (2001) is commonly used in high-level computer vision tasks, for example, image labeling He *et al.* (2004)Triggs and Verbeek (2008), image segmentation Plath *et al.* (2009)Wang *et al.* (2006)Zheng *et al.* (2015)Vemulapalli *et al.* (2016), object detection Quattoni *et al.* (2005)Shu *et al.* (2013) and so on. The main difference between Markov Random Field and Conditional Random Field is on the unary potentials. For Markov Random Field, the unary potential part is a generative module where all variables are unknown, while for Conditional Random Field, the unary potential is a discriminative module where two kinds of variables are involved: input unknown variables and output known variables. So for the latter model, the unary potential part can easily be replaced by using any existing classification framework, for example, support vector machine Noble (2006); Suykens and Vandewalle (1999), random forest Liaw and Wiener (2002); Breiman (2001), logistic regression Hosmer Jr *et al.* (2013); Press and Wilson (1978) and so on. This allows the final model to be more flexible.

## 2.2 Learning to Rank

This set of models focuses on modeling the difference between different data points. The main task is to learn the score function in the data space. With a higher score, the data point is more preferable to others with lower scores. One common ranking model is RankSVM Joachims (2002)Chapelle and Keerthi (2010). Let us take one simple version as an example. There is one dataset $\{x_i, i \in \{1, 2, ..., n\}\}$. The $i$-th data is $x_i$. If $x_{i_1}$ has a higher score than that of $x_{i_2}$, then $(i_1, i_2)$ is one ranking pair. All the ranking pairs consist of one ranking set. I denote it as $P$. Then the cost function for the RankSVM is as Eqn.2.3.

$$\begin{aligned} \min \quad & \frac{1}{2}\|\omega\|^2 + C\sum \xi_{i_1,i_2} \\ s.t. \quad & \omega^T x_{i_1} \geq \omega x_{i_2} + 1 - \xi_{i_1,i_2} \qquad \forall (i_1, i_2) \in P \\ & \forall \xi_{i_1,i_2} \geq 0 \end{aligned} \tag{2.3}$$

In this model (Eqn.2.3), the cost function is to learn the pre-defined cost function which is $\omega^T x + b$ where $b$ is a constant.

Chapter 3

# SIMULTANEOUS SEMANTIC SEGMENTATION OF A SET OF PARTIALLY LABELED IMAGES

## 3.1 Introduction

In the era of Internet and social media, there are more and more images posted online. Often, such on-line data lack sufficient textual annotation desired by learning-based algorithms. To make such data more useful, efforts have been devoted towards tasks like image taggingChen *et al.* (2013)He *et al.* (2014)Saito *et al.* (2013) and image classification Lapin *et al.* (2014)Zhang *et al.* (2014c)Voravuthikunchai *et al.* (2014), targeting at producing labels for the images. The finest granularity one could achieve in this labeling effort is to perform semantic segmentation Arbeláez *et al.* (2012), which may classify each pixel in one image into a proper class/label. Both fully-supervised and weakly-supervised approaches exist.

In the fully-supervised setting, a set of images with pixel-level labels are available. In Tighe and Lazebnik (2013b), all pixels in one superpixel are assumed to have the same label and Markov Random Field (MRF) was used to capture the context information to help improve the local superpixel-level labeling. Limited availability of fully-labeled data is a practical constraint for such approach. In Tighe and Lazebnik (2013a)Tighe and Lazebnik (2013b), region-based cues are used to build exemplar-SVMs to gain the final labeling. However, there is one obvious disadvantage: users have to label each pixel in the dataset, which is time-consuming and involves a lot of manual work. In the weakly-supervised setting, data with only image-level labels are assumed. Most existing work further assumes that the labels are “complete” in the sense that the image-level label set for a given image contains all possible labels we may assign to any pixel in that image. This setting has been used in Xie *et al.* (2014a)Liu *et al.* (2013)Vezhnevets *et al.* (2012b).

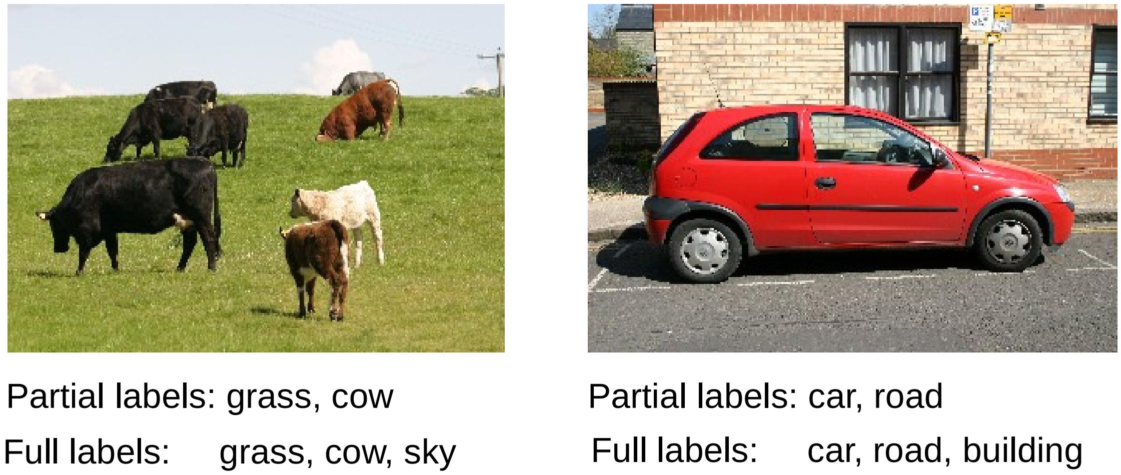

Partial labels: grass, cow
Full labels: grass, cow, sky
Partial labels: car, road
Full labels: car, road, building

Figure 3.1: Two images with partially and fully image-level labels.

The abundance of images with tags on social media platforms provides the opportunity for obtaining large-scale training sets without laborious manual labelling. However, in reality, even if we may be able to obtain a lot of images with a desired set of semantic tags (and use the tags as semantic labels for simplicity), the majority of on-line images would still have only *incomplete* image-level labels, especially for user-generated images. That is, it is unrealistic to expect tags associated with an on-line image would happen to cover *all* semantic concepts we need to employ for segmentation. Therefore, in order to utilize the vast on-line images, we face the task of how to label each pixel in each image (i.e., semantic segmentation), given a set of images with partial image-level labels. Figure shows a demo of one image with partially image-level labels, while our task is illustrated in Figure 3.2. One similar work is Zhang *et al.* (2015), which only considers using information from one image only and does not consider the fact that visually similar superpixels across different images also are likely to have the same labels. In this chapter, I work on this problem from one new aspect by proposing an approach based on conditional random fields (CRFs),

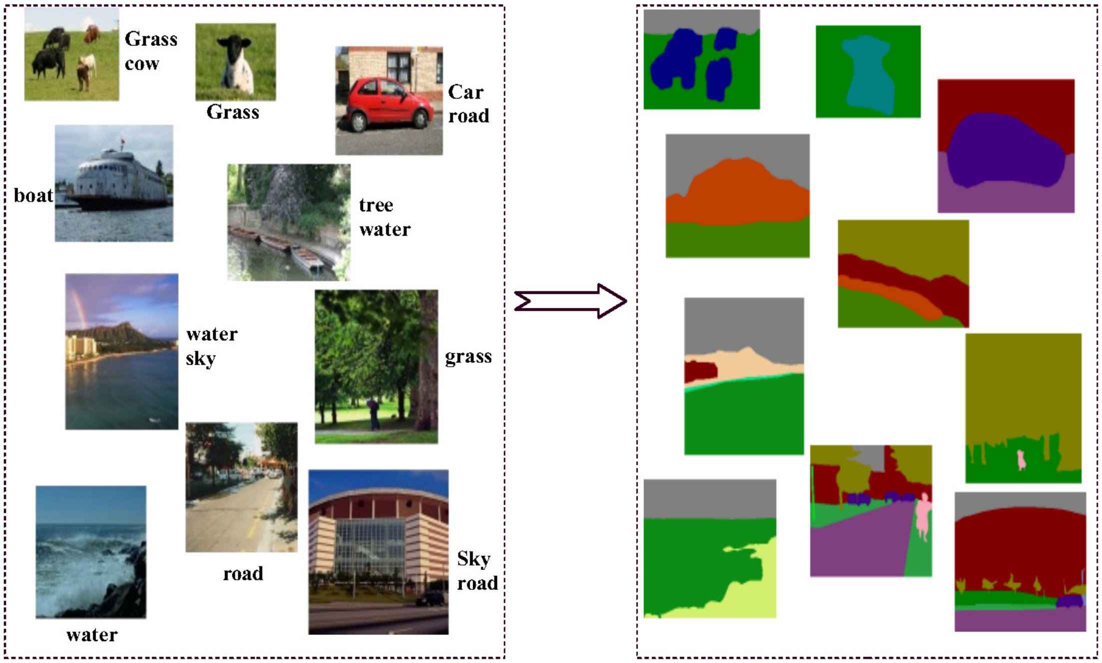


Figure 3.2: Illustrating the problem studied in this chapter: the left panel represents the input to our algorithm, which are a set of images with partial image-level labels (one demo shown in Figure 3.1), and the right panel is the output of segmented images with labeled pixels. A formal problem definition is shown in Section 3.3.

which attempts to employ all possible sources of information in the dataset to deal with the challenge of incomplete labels.

Contributions of this chapter are as follows. First, I propose a novel formulation for a new problem of semantic segmentation with partial image-level labels. Second, under the proposed multi-image model, I propose an efficient solution and demonstrate with comparative experiments its effectiveness.

The organization of the remainder of the chapter is as follows. I first give a brief literature review on related works in Section 3.2. Then, a detailed description of the problem and our proposed approach are provided in Section 3.3. To show the performance of our proposed method, experiments are reported in Section 3.4. We conclude our work and present our future work in Section 3.5.

## 3.2 Related Works

We briefly review below two classes of related research on semantic segmentation: those relying on fully-supervised learning and those utilizing only weakly-supervised learning. As is evident from the following discussion, the distinction between these two classes of approaches is mainly on the granularity of labelling for the training data.

### *3.2.1 Fully-supervised Semantic Segmentation*

As described in Section 3.1, in fully-supervised semantic segmentation, labels of each pixel or superpixel in the training set are known. There are a lot of existing efforts on this regard. In Shotton *et al.* (2009), Jamie Shotton *et al.* proposed semantic texton forests to do semantic segmentation using a bag-of-semantic-textons model, where only simple features of superpixels were used. To improve the performance, some other approaches attempt to consider neighboring information of different superpixels. In Kohli *et al.* (2009), Pushmeet Kohli *et al.* proposed to use higher order CRFs to capture such information of a set of pixels. Since high-order CRF models do not model the relevance of semantic labels, in Myeong and Lee (2013), Heesoo Myeong *et al.* proposed to use high-order semantic relations to capture the context information in images and then transfer semantic labels from a labeled image to another unlabeled image. Besides tree-structure algorithms and graphical models (like CRF, MRF), active learning and deep learning are also applied to semantic segmentation recently. In Roig *et al.* (2013), Gemma Roig *et al.* proposed a MAP inference method based on active learning, which is in fact one semi-supervised method. In Sharma *et al.* (2015), to improve the Recursive Context Propagation Network (RCPN), two revisions were made: one is to solve the potential problem because of the special structure of RCPN, which can help reduce the complexity of the network structure;

the other is to consider the context information by building a Markov Random Field on the modified structure. This is one recent work on applying deep network to capture the context information of different superpixels for semantic segmentation.

Obviously, one key limitation of the fully-supervised approaches is the requirement of a set of images with pixel-level (or superpixel-level) labels. Due to the cost associated labeling, generally speaking one cannot assume the availability of high-quality and large-scale training data.

### *3.2.2 Weakly-supervised Semantic Segmentation*

Because of the strong requirement of fully-supervised semantic segmentation, research on finding new techniques to solve weakly semantic segmentation becomes popular. Liu *et al.* worked on dual clustering for semantic segmentation by constructing two clusterings on smoothness and also the relation between image features and superpixel-level labels Liu *et al.* (2013). Besides the dual clustering method, many other approaches are also proposed to solve weakly supervised semantic segmentation. For example, Vezhnevets et al proposed to use active learning in Vezhnevets *et al.* (2012a), and multiple instance multi-task learning to solve weakly semantic segmentation in Vezhnevets and Buhmann (2010). It may be difficult to learn superpixel-level labels from only one image. In Vezhnevets *et al.* (2011), a multi-image model was proposed, which builds a graphical model on the entire dataset. More recently, a graphical model was also proposed in Chang *et al.* (2014), where multiple instance learning and CRF are combined. Besides CRF-based methods, structural information from different superpixels was also considered in Zhang *et al.* (2013)Zhang *et al.* (2014b)Zhang *et al.* (2014a), using the concept of graphlets. Recently, semantic relevance has also been studied in the weakly-supervised cases. For example, in Xie *et al.* (2014b), hypergraphs were used to capture the high-order semantic relevance, instead of only the second-order relevance in Xie *et al.* (2014a), and in Pinheiro and Collobert (2015), deep learning techniques are used to find the pixel-level labeling. In Zhang *et al.* (2015), Wei Zhang *et al.* studied one new practical case in which each image is assumed to have part of image-level labels and also maybe some incorrect labels.

While apparently less stringent than the fully-supervised cases, the image-level labels in existing methods of weakly-supervised semantic segmentation are still assumed to be complete, i.e., the set of labels of a given image captures all possible semantic labels that can be assigned to pixels of that image. As discussed previously, this limitation makes it difficult to utilize vast amount of on-line pictures that would otherwise be useful for the learning task. Our study in this chapter is intended to address this issue by considering using information from the entire dataset instead of only one image. We will formally define the problem and present our solution in the next section.

## 3.3 Proposed Approach

Based on the previous discussion, I formally define the following problem of this study: Given a set of images with incomplete image-level labels, to predict all pixel-level labels for each image in the set. The image-level labels indicate possible objects in one image, while the pixel-level labels are the final desired segmentation and classification. The incompleteness of labels for an image means that this image may contain some objects/regions which cannot be assigned to any of the given classes in its label set. For example, an image with four objects, *car*, *street*, *sky*, and *grass*, may have only a set of image-level labels, say *car* and *sky*. Still, in the final segmentation, the correct results should properly label those regions corresponding to the missing labels (*street* and *grass*). Apparently, the missing information needs to be figured out by considering the entire set of images. This is schematically illustrated in Figure 3.2. In this work, I employ the concept of superpixel Ren and Malik (2003), and assume that pixels within the same superpixel share the same label. This helps simplify the problem to some extent for better tractability.

I use the following notations in the rest of the presentation. Denote one image set with $N$ images by $\mathscr{A} = \{I_i, i \in \{1, \cdots, N\}\}$, which has corresponding partial image-level labels $\mathscr{L} = \{L_i, i \in \{1, \cdots, N\}\}$. Pixels are denoted by $p_{i,j}, j \in \{1, \cdots, M_i\}, i \in \{1, \cdots, N\}$ where $p_{i,j}$ is the $j^{th}$ pixel in the image $I_i$ which has $M_i$ pixels in total. Similarly, superpixels of the image $I_i$ are denoted by $x_i = \{x_{i,j}, j \in \{1, \cdots, n_i\}\}$ where $x_{i,j}$ is the $j^{th}$ superpixel in the image $I_i$ which has $n_i$ superpixels in total. Also I use $L_{i,j}$ to denote the label of the $j^{th}$ superpixel's label in the image $I_i$.

### *3.3.1 Formulating the Problem*

In our problem, the input images do not have superpixel-level labels. Further, the images do not have a complete set of semantic labels. Evidently, in general the full information needed for labelling an image needs to be inferred from other images. The multi-image model introduced in Vezhnevets *et al.* (2011) may be employed except that complete labelling was assumed therein. Our basic strategy in modeling the problem with incomplete labels is to construct a conditional random field (CRF) for capturing these types of probabilistic associations: visually-similar superpixels are likely to have the same labels (but two similar superpixels may have different likelihoods belonging to the same label, depending on if they are from the same image or from different images), nearby superpixels tend to share labels, and the final label set of an image is a superset of the given (incomplete) label set. Graphically, a basic component of the overall CRF model may be illustrated by Figure 3.3.

In Figure 3.3, $x_{i,j}$ is the $j^{th}$ superpixel of the image $I_i$ in the dataset. $S_{i,j}$ is the set of spatial neighbors of $x_{i,j}$, defined as the superpixels which are located next to $x_{i,j}$ in the image $I_i$. $M_{i,j}$ is the set of visually-similar neighbors of $x_{i,j}$, defined as superpixels which are located in those images sharing common image-level labels as $I_i$. $V_{i,j}$ is the set of visually-similar neighbors of $x_{i,j}$, defined as superpixels which are located in the

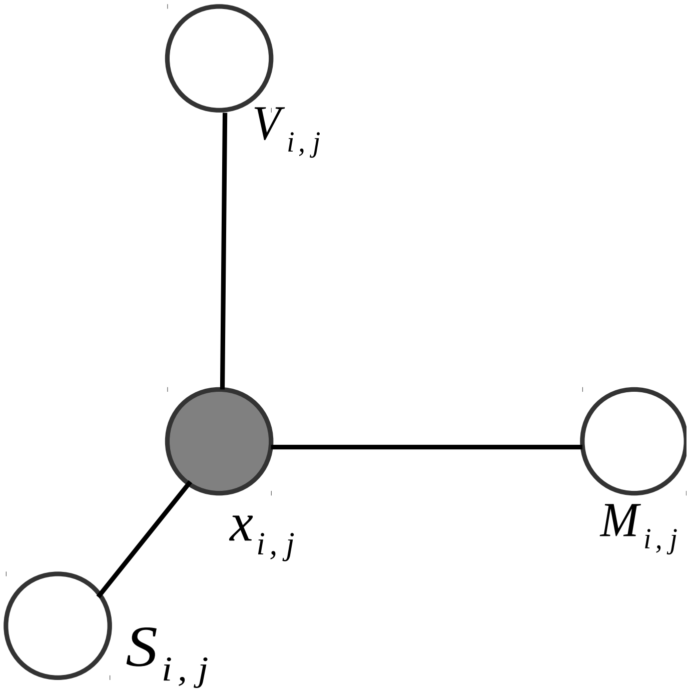


Figure 3.3: Illustrating the basic component of the proposed CRF model. Each superpixel is related to others via the shown connections. See text for definitions of the symbols. The *entire set of image* forms an overall CRF by combining all the basic components corresponding all superpixels.

images without common image-level labels with $I_i$. To help illustrate how the nodes and connections on the final CRF link the entire image set together, we depict in Figure 3.4 a visual example with exemplar images and their superpixels explicitly shown.

Based on the structure described above, we can have the complete energy function for

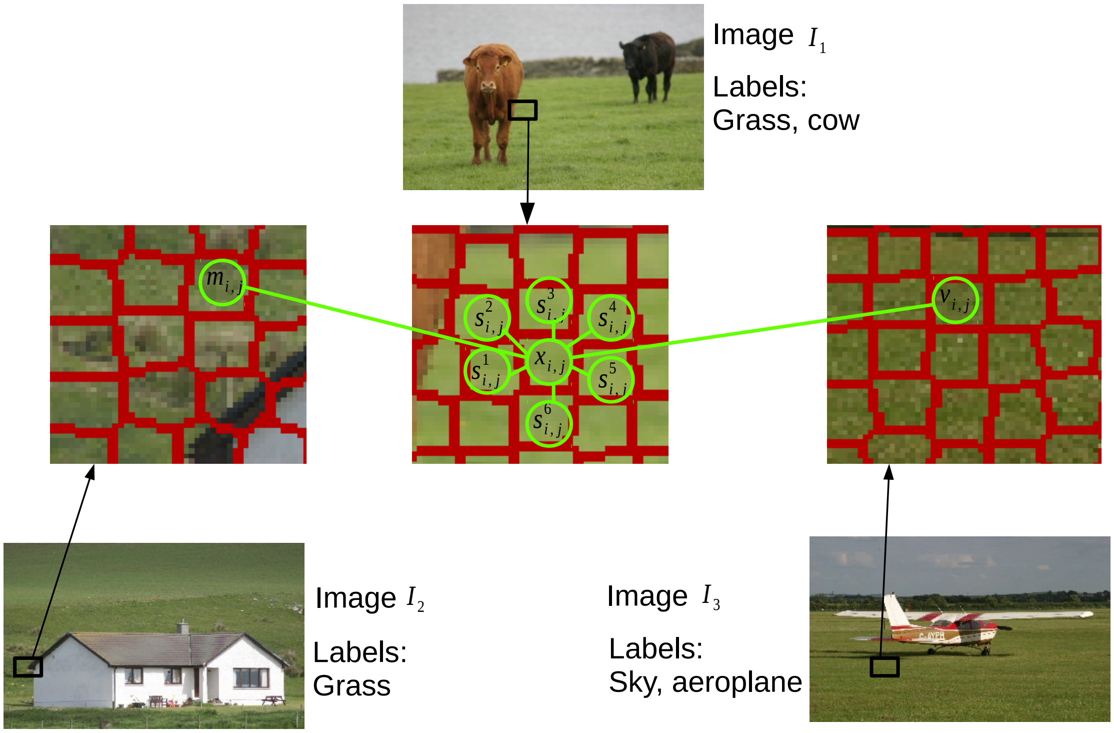


Figure 3.4: Illustrating basic components of the proposed CRF model with sample images. Shown are some superpixels of three images $I_1, I_2, I_3$. These superpixels are separated by red boundaries and their positions in their corresponding images are marked by the black rectangles. $I_1$ and $I_2$ have one common image-level label, while $I_1$ and $I_3$ have no common image-level labels. A basic CRF component is shown in light green color and is built on $x_{i,j}$. Each circle represents one node in CRF. In this example, we only set $M_{i,j} = \{m_{i,j}\}$ and $V_{i,j} = \{v_{i,j}\}$ and their size is one. It is easy to see there are six elements in $S_{i,j}$, which is $\{s^k_{i,j}, k \in \{1, 2, \cdots, 6\}\}$.

our CRF-based model as given in Eqn.3.1:

$$
\begin{aligned}
&E(\{L_{i,j}, j \in \{1, \cdots, M_i\}, i \in \{1, \cdots, N\}\}, \theta, \alpha) = \\
&\sum_{x_{i,j}, \forall i,j} (\phi(x_{i,j}, L_{i,j}, \theta) + \lambda(L_{i,j}, I_i)) + \\
&\alpha_1 \sum_{(x_{i,j}, x'_{i,j}) \in S_{i,j}, \forall i,j} \varphi(L_{i,j}, L'_{i,j}) + \\
&\alpha_2 \sum_{(x_{i,j}, x'_{i,j}) \in M_{i,j}, \forall i,j} \varphi(L_{i,j}, L'_{i,j}) + \\
&\alpha_3 \sum_{(x_{i,j}, x'_{i,j}) \in V_{i,j}, \forall i,j} \varphi(L_{i,j}, L'_{i,j})
\end{aligned}
\tag{3.1}
$$

where $\alpha = [\alpha_1, \alpha_2, \alpha_3]$ controls the contributions of each potential terms, $\phi(x_{i,j}, L_{i,j}, \theta)$ is the unary potential which gives the energy caused by the fact that the label $L_{i,j}$ is assigned to the superpixel $x_{i,j}$. $\lambda(L_{i,j}, I_i)$ relates to how likely $I_i$ has the label $L_{i,j}$. It can be the negative of the possibility that the image $I_i$ has the label $L_{i,j}$, computed by Chen *et al.* (2013). For the pairwise potential, we use the Potts model, where the function $\varphi(\cdot)$ is given as Eqn.3.2.

$$\varphi(L_{i,j}, L^{'}_{i,j}) = \begin{cases} 1 & \text{if } L_{i,j} \neq L^{'}_{i,j} \\ 0 & \text{otherwise} \end{cases} \tag{3.2}$$

### *3.3.2 An Inference Algorithm*

Exact solutions for achieving the extrema of Eqn.3.1 would require exponential complexity and thus cannot be obtained unless it is for datasets of trivial complexity. Approximate approaches to inference under similar graphical models have been developed over the years. Examples include Loopy Belief Propagation Murphy *et al.* (1999), Graph cut Delong *et al.* (2012), Simulated Annealing AARTS/KORST. (1990), and etc. In this work, we adopt Iterated Conditional Modes (ICM) Kittler and Föglein (1984) in developing an inference algorithm, owing to its simplicity and in turn efficiency in dealing with a large model like ours. The key idea of the ICM-based algorithm is based on the iterative update: when computing the label of one superpixel, labels of the others are assumed to be fixed.

For each superpixel $x_{i,j}$, its label $L_{i,j}$ is computed by (Eqn.3.3):

$$\begin{aligned} L_{i,j} = \arg\min_{l} \phi(x_{i,j}, l, \theta) + \lambda(l, I_i) + \\ \alpha_1 \sum_{(x_{i,j}, x'_{i,j}) \in S_{i,j}} \varphi(l, L'_{i,j}) + \\ \alpha_2 \sum_{(x_{i,j}, x'_{i,j}) \in M_{i,j}} \varphi(l, L'_{i,j}) + \\ \alpha_3 \sum_{(x_{i,j}, x'_{i,j}) \in V_{i,j}} \varphi(l, L'_{i,j}) \end{aligned} \tag{3.3}$$

The entire algorithm based on the above core ICM iteration is given in Algorithm 1.

### *3.3.3 Key Implementation Details*

We now present a few key technical details that are necessary to fully implement the proposed solution. We use the SLIC algorithm proposed in Achanta *et al.* (2012) to obtain superpixels for images in our experiments and also compute the histogram-based features for superpixels and images, following the method of Tighe and Lazebnik (2013b). Before constructing the entire energy function of Eqn.3.1, we first train one SVM classifier using a very small image set. In this small image set, there are about two images per label and full pixel-level labels of each image are provided. Labeling this subset requires less manual work. More details are shown in Section 3.4. This pre-trained SVM classifier supplies a measurement for the unary potential in the proposed model, i.e., the function $\phi(\cdot)$ given in Eqn. 3.4.

$$\phi(x_{i,j}, L_{i,j}, \theta) = \begin{cases} \rho & \text{if } L_{i,j} \neq L'_{i,j}(\theta) \\ 0 & \text{otherwise} \end{cases} \tag{3.4}$$

where $L'_{i,j}(\theta)$ is the predicted label of $x_{i,j}$ by the pre-trained SVM with model parameters $\theta$, and $\rho$ is the penalty.

**Algorithm 1** An Algorithm Based On ICM

```
Input: Energy function (Eqn.3.1), one potential label set $\tilde{L}$ of each superpixel $x_{i,j}$
Output: the label $L_{i,j}$ of each superpixel $x_{i,j}$, $j \in \{1, \cdots, M_i\}, i \in \{1, \cdots, N\}$
BEGIN:
initialize each $x_{i,j}$ using random element from $\tilde{L}$ and store initialized labels of each superpixel in $Y_1, Y_2$.
while check the stop-condition do
    for each superpixel $x_{i,j}$, $j \in \{1, \cdots, M_i\}, i \in \{1, \cdots, N\}$ do
        tmp = $\emptyset$ and Consider $S_{i,j}$, $M_{i,j}$ and $V_{i,j}$ of $x_{i,j}$.
        for each $l$ in $L$ do
            compute the local energy (denoted as $e$) by assuming each superpixel has the label as that in $Y_1$ except that $x_{i,j}$ has the label $L_{i,j} = l$
            tmp = tmp $\cup e$.
        end for
        Set the label of $x_{i,j}$ in $Y_2$ as $l^{'}$ which has the smallest local energy.
    end for
    $Y_1 = Y_2$.
end while
```

For the term $\lambda(L_{i,j}, I_i)$, we compute it using the method proposed in Chen *et al.* (2013), which does image-tagging and can provide a ranked list of all possible image-level labels which are likely to be shown in the corresponding image. $\lambda(L_{i,j}, I_i)$ is the negative value of the likelihood that the image $I_i$ has the label $L_{i,j}$.

For pairwise potentials, we need to consider different neighboring relations. For one superpixel $x_{i,j}$, there are three sets of neighbors we need to compute: $S_{i,j}$, $M_{i,j}$ and $V_{i,j}$. For one given superpixel $x_{i,j}$, the spatial neighbor set $S_{i,j}$ can be estimated using image erosion/dilation (note that typically superpixels are irregular in shape). This is illustrated in Figure 3.5. For the other two sets of neighbors, we can obtain them by Algorithm 2, in which the normalized Euclidean distance is used to compute the similarity between different images and superpixels, based on the image/superpixel features defined above. We emphasize that such neighboring relations are defined based on the proposed CRF model and thus they reflect physical constraints imposed by the given labels (and their interaction) and geometrical proximity, in addition to visual similarity.

**Algorithm 2** Algorithm to compute $M_{i,j}$ and $V_{i,j}$

```
Input: $\{I_i, L_i\}$, $\{x_{i,j}\}$, $j \in \{1, \cdots, M_i\}$, $i \in \{1, \cdots, N\}$, $D_1(\cdot)$ which is the function to compute the distance between two images and $D_2(\cdot)$ which is to compute the distance between two superpixels.
Output: $M_{i,j}$, $V_{i,j}$, $j \in \{1, \cdots, M_i\}$, $i \in \{1, \cdots, N\}$

BEGIN:
// To compute SM$_i$, SV$_i$.
for i = 1,$\cdots$, N do
    for j = 1, $\cdots$, N, $i \neq j$ and $L_i \cap L_j \neq \emptyset$ do
        Compute the similarity $D_1(I_i, I_j)$.
    end for
    Find the top $q$ most similar images, denoted as SM$_i$.
    for j = 1, $\cdots$, N, $i \neq j$ and $L_i \cap L_j == \emptyset$ do
        Compute the similarity $D_1(I_i, I_j)$.
    end for
    Find the top $q$ most similar images to $I_i$, denoted as SV$_i$.
end for
```

**Algorithm 2** Algorithm to compute $M_{i,j}$ and $V_{i,j}$ (continued)

```
for each superpixel $x_{i,j}$, $j \in \{1, \cdots, M_i\}$, $i \in \{1, \cdots, N\}$ do
    // we have SM$_i$ and SV$_i$ of $I_i$
    // and will construct SPM$_{i,j}$ and SPV$_{i,j}$
    SPM$_{i,j} = \emptyset$, MSS$_{i,j} = \emptyset$, $\forall i, j$.
    for each superpixel $x^{'}_{i,j}$ in each image $I' \in$ SM$_i$ do
        Find the top $p$ most similar superpixels to $x_{i,j}$ based on $D_2(x_{i,j}, x^{'}_{i,j})$
        Denote these $p$ superpixels as MSS$_{i,j}$ and also we set SPM$_{i,j}$ = SPM$_{i,j} \cup$ MSS$_{i,j}$
    end for
    Find top $k$ most similar superpixels to $x_{i,j}$ from SPM$_{i,j}$, which are $M_{i,j}$ of $x_{i,j}$.

    SPV$_{i,j} = \emptyset$, MSS$_{i,j} = \emptyset$, $\forall i, j$.
    for each superpixel $x^{'}_{i,j}$ in each image $I' \in$ SV$_i$ do
        Find the top $p$ most similar superpixels to $x_{i,j}$ based on $D_2(x_{i,j}, x^{'}_{i,j})$
        Denote these $p$ superpixels as MSS$_{i,j}$ and SPV$_{i,j}$ = SPV$_{i,j} \cup$ MSS$_{i,j}$
    end for
    Find top $k$ most similar superpixels to $x_{i,j}$ from SPV$_{i,j}$, which are $V_{i,j}$ of $x_{i,j}$
end for
```

### *3.3.4 Comparison With MIM*

The proposed method bears some similarity to the Multi-Image Model (MIM) of Vezhnevets *et al.* (2011), since both consider a set of images simultaneously. To appreciate the key difference easily, we provide the energy function of the MIM below (Eqn.3.5):

$$\begin{aligned} &E(\{L_{i,j}, j \in \{1, \cdots, M_i\}, i \in \{1, \cdots, N\}\}, \theta) = \\ &\sum_{x_{i,j}, \forall i,j} (\psi_1(x_{i,j}, L_{i,j}, \theta) + \pi(L_{i,j}, I_i)) + \\ &\sum_{(x_{i,j}, x^{'}_{i,j}) \in S_{i,j}, \forall i,j} \varphi_1(L_{i,j}, L^{'}_{i,j}, x_{i,j}, x^{'}_{i,j}) + \\ &\sum_{(x_{i,j}, x^{'}_{i,j}) \in M_{i,j}, \forall i,j} \varphi_1(L_{i,j}, L^{'}_{i,j}, x_{i,j}, x^{'}_{i,j}) \end{aligned} \tag{3.5}$$

where $\pi(L_{i,j}, I_i)$ is zero if the label $L_{i,j}$ is one image-level label of the image $I_i$ and it is set to infinity otherwise. Moreover, $\varphi_1(\cdot)$ is given as follows:

$$\begin{aligned} &\varphi_1(L_{i,j}, L^{'}_{i,j}, x_{i,j}, x^{'}_{i,j}) = \\ &\begin{cases} 1 - D(x_{i,j}, x^{'}_{i,j}) & \text{if } x_{i,j}, x^{'}_{i,j} \text{are different} \\ 0 & \text{otherwise} \end{cases} \end{aligned} \tag{3.6}$$

where $D(\cdot)$ is one similarity metric.

Eqn.3.5 clearly indicates one strong requirement on the labels, imposed by the choice of $\pi(\cdot)$. Because of that function, MIM cannot be used to solve the general problem defined in this chapter. In our formulation, to solve the more general and practical problem, we relaxed the strong requirement in MIM by introducing a new $\pi(\cdot)$ function *plus* one additional pairwise potential to better capture visual similarity of superfixels (those across images and do not have common image-level labels). These resulted in the new model of Eqn.3.1. In fact, compared with both formulations, we can see that MIM is one special case of our approach, which is used to deal with the less challenging situation where images have completely image-level labels.

## 3.4 Experiments

In this section, we demonstrate the effectiveness of the proposed approach based on comparative experiments using the following three datasets: one synthetic dataset, the MSRC-21 dataset Shotton *et al.* (2009) and the Siftflow dataset Tighe and Lazebnik (2013b). For the synthetic dataset and the MSRC-21 dataset, we make comparison with the approach in Vezhnevets *et al.* (2011), which is among the state-of-art methods in the literature. For the Siftflow dataset, we provide our experimental results and compare with existing approaches in the fully-supervised case and the ordinary weakly-supervised case. The comparison is based on two metrics: per-pixel accuracy (denoted as *pp* and shown in Eqn.3.7) and average per-class accuracy (denoted as *p̄c* and shown in Eqn.3.9). To compute these measures, we need the size of each superpixel $x_{i,j}$, which is denoted by $size(x_{i,j})$.

$$pp = \frac{\sum_{i,j} \delta(L_{i,j} - L^{'}_{i,j}) size(x_{i,j})}{\sum_{i,j} size(x_{i,j})} \tag{3.7}$$

$$pc_l = \frac{\sum_{i,j} \delta(L_{i,j} - l)\delta(L_{i,j} - L^{'}_{i,j}) size(x_{i,j})}{\sum_{i,j} \delta(L_{i,j} - l) size(x_{i,j})} \tag{3.8}$$

$$\bar{pc} = \frac{1}{|\bigcup L_i|} \sum_l \mathrm{pc}_l \tag{3.9}$$

In the above definitions, $L^{'}_{i,j}$ is the predicted label and $L_{i,j}$ is the ground truth of the label of $x_{i,j}$, and $pc_l$ is the pixel-level accuracy for all the pixels whose label is $l$. Also $|\bigcup L_i|$ is the total number of potential labels .

### 3.4.1 Synthetic Dataset

The simulation is designed as follows. First, we generate one synthetic dataset that has 30 pairs of observation images and labelmaps. An observation image is a 200×200 grayscale image while its labelmap is a 200×200 image whose pixel values are the labels of its corresponding observation. For each observation image, we split it into 20×20 superpixels, each of which has 10×10 pixels. Moreover, we assume that all pixels in one superpixel have the same label and labels are from this set: $\{1, 2, 3, 4, 5\}$.

To generate each pair of one observation image and its labelmap, we run the following procedure:

1. We first generate one labelmap randomly and make sure that labels of pixels in the same superpixel are the same.

2. The corresponding observation image is generated based on the new labelmap.

3. The inference algorithm runs for 200 iterations to obtain the final pair of observation image and labelmap.

    (a) For each iteration, we use the current labelmap and the observation image to generate a better labelmap whose energy is smaller. Then based on the new generated labelmap, we generate the new observation image.

During the above procedure, we set the total number of iterations to be 200 since at this iteration the observation-labelmap pair is already stable. Besides the number of iterations, we set the relationship between one observation image and its labelmap as the Gaussian distribution whose standard variation is set to be 10. Samples of the constructed dataset are shown in Figure 3.6. The average size of the complete image-level labels is 3.46. To generate partial image-level labels, we randomly remove one label from the complete image-level labels. The parameters $k$, $q$ and $p$ we set in this simulation are 21, 3, 5, respectively.

The synthetic dataset was then used to compare the performance of the proposed approach and the MIM method. The MIM method would simply assume whatever labels given for an image is complete. The final results are summarized in Table 3.1. From these results, it is obvious that the MIM method lags the proposed approach by a large margin. We also note the difficulty of the task (even if the dataset is synthetic), since a lot of source of uncertainties were introduced in the process of creating the data. This explains why the overall accuracy numbers are not very high for either approach.

Table 3.1: Comparing with the MIM model on the synthetic dataset.

| | pp | $\bar{pc}$ |
|---|---|---|
| MIM Vezhnevets *et al.* (2011) | 51.15% | 29.81% |
| Proposed | 76.74% | 42.72% |

### *3.4.2 MSRC-21 Dataset*

In this dataset, there are 591 images and 21 objects [1] in total. We split the dataset into two parts: Set one and Set two, both are the same as those used in Shotton *et al.* (2009). As a result, there are 276 images in Set one, 256 in Set two. Also we call the union of Set one and Set two as the Entire Set. To get the pre-trained SVM classifier, we randomly choose 42 images out of 59 images which consist of the validation set as in Shotton *et al.* (2009). The average numbers of the complete image-level labels for Set one, Set two and the Entire Set are 2.4710, 2.4492 and 2.4605, respectively. To generate partial image-level labels, we randomly remove one label from each complete image-level label set. So the average sizes of Set one, Set two and Entire Set decrease by 40.4 %, 40.8% and 40.6%, respectively. In this experiment, parameters $k$, $p$ and $q$ are set to be $10$, 3 and $8$, respectively.

The per-class accuracies from the proposed and the MIM method for Set one, Set two, and the Entire Set are plotted respectively in Figure 3.7, Figure 3.8 and Figure 3.9. Overall, the performance gains of the proposed method over MIM are 5%, 3% and 2% respectively for Set one, Set two, and the Entire Set.

[1] There are 23 objects in total, but 2 of them are not considered by Microsoft research. So we only use 21 objects. Details are shown in the dataset which is available on Microsoft research.

In addition to per-class accuracy, we also provide the per-pixel accuracy in Table 3.2, where it is clear that the proposed approach was able to outperform MIM by large margins on all the sets of data.

Table 3.2: The per-pixel accuracies *pp* of our approach and MIM in Vezhnevets *et al.* (2011).

| | Set one | Set two | Entire Set |
|---|---|---|---|
| MIM in Vezhnevets *et al.* (2011) | 43.33% | 39.44% | 41.82% |
| Proposed | 56.69% | 52.80% | 53.08% |

The above results demonstrated the effectiveness of the proposed approach in dealing with incomplete image-level labels. It is worth pointing out that the MIM method reported higher performance numbers in Vezhnevets *et al.* (2011), where it was studied as an ordinary weakly-supervised approach with complete image-level labels for training. Our experimental setting is more realistic for simulating the scenario of learning with Web images. In this experiment, considering the dropped label per image, the label set suffers a loss of around $40\%$ labeling information compared with the case where images have complete image-level labels. The proposed approach, even if with only a very simple ICM-based inference algorithm, was shown to be able to better deal with the incomplete label data.

### *3.4.3 Siftflow Dataset*

In this experiment, we show the performance of our algorithm on the Siftflow dataset Tighe and Lazebnik (2013b). This dataset consists of 2688 images and 33 labels. We use the entire training set which has 2488 images, as defined in Russell *et al.* (2008). The average number of image-level labels for each image in the entire Siftflow dataset is 4.4297 and for the part we use, on average, there are 4.3881 labels per image. To simulate incomplete image-level labeling, we create partial image-level labels for each image by randomly removing one label from the original label set. This means we remove $22.79\%$ label information on average for each image. During the experiment, parameters $k$, $p$ and $q$ are set to be $10$, $3$ and $8$, respectively. Our results are: $pp = 57.09\%$ and $\bar{pc} = 22.34\%$. Since the related work do not report the per-pixel accuracy (*pp*) on this dataset, we only report the per-class accuracy (by quoting) in Table 3.3, including the results from some fully-supervised methods (Shotton *et al.* (2009)Liu *et al.* (2009)) and weakly-supervised methods assuming complete image-level labels (Vezhnevets *et al.* (2011)Vezhnevets *et al.* (2012b)Liu *et al.* (2013)Zhang *et al.* (2014a)). From the table, we see that our approach was able to deliver nearly comparable performance, although we subject our approach to the heavy loss of information, while the competing methods either utilize pixel-level labels or assume and use complete image-level labels.

Table 3.3: Average per-class accuracy $\bar{pc}$ from our approach and those from a set of competing approaches, either fully-supervised or weakly-supervised with complete image-level labels. The results above are in percentage.

| | Vezhnevets *et al.* (2011) | Vezhnevets *et al.* (2012b) | Shotton *et al.* (2009) | Liu *et al.* (2009) | Liu *et al.* (2013) | Zhang *et al.* (2014a) | Ours |
|---|---|---|---|---|---|---|---|
| $\bar{pc}$ | 14% | 21% | 24% | 24% | 26% | 27.73% | 22.34% |

## 3.5 Conclusion & Future Work

We identified a key limitation in existing methods for semantic segmentation and proposed a new multi-image formulation for addressing the limitation. An inference algorithm was designed for finding a solution under the proposed multi-image model. To demonstrate the effectiveness of our algorithm, we performed experiments on both synthetic data and real datasets including MSRC-21 and Siftflow. While current results have shown advantages of the proposed method, there are still a few leads for future exploration. In particular, current results indict that some classes have low per-class accuracy, possibly due to their rare presence in the images. Such information (some classes being rare), if known *a priori*, may be explicitly factored into the formulation so that rare classes do not get overshadowed by other more common classes.

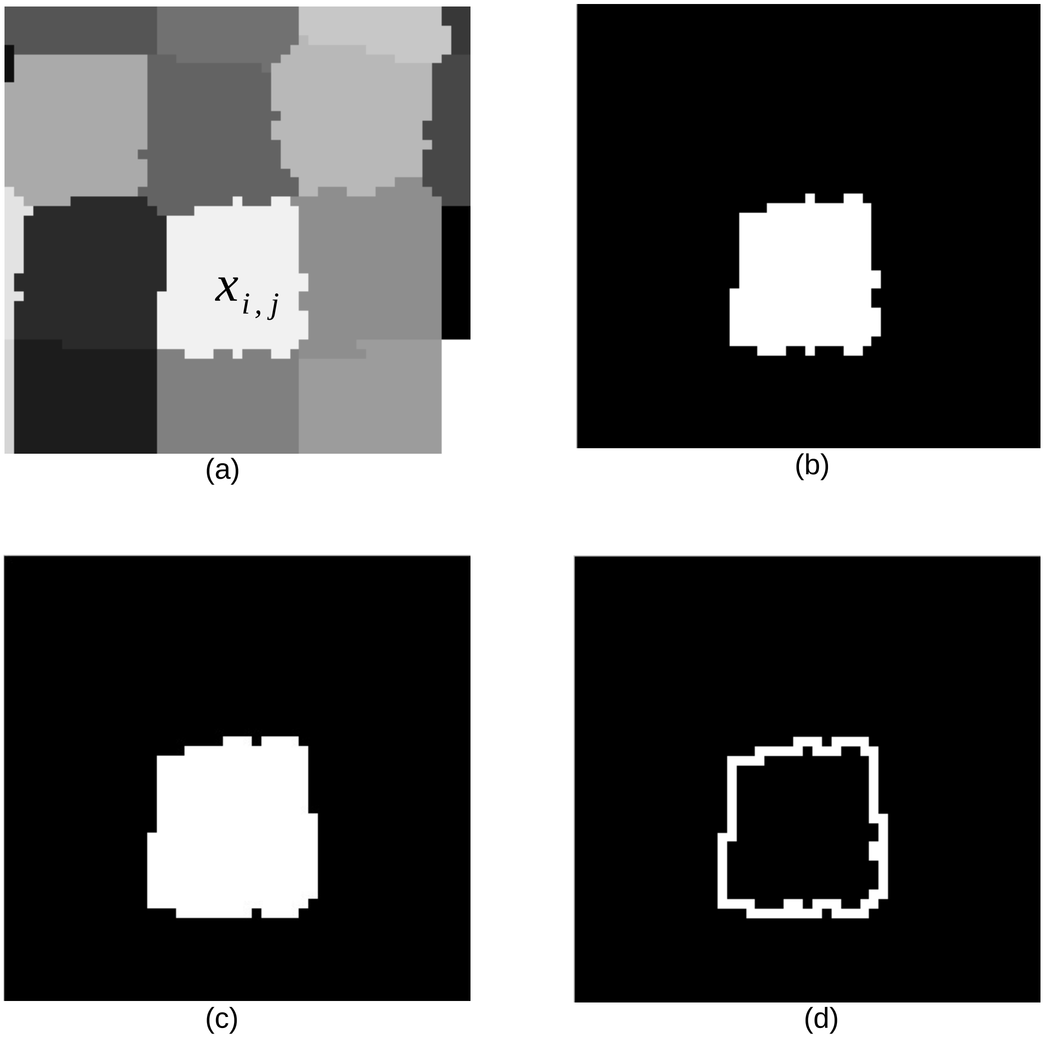


Figure 3.5: Illustrating how to find the spatial neighbors of one given superpixel $x_{i,j}$ shown in $(a)$. First we need to get the image $(b)$ which is the mask of $x_{i,j}$. Then we can apply the image dilation to $(b)$ to get the image $(c)$. By computing the difference of images $(b)$ and $(c)$, the final mask $(d)$ is obtained. Comparing $(d)$ and the original image $(a)$, we can easily get $S_{i,j}$ which consists of super-pixels which overlap with the final mask $(d)$.

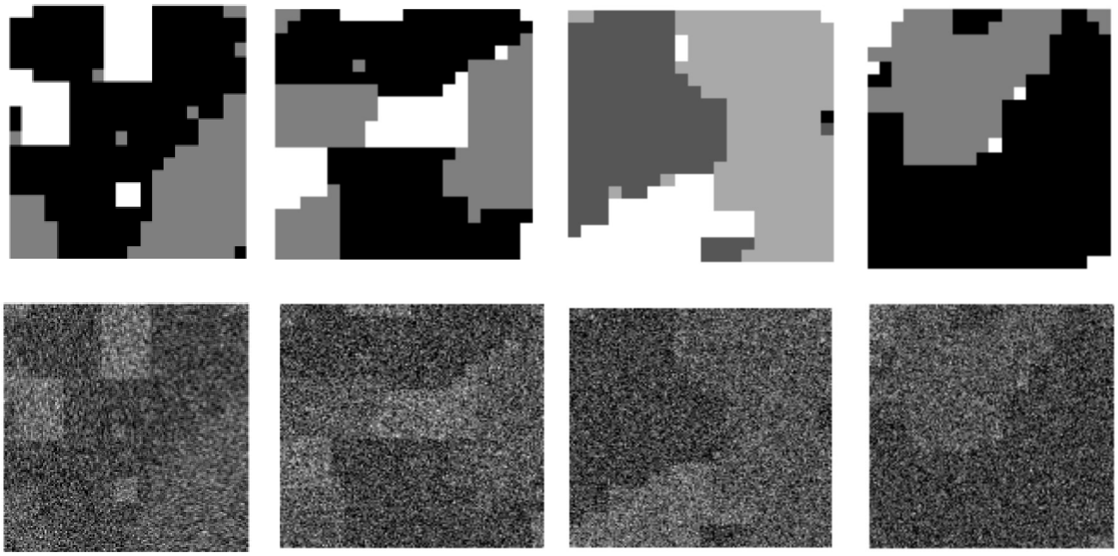


Figure 3.6: This figure shows some pairs of the observation and the labelmap generated in the synthetic dataset. The first row consists of labelmaps while the second one consists of observation images. For each column, it is a pair of one labelmap and its observation.

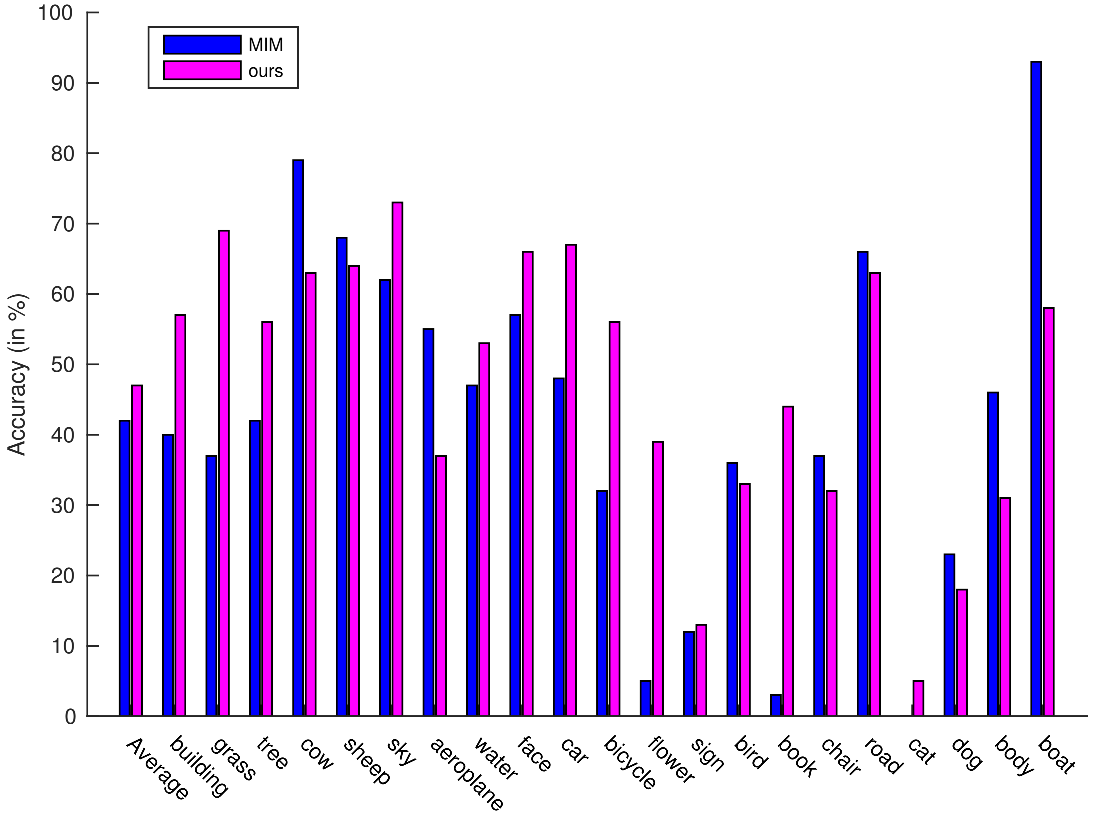


Figure 3.7: Comparison of per-class accuracies for Set one. The first column is the average performance of two algorithms. The left 21 columns are for each object.

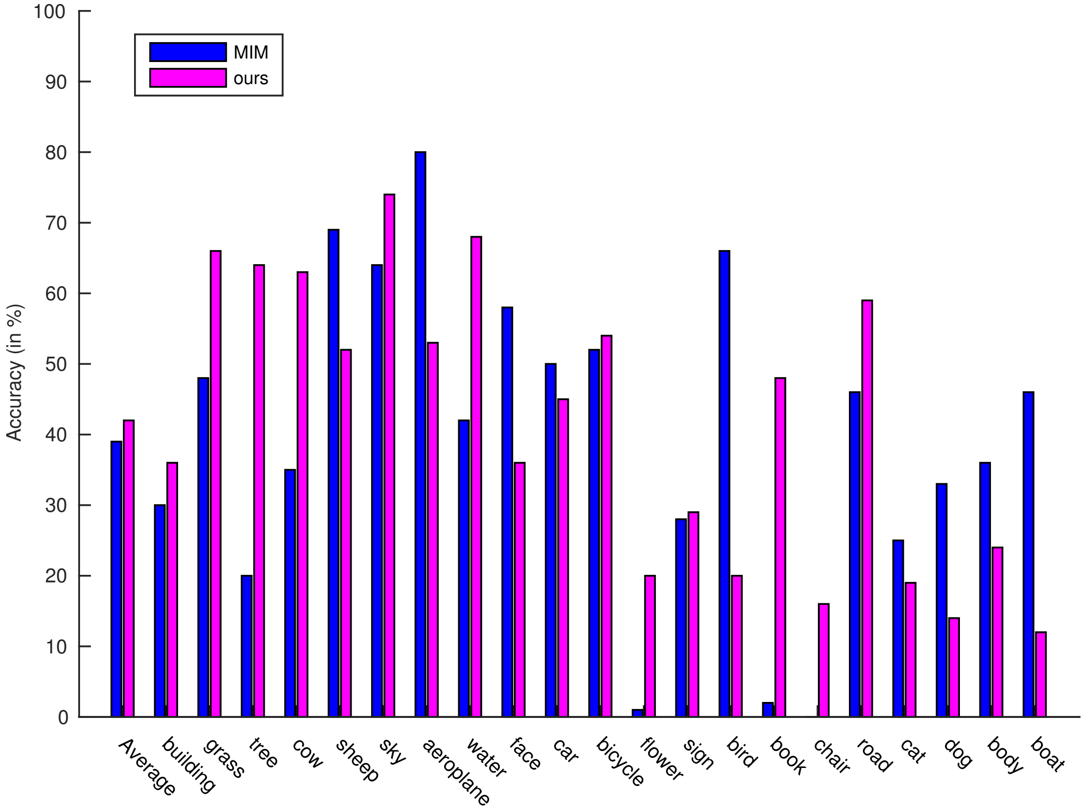


Figure 3.8: Comparison of per-class accuracies for Set two. The first column is the average performance of two algorithms. The left 21 columns are for each object.

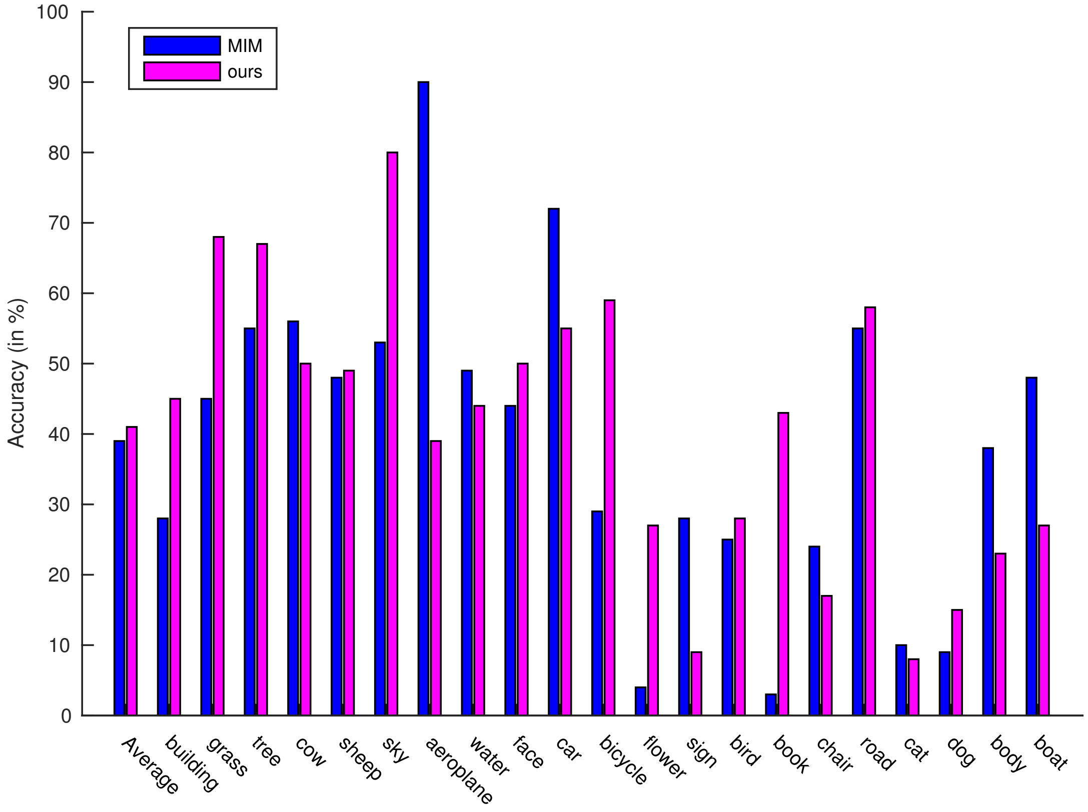


Figure 3.9: Comparison of per-class accuracies for the Entire Set. The first column is the average performance of two algorithms. The left 21 columns are for each object.

Chapter 4

# FEATURE DESIGN FOR TEXT BASED DATA ON SOCIAL NETWORK SITES

This chapter describes how to design features for text based data on social networks. There are two pieces of research involved: one is the feature design for answer quality prediction on community-based question answering, and another one is a classification problem on Twitter website.

## 4.1 Best Answers Prediction in Community-based Question-Answering Services

### *4.1.1 Introduction*

Community-based question-answering (CQA) services help people solve many difficult questions. The importance and huge societal impact of such services are evidenced by the heavy traffic observed on popular CQA sites like Yahoo Answers (answers.yahoo.com), Baidu Zhidao (zhidao.baidu.com), and Stack Overflow (stackoverflow.com). On a CQA site, a person (the asker) posts a question and waits for answers from other users (the answerers). If multiple answers are provided, the asker can select the most suitable one, which is called the *accepted answer* or the *best answer*. Questions that do not have a designated best answer are stamped as ”not-answered”. Not every asker always selects the best answer for his/her question. This could be simply due to lack of action, or due to the difficulties in deciding on the best answer. As a result, many questions are left as ”not-answered” (e.g., see Yang *et al.* (2011)). Not-answered questions do not facilitate knowledge exchange, as other users would hesitate to rely on them for information, given their ”not-answered” labels, even if in reality there may be many good candidate answers posted. Some sites also delete such not-answered questions after certain time of their posting, resulting in lost knowledge if there is indeed a suitable answer posted already. Towards addressing these problems, this chapter focuses on learning from labeled data to predict whether an answer should be selected as the best answer. The study on best answer prediction can also contribute to the understanding of answer quality and help users improve their answers.

For a candidate answer $A_c$ to be considered as the best answer, in general three factors need to be assessed: (1) the quality of the answer content (e.g., its readability); (2) whether the answer contributes to solving the given question $Q$; and (3) how it competes with other answers $A_i$. These are schematically illustrated in Figure 4.1). We call the third factor *contextual information* since it is *relative* in nature. While there have been some reported studies (Adamic *et al.* (2008); Shah and Pomerantz (2010); Blooma *et al.* (2010), to be detailed in the next section) on predicting the best answer, it remains to be fully explored to consider all these factors coherently and to evaluate the importance of the contextual information in solving the problem. This is the objective of this study.

The major contribution of the work is twofold. Firstly, based on the analysis of a large CQA dataset, we designed features to measure the three key factors in selecting the best answer, especially contextual information. Secondly, through designing and evaluating a learning approach using these features to predict whether an answer may be selected as the best answer, we studied the importance of the factors based on their contribution to making the correct prediction.

### *4.1.2 Related Work*

There are a few related studies in the literature. Liu *et al.* worked on predicting the asker's satisfaction with the answers Liu *et al.* (2008). The features used do not measure contextual information among the answers. Harper *et al.* studied answer quality by answering two research questions: how the answer quality in different CQA sites is different from each other and how askers receive better answers Harper *et al.* (2008). They found that fee-based CQA sites are more likely to receive high quality answers. Jeon *et al.* con-

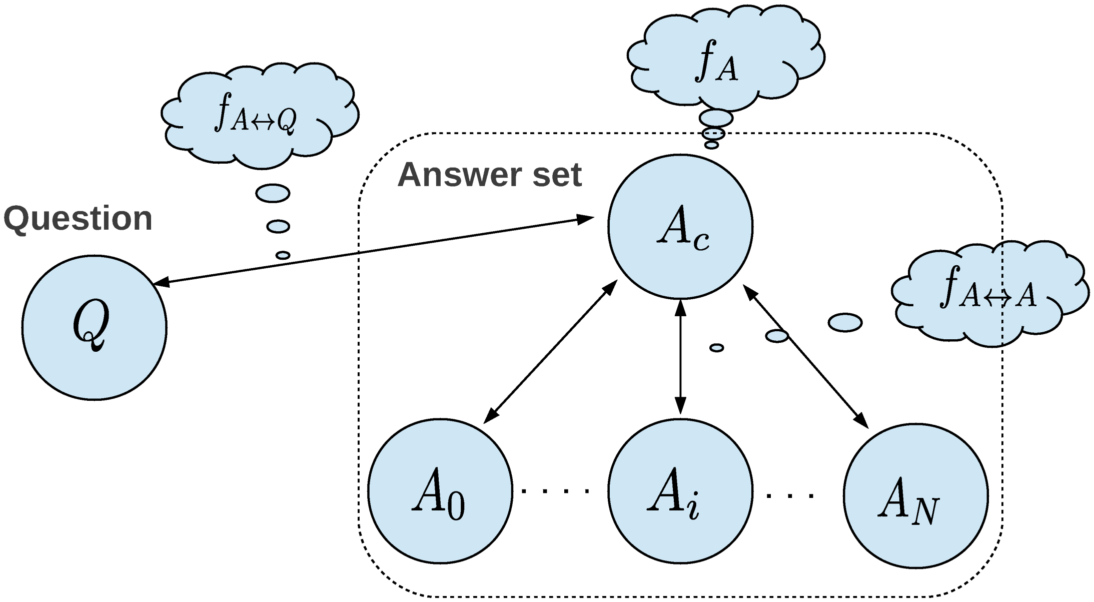


Figure 4.1: It illustrates three factors in assessing the likelihood of an answer $A_c$ under consideration as the best answer: the dash-lined rectangle indicates the answer set to the question $Q$. $f_{A \leftrightarrow Q}$ is the set of features measuring relevance of $A_c$ to $Q$, $f_A$ is the set of features measuring the inherent quality of $A_c$, and $f_{A \leftrightarrow A}$ is the set of features measuring the competition between $A_c$ and the other answers $A_0, \cdots, A_N$.

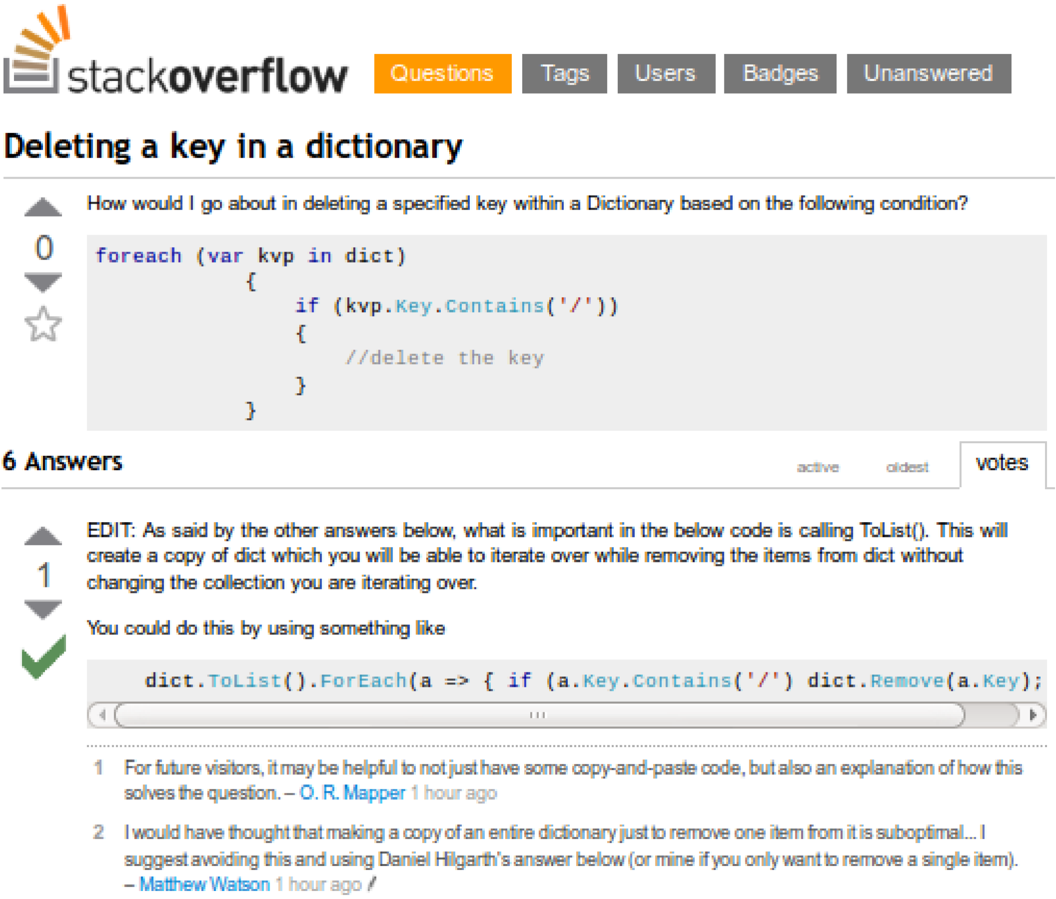
tinued to work on the further effect of price on answer quality in fee-based CQA sites Jeon *et al.* (2010). For the answer quality in different CQA sites, Fichman also made a detailed comparison Fichman (2011). Shah *et al.* worked on the best answer prediction Shah and Pomerantz (2010). In their work, they extracted features which contain information from the questions, the answers, and the users. But there is no consideration on the relationship between the answers and the questions, or relationship among the answers. This is the same case with the work in Blooma *et al.* (2010). Yang *et al.* worked on predicting whether a question will receive the best answer and analyzed which factors contribute to solving the problem Yang *et al.* (2011). Adamic *et al.* studied activity characteristics and mentioned how to predict whether one answer is the best answer given the question with its answers

Adamic *et al.* (2008), using content feature proposed in Agichtein *et al.* (2008). In both cases, not all the factors were considered and especially the contextual information among the answers was not explicitly employed.

### *4.1.3 Stack Overflow Description*

This study is based on Stack Overflow, a CQA site for computer programming, which was selected for its good quality control on the questions (and accordingly the answers) since any post unrelated to programming will be deleted automatically or via voting by senior users. Each question has three main parts: *title*, *body* and *tags*. In the body part, askers can describe their problems in detail. They may use figures or URL links etc. For tags, they may choose at most five existing terms that are most related to the question, or they can create new tags. Each question may receive multiple answers. For each question or answer, users can add *comments* to further discuss it. If one comment is good for solving the problem, it will be awarded with a *score* which shows in front of the comment. For each post (a question or an answer), it will have *upvotes* or *downvotes* from senior users and the corresponding askers or answerers will earn or lose reputation correspondingly. For a question, after it receives multiple answers, the asker can select one which in his or her opinion is most suitable for his or her question. The selected answer is called *Accepted Answer*, which is used in this study interchangeably as the best answer. Figure 4.2 illustrates one sample on Stack Overflow.

The dataset we used in this chapter was downloaded from Stack Overflow for questions and answers posted before August 2012. The original dataset has contains 3,453,742 questions and 6,858,133 answers. In our experiment, we first select questions posted in June 2011 and then track all the answers or comments until August 2012. That is, each question was posted for more than one full years before the answers were collected. In this way, we may assume that all the questions were given enough time to gather good answers.



Deleting a key in a dictionary

How would I go about in deleting a specified key within a Dictionary based on the following condition?

```
foreach (var kvp in dict)
            {
                if (kvp.Key.Contains('/'))
                {
                    //delete the key
                }
            }
```

EDIT: As said by the other answers below, what is important in the below code is calling ToList(). This will create a copy of dict which you will be able to iterate over while removing the items from dict without changing the collection you are iterating over.

You could do this by using something like

```
    dict.ToList().ForEach(a => { if (a.Key.Contains('/') dict.Remove(a.Key);
```

1 For future visitors, it may be helpful to not just have some copy-and-paste code, but also an explanation of how this solves the question. – O. R. Mapper 1 hour ago

2 I would have thought that making a copy of an entire dictionary just to remove one item from it is suboptimal... I suggest avoiding this and using Daniel Hilgarth's answer below (or mine if you only want to remove a single item). – Matthew Watson 1 hour ago

Figure 4.2: This is a sample to show the questions and answers on Stack Overflow site.

This resulted in a subset of 103,793 questions and 196,145 answers, on which the later experimental results were based.

### 4.1.4 Features Description

As described above, our goal is to predict whether an answer will be selected as the best answer. We now design features for a given answer (with its corresponding question and other answers). The questions and answers are first preprocessed via standard procedures as illustrated in Figure 4.3, where the original text streams (sentences) are represented by the vector-space unigram model with TF-IDF weights Shtok *et al.* (2012). In subsequent discussion, this pre-process result will contribute to the extraction of the following features

(Table 4.1, 4.2, 4.3), corresponding to the three factors (Figure 4.1) discussed previously.

### Features Extracted from Answer Context

To describe the context information, we use three features $f_{A \leftrightarrow A}$: similarity between the answer $A_c$ under consideration and other answers $A_i$ to the same question, the number of $A_i$, and the order $A_c$ was created $ans\_index$ (e.g. by sorting the creation time, we know that $A_c$ is the 4th answer to its question). The similarity feature has three dimensions: average, minimum and maximum similarity between $A_c$ and $A_i$ as defined below:

$$ave\_Ans\_sim = \frac{\sum\limits_{i \neq c} sim(A_c, A_i)}{num(A_{i \neq c})} \tag{4.1}$$

$$min\_Ans\_sim = \min_{i \neq c} sim(A_c, A_i) \tag{4.2}$$

$$max\_Ans\_sim = \max_{i \neq c} sim(A_c, A_i) \tag{4.3}$$

where $sim(\cdot, \cdot)$ is the cosine similarity as in Figure 4.3 and $num(A_{i \neq c})$ is the total number of other answers $A_i$.

### Features Extracted from Question-Answer Relationship

This group of features $f_{A \leftrightarrow Q}$ are based on the similarity between $A_c$ and $Q$, which is $sim(A_c, Q)$, and also the time lag between the postings of the question and the answer, which is $timeSlot(A_c, Q)$. Since each question consists of a title and a body, to compute the similarity, we combine the title and the body before calculating the cosine similarity. Because the question can receive an answer at any time if it is not locked or closed, the time lapse between question and answer varies dramatically (e.g., from a few seconds to one year in our data). Thus, we represent this lag using logarithm scale.

$$QA\_sim = sim(A_c, Q) \tag{4.4}$$

$$timeSlot = timeSlot(A_c, Q) \tag{4.5}$$

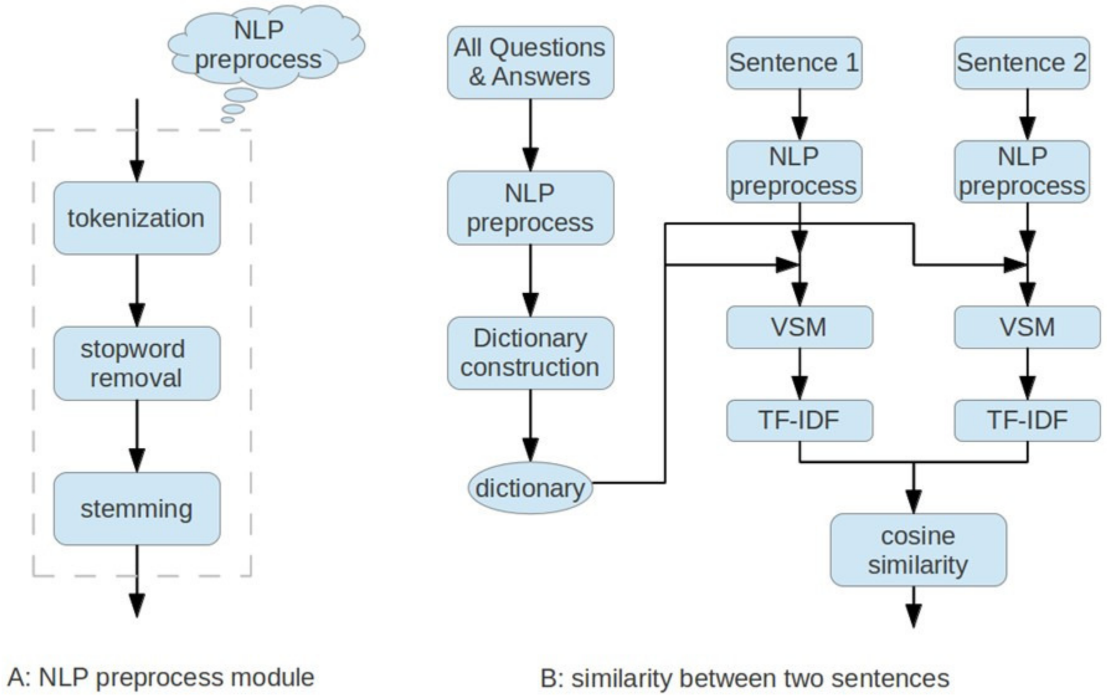


Figure 4.3: This figure shows the process to compute the similarity between two sentences. Part A is the pre-process module which is used in Part B. Part B is the flow chart to show how to compute the similarity.

### Features Extracted from Answer Content

To describe the content quality of an answer, multiple features $f_A$ are defined below:

- Features from the answer body: the length of answer body, whether it has illustration pictures/codes, whether it refers to other web pages using URL, etc. Moreover, if one answer has a clear paragraph structure instead of messing everything up into one paragraph, it will be easy to read and then likely to be selected as a best answer. Thus, the readability of the answer also affects whether the answer will be selected as best answer and we define it as features related with paragraph length (Eq.4.6).

$$\textit{readability} = [\max_i(L_i), \frac{1}{M}\sum_{i=1}^{M} L_i] \tag{4.6}$$

where $L_i$ is the length of $i_{th}$ paragraph of the answer and $M$ is the total number of paragraphs.

- Features from an answer's comments: The features are the number and average score of the comments and the variance of the scores.

Table 4.1, 4.3, 4.2 summarizes the above three types of features. Together, we compute a 16-dimensional feature vector for a candidate answer under consideration.

### *4.1.5 Prediction via Classification*

With the features extracted for a candidate answer, we predict if it may be selected as the best answer through learning a classifier using labelled data: feature vectors corresponding to best answers and non-best-answers according to the ground-truth are used to learn a 2-class classifier. The classifier we used is based on the random forest algorithmBreiman (2001). Random forest is an efficient algorithm to classify large dataset. It also provides an efficient approach to computing feature importance, which is useful for us to analyze the importance of each feature Table 4.1, 4.3, 4.2.

### *4.1.6 Experimental Results*

The experiments were based on the Stack Overflow dataset described earlier. Among the 103,793 questions and 196,145 answers used, there are 4,950 questions that do not have any answer and 45,715 questions with only one answers. For questions with only one, 16,986 of them have no best answers while 28,729 having the best answers. We used all 196,145 answers in our experiment, with the best answers as positive samples and the negative samples being the answers that are not best answers.

We use random forest classifier to do classification and twofold cross-validation. The average accuracy is shown in Table 4.4. We emphasize that the focus of this study is

Table 4.1: Features designed for an answer $A_c$ to a question $Q$. $A_i$ are other answers to $Q$. This table shows features extracted based on the answer only.

| group | index | symbol | feature description |
|---|---|---|---|
| $f_A$ | 0,1 | *ave_comment*, *var_comment* | they are the average and variance of the scores of the comments to $A_c$. |
| | 2 | *comment_num* | $A_c$'s comments number. |
| | 3, 4, 5 | *URL_tag*, *pic*, *code* | they show whether $A_c$ has a URL tag, illustration figures, or codes. |
| | 6 | *ans_len* | it is the length of $A_c$. |
| | 7, 8 | *readability* | they show whether $A_c$ is easy to read, see Eq.4.6 |

on analyzing only features extracted from the questions and answers without using user-specific information. User-specific information, when available, can be used to further improve the performance as done in (Yang *et al.* 2011).

The distribution of the feature importance is shown in Figure 4.4. Both Figure 4.4 and Table 4.4 indicate that features from the answer context $f_{A\leftrightarrow A}$ contribute the most. We also compute the average feature importance from the three groups of features. For features from the answer context, the average feature importance is 0.1202. For the features from the question-answer relationship, the average feature importance is 0.05871. For the features from the answer content, the average feature importance is 0.03128. This also shows the importance of $f_{A\leftrightarrow A}$. In the following, we discuss feature importances based on Figure 4.4, respectively.

Table 4.2: Features designed for an answer $A_c$ to a question $Q$. $A_i$ are other answers to $Q$. This table shows features extracted based on the information from both question and answer.

| group | index | symbol | feature description |
|---|---|---|---|
| $f_{A \leftrightarrow Q}$ | 9 | *QA_sim* | the similarity between $A_c$ and $Q$. (Figure 4.3). |
| | 10 | *timeSlot* | the difference between $A_c$'s creation time and $Q$'s. |

In the group $f_{A \leftrightarrow A}$, the most important feature is *competitor_num*. This suggests that the more competitors the answer $A_c$ has, the less likely is may be selected as the best answer. The feature *min_Ans_sim* has slightly less but comparable importance as *competitor_num*. This shows that the best answer is usually most different from the others. However it does not mean the best answer and the competitors should be totally different. Since all the answers aim at answering the same questions, they also should have similarity. We can see this from the importance of $ave\_Ans\_sim$.

In the group $f_{A \leftrightarrow Q}$, the feature $timeSlot$ contributes more than the feature $QA\_sim$. This shows that earlier answers have a higher chance to be selected as the best answer.

Within the group $f_A$, $comment\_num$ and $ans\_len$ contribute more than the others. This suggests that the best answer is usually the one with more details and comments. This is reasonable and intuitive. The $readability$ feature also contributes significantly, suggesting that answers that are easy to read are likely to be selected.

Table 4.3: Features designed for an answer $A_c$ to a question $Q$. $A_i$ are other answers to $Q$. This table shows features extracted based on the information from the interaction between answers.

| group | index | symbol | feature description |
|---|---|---|---|
| $f_{A \leftrightarrow A}$ | 11, 12, 13 | *ave_Ans_sim,*<br>*min_Ans_sim,*<br>*max_Ans_sim* | the average, minimum, maximum of similarities between $A_c$ and $A_i$. |
| | 14 | *competitor_num* | the number of $A_i$. |
| | 15 | *ans_index* | the order that $A_c$ was created. E.g. it is the $2_{nd}$ answer to the question. |

Table 4.4: Prediction accuracy for different feature groups. $f_{A \leftrightarrow A}$, $f_{A \leftrightarrow Q}$, $f_A$ are three groups of features we described in the previous sections.

| Features | $f_{A \leftrightarrow A}$ | $f_{A \leftrightarrow Q}$ | $f_A$ | $all$ |
|---|---|---|---|---|
| Accuracy | 70.71% | 60.27% | 65.59% | 72.27% |

### 4.1.7 Conclusion and Future work

We studied the problem of predicting the best answer on CQA sites. Our experiments and analysis with a reasonably large dataset have shown that some features, and in particular those reflecting the contextual information among the answers, are more important for the task. The results also suggest that the features designed in the chapter appear to be able to do the job reasonably well. In the future, we plan to study the importance of user-centric information (e.g., usage history, location etc.) for the prediction problem.

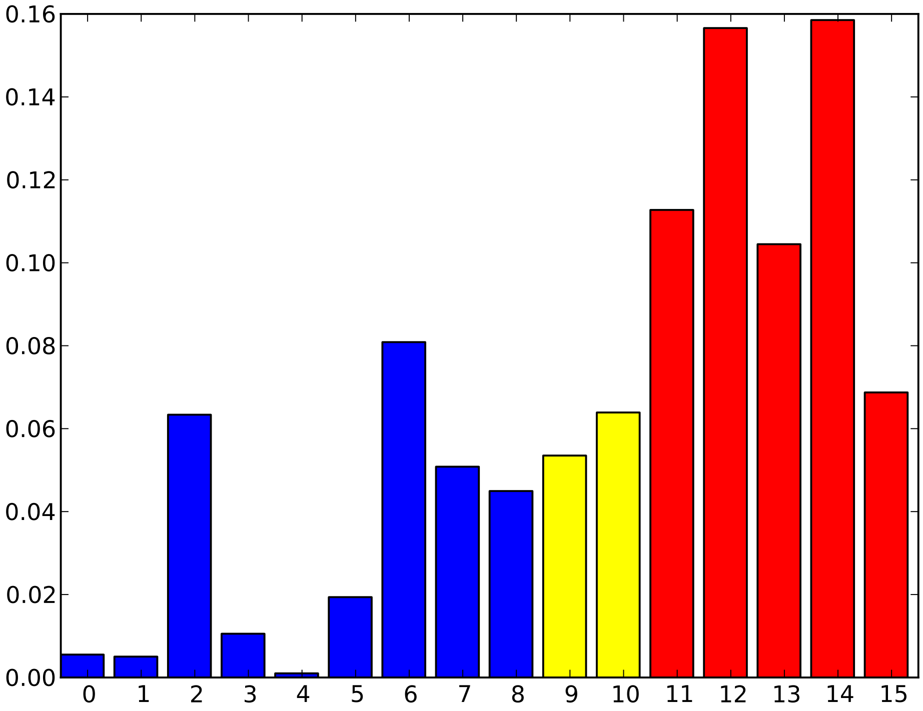


Figure 4.4: The distribution of feature importances. The bars correspond to 16 features defined in Table 4.1, 4.3, 4.2, respectively.

## 4.2 Finding Needles of Interested Tweets in the Haystack of Twitter Network

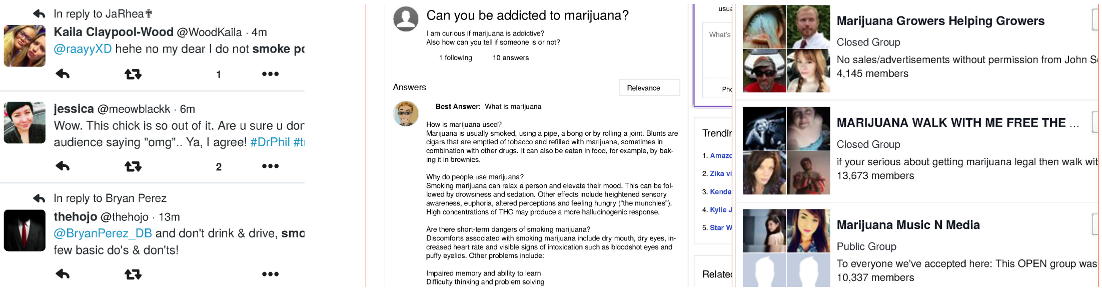
Drug use and abuse is a serious societal problem. The fast development and adoption of social media and smart mobile devices in recent years bring about new opportunities for advancing computer-based strategies for understanding and intervention of drug-related behaviors. However, the existing literature still lacks principled ways of building computational models for supporting effective analysis of large-scale, often unstructured social media data. Part of the challenge stems from the difficulty of obtaining so-called ground-truth data that are typically required for training computational models. This chapter presents a progressive semi-supervised learning approach to identifying Twitter tweets that are related to personal and recreational use of marijuana. Based on a small, labeled dataset, the proposed approach first learns optimal mapping of raw features from the tweets for classification, using a method of weakly hierarchical lasso. The learned feature model is then used to support unsupervised clustering of Web-scale data. Experiments with realistic data crawled from Twitter are used to validate the proposed approach, demonstrating its effectiveness.

### *4.2.1 Introduction*

Drug use/abuse is among the serious societal problems in the modern age. According to a 2011 report Center (2011), in the United States alone, illicit drug use costs the society more than $193 billion annually and the number is increasing. The impact is also widespread: In 2013, about 24.6 million Americans 12 years old or older were illicit drug users Abuse and Administration (2014). Accordingly, a lot of research efforts have been devoted to understanding drug-use-related behaviors and the analysis of potential benefits and limitations of various intervention strategies. A key step in such drug-use-related research is the collection of user behavior data.

Most contentional approaches to user data collection are based on recruitment of participants who would provide inputs to a drug-use-related study, e.g., by answering questionnaires carefully designed to gather various types of behavioral and/or demographical data. For example, to study the relationship between reproductive strategy and views on recreational drug use, Katinka Quintelier *et al.* recruited students from Belgium, Netherland and Japan to fill out paper surveys for data collection. The total number of participants is 476 Quintelier *et al.* (2013). In Lacson *et al.* (2012), John Charles Lacson *et al.* evaluated the association between marijuana use and nonseminoma study. They collected data from 163 patients. There are some well-known limitations in such efforts. For example, the sample size is typically small, as it is in general very costly to involve a large population in such studies. More importantly, such questionnaires in general rely on a participant's explicit recall of his/her drug-use behavior, which could be a limiting factor on its own (e.g., issues like incorrect memory or intentional omission of some facts). (

The phenomenal growth of social media and smart mobile devices has led to more and more drug-use-related data appearing online. For example, there are many drug-related discussion groups on Facebook [1] , many drug-use-related questions asked and answered on Yahoo!Answers [2] , as well as many drug-related tweets on Twitter [3] (see Figure 4.5). )



In reply to JaRhea✟
Kaila Claypool-Wood @WoodKaila · 4m
@raayyXD hehe no my dear I do not **smoke p**
1

jessica @meowblackk · 6m
Wow. This chick is so out of it. Are u sure u dor
audience saying "omg".. Ya, I agree! #DrPhil #t
2

In reply to Bryan Perez
thehojo @thehojo · 13m
@BryanPerez_DB and don't drink & drive, **smo**
few basic do's & don'ts!

Can you be addicted to marijuana?
I am curious if marijuana is addictive?
Also how can you tell if someone is or not?
1 following 10 answers

Answers
Relevance

**Best Answer:** What is marijuana

How is marijuana used?
Marijuana is usually smoked, using a pipe, a bong or by rolling a joint. Blunts are cigars that are emptied of tobacco and refilled with marijuana, sometimes in combination with other drugs. It can also be eaten in food, for example, by baking it in brownies.

Why do people use marijuana?
Smoking marijuana can relax a person and elevate their mood. This can be followed by drowsiness and sedation. Other effects include heightened sensory awareness, euphoria, altered perceptions and feeling hungry ("the munchies"). High concentrations of THC may produce a more hallucinogenic response.

Are there short-term dangers of smoking marijuana?
Discomforts associated with smoking marijuana include dry mouth, dry eyes, increased heart rate and visible signs of intoxication such as bloodshot eyes and puffy eyelids. Other problems include:

Impaired memory and ability to learn
Difficulty thinking and problem solving

**Marijuana Growers Helping Growers**
Closed Group
No sales/advertisements without permission from John S
4,145 members

**MARIJUANA WALK WITH ME FREE THE ...**
Closed Group
if your serious about getting marijuana legal then walk wit
13,673 members

**Marijuana Music N Media**
Public Group
To everyone we've accepted here: This OPEN group was
10,337 members

Figure 4.5: The left one illustrates related tweets on Twitter, the middle one shows an example of one question and its answer related with marijuana on Yahoo! Answers, and the right one shows several groups related with marijuana on Facebook.

Such user-generated social media may be collected at a much larger scale (than an explicit user survey) and thus have the potential of offering realistic insights into understanding of substance-use behaviors, their situational factors, and social contexts. A few recent efforts illustrate this nicely. In Lee (2014), Christine Lee *et al.* found that the substance-use related behaviors have similar patterns in data from traditional survey-based approaches and those from social media. In Whitehill *et al.* (2015), Jennifer Whitehill *et al.* studied the relationship between mobile usage of social networking sites (e.g. Facebook and Twitter) and the alcohol use in a large street festival. In van Hoof *et al.* (2014), Joris Hoof *et al.*

[1] https://www.facebook.com/

[2] https://answers.yahoo.com/

[3] https://twitter.com/

conducted one study on analyzing Facebook profiles to show that some Facebook profile elements can be the indicators of real-life behaviors. In Stoddard *et al.* (2012), Sarah Stoddard *et al.* examined the influence of young people's social networking behaviors on their alcohol and other drug use. They found that peer influence is an important factor in alcohol and marijuana use not only in person but also on-line.

While having demonstrated to some extent the potential of using social media for substance-use research, these existing efforts also revealed the challenges of building computational models for analyzing largely-unstructured social-media. For example, some user attributes that may be readily available from an explicit survey now need complex inference strategies to figure them out. Further, any approach that relies on training from some labelled dataset cannot be easily extended to large-scale analysis. In this chapter, we address some of these challenges in the context of illicit marijuana use and its manifestation on Twitter. Specifically, we propose one semi-supervised approach to studying the user behaviors of the illicit marijuana use using noisy, unstructured and large-scale Twitter data. We first study the feature selection scheme via one classification task, which is to predict whether one Twitter tweet is related to personal and recreational marijuana use based on a small labeled dataset. Building on top of the results from the small labelled dataset, we then develop an unsupervised clustering scheme for processing Web-scale data to further improve the analysis. To our knowledge, this is the first work to study marijuana use behaviors using large-scale Twitter data, and the proposed semi-supervised approach is shown to be effective and efficient. We will make the dataset public for other researchers to further evaluate.

The rest of the chapter is organized as follows. A brief review of related work is given in Section 4.2.2. In Section 4.2.3, the research problem of our effort is defined and the features we use are also described. We show how to learn the feature selection scheme in Section 4.2.4 and how to use the learned scheme to improve the clustering of the large-scale dataset in Section 4.2.5. Finally, we show the experimental results in Section 4.2.6 and conclude this chapter and present some limitations of our study in Section 4.2.7.

### *4.2.2 Related Works*

In this section, we briefly review some related work on study of use of marijuana and other substance, including both traditional methods of recruiting participants and more recent approaches using social media data.

#### **Participant-recruitment Based Research**

In Bachman *et al.* (1991), Bachman *et al.* used questionnaires to study the racial/ethnic differences in smoking, alcohol use and drug use in American high school seniors from 1976 to 1989. The data collection lasted for many years. Johnston *et al.* conducted follow-up surveys on young adults regarding their behaviors related to drug use in Johnston (2010). They discussed several trends in use patterns of typical drugs, alcohol, and cigarette smoking among young people and also the difference of drug use between the college and non-college populations, male and female and so on. Schuster *et al.* recruited 9th and 10th graders from sixteen Chicago high schools to study the gender specific associations between marijuana use and risky sexual behaviors and other depressive symptoms in Schuster *et al.* (2013). Marijuana use may also affect the development of intelligence. To show this, a longitudinal study of 614 families for several years by Jackson *et al.* was reported in Jackson *et al.* (2016). The result shows that there is little direct evidence that marijuana use in adolescent has a negative effect on IQ.

As noted earlier, these population-survey-based efforts are usually very time-consuming merely for the stage of data collection. Another point to note is that the above-mentioned efforts focused more on finding features or trends from the data rather than developing computational approaches for modeling user behaviors.

### Research Using Social Media

Social media and mobile Internet use in teens and young adults was studied by Lenhart *et al.* in Lenhart *et al.* (2010). They found that 47% online adults and 72% online 18-29-year-olds use social networking websites. Ramo *et al.* found that it is useful and cost-effective to use Facebook as recruitment source to do research on substance use Ramo and Prochaska (2012). More than 200 online forums or websites in 7 European countries were monitored to identify emerging trends in recreational drug use by Deluca *et al.* Deluca *et al.* (2012). The non-medical use of Adderall (one psychostimulant drug) among college students using Twitter were studied in Hanson *et al.* (2013), where the frequencies, percentages and means were analyzed, and the experiments showed that their findings were similar to traditional survey-based methods. To study the smoking behavior on Twitter, Myslin *et al.* collected tweets from Twitter and performed content and sentiment analysis Myslín *et al.* (2013). Cavazos-Rehg *et al.* also performed content analysis of tweets but with a pro-marijuana Twitter handle (@stillblazingtho) plus the demographics of the handle's followers Cavazos-Rehg *et al.* (2014). Volkow *et al.* reported risks of the recreational use of marijuana like the risk of addiction, effect on brain development, relation to mental illness and so on in Volkow *et al.* (2014). They also showed that there are about 12% of people who use marijuana as non-medical drug, especially among the young people. Krauss *et al.* studied the hookah smoking behavior on Twitter in Krauss *et al.* (2015). They coded each tweet using a Likert scale from 1 to 5, and relied on collecting the crowd-sourcing results. Leah *et al.* reported their research on how posts on Twitter changed after legalizing recreational use of marijuana in two states Thompson *et al.* (2015). Katsuki *et al.* studied the youth non-medical use of prescription medications (NUPM) on Twitter in order to model the frequency of NUPM-related tweets and identified the illegal access to drug abuse via online pharmacies in Katsuki *et al.* (2015). They labeled the tweets to see if they are related with

NUPM behavior and also whether a user has positive or negative attitudes towards NUPM. Then a Support Vector Machine (SVM) was used to do classification on the tweets.

While demonstrating the great potential of using social media for substance-use-related analysis, these existing efforts have yet to be extended to Web-scale data. In particular, we have not seen specific computational models for analyzing Web-scale Twitter data for understanding marijuana-use-related behaviors. As noted earlier, part of the challenge lies in the difficult of obtaining labeled training data. To address these issues is among the motivations for our work in this chapter.

### *4.2.3 Problem Definition*

Twitter is one popular social networking service by which people can post photos, videos and up to 140 characters of text. These posts are called *tweets*. To study the behavior of marijuana users on Twitter, a fundamental problem is to identify tweets that are related to some underlying users who use marijuana. This problem is more subtle than it appears. For example, one cannot simply rely on using the keyword "marijuana" to search the tweets for solving the problem. There are several complicating factors. First, many "street names" are used to describe marijuana and in fact most recreational marijuana users never use the term "marijuana" explicitly. Second, there may be many tweets that involve medical or research-oriented references to marijuana but they are not at all useful for a study on illicit marijuana use. Considering these factors, we propose to classify a tweet into one of following three categories:

- Class One: Tweets in this class are related to personal recreational use of marijuana. They are posted by individual users instead of some official accounts (for example, those for newspaper, companies, or medical institutes).

- Class Two: In this class, all tweets are related to marijuana but not in the sense of recreational use. For instance, they may discuss the medical or prescription use of marijuana, or report some news involving marijuana.

- Class Three: This is for those tweets having no identifiable relationship with marijuana use.

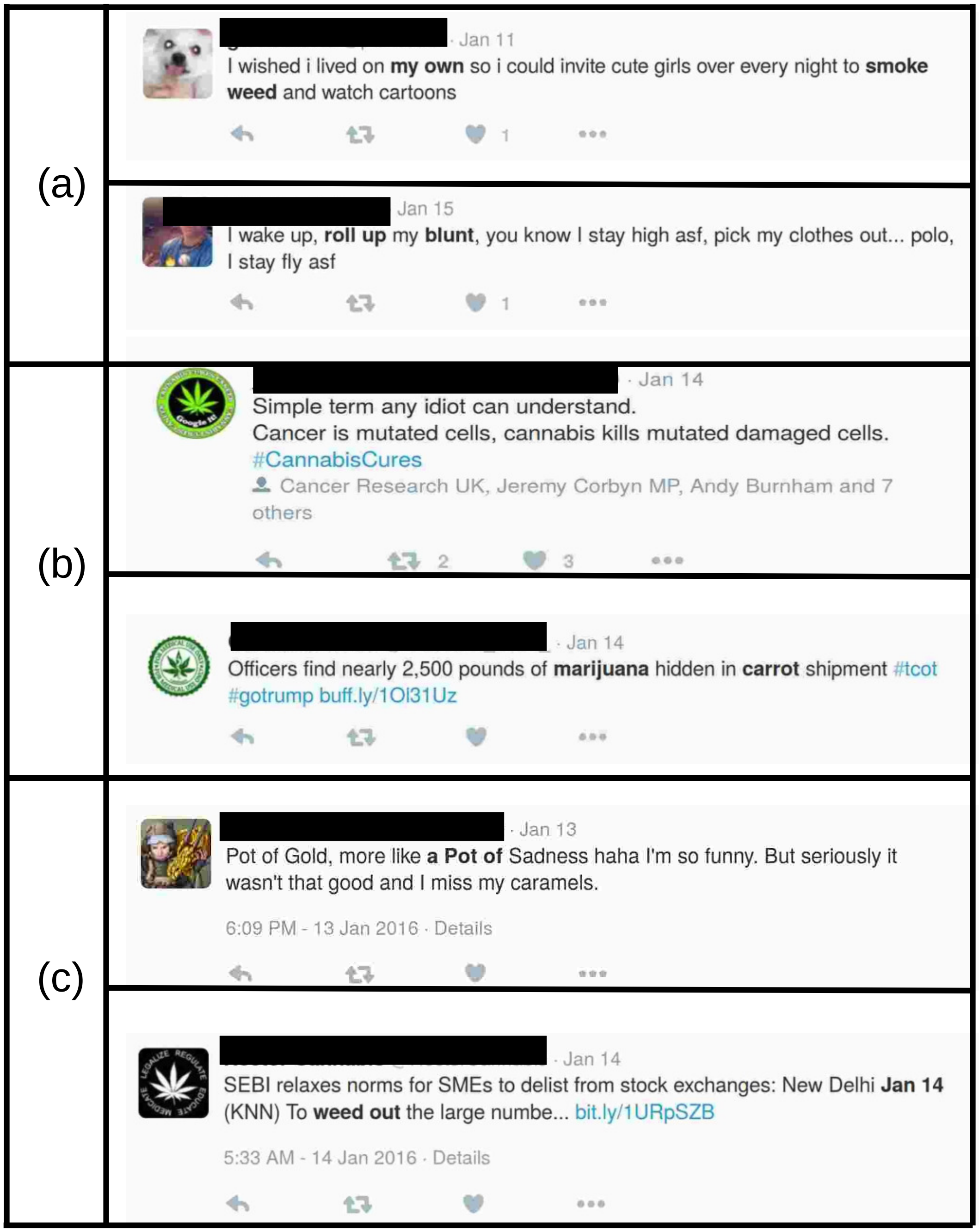
Figure 4.6 illustrates several real examples for each of the three classes defined above.

Various text-based features may be extracted for the task of classifying the tweets. Also, as evident from the related work, it is important to consider social interactions among the underlying users. Furthermore, all these features are not mutually independent, and their intricate correlation may provide additional evidence for improved classification. Considering these, and with the goal of classifying large-scale tweets in mind, we now discuss our overall approach, which is illustrated in Figure 4.7. In the approach, we first extract a set of basic features from each tweet. Then, utilizing a small labelled training set, we learn a good feature mapping that takes into consideration both some basic features *and* their interactions, based on weakly-hierarchical lasso. The learned feature mapping model is used to process the large-scale data and perform clustering. As the features are optimized for classifying the tweets into the predefined three classes, the hypothesis is that the unsupervised clustering results give arise to clusters corresponding to the three classes (which will be evaluated in the Experiments section). 4.7.

In the following, we first present the basic set of features designed for our task. These features are extracted from either the content of the underlying tweets or the social interactions among the corresponding users, as elaborated below.

### Content-based Features

- The length of the tweet: For each tweet, its length can be one useful feature. For example, the tweets from ordinary users may be generally shorter than those from



[redacted] · Jan 11
I wished i lived on **my own** so i could invite cute girls over every night to **smoke weed** and watch cartoons
1

[redacted] Jan 15
I wake up, **roll up** my **blunt**, you know I stay high asf, pick my clothes out... polo, I stay fly asf
1

[redacted] · Jan 14
Simple term any idiot can understand.
Cancer is mutated cells, cannabis kills mutated damaged cells.
#CannabisCures
Cancer Research UK, Jeremy Corbyn MP, Andy Burnham and 7 others
2 3

[redacted] · Jan 14
Officers find nearly 2,500 pounds of **marijuana** hidden in **carrot** shipment #tcot #gotrump buff.ly/1Ol31Uz

[redacted] · Jan 13
Pot of Gold, more like **a Pot of** Sadness haha I'm so funny. But seriously it wasn't that good and I miss my caramels.
6:09 PM - 13 Jan 2016 · Details

[redacted] · Jan 14
SEBI relaxes norms for SMEs to delist from stock exchanges: New Delhi **Jan 14** (KNN) To **weed out** the large numbe... bit.ly/1URpSZB
5:33 AM - 14 Jan 2016 · Details

Figure 4.6: Demos to show three classes: (a) is for Class One, (b) is for Class Two and (c) is for Class Three.

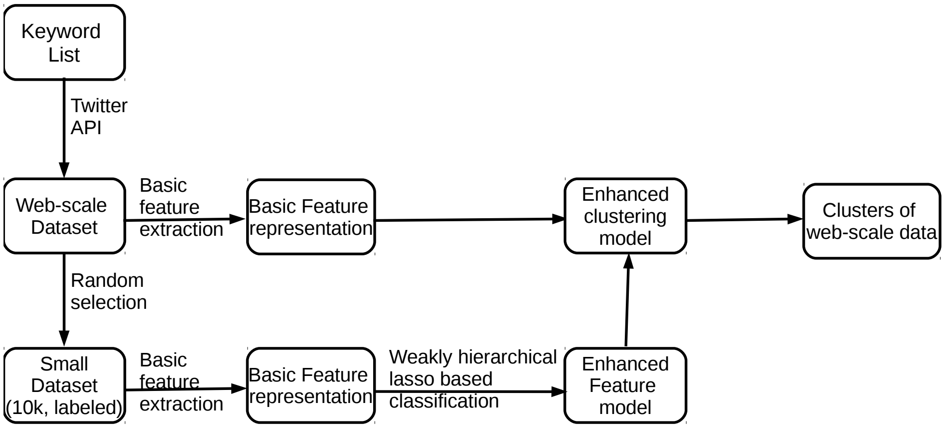

Keyword List
Twitter API
Web-scale Dataset
Basic feature extraction
Basic Feature representation
Enhanced clustering model
Clusters of web-scale data
Random selection
Small Dataset (10k, labeled)
Basic feature extraction
Basic Feature representation
Weakly hierarchical lasso based classification
Enhanced Feature model

Figure 4.7: It shows the entire framework of our methodology.

official accounts.

- Favorite Count & Retweeted Count: It shows how many people think the tweet is favorite and the number people who retweet this post. This is in general useful for measuring how influential the tweet is.

- The number of Hash-Tags: This calculates how many trends one tweet mentions. Our original dataset were obtained by crawling using selected street names of marijuana. The tweets with more trends are likely to be classified as Class Three or Two, instead of Class One.

- TF-IDF on Unigram: Unigram is one common feature used to capture characteristics of one tweet. We build TF-IDF for unigrams of each tweet and use it as one feature.

### User-based Features

- Number of followings and followers: Each user on Twitter can follow others or be followed. However for some official accounts or famous people, they are likely to have a smaller number of followings but a large number of followers. These users are unlikely to post tweets related to personal and recreational use of marijuana.

- Number of Tweets: This records how many tweets one user has already posted, capturing the level of Twitter activity of the user.

### *4.2.4 Learning Feature Mapping From A Small Dataset*

Considering the computational efficiency needed for processing Web-scale data, we may employ a linear classifier as the baseline for doing the classification, as given by Eqn.4.7.

$$y_i = f(x_i w) \tag{4.7}$$

where the $i^{th}$ data point is $x_i \in \mathbb{R}^{1\times d}, i \in \{1, \cdots, N\}$ which is normalized, and its label is $y_i \in \{1, 2, 3\}$ and the coefficient to learn is $w \in \mathbb{R}^{d\times 1}$. In this chapter, the discriminant function is chosen to be one-vs-one linear SVM. The implementation details are provided as follows. We first train one linear regression model by optimizing Eqn.4.8.

$$\min_{w} \quad \|Xw - y\|_2^2 + \frac{1}{2}\|w\|_2^2 \tag{4.8}$$

where $X \in \mathbb{R}^{N\times d}$ and $y \in \mathbb{R}^{N\times 1}$. Then we apply one-vs-one linear SVM to $s = Xw \in \mathbb{R}^{N\times 1}$ to find the label for each tweet.

$$\min_{v} \quad \|v\|_2^2 + C\sum_{i=1}^{N} \xi_i \tag{4.9}$$

$$\text{s.t.} \quad y_i(s_i * v + b) \geq 1 - \xi_i \;\; \forall i \tag{4.10}$$

where $\xi$ is non-negative.

However, in practice, the linear model is inadequate for capturing the high degree of non-linearity that typically exists in our problem, which has been shown in our experiments. To allow some level of nonlinearity while maintaining computational efficiency, we introduce to the problem $2^{nd}$-order interaction terms with a weakly hierarchical structure, as described in Bien *et al.* (2013)Liu *et al.* (2014). The resultant model is given in Eqn.4.11.

$$\begin{aligned} y &= f(z) \\ z &= xw + \frac{1}{2}\sum_{i}^{d}\sum_{j}^{d} x_i x_j Q_{i,j} \end{aligned} \tag{4.11}$$

where $z$ is called the $z$-term of $x$ (for simplicity) and the discriminant function $f(\cdot)$ is given in Eqn.4.9 (one-vs-one linear SVM in this chapter) and $x_i$ is the $i^{th}$ dimension of the data point $x$ and $Q_{i,j} \in R$ is the coefficient for the interaction between $i^{th}$ and $j^{th}$ dimensions of the feature space.

To solve the classification problem under this new model, we formulate the following optimization problem in Eqn.4.12.

$$\begin{aligned} \min_{w,v,Q} \quad & \frac{1}{2}\sum_{i}(f(z_i, v) - y_i)^2 + \lambda_1 \|w\|_1 + \frac{\lambda_3}{2}\|Q\|_1 \\ \text{s.t.} \quad & \|Q_{\cdot,j}\|_1 \leq |w_j| \quad \text{for} \quad j = \{1, \cdots, d\} \end{aligned} \tag{4.12}$$

where $z_i$ is the $z$-term of $x_i$ as defined in Eqn.4.11, $\|Q\|_1 = \sum_{i,j} |Q_{i,j}|$ and $v$ is the model parameter of the discriminant function (the one-vs-one linear SVM).

### Solving the Optimization Problem

Solving Eqn.4.12) directly is difficult. Hence we simplify this optimization problem by a two-step process: We first learn parameters $w$ and $Q$ and then learn the model parameter $v$ of the discriminant function.

For parameters $w$ and $Q$, we model them as one regression model as Eqn.4.13 when we do not consider the discriminant function.

$$\begin{aligned} \min_{w,Q} \quad & \frac{1}{2}\sum_{i}(z_i - y_i)^2 + \lambda_1 \|w\|_1 + \frac{\lambda_3}{2}\|Q\|_1 \\ \text{s.t.} \quad & \|Q_{\cdot,j}\|_1 \leq |w_j| \quad \text{for} \quad j = \{1, \cdots, d\} \end{aligned} \tag{4.13}$$

where $z_i$ is the $z$-term of $x_i$ as defined in Eqn.4.11. Then after $w$ and $Q$ are obtained, we learn $v$ of the discriminant function by optimizing Eqn.4.9.

Converting Eqn.4.12 into Eqn.4.13 and Eqn.4.9 allows us to solve the original optimization problem. By solving Eqn.4.13 and Eqn.4.9, we can obtain the model parameters $v$, $w$ and $Q$ which satisfy the original problem (Eqn.4.12) as well. However, since we add more constraints on these parameters in the process of simplification, the obtained $v$, $w$ and $Q$ are only the local optima of Eqn.4.12.

While the details for solving Eqn.4.13 can be found in Bien *et al.* (2013), a brief description is given below. From Eqn.4.13, we can see that this optimization problem is non-convex because of the existence of constraints, and as a result, we cannot solve it using convex optimization approaches. Thus in Bien *et al.* (2013), one convex relaxation by setting $w = w^+ - w^-$ is given, where $w^+$ and $w^-$ are nonnegative. The convex relaxation version is given as Eqn.4.14.

$$\min_{w^+,w^-,Q} \quad \frac{1}{2}\sum_i (\hat{z}_i - y_i)^2 + \lambda_1(w^+ + w^-) + \frac{\lambda_3}{2}\|Q\|_1 \tag{4.14}$$

$$\text{s.t.} \quad \|Q_{\cdot,j}\|_1 \leq w_j^+ + w_j^- \quad \text{for} \quad j = \{1, \cdots, d\}$$

$$w_j^+, w_j^- \geq 0 \quad \text{for} \quad j = \{1, \cdots, d\} \tag{4.15}$$

where $\hat{z}_i = x_i \cdot (w^+ - w^-) + \frac{1}{2}\sum_j^d \sum_k^d x_{i,j}x_{i,k}Q_{i,j}$. A lot of convex optimization approaches can be used to solve Eqn.4.14, such as FISTA Beck and Teboulle (2009).

After we obtain the parameters $w$ and $Q$, we can learn the parameter $v$ of the discriminant function, which is given by Eqn.4.16.

$$\min_v \quad \frac{1}{2}\|v\|_2^2 + C\sum_{i=1}^N \xi_i \tag{4.16}$$

$$\text{s.t.} \quad y_i(z_i * v + b) \geq 1 - \xi_i \ \forall i \tag{4.17}$$

where $\xi$ is non-negative. This can be solved by working on its duality problem as in

Eqn.4.18, using sequential minimal optimization Platt *et al.* (1998).

$$\max_{\alpha} \quad \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} y_i y_j \alpha_i \alpha_j z_i z_j \tag{4.18}$$

$$\text{s.t.} \quad 0 \leq \alpha_i \leq C \ \ \forall i \tag{4.19}$$

$$\sum_i y_i \alpha_i = 0 \tag{4.20}$$

### *4.2.5 Clustering with The Learned Feature Mapping*

A supervised approach cannot be directly applied to Web-scale datasets as manually-labeled data are in general in a much smaller scale. A semi-supervised approach would rely on unsupervised clustering to first identify the structures of the data and then employ a small amount of labeled data to annotate the structures. For example, using K-means clustering, we can group a dataset into different clusters. For data points in each cluster, if we assume that they have the same labels, we can randomly select a small number of data points for labeling and then use the labels to annotate the clusters. Assuming $k$ groups in a dataset, a basic K-means algorithm is equivalent to solving the following problem (Eqn.4.21):

$$\min_{\pi_j, j \in \{1,\cdots,k\}} \sum_{j=1}^{k} \sum_{v \in \pi_j} \|x_v - c_j\|_2^2 \tag{4.21}$$

where $c_j$ is the $j^{th}$ centroid and $\pi_j$ is the $j^{th}$ cluster.

As we have presumably found a feature mapping scheme in the previous section by maximizing classification accuracy for the labelled data, it is natural to use the learned feature mapping for the clustering stage. Denote the dataset as $\{x_i, i \in \{1, \cdots, N\}\}$. Consider the influence of the 2-order feature interaction, the dataset representation is converted as $\{\tilde{x}_i, i \in \{1, \cdots, N\}\}$ where $\tilde{x}_i$ is given by Eqn.4.22.

$$\tilde{x}_i = (x_i, vec(S_i)) \tag{4.22}$$

where the element at $(j, k)$ in the matrix $S_i$ is the product of the $j^{th}$ and $k^{th}$ dimension which is $x_{i,j}x_{i,k}$. It is easy to see $\tilde{x}_i \in \mathbb{R}^{1\times(d+d^2)}$. For the new representation, the interaction of the feature dimension is captured by parameters $w$ and $Q$ which are learned from the small labeled dataset (see Section 4.2.4). By treating the learned parameters as a kernel, we can have the new clustering as Eqn.4.23.

$$\min_{\pi_j, j\in\{1,\cdots,k\}} \sum_{j=1}^{k} \sum_{v\in\pi_j} (\tilde{x}_v - c_j) M (\tilde{x}_v - c_j)^T \tag{4.23}$$

where the learned metric matrix $M = diag((w; vec(Q))) \in \mathbb{R}^{(d+d^2)\times(d+d^2)}$.

### *4.2.6 Experiments*

In this section, we evaluate the performance of our approach with comparison with several typical existing methods. We report two main experiments: the first one evaluate the the feature mapping scheme learned from the small labeled dataset, and the other one is about how to apply the learned scheme to the large-scale data.

#### Dataset Construction

For constructing a small labelled dataset, instead of crawling random tweets online, we first use a list of keywords as one filter to remove unrelated tweets. These keywords are defined based on several Web sources and some government documents. The overall process for crawling tweets to form the evaluation datasets is summarized below:

- Obtain a list of street names for marijuana based on some marijuana-related research and government Websites; Rank the street names based on their frequency of occurrences on the list of Websites.

- Choose top $k_1$ names and then for each one, we can crawl $n$ tweets.

- Label these tweets and compute class distribution.

- Based on the class distribution, we can choose top $k_2$ names as the final keywords.

In this chapter, we use one famous Drug Rehabs online treatment center as the online-forum [4] to find a list of street names for the marijuana. Meanwhile, we also use one official document from a government source as another guideline [5] . It is worth noting that, since some of the street names for marijuana are common words (e.g., weed, pot), crawling tweets based on the above list of street names inevitably results in all three classes of tweets (not only class 1 and class 2). Hence the last step is to estimate a more proper list for getting a good distribution for the three classes. We used parameters $k_1 = 30$, $k_2 = 10$ and $n = 50$. The final keyword list was determined to be: *marijuana*, *weed*, *blunt*, *cannabis*, *pot*, *reefer*, *buds*, *420*, *mary jane*, *blaze*.

With the final list, the Twitter API [6] is utilized to crawl data. The time period we crawled is from January 09 to January 15 in 2016 and all tweets are in English. We crawled a total of 1,166,441 tweets. Among these we randomly labeled 10,000 with comparable proportion for each class (see Table 1 for exact composition in terms of class labels). This small labelled dataset was annotated by two people reading the tweets to decide their labels, using the interface shown in Figure 4.8.

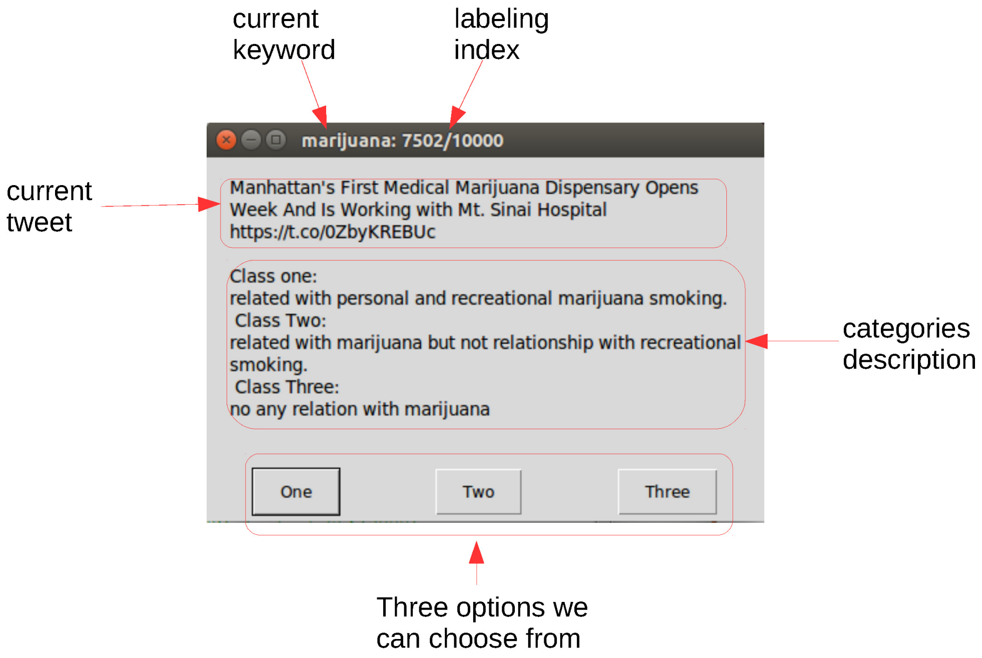
### Experiment Settings

Two experiments are performed to show the effectiveness and efficiency of our approach. In the first one, based on the small labeled dataset, we learn the optimal feature structure based on weakly hierarchical lasso and then compare with commonly used approaches like linear classifier (Eqn.4.7) and linear SVM. These two baselines are chosen because in the large-scale dataset, linear algorithms are commonly used. Moreover, random guess also is

[4] www.rehabs.com

[5] http://www.vva.org/documents/VAD_Materials/Supplemental%20Materials/street_terms.pdf

[6] https://dev.twitter.com/rest/public



Manhattan's First Medical Marijuana Dispensary Opens Week And Is Working with Mt. Sinai Hospital https://t.co/0ZbyKREBUc

Class one:
related with personal and recreational marijuana smoking.
Class Two:
related with marijuana but not relationship with recreational smoking.
Class Three:
no any relation with marijuana

Figure 4.8: The simple interface for manual labelling of crawled tweets.

chosen as one baseline. In the second experiment, the learned feature mapping is applied to the large-scale dataset for clustering.

### Learning the Feature Mapping

In this part, to compare with the baselines, we split the 10,000 tweets randomly into two parts: training set of 8,000 tweets and testing set with 2,000 tweets. The distributions of each class in both sets are shown as Table.4.5. All the feature vectors are normalized. Since in the our approach, we need to compute the feature interaction terms which is defined as the $z$-term in Eqn.4.11, we have to reduce the dimension of the original feature vectors. In this experiment, we use LDA Gu *et al.* (2011) to do dimension reduction of TF-IDF of Unigram in the feature sets for our approach. For random guess, we randomly assign one

Table 4.5: The statistics of training set and the testing set. C1: Class One, C2: Class Two, C3: Class Three.

| | C1 | C2 | C3 | all |
|---|---|---|---|---|
| training | 3,061 | 2,017 | 2,922 | 8,000 |
| testing | 769 | 500 | 731 | 2,000 |

label to every data point and then compute the accuracy based on Eqn.4.24.

$$e = \frac{\sum_{i=1}^{N_t} I(y_i == \hat{y}_i)}{N_t} \tag{4.24}$$

where $y_i$ is the ground-truth label of the tweet $x_i$ and $\hat{y}_i$ is the predicted label and

$$I(x) = \begin{cases} 1 & \text{if } x \text{ is true} \\ 0 & \text{otherwise} \end{cases} \tag{4.25}$$

Table 4.6: The confusion matrix for *LC*

| | C1 | C2 | C3 |
|---|---|---|---|
| C1 | 0.4616 | 0.3108 | 0.2276 |
| C2 | 0.3820 | 0.3920 | 0.2260 |
| C3 | 0.1751 | 0.3146 | 0.5103 |

The experiment results are shown in Table 4.9 and The confusion matrix of our approach is shown in Table 4.6, 4.7, 4.8.

From Table. 4.9, we can easily see that our algorithm stands out. Compared with the modified linear classifier (Eqn. 4.7 and Eqn. 4.9) with our algorithm, the difference is that we consider the interaction terms (the $z$-term) defined in Eqn.4.11. Thus these results also show that it is necessary to consider feature selection scheme using weakly hierarchical lasso. Furthermore, our approach performs better than linear SVM. This is

Table 4.7: The confusion matrix for the algorithm *SVM*.

| | C1 | C2 | C3 |
|---|---|---|---|
| C1 | 0.6060 | 0.1508 | 0.2432 |
| C2 | 0.1820 | 0.6120 | 0.2060 |
| C3 | 0.1259 | 0.0780 | 0.7962 |

Table 4.8: The confusion matrix for the algorithm *Ours*.

| | C1 | C2 | C3 |
|---|---|---|---|
| C1 | 0.9831 | 0.0130 | 0.0039 |
| C2 | 0.0020 | 0.9920 | 0.0060 |
| C3 | 0.0014 | 0.0410 | 0.9576 |

Table 4.9: The table shows the performance of each baseline and our method. RG: random guess; LC: linear classifier; SVM: linear SVM.

| *RG* | *LC* | *SVM* | *Ours* |
|---|---|---|---|
| 0.326 | 0.462 | 0.677 | 0.976 |

also easy to understand because of the nonlinearity introduced in our formulation (Eqn. 4.11). Nonlinearity comes from the $z$-term.

To further show the performance of each algorithm, the confusion matrices are shown in Table. 4.6, 4.7, 4.8. It shows that our approach performs best in all of the three classes. From Table 4.6, we can see that *LC* cannot distinguish Class 1 and Class 2. For example, for Class 2, almost the same number of tweets are classified into Class 1 and Class 2. The baseline with *SVM* performs better than *LC*, but the error is still significant.

Our approach effectively solves the problem of how to fuse features and provides the optimal feature selection/combination scheme. It is possible to analyze which features (or their interactions) are most influential. Table 4.10 shows the top three main factors which affect the classification performance and their corresponding coefficients. From this table, it can be seen that the number of retweets and also the TF-IDF of Unigram play important roles in distinguish these three classes. We can also see that the content of the tweets is most important for classification. Based on the results of Table 4.10, the top interactions are from the two TF-IDF feature dimensions. This is also demonstrated by the experiment results (see Table 4.11).

Table 4.10: Illustration of top-3 main factors. The second one and the third one are from TF-IDF of Unigram.

| retweet_num | TF-IDF1 | TF-IDF2 |
|---|---|---|
| 4.41e-05 | 4.53e-01 | 3.62e-01 |

Table 4.11: This table shows top-3 interaction factors and their corresponding coefficients.

| TF-IDF1 * TF-IDF1 | TF-ID2 * TF-ID2 | TF-ID1 * TF-IDF2 |
|---|---|---|
| -2.258e-1 | 1.844e-1 | 7.884e-2 |

### Clustering Structure on the Web-Scale Data

In this part, we apply the learned feature mapping scheme to the large dataset, which contains not only the labeled data points but also unlabeled ones. To show the clustering structure of the partially labeled dataset, we perform two experiments: one using one baseline which is KMeans and the other one is our method based on Eqn. 4.23. For a good clustering outcome, we assume in each cluster, a majority of data points belong to the same class. To evaluate the performance of the results, we present two metrics (Eqn. 4.26) to show whether any class is dominant in a given cluster. In each cluster, there may be three classes with sizes $n_0$, $n_1$ and $n_2$ (in non-increasing order) respectively. If one class does not exist, it means its size is zero.

$$m_1 = \frac{n_2}{n_0} \qquad m_2 = \frac{n_1}{n_0} \tag{4.26}$$

In our experiment, the large dataset is partially labeled and thus when we compute $m_1$ and $m_2$, we only consider the labeled data in each cluster. Then the average is computed for the entire dataset. These two metrics are presented to measure what is the difference between the dominant class and the others. If the values of these metrics are small, then they shows that compared with the size of the dominant class, the others are small.

In our experiment, the number of clusters is chosen from a pre-defined set which is $k \in \{10, 100, 200, 300, 400, 500, 1000\}$. In this way, we can learn the effect of the number of clusters on the clustering performance. The experiment results are shown in Table 4.12.

From Table 4.12, we can see that our clustering approach by employing the learned feature mapping scheme performs better than the baseline. As the number of clusters goes up, $\bar{m}_1$ and $\bar{m}_2$ of KMeans and our approach become small, which means that the percentage of the dominant class becomes large. Compared with the baseline, the percentage of the dominant class is much larger since the corresponding metrics' values are smaller. The average percentage of the dominant class is shown in Fig.4.9.

### *4.2.7 Conclusion and Future Work*

In this chapter, we presented one semi-supervised approach to analysis of Twitter data related to marijuana use, using web-scale data. The entire approach has two steps: learning the optimal feature mapping scheme and grouping the entire data using an improved clustering algorithm. In the first step, we proposed a new linear classifier with weakly hierarchical lasso and solved it by relaxing the objective function to an easier form. In the

Table 4.12: Experiment results on studying the clustering structure of partially labeled dataset. (a) for the baseline and (b) for ours. They show the size of the other class compared with the dominant one.

(a) The baseline

| k | $\bar{m}_1$ | $\bar{m}_2$ |
|---|---|---|
| 10 | 0.411 | 0.647 |
| 100 | 0.320 | 0.594 |
| 200 | 0.333 | 0.594 |
| 300 | 0.318 | 0.602 |
| 400 | 0.291 | 0.542 |
| 500 | 0.282 | 0.542 |
| 1000 | 0.239 | 0.495 |

(b) our approach

| k | $\bar{m}_1$ | $\bar{m}_2$ |
|---|---|---|
| 10 | 0.381 | 0.555 |
| 100 | 0.280 | 0.487 |
| 200 | 0.240 | 0.436 |
| 300 | 0.263 | 0.485 |
| 400 | 0.243 | 0.423 |
| 500 | 0.228 | 0.427 |
| 1000 | 0.116 | 0.320 |

second step, we showed how to apply the learned feature mapping scheme to the clustering algorithm. Finally, we carried out experiments on large-scale data from Twitter. The experimental results demonstrated the effectiveness and efficiency of our approach.

There are still some limitations we need to work on. For example, when we learn the feature mapping scheme, we relax the problem to be one easier one, and thus the learned parameters are only locally optimal. Another problem is that the dataset could still be bigger, possibly covering a longer period than the one-week period used in our data collection. Furthermore, how to incorporate features reflecting temporal patterns of user behaviors is worth studying.

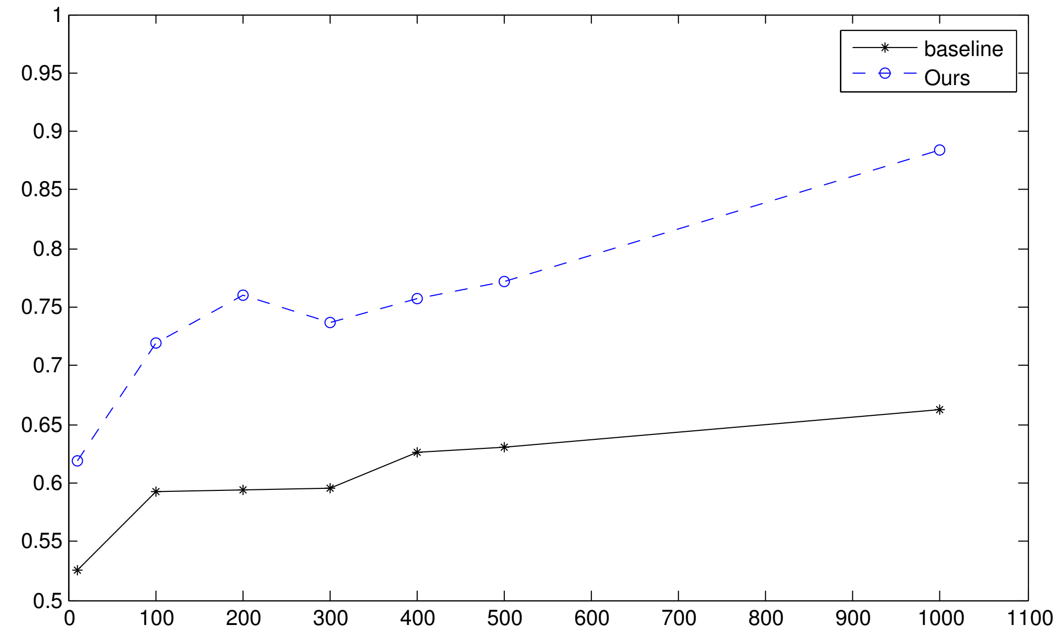


Figure 4.9: It shows the average percentages of the dominant class plotted based on the experiment result at each $k \in \{10, 100, 200, 300, 400, 500, 1000\}$

Chapter 5

# WEAKLY HIERARCHICAL LASSO BASED LEARNING TO RANK IN BEST ANSWER PREDICTION

As one form of user-generated content, posts on community question and answering (CQA) sites are often very noisy. One way of extracting useful knowledge from these CQA sites is to identify pairs of questions and their best answers. In reality, this is not a trivial task as many askers eventually do not mark the best answers even if some answers have perfectly solved their problems. To solve this problem, research on best answer prediction appeared and has been working on for a long time. User-generated answers often consist of multiple "views", each capturing different (albeit related) information (e.g., expertise of the asker, length of the answer, etc.). Such views interact with each other in complex manners that should carry a lot of information for distinguishing a potential best answer from others. Little existing work has exploited such interaction for better prediction. In this chapter, we propose a new learning-to-rank method, ranking support vector machine (RankSVM) with weakly hierarchical lasso, to explicitly model view interaction in best answer prediction. The key idea is to treat each feature dimension as one view of the task and then involve the second-order view interactions via constructing weakly hierarchical structure for predicting best answers. To find a solution under the proposed model, we apply an iterative shrinkage and thresholding algorithm for solving the non-convex problem. The evaluation of the approach was done using two datasets: MQ2007 and Stack Overflow. Experimental results demonstrate that the proposed approach has superior performance compared with current state-of-the-art methods.

## 5.1 Introduction

In the era of Internet and social media, community question and answering (CQA) sites, like Baidu Zhidao [1] , Yahoo! Answers [2] and StackOverflow [3] , are seeing phenomenal growth. As one form of user-generate content, data from CQA sites are typically very noisy, which does not lead to ready usage either by humans or by computers. Consequently, how to extract useful information from the noisy CQA data to form valuable knowledge base has become an important research task Anderson *et al.* (2012). One popular task on this regard is best answer prediction, on which our chapter focuses.

Given a question with multiple answers, one way to solve best answer prediction is to reformulate it into a binary classification problem which is whether, in a question-answer pair, the answer is the best one or not. There have been some research efforts in this setting like Agichtein *et al.* (2008), Shah and Pomerantz (2010). In these efforts, features were extracted from different views of the data to generate a good representation for the question-answer pairs, and the final feature vector was formed by concatenating them together. As a result, each feature dimension carries some information of the CQA data. But there are a couple of limitations inherent to these existing techniques. First, a binary classifier is not natural to this research problem, which often involves multiple answers for one given question. It is possible for a trained classifier to declare many or *even all* answers are the best ones (if they happen to lead to feature vectors lying on the positive side of the decision boundary). Also it is counter-intuitive as a human user would normally compare all received answers and decide on a single best one. The binary classification does not model directly on the difference of multiple answers, compared with learning-to-rank techniques. Second, the interaction between features from different views may carry a lot of

[1] `http://zhidao.baidu.com/`

[2] `https://answers.yahoo.com/`

[3] `http://stackoverflow.com/`

information for distinguishing a potential best answer from others, however current existing methods do not readily support incorporation of such interactions, which by itself is a challenging task.

In anther setting, best answer prediction is modeled as one ranking problem, which is conceptually more intuitive. This kind of modeling results from the fact that the best answer to one question is defined/discovered relatively by comparing it with all the other given answers. A ranking-based setting may benefit even more from considering the latent interactions between features designed from different views of the CQA data. Unfortunately, similar to the binary-classification cases, the existing learning-to-rank techniques have not attempted to explicitly to model such interactions among different views of the data Dalip *et al.* (2013)Cai and Chakravarthy (2013)Chapelle and Keerthi (2010).

In this chapter, we focus on how to incorporate the interaction structure of features into one existing algorithm framework to improve the performance of best answer prediction. Similar to Cai and Chakravarthy (2013)Hieber and Riezler (2011), we adopt the learning-to-rank formulation for its natural match to the prediction problem. Considering the interaction structure (or the hierarchical structure of feature dimensions in our study) and the ranking framework, we propose a new learning-to-rank formulation based on weakly hierarchical lasso.

The contributions of our work are summarized as follows: Firstly, we propose a new RankSVM model by constructing the weakly hierarchical structure between features from different views. Secondly, to solve the new formulation, we propose an efficient algorithm and evaluate via experiments its efficiency and effectiveness with comparisons with other existing methods.

## 5.2 Related Work

In this section, we review briefly related research on community question and answering, and discuss the difference between the reviewed work and our proposed method.

### *5.2.1 Content Quality Analysis*

Compared with traditional on-line search, as one supplementary approach to solving our daily problems, CQA sites contain a lot of valuable knowledge. Thus, since the first CQA site was launched, finding high quality content from these sites has become important. For example some early work was done in Jeon *et al.* (2006) where Jiwoon Jeon *et al.* crawled data from Naver Q&A site and manually labeled each pair of questions and their corresponding answers as *bad, medium, good.* They proposed to use non-textual features to represent each question-answer pair and used kernel density estimation and the maximum entropy approach to model the problem of answer quality. To have a better representation of questions and answers on CQA sites, more sources of information were used to extract new features like interactions between questions and answers and users, as studied in Agichtein *et al.* (2008), where Eugene Agichtein *et al.* proposed to use non-content information to model question and answer pairs on CQA sites including the interaction features. Then different classifiers like support vector machine, log-linear classifier and stochastic gradient boosted trees were applied to learn the prediction model, whose efficiency and effectiveness were evaluated using data from Yahoo! Answers. The importance of social information for predicting answer quality was studied in Shah and Pomerantz (2010), where Chirag Shah *et al.* found the importance of user information by studying the quality labeled manually. Besides research on the answer quality, question quality is also studied. In Li *et al.* (2012), Baichuan Li *et al.* worked on the question quality prediction problem. They first studied what factors may affect question quality and then proposed a model termed Mutual Reinforcement-based Label Propagation to predict question quality. In Yao *et al.* (2015), it was found that the voting scores of questions have a strong positive correlation with that of the corresponding answers and they proposed a set of co-prediction algorithms to predict the voting scores of questions and answers.

The above work focused on content quality prediction (question quality and answer quality), which is modeled as one classification problem. These existing efforts mainly focused on finding a better representation of the data by introducing various features to facilitate the prediction problem.

### *5.2.2 Best Answer Prediction and Answer Ranking*

Pairs of questions and their best answers can be easily used to answer similar questions, as the research in Shtok *et al.* (2012) shows. With the fast growth of CQA sites, there are a lot of questions which have high quality answers but no best ones eventually marked. To this end, a lot of research efforts have been devoted to best answer prediction and answer ranking. In Adamic *et al.* (2008), Lada Adamic *et al.* analyzed Yahoo! Answers for best answer prediction. They used simple four-dimensional features and reported that the length of answers is the most important factor of answer quality. The problem they are worked on is to predict whether a given answer is the best one of the given question. They did not consider interaction information like relationship between questions and answers and users. It is not natural to model best answer prediction as a classification problem since the best answer is relatively defined. Thus there have been a lot of efforts on modeling best answer prediction as a ranking problem. In Surdeanu *et al.* (2008), Mihar Surdeanu *et al.* proposed a ranking model for non-factoid questions and studied whether ranking algorithms can be used to rank answers for given questions. They also showed the importance of different features in the answer ranking problem. This work was further extended in Surdeanu *et al.* (2011). Instead of simply applying learning to rank algorithms, some researchers worked on improving the performance by using piggybacking and ranking aggregation techniques. In Hieber and Riezler (2011), Felix Hieber *et al.* applied RankSVM algorithms to best answer prediction with piggybacking being used to improve the performance. In their work, interaction features were used, like the similarity between questions and answers. Piggybacking is used to for obtaining a better representation of the questions so that similarity between the questions and answers can help improve the ranking performance of RankSVM. One example work to use ranking aggregation is Agarwal *et al.* (2012), where Arvind Agarwal *et al.* made a comparison between different learning to rank algorithms and proposed to use ranking aggregation techniques to improve them. But that work focused on the factoid question and answers instead of CQA. In contrast, our work employs hierarchical interactions in the feature space.

There are also some efforts on studying the influence of different combinations of features on the prediction accuracy and also comparison across different CQA sites Burel *et al.* (2012). Point-wise ranking techniques were also used to rank answers to each question. In Dalip *et al.* (2013), Daniel Dalip *et al.* assumed that the voting scores to be the quality scores of answers. Then random forest was used to model the relationship between the scores and features. The final predicted rating scores were used to rank each questions. To evaluate the performance, normalized discounted cumulative gain at top k (NDCG@K) is used. However, there is noise in the rating scores as shown in Ravi *et al.* (2014), and thus in our work we do not use this assumption. The information between answers to each question may help capture the *relative* information for better prediction, as shown in Tian *et al.* (2013), where Tian *et al.* proposed to extract features from the context information between answers to each question. There are many other efforts on finding/defining new features for best answer prediction. For example, temporal features are proposed in Cai and Chakravarthy (2013).

One common observation in the most of the existing work is that, when new features are derived, all of them are concatenated to one vector to be the final feature vector. For example, in Adamic *et al.* (2008), these features are used: reply length, thread length, the total number of best answers of one user, the total number of replies one user has. They can be denoted as $x_1, x_2, x_3, x_4$. Then the final feature vectors are the simple concatenation of these features which are $(x_1, x_2, x_3, x_4)$. In our work, we focus on proposing a new model which can capture the feature interactions based on hierarchical lasso.

## 5.3 Problem Description and Formulation

The research problem in this chapter is formally defined as follows: given a question with all of its received answers, to predict which one is the best one. To select the best answer, one has to compare it with the others, so that the best answer is relatively defined. Thus instead of using the classification framework, we employ the learning-to-rank strategy. The basis of our proposed approach is RankSVM Chapelle and Keerthi (2010). While existing work focuses on designing new features, we study this prediction problem from the following angle: modeling the interaction of features from different views of data beyond simple concatenation of them. To achieve this goal, we employ weakly hierarchical lasso Bien *et al.* (2013) in constructing a new ranking model.

Notations of this chapter are described in the following. Denote a dataset with $N$ questions as $\{q_i,\ i \in \{1, \cdots, N\}\}$. For each question $q_i$, it receives a group of answers which are $\{A_{i,j},\ j \in \{1, \cdots, M_i\}\}$ where $M_i$ is the total number of answers to $q_i$. The feature vector $x_{i,j} \in \mathbb{R}^{1\times d}$ is used to represent the $j^{th}$ answer to the $i^{th}$ question. Moreover, the $k^{th}$ dimension of one feature vector $x_{i,j}$ is defined as $x_{i,j,k}$ where $k \in \{1, \cdots, d\}$. $x_{i,j}$ is the simple concatenation of features extracted from different views of our problem, as done in the existing work. It is named as the *main effect*. Then for each $x_{i,j}$, we compute the second-order interaction which is denoted as $z_{i,j} \in \mathbb{R}^{1\times d^2}$, which is called the second-order *interaction term*. The final feature vector by considering the main effect and the interaction term is denoted as $\hat{x}(i,j) = [x_{i,j}, z_{i,j}] \in \mathbb{R}^{1\times(d+d^2)}$. The interaction term is defined as follows (see Eqn.5.1):

$$\begin{aligned} z_{i,j} &= [z_{i,j}^{(1)}, z_{i,j}^{(2)}, \cdots, z_{i,j}^{(d)}] \qquad (5.1)\\ z_{i,j}^{(m)} &= [x_{i,j,m}\cdot x_{i,j,1},\ x_{i,j,m}\cdot x_{i,j,2},\ \cdots, x_{i,j,m}\cdot x_{i,j,d}] \end{aligned}$$

where $i \in \{1,\cdots,N\}$, $j \in \{1,\cdots,M_i\}$ and $m \in \{1,\cdots,d\}$.

In our work, instead of classification methods, learning-to-rank techniques are used to model the *relativeness* of the best answers. Each relatively ranked pair is represented as $(q_i,\ A_{i,j_1},\ A_{i,j_2})$ where the quality of $A_{i,j_1}$ is higher than that of $A_{i,j_2}$. For simplicity, we may use $(i,j_1,j_2)$ as the short version of $(q_i,\ A_{i,j_1},\ A_{i,j_2})$ in the following equations. The set $P_i$ contains all these pairs of answers to the question $q_i$. Furthermore, the entire set of these relatively ranked pairs is denoted as $P$ in Eqn.5.2.

$$P = \bigcup_{i\in\{1,\cdots,N\}} P_i \qquad (5.2)$$

RankSVM, as one state-of-the-art pair-wise learning-to-rank algorithm used in best answer prediction Cai and Chakravarthy (2013)Hieber and Riezler (2011), is used as the basic building block of our new ranking model.

The RankSVM formulation is given below (Eqn. 5.3):

$$\begin{aligned} \min_{w\in\mathbb{R}^{d\times 1}} \quad & \frac{1}{2}\|w\|_2^2 + C\sum \xi_{i,j_1,j_2} \qquad (5.3)\\ \text{s.t.} \quad & S_1(i,j_1) \geq S_1(i,j_2) + 1 - \xi_{i,j_1,j_2}, \quad \forall(i,j_1,j_2)\\ & \xi_{i,j_1,j_2} \geq 0, \quad \forall(i,j_1,j_2) \end{aligned}$$

where $(i,j_1,j_2)$ is one ranked QA pair in $P$ and $S(i,j)$ is the quality score function of the $j^{th}$ answer to $q_i$ and defined in Eqn.5.4.

$$S_1(i,j) = x_{i,j}\ w + w_0 \qquad (5.4)$$

where $w_0 \in \mathbb{R}$.

To improve the performance of RankSVM, our model involves the second-order interactions via constructing one weakly hierarchical structure in the feature space. The formulation of the new ranking model is shown in Eqn.5.5. Compared with the existing work, we model the latent interaction structure between features from different views of the data, instead of simple concatenation. The hierarchical structure of the feature space is constructed through the first group of constraints (a.k.a $\|Q_{\cdot,j}\|_1 \leq |w_j|, j \in \{1, \cdots, d\}$) in Eqn.5.5.

$$\begin{aligned} \min_{\substack{w \in \mathbb{R}^{d\times 1}, \\ Q \in \mathbb{R}^{d \times d}}} \quad & \|w\|_1 + \frac{1}{2}\|Q\|_1 + C \sum_{(i,j_1,j_2)\in P} \xi_{i,j_1,j_2} \\ \text{s.t.} \quad & \|Q_{\cdot,j}\|_1 \leq |w_j|, \quad j \in \{1, \cdots, d\} \\ & \xi_{i,j_1,j_2} \geq 0, \quad \forall (i, j_1, j_2) \in P \\ & S(i, j_1) > S(i, j_2) + 1 - \xi_{i,j_1,j_2}, \quad \forall (i, j_1, j_2) \in P \end{aligned} \tag{5.5}$$

where $Q_{\cdot,j}$ is the $j^{th}$ column of $Q$, $\|Q\|_1 = \sum_i \sum_j |Q_{i,j}|$ and $S(\cdot,\cdot)$ is the ranking score for each answer to one question defined in Eqn.5.6. For example $S(i, j)$ is the ranking score for answer $A_{i,j}$ to $q_i$.

$$S(i, j) = x_{i,j}w + \frac{1}{2}z_{i,j}\ vec(Q) + w_0 \tag{5.6}$$

where $vec(Q)$ is the vectorized version of $Q$ and $z_{i,j}$ is shown in Eqn.5.1 and $w_0 \in \mathbb{R}$.

To help illustrating the proposed model, we depict the hierarchical structure based on one example shown in Figure 5.1, in which we only show three features: the length of the answer ($A_{len}$), the number of URLs in the answer ($N_{url}$), the number of pictures used in the answer ($N_{pic}$). In this illustration, we can see that the upper layer contains all main effects (a.k.a $x_{i,j}$ ) while the second layer shows the interaction terms (a.k.a $z_{i,j}$ in Eqn.5.1) excluding the square values of themselves. When one term contributes to the objective function, no matter it belongs to main effects or interaction terms, its corresponding coefficient is set to be non-zero. For each interaction term, if it contributes to the objective function, then at least one of its corresponding main effects contributes to the objective function. Satisfying these hierarchical constraints, it is easy for us to conclude that the interaction terms contribute less than their corresponding main effects. Specifically, in this figure, if the coefficient of $A_{len} \cdot N_{url}$ is non-zero, then the coefficient of $A_{len}$ is non-zero but that of $N_{url}$ can be zero.

From Eqn. 5.5, the weakly hierarchical lasso is involved via the first group of constraints (a.k.a $\|Q_{\cdot,j}\|_1 \leq |w_j|, j \in \{1, \cdots, d\}$).

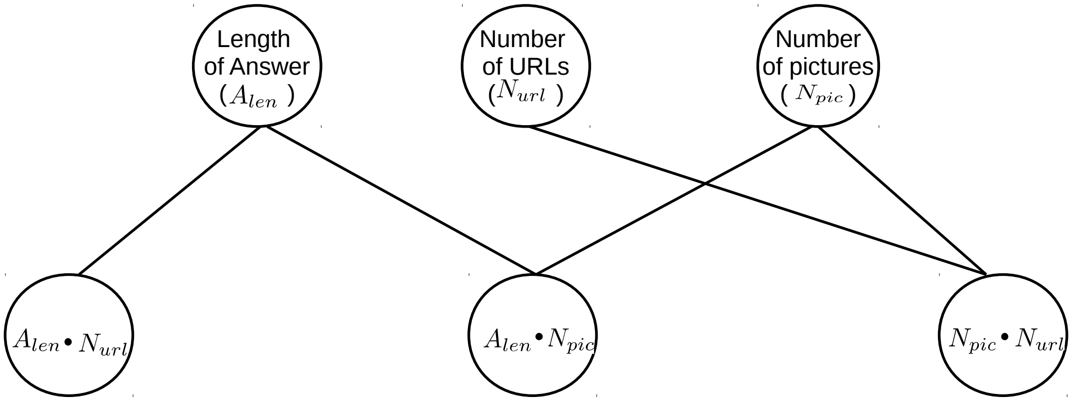

Length of Answer ($A_{len}$)
Number of URLs ($N_{url}$)
Number of pictures ($N_{pic}$)
$A_{len} \bullet N_{url}$
$A_{len} \bullet N_{pic}$
$N_{pic} \bullet N_{url}$

Figure 5.1: One illustration to show hierarchical structure in the feature space, where "·" represents the scalar multiplication. The first layer contains the main effect, while the second layer consists of the $2^{nd}$ order of interaction.

## 5.4 Solving the Proposed Model

To develop a solution to our proposed model in Eqn. 5.5, we first reformulate the problem as follows. Consider this group of constraints (Eqn.5.7) in the proposed model in Eqn. 5.5.

$$S_{i,j_1} > S_{i,j_2} + 1 - \xi_{i,j_1,j_2} \tag{5.7}$$

Together with Eqn.5.6, we have the following computation:

$$\begin{aligned}
&S_{i,j_1} > S_{i,j_2} + 1 - \xi_{i,j_1,j_2} \\
&S_{i,j_1} = x_{i,j_1} w + \frac{1}{2} z_{i,j_1}\ vec(Q) + w_0 \\
&S_{i,j_2} = x_{i,j_2} w + \frac{1}{2} z_{i,j_2}\ vec(Q) + w_0
\end{aligned} \tag{5.8}$$

If we assume the relatively ranked pair $(q_i, A_{i,j_1}, A_{i,j_2})$ is the $m^{th}$ element in the set $P$ of Eqn.5.2, then Eqn.5.8 can be simplified and the following is obtained:

$$\tilde{x}_m w + \frac{1}{2} \tilde{z}_m \cdot vec(Q) > 1 - \tilde{\xi}_m \tag{5.9}$$

where $\tilde{x}_m$, $\tilde{z}_m$ should satisfy the following constraints in Eqn.5.10.

$$\begin{aligned}
\tilde{x}_m &= x_{i,j_1} - x_{i,j_2} \\
\tilde{z}_m &= z_{i,j_1} - z_{i,j_2}
\end{aligned} \tag{5.10}$$

As a result, Eqn.5.5 is converted to the following:

$$\begin{aligned}
\min_{w,Q} \quad & \|w\|_1 + \frac{1}{2}\|Q\|_1 + C \sum_{m \in \{1,\cdots,|P|\}} \tilde{\xi}_m \\
\text{s.t.} \quad & \tilde{x}_m w + \frac{1}{2} \tilde{z}_m \cdot vec(Q) > 1 - \tilde{\xi}_m, \quad m \in \{1, \cdots, |P|\} \\
& \|Q_{\cdot,j}\|_1 \leq |w_j|, \quad j \in \{1, \cdots, d\} \\
& \tilde{\xi}_m \geq 0, \quad m \in \{1, \cdots, |P|\}
\end{aligned} \tag{5.11}$$

where $|P|$ is the size of the set $P$.

Now we can reformulate Eqn.5.11 into Eqn.5.12:

$$\begin{aligned}
\min_{w,Q} \quad & \|w\|_1 + \frac{1}{2}\|Q\|_1 + C \cdot L(w, Q) \\
\text{s.t.} \quad & \|Q_{\cdot,j}\|_1 \leq |w_j|, \ \ j \in \{1, \cdots, d\}
\end{aligned} \tag{5.12}$$

where $L(w, Q)$ is given in the following:

$$L(w, Q) = \sum_{m=1}^{|P|} \max(0, 1 - (\tilde{x}_m w + \frac{1}{2}\tilde{z}_m\ vec(Q)))^2 \tag{5.13}$$

Set $\lambda = \frac{1}{C}$, the final model is obtain as given in Eqn.5.14

$$\begin{aligned} \min_{w,Q} \quad & L(w, Q) + \lambda \cdot \|w\|_1 + \frac{\lambda}{2}\|Q\|_1 \\ \text{s.t.} \quad & \|Q_{\cdot,j}\|_1 \leq |w_j|, \quad j \in \{1, \cdots, d\} \end{aligned} \tag{5.14}$$

To this point, our objective function has been reformulated into the standard form as in the weakly hierarchical lasso problem defined in Bien *et al.* (2013) and Liu *et al.* (2014).

To solve Eqn. 5.14, the scheme in Liu *et al.* (2014) can be applied since it can directly solve the weakly hierarchical lasso without adding more penalty compared with approach in Bien *et al.* (2013). Since the optimization process in Liu *et al.* (2014) is based on a general iterative shrinkage and thresholding algorithm (GIST) in Gong *et al.* (2013), before we use the method in Liu *et al.* (2014), we need to prove that $L(w, Q)$ in Eqn. 5.14 is continuously differentiable with Lipschitz continuous gradient.

Before proceeding with the proof, we introduce following notations:

$$\begin{aligned} \hat{x} &= (\tilde{x}, \tilde{z}) \\ \hat{w} &= \begin{pmatrix} w \\ \frac{1}{2}vec(Q) \end{pmatrix} \end{aligned} \tag{5.15}$$

As a consequence, $\hat{x} \in \mathbb{R}^{1\times(d+d\cdot d)}$ and $\hat{w} \in \mathbb{R}^{(d+d\cdot d)\times 1}$. $L(w, Q)$ is converted from Eqn.5.13 as Eqn.5.16.

$$\hat{L}(\hat{w}) = \sum_{m\in\{1,\cdots,|P|\}} \max(0, 1 - \hat{x}_m \cdot \hat{w})^2 \tag{5.16}$$

To show $\hat{L}(\hat{w})$ is differentiable with Lipschitz continuous gradient, this requirement needs to be satisfied: there exists a positive constant $\beta$ such that

$$\|\frac{d\hat{L}}{d\hat{w}}(w_1) - \frac{d\hat{L}}{d\hat{w}}(w_2)\|_2 \leq \beta\|w_1 - w_2\|_2 \tag{5.17}$$

Let us first consider one additive component of $\hat{L}(\hat{w})$. The point-wise maximum function can be written as Eqn.5.18.

$$\begin{aligned} l(\hat{w}) &= \max(0, 1 - \hat{x}_m \cdot \hat{w})^2 \\ &= \begin{cases} 0 & \text{if } 1 - \hat{x}_m \cdot \hat{w} < 0 \\ (1 - \hat{x}_m \cdot \hat{w})^2 & \text{if } 1 - \hat{x}_m \cdot \hat{w} \geq 0 \end{cases} \end{aligned} \tag{5.18}$$

It is easy to see that when $w_1, w_2 \in \{w | 1 - \hat{x}_m \cdot w < 0\}$ and $w_1, w_2 \in \{w | 1 - \hat{x}_m \cdot w \geq 0\}$, Eqn.5.17 is satisfied. Now considering $w_1 \in \{w | 1 - \hat{x}_m \cdot w < 0\}, w_2 \in \{w | 1 - \hat{x}_m \cdot w > 0\}$, it is easy to see that the left part of Eqn.5.17 becomes $\|(1 - \hat{x} \cdot w_2)\hat{x}_m\|$. Moreover, define $\hat{w}^*$ as $1 - \hat{x}_m \cdot w^* = 0$ and this inequality is satisfied: $\|w_1 - w_2\| \geq \|w^* - w_2\|$. Now to obtain the constant $\beta$, the following induction is performed:

$$\begin{aligned} &\|(1 - \hat{x}_m \cdot w_2)\hat{x}_m\| \leq \beta \|w_1 - w_2\| \\ \Leftarrow &\frac{\|(1 - \hat{x}_m \cdot w_2)\hat{x}_m\|}{\|w_1 - w_2\|} \leq \beta \\ \Leftarrow &\frac{\|(1 - \hat{x}_m \cdot w_2)\| \|\hat{x}_m\|}{\|w^* - w_2\|} \leq \beta \\ \Leftarrow &\frac{\|(1 - \hat{x}_m \cdot w_2)\| \|\hat{x}_m\|^2}{\|w^* - w_2\| \|\hat{x}_m\|} \leq \beta \\ \Leftarrow &\frac{\|(1 - \hat{x}_m \cdot w_2)\| \|\hat{x}_m\|^2}{\|1 - \hat{x}_m \cdot w_2\|} \leq \beta \\ \Leftarrow &\beta \geq \|\hat{x}_m\|^2 \end{aligned} \tag{5.19}$$

Similarly, it is easy to obtain that $\beta \geq \|\hat{x}_m\|^2$ also satisfies the case where $w_2 \in \{w | 1 - \hat{x}_m \cdot w < 0\}, w_1 \in \{w | 1 - \hat{x}_m \cdot w > 0\}$. Thus, there exists a proper positive constant $\beta$ so that $l(\hat{w})$ meets the requirement Eqn. 5.17. In conclusion, $l(\hat{w})$ is continuously differentiable with Lipschitz continuous gradient. With this result, we will further introduce and prove the following lemma, together with which we will able to show the desired property for $L(w, Q)$ is satisfied.

**Lemma 5.4.1.** *For each function $f(w)_i, i \in \{1, \cdots, N\}$ which is continuously differentiable with Lipschitz continuous gradient, their summation $f(w) = \sum_{i=1}^{N} f_i(w)$ is continuously differentiable with Lipschitz continuous gradient.*

*Proof.*

$$\begin{aligned} & \|\frac{d}{dw}f(w_1) - \frac{d}{dw}f(w_2)\| \\ = \quad & \|\sum_{i=1}^{N}\frac{d}{dw}f_i(w_1) - \sum_{i=1}^{N}\frac{d}{dw}f_i(w_2)\| \\ = \quad & \|\sum_{i=1}^{N}(\frac{d}{dw}f_i(w_1) - \frac{d}{dw}f_i(w_2))\| \\ \leq \quad & \sum_{i=1}^{N}\|\frac{d}{dw}f_i(w_1) - \frac{d}{dw}f_i(w_2)\| \\ \leq \quad & \beta\|w_1 - w_2\| \end{aligned} \tag{5.20}$$

Denote that there exists positive constant $\beta_i$ such that $f_i(w)$ satisfies Eqn.5.17 where $i \in \{1, \cdots, N\}$. Thus Eqn.5.20 is valid when $\beta$ meets this requirement:

$$\beta = \max_i \beta_i \tag{5.21}$$

□

Since $\max(0, 1 - \hat{x}_m \cdot \hat{w})^2$ satisfies Eqn. 5.17 and $\hat{L}(\hat{w}) = \sum_{m \in \{1, \cdots, |P|\}} \max(0, 1 - \hat{x}_m \cdot \hat{w})^2$, according to Lemma 5.4.1, $\hat{L}(\hat{w})$ satisfies Eqn. 5.17, same as $L(w, Q)$ defined in Eqn. 5.13. Thus, $L(w, Q)$ is continuously differentiable with Lipschitz continuous gradient.

Now it is feasible to apply the algorithm in Liu *et al.* (2014) to solve Eqn.5.14 which is equivalent to solving this proximal operator problem of Eqn.5.22.

$$\begin{aligned} (w^{(k+1)}, Q^{(k+1)}) = \arg\min_{w,Q} \frac{1}{2}\|w - v^{(k)}\|_2^2 + \frac{1}{2}\|Q - U^{(k)}\|_2^2 \\ + \frac{1}{t^{(k)}}(\lambda\|w\|_1 + \frac{\lambda}{2}\|Q\|_1) \\ \text{s.t.} \quad \|Q_{\cdot,j}\|_1 \leq |w_j| \quad \forall j \in \{1, \cdots, d\} \end{aligned} \tag{5.22}$$

where $v^{(k)}, U^{(k)}$ are defined as follows:

$$v^{(k)} = w^{(k)} - \frac{1}{t^{(k)}} \cdot \bigtriangledown_w L(w^{(k)}, Q^{(k)}) \tag{5.23}$$

$$U^{(k)} = U^{(k)} - \frac{1}{t^{(k)}} \cdot \bigtriangledown_Q L(w^{(k)}, Q^{(k)}) \tag{5.24}$$

where $t^{(k)} > 0$ which is the step size.

Considering $w, Q$ are products of their signs and also absolute values, Eqn.5.22 can be re-written into Eqn.5.25.

$$\begin{aligned}(w^{(k+1)}, Q^{(k+1)}) = \arg\min_{w,Q} \frac{1}{2}\|w - v^{(k)}\|_2^2 + \frac{1}{2}\|Q - U^{(k)}\|_2^2 \\ + \frac{1}{t^{(k)}}(\lambda\|w\|_1 + \frac{\lambda}{2}\|Q\|_1) \\ \text{s.t.} \quad \tilde{Q_{\cdot,j}} \leq \tilde{w_j} \quad \forall j \end{aligned} \tag{5.25}$$

where $Q_{\cdot,j} = sign(Q_{\cdot,j})\ \tilde{Q_{\cdot,j}}$ and $w_j = sign(w_j)\ \tilde{w_j}$. The above equation can be solved in a closed form as proved in Liu *et al.* (2014). The pseudocode of our entire algorithm is shown in the following. which is summarized in Algorithm 3.

## 5.5 Experiments

In this section, we present experimental results based on MQ2007 and StackOverflow to show the performance of our proposed model and the comparison with existing state-of-the-arts.

### *5.5.1 MQ2007 Dataset*

Our proposed method is derived from RankSVM which is one ranking algorithm used in the state-of-the-art of the best answer prediction Surdeanu *et al.* (2008)Surdeanu *et al.* (2011)Hieber and Riezler (2011)Cai and Chakravarthy (2013). To show the importance of the weakly hierarchical lasso, we compare our proposed model and RankSVM using one benchmark dataset for learning to rank: MQ2007.

**Algorithm 3** The pseudo-code to solve our model

```
INPUT: data matrix $X$ and ranking information of all data
OUTPUT: model parameters $w$ and $Q$
BEGIN:
    compute the set $P$ based on Eqn.5.2.
    compute the data difference $\{\tilde{x}_m, m \in \{1, \cdots, |P|\}\}$ and $\{\tilde{z}_m, m \in \{1, \cdots, |P|\}\}$ as Eqn.5.10.
    provide initial values for $w$ and $Q$.
    choose one $t$ via BB Rule Barzilai and Borwein (1988).
while $w$, $Q$ satisfy the stop criteria do
    while $t^k$ does not satisfy the stop criteria do
        update $v^k$ according to Eqn.5.23.
        update $U^k$ according to Eqn.5.24.
        obtain new $w^{(k+1)}$ and $Q^{(k+1)}$ based on Eqn.5.25, which can be in the closed form as Liu et al. (2014).
        update the step size $t^{(k)} = \alpha * t^{(k)}$ where $\alpha$ is the constant update ratio.
    end while
    k = k + 1;
end while
```

This dataset is one part of LETOR4.0 released by Microsoft Research Qin and Liu (2013). It was constructed based on the Gov2 web page collection using one query dataset from TREC 2007 [4] . This data set uses five-fold cross-evaluation so that five folds are provided. We only use the training set to train models and testing set to test them. The statistics of these five folds are shows in two tables: Table 5.1 for all training sets and Table

[4] http://trec.nist.gov/data/million.query07.html

5.2 for all testing sets.

Table 5.1: The statistics of training sets in MQ2007.

| | fold 1 | fold 2 | fold 3 | fold 4 | fold 5 |
|---|---|---|---|---|---|
| number of queries | 1017 | 1017 | 1014 | 1014 | 1014 |
| average number of retrieved documents | 41.453 | 41.257 | 40.750 | 40.905 | 41.376 |

Table 5.2: The statistics of testing sets in MQ2007.

| | fold 1 | fold 2 | fold 3 | fold 4 | fold 5 |
|---|---|---|---|---|---|
| number of queries | 336 | 339 | 339 | 339 | 339 |
| average number of retrieved documents | 40.631 | 41.336 | 42.153 | 40.870 | 40.746 |

For each query in MQ2007, relevant documents are provided and labeled with relevant scores. Moreover features are extracted for each document. Thus in MQ2007, each data has such information: query ID, ranking order and the 46-dimensional feature vector which contains information like term frequency, inverse document frequency, Document length, BM25 Robertson *et al.* (1995) as described in Qin and Liu (2013). Before conducting this experiment on this dataset, we compute z-scores for each data dimension and re-construct training files and testing files based on z-scores. To use MQ2007, we exact relatively ranked pairs from each retrieved ranking lists and then apply pairwise-ranking algorithms to these pairs. Similarly to $P$ in Eqn. 5.2, all relatively ranked pairs together form one set $R$ shown in Eqn. 5.26:

$$R = \{(R_{i,1}, R_{i,2}), i \in \{1, \cdots, L\}\} \tag{5.26}$$

where $L$ is the total number of relatively ranked pairs in MQ2007, $(R_{i,1}, R_{i,2})$ is the $i^{th}$ one in $R$ and the retrieved document $R_{i,1}$ is more relevant to its query than $R_{i,2}$. To evaluate the performance, we use the following evaluation measure defined in Eqn. 5.27.

$$e_0 = \frac{\sum_{\forall(R_{i,1},R_{i,2})\in R} I(s_{i,1} > s_{i,2})}{L} \tag{5.27}$$

where $s_{i,1}, s_{i,2}$ are predicted ranking scores of $R_{i,1}$ and $R_{i,2}$ respectively, and $I(\cdots)$ is defined in the following: Essentially, this metric measurement is to compute how many ranking pairs are correctly predicted.

Experiment results based on the evaluation Eqn.5.27 are shown in Table. 5.3. Five-cross evaluation is performed and for each fold, it is one cross-evaluation. The average performance of both models is also listed. From Table. 5.3, it is easy to see that our proposed RankSVM with weakly hierarchical lasso stands out. On average, our proposed model is 18.67% better than that of RankSVM. In other words, this experiment also shows the second-order feature interactions from different views can play an important role in learning to rank on the application of the web document retrieval.

Table 5.3: The results of RankSVM and our proposed model on MQ2007 are shown.

| | Fold1 | Fold2 | Fold3 | Fold4 | Fold5 | mean |
|---|---|---|---|---|---|---|
| RankSVM | 0.551 | 0.469 | 0.526 | 0.513 | 0.473 | 0.505 |
| Ours | 0.699 | 0.682 | 0.704 | 0.686 | 0.688 | 0.692 |

### 5.5.2 Stack Overflow

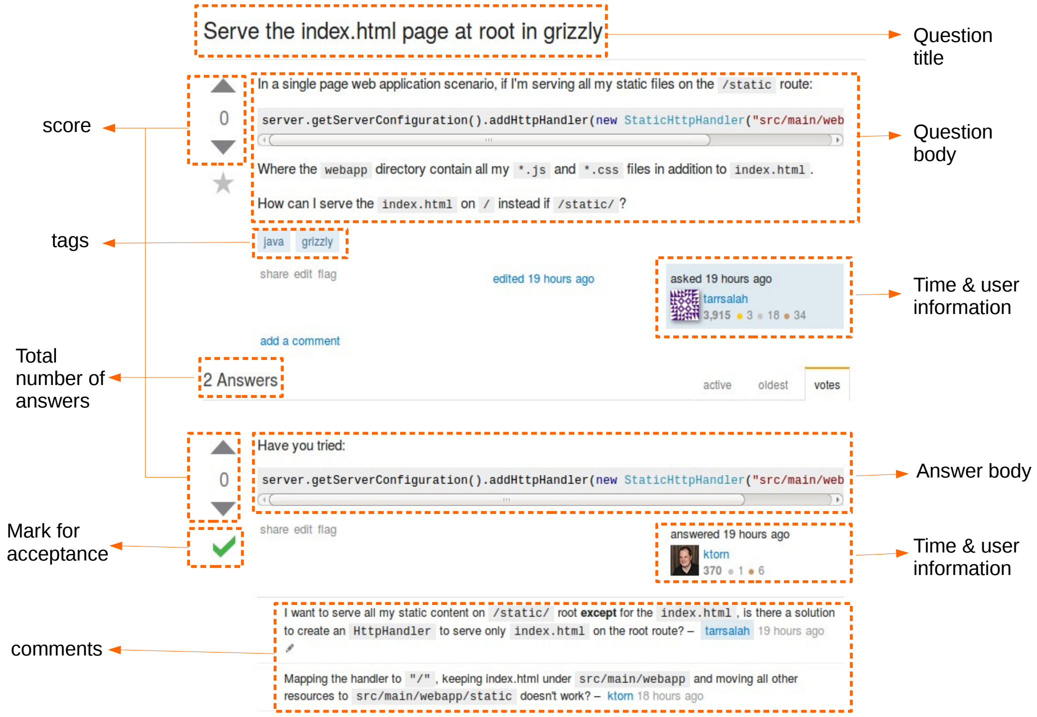
In the first experiment, we showed that the weakly hierarchical lasso really can improve the ranking ability of the framework of RankSVM. In this section, the performance of our model on the problem of best answer prediction is presented. The dataset we use is one active and popular CQA site on computer programming. All information about this site is available to download [5] . The description of StackOverflow is shown as follows.

### Data Description

Founded in 2008, StackOverflow is active and well maintained. On this site, users can post questions and everyone can provide answers even including the askers. For each question and each answer, users can comment on it. For one question or answer, users can vote up or down based on its quality except the user who posts it. For one comment, users can only vote up if they think the comment is useful, but cannot vote down. Same as one question or one answer, the one cannot vote up his or her own comments. For one question or one answer, it can receive up-votes and also down-votes. Then the number of up-votes minus the number of down-votes is the vote score. It is easy to see that the vote score are integers and can be negative.

Each question can receive multiple answers and only the asker can decide which one can be marked as the *accepted answer* which we call the *best answer*. This choice is not permanent, which means the asker can change his or her mind at any time and mark another answer as the *best answer*. There is one fact we need to point out. One question may receive multiple correct answers but only one of them can be marked as the *best answer*. So the *best answer* has the relatively best quality instead of absolutely best one. This is the reason why we use the learning to rank techniques instead of the classification methods. For users, they can earn reputations if their posts (e.g. questions ,answers, and comments) obtain upvotes or answers are accepted or suggestions on editing others' posts are accepted. Otherwise, they lose reputations if their posts receive downvotes or are reported as spam or offensive. Figure 5.2 shows one sample of one question with its answers from StackOverflow. Till May 8, 2015, the statistics of this site are as in Table. 5.4.

[5] `http://blog.stackoverflow.com/2009/06/stack-overflow-creative-commons-data-dump/`



Serve the index.html page at root in grizzly

In a single page web application scenario, if I'm serving all my static files on the `/static` route:

```
server.getServerConfiguration().addHttpHandler(new StaticHttpHandler("src/main/web
```

Where the `webapp` directory contain all my `*.js` and `*.css` files in addition to `index.html`.

How can I serve the `index.html` on `/` instead if `/static/` ?

asked 19 hours ago
tarrsalah
3,915 ● 3 ● 18 ● 34

Have you tried:

```
server.getServerConfiguration().addHttpHandler(new StaticHttpHandler("src/main/web
```

answered 19 hours ago
ktorn
370 ● 1 ● 6

I want to serve all my static content on `/static/` root **except** for the `index.html`, is there a solution to create an `HttpHandler` to serve only `index.html` on the root route? – tarrsalah 19 hours ago

Mapping the handler to `"/"`, keeping index.html under `src/main/webapp` and moving all other resources to `src/main/webapp/static` doesn't work? – ktorn 18 hours ago

Figure 5.2: Illustration of one sample question from Stack Overflow.

Table 5.4: The information of Stack Overflow till May 8, 2015.

| | |
|---|---|
| number of users | 4,232,639 |
| number of votes | 62,357,544 |
| number of comments | 44,557,809 |
| number of questions | 9,365,722 |
| number of answers | 15,632,696 |

### Experiment Settings

In our experiment, part of StackOverflow dataset is used. We downloaded all questions posted from October 1, 2012 to December 31, 2012 and all related information like answers was tracked until January 2014. This time period was chosen because of these reasons: First, questions and answers in this time period are not very out-dated; Second, few user activities on posts in this period are active. Thus, we assume that the best answer to one question is the final one. The dataset we use was dumped on January 2014 [6] . Before feature extraction, posts without users' IDs are removed. Then, only questions which have best answers and at least two more answers are considered. The final processed dataset has 52,104 questions and 190,165 answers. On average, there are 3.65 answers per question. During the experiments, our data set is randomly split into two parts evenly: training and testing.

To be specific, details as follows show how to generate relatively ranked pairs. For each question, only its best answer is considered as the high quality answer while others are treated as low-quality answers. Then each pair is generated in this way: one best answer and one of other answers to the same question. After all pairs are generated, feature extraction is performed based on information from three main aspects of each pair of questions and answers: content, interactions, users. These are briefly described below.

The First group of features are extracted based on the content of the answer in each pair of questions and answers. Part of these features are based on comments to the answers like *average score of comments, variance of the comments' scores, number of comments*. Comment-based features at least show that the corresponding answer is interesting and incur a good discussion towards problem solving. Besides these, whether one answer has pictures, URL or codes are also factors to show that the current answer has a high quality, since these components are able to show more information than text. Moreover, the length of answers Adamic *et al.* (2008)Agichtein *et al.* (2008) and its *readability* Tian *et al.* (2013) also play an important role on answer quality.

Apart from the content information, features based on *interaction* are also considered, for example, the interaction between questions and answers, and that between different answers to one question. The first one is easy to understand since one answer has to be

[6] http://blog.stackoverflow.com/category/cc-wiki-dump/

similar to its corresponding question, and thus the similarity between questions and answers is used as one feature. The second one is designed based on the assumption that users prefer the answers which is easy to understand. Computation of these features are shown in Tian *et al.* (2013). This is different from the feature interaction in our model. This one is on the feature-design level which focuses on exploring new information sources to design new features, while our case focuses on the model-design level.

User information also has an impact on the quality of answers. One answer is likely to have a high quality if the answerer is one expert. To represent the expertise of one user, these features are extracted based on users' previous activities, for example the number of answers one provides, how many questions one asks, the number of best answers he or she posts.

Our experiment is conducted by considering different groups of features and then results are presented respectively. In this way, it is easy to see the performance of different algorithms when we only consider informations from different aspects of our research problem (i.e. different groups of features). Finally, the experiment is conducted on the entire feature set we have. The three groups of features we consider in this experiment are: *content*, *interactions* and *user information*.

### Experiment Results & Discussion

To show the performance of our proposed algorithm, we compare our model with approaches used in state-of-the-art. As mentioned in Section Introduction, there are two main trends in best answer prediction: one is to use classification techniques and then decision values are used as quality scores while the other one is to use ranking approaches directly. For the former case, linear support Vector Machine (SVM) is common used because data in social media is in large scale so that nonlinear algorithms are not computational efficient. In our experiment, linear SVM is the first baseline we choose. For the latter case, RankSVM Chapelle and Keerthi (2010) is used which is one main ranking algorithm used in the area of best answer prediction Cai and Chakravarthy (2013). The code for RankSVM is from Microsoft Research [7] . On CQA sites, there are no direct information we can use as the metric to measure answer quality without manually labeling. For example, scores of each answer might be one proper metric. But this metric is not accurate. It is easy to see that it is easy for the answer which is posted early to have the high score. In fact, on Stack Overflow, there are a lot of answers having the higher scores than the corresponding best answers [8] . Thus in our experiments, we only treat the best answers as the high-quality ones and others as low-quality. As a result, in our experiment, it is the pairwise ranking problem so we do not compare with listwise ranking algorithms.

To make comparison between different models, two evaluation metrics are used: one is defined in Eqn. 5.28 and the other one is defined in Eqn. 5.29.

$$e_1 = \frac{\sum_{\forall(q_i, A_{i,j_1}, A_{i,j_2}) \in P} I(s_{i,j_1} > s_{i,j_2})}{|P|} \tag{5.28}$$

where $s_{i,j_1}, s_{i,j_2}$ are predicted scores of $A_{i,j_1}, A_{i,j_2}$ respectively. The relatively ranking set $P$ is defined in Eqn. 5.2 and the function $I(\cdot)$ is shown in Eqn. 5.30.

$$\begin{aligned} g(i) &= \arg\max_j \{s_{i,j}, j \in \{1, \cdots, M_i\}\} \\ e_2 &= \frac{\sum_i I(j_{i,0} == g(i))}{N} \end{aligned} \tag{5.29}$$

where $j_{i,0}$ is the index of the best answer of the $i^{th}$ question, $s_{i,j}$ is the predicted score of the $j^{th}$ answer of the $i^{th}$ question and the function $g(\cdot)$ returns the index of the best answer

[7] http://research.microsoft.com/en-us/um/beijing/projects/letor/baselines/ ranksvm-primal.html

[8] https://data.stackexchange.com/stackoverflow/query/380215/where-accepted-answer-does-not-have-the-highest-score

of one given question and the function $I(\cdot)$ is given by Eqn.5.30.

$$I(x) = \begin{cases} 1 & \text{if } x \text{ is true} \\ 0 & \text{otherwise} \end{cases} \tag{5.30}$$

From the definitions, it is easy to see this fact: $e_1$ shows how good one algorithm is when it considers the pairwise ranking regardless of whether one algorithm can find the best answer to one question or not, while $e_2$ shows the performance of each algorithm when applied to best answer prediction. In other words, $e_1$ measures what percentage of relatively ranked pairs are predicted correctly, which focuses on the answer-level comparison. However $e_2$ measures what percentage of questions have the correctly predicted best answers.

To show the performance of different models on the pairwise ranking in best answer prediction, experiments were conducted to collect the metric $e_1$. The experimental results are shown in Table. 5.5. Table. 5.5 presents the performance of algorithms used as learning to rank. From the results, we can see that our model performs best not only when only individual feature groups are considered but also when all features are considered. This shows that our model can be one good pairwise ranking algorithm in the area of community question and answering. From the results of SVM, we can see that when only $f_c$ is considered,

Table 5.5: This table shows the results of different algorithms on Stack Overflow when considering the measurement metric $e_1$. Three groups of features: $f_c$ content, $f_i$ interactions, $f_u$ user information.

| | $f_c$ | $f_i$ | $f_u$ | *all* |
|---|---|---|---|---|
| SVM | 0.671 | 0.541 | 0.480 | 0.544 |
| RankSVM | 0.411 | 0.534 | 0.543 | 0.476 |
| Ours | 0.689 | 0.552 | 0.570 | 0.693 |

the performance is best. However, when simple concatenation of all features from different views is applied, the final one gives worse performance instead of better one. Similarly, for RankSVM, its performance is best when only $f_u$ is considered. However after considering all features, the performance drops. For our approach, because we consider the interaction structure of features from different views, the final performance is best. This shows that there exists on latent interaction structure in the feature space. Incorporating weakly hierarchical lasso, we can capture this interaction structure. This shows the effectiveness of our proposed model.

To show comparison of performance on best answer prediction, experiments were run to collect metric $e_2$. Table. 5.6 presents the performance of different models. From the results, it is easy to see that our model performs best in the problem of best answer prediction not only when considering different groups of features independently but also when considering all features jointly. Similar to Table.5.5, the performance of SVM and RankSVM drop a lot when all features are considered by simple concatenation. For our model, it does not have this problem because of the fact that we incorporate the information from the latent interaction of features from different views.

Table 5.6: Experiment results ($e_2$) of different algorithms' performance. Three groups of features: $f_c$ content, $f_i$ interactions, $f_u$ user information.

| | $f_c$ | $f_i$ | $f_u$ | *all* |
|---|---|---|---|---|
| SVM | 0.479 | 0.331 | 0.294 | 0.349 |
| RankSVM | 0.223 | 0.321 | 0.361 | 0.286 |
| Ours | 0.494 | 0.334 | 0.377 | 0.498 |

Consequently, we conclude that the proposed models perform better than those in the state-of-the-art. Performance of experiments using both metrics shows the effectiveness of hierarchical interactions between different views in the problem of best answer prediction.

## 5.6 Conclusion & Future Work

We present a new learning-to-rank approach to best answer prediction on CQA sites. Incorporating the weakly hierarchical lasso, our proposed model is able to effectively exploit the interactions of features from different views of the data. To find a solution under this new model, we reformulate it into one existing optimization framework. Experiments on Stack overflow are used to evaluate the proposed approach, with comparison to other methods in state-of-the-art. The experimental results demonstrate the effectiveness and superior performance of our approach. Although our algorithm is designed originally for best answer prediction, it can be treated as one ranking algorithm and used in most ranking situations. Thus the application of our algorithm in different areas can be one piece of future work. Moreover, in our algorithm, one limitation is that we study the interaction structure of different feature dimensions, instead of different groups of feature dimensions. Another interesting future work is to extending our algorithm by considering the hierarchical structure of different groups of feature dimensions.

Chapter 6

# CONCLUSION AND FUTURE WORK

In this chapter, I summarize my major contributions of this dissertation work and also suggest directions for future research work to support visual question answering.

## 6.1 Major Contributions

Visual Question Answering, as an important and promising emerging field, is primarily established on research involving Computer Vision, Natural Language Processing and Reasoning. To support this research area, I work on dealing with two main related research problems: one is weakly supervised semantic segmentation, and the other one is best answer prediction. The contributions of this dissertation are summarized as follows.

**Best answer prediction in community based question answering sites** Contributions of this dissertation to best answer prediction in community based question answering sites involves two dimensions. Firstly, I design a new way to measure the answer quality based on the analysis of large-scale dataset. As one type of user-generated content, community based question answering sites contain a large number of questions without best answers. Pairs of questions and their best answers can be good re-usable resources to help other people solve similar problems. So this fact that a lot of best answers are missing results in a lot of waste. My findings can help have a better representation for the data collected for best answer prediction problem. It can be treated as a measurement to help generated community based knowledge. Furthermore, I also propose a new research method to predict which answer is the best one. This new ranking model not only can capture the nature of best answer prediction problem but also can be applied to other ranking problems. With the help of the weakly hierarchical lasso, the proposed method is able to model the hidden structure of the input data's feature space. To support my conclusion, I conduct experiments on large-scale datasets. One important dataset I use is from the StackOverFlow site, which is a question-answering site for programmers.

**Image Understanding via Graphical Model** Visual question answering cannot be solved in a good way without the help of image understanding. Thus, another research topic I concentrate on is image understanding. I have a thoroughly literature review and identify that it is meaningful to study the weakly semantic segmentation problem, in which I need to predict the label for each pixel of each image given images with only partially image-level labels. Solving this new weakly semantic segmentation problem, I can provide a large-scale images with pixel-level labels to help existing supervised learning problems in image recognition area like object detection, human detection, supervised semantic segmentation and *et al.* To solve this new problem, the overlapping information of image-level label sets from different images contributes a lot. This dissertation employs a popular framework of graphic model (conditional random field) to capture the underlying neighborhood information existing in the input image space. This new graphic model considers the neighborhood information inside one image and that between different image across the entire dataset. To support the theoretic findings, several experiments are performed on different common used datasets.

## 6.2 Preliminary Exploration into a Deep Learning Approach

One of my current project is about deep learning and weakly semantic segmentation. This is the extended work for Chapter 3. In this project, the deep learning technique is used to learn the feature representation for the super-pixels in images. The baseline method takes as inputs images and also their corresponding super-pixel maps (Kwak *et al.*, 2017).

In the training part, only image-level labels are considered. Using a large training set, the training network is able to learn the best feature representation for the superpixels. In the testing part, the trained features are sent to a fully connected layer to generate a labelmap for the testing images.

This baseline only considers the constraint of pixels in one superpixel, which assumes that all pixels in a superpixel have the same label. However, the neighbor information between different superpixels is not considered. To solve this problem, I add a conditional random field model as one post-processing step to help capture the neighbor information. In the training part, the model is as Fig. 6.1. Different training modules in the training networks are as follows. $f_{enc}$ is the encoder which is the VGG16 [1] networks. This encoder is pre-trained on the ImageNet dataset. $f_{ups}$ is the module which converts the feature map $z$ to become the same size as the input images. The upsampling layer is based on the research from (Zeiler and Fergus, 2014). Then the most important layer is the superpixel pooling layer, which considers all pixel-level features from upsampled feature map in the same superpixel, and then average these feature vectors. This layer is corresponding to the assumption that all pixels in one same superpixel should have the same labels.

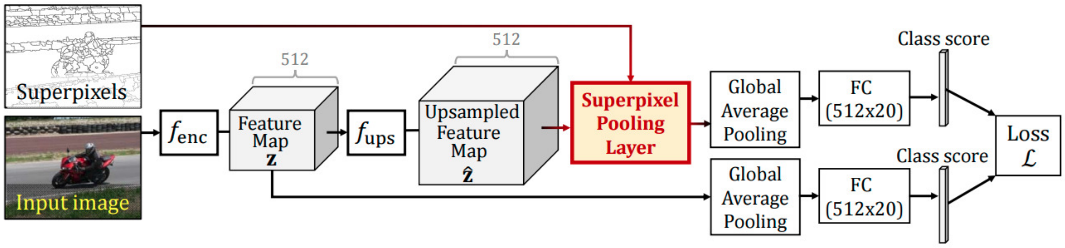


Figure 6.1: This is the framework for the training stage (this figure is from Kwak *et al.* (2017)).

After adding the CRF smoothing module, the testing part becomes as Fig. 6.2. In

[1] http://www.robots.ox.ac.uk/~vgg/research/very_deep/

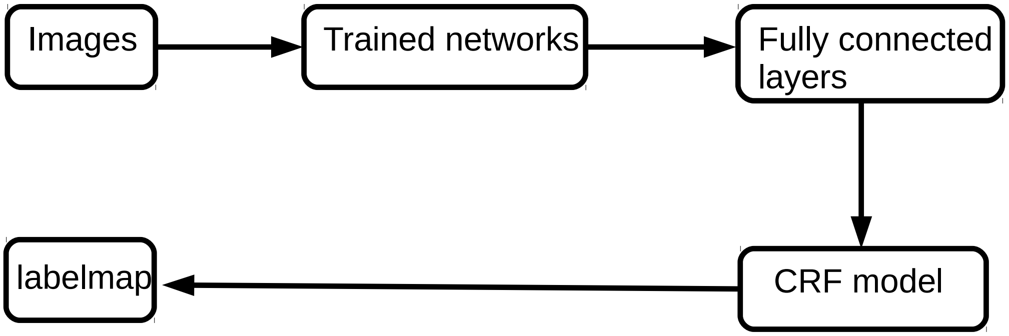


Figure 6.2: This is the framework for the new testing stage.

testing part, the trained network is the similar to the training network which consists of $f_{enc}$, $z$ map and $f_{ups}$, $\hat{z}$ and the superpixel pooling layer.

In the new model mentioned above, the CRF module in the testing part is treated as the post-processing, which needs to be trained separately. To automate the entire process, I include CRF into the network module. In this way, it can be trained together with the training process of the network part. New framework is shown in Fig.6.3. It has two main parts: network channel and CRF channel. In the former one, it has three modules from Kwak *et al.* (2017): $f_{enc}$, $f_{ups}$ and superpixel pooling layer, which are used to involve the superpixel information into model training process. The output of the superpixel pooling layer goes to the SegmentNet module (Pinheiro and Collobert (2015)) to generate response maps, which are merged with the output of CRF channel to obtain merged response maps. These maps go through aggregation layer to generate the loss which is used to do back-propagation, and meanwhile, are used to update the CRF channel's output. Segment-Net module and the aggregation layer are from Pinheiro and Collobert (2015). The first one is one 4-layer network which needs to be learned from training process and the second one is to map the response maps to be image-level labels, which is *Log-Sum-Exp* layer from Boyd and Vandenberghe (2004). There are several technical challenges existing in this new framework. First, feature maps after $f_{enc}$ have different sizes from those of input images,

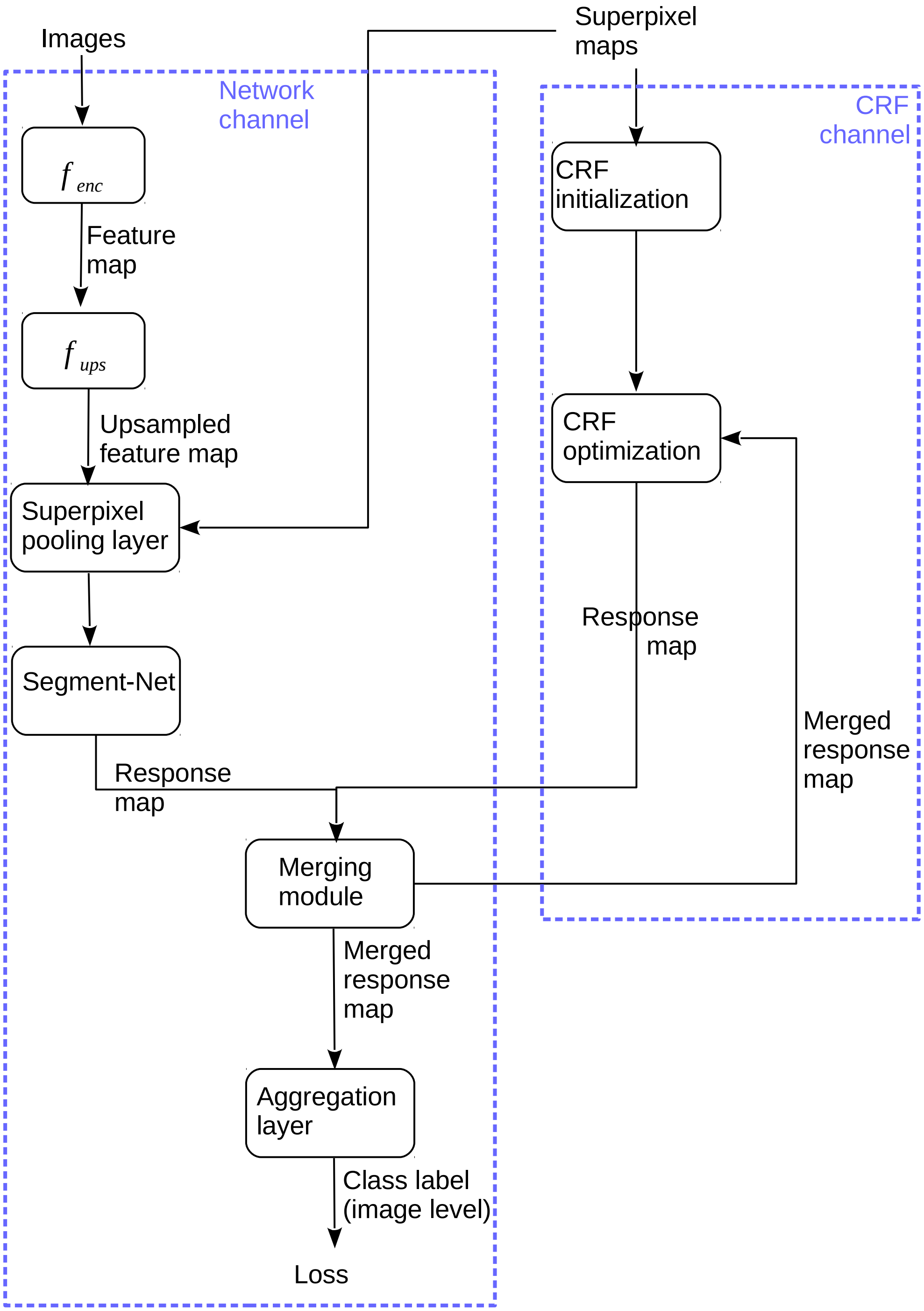


Figure 6.3: New framework with CRF feedback loop.

as a result it is unable to involve superpixel information and also hard to merge outputs of network channel and the CRF one. To solve this, $f_{ups}$ module is used. Second, images in the dataset do not have pixel-level labels but only have partial image-level ones. Because of this fact, it is difficult to start CRF module. To deal with this difficulty, I first label a small dataset which contains images with all pixel-level labels. For each label, it only has about two images so that in total there are around two hundred ones. Using this dataset, I generate the initial labelmaps for CRF module. Third, the output of CRF channel is discrete, which makes the final loss be not differentiable.

To test the new algorithm, a simulation experiment is designed. For each image in the simulated dataset, it is generated in the following process.

1, Randomly generate the size of image-level labels.

2, Randomly generate image-level labels with the pre-defined size from Step 1.

3, The dimension of the labelmap is 224×224 and each superpixel's size is 14×14.

4, For each pixel, its observation is generated according to Gaussian distribution with its label related mean value.

5, Finally, the superpixel map is generated, whose index starts from 0 instead of 1. For all pixels in the same superpixel, their indexes are the same.

For all images, their labels are from this set $\{0, 1, 2, 3, 4, 5\}$ and the corresponding mean values for the Gaussian distribution is from $\{5, 10, 15, 20, 25, 30\}$. One demo for this dataset is shown in Fig.6.4. Training set has 1500 images and testing set has 300 images. In our real image case, training images only have partial image-level labels so during the simulation, I randomly drop several image-level labels. The preliminary exploration for the deep learning based approach shows that the network can be learned from the simulated data.

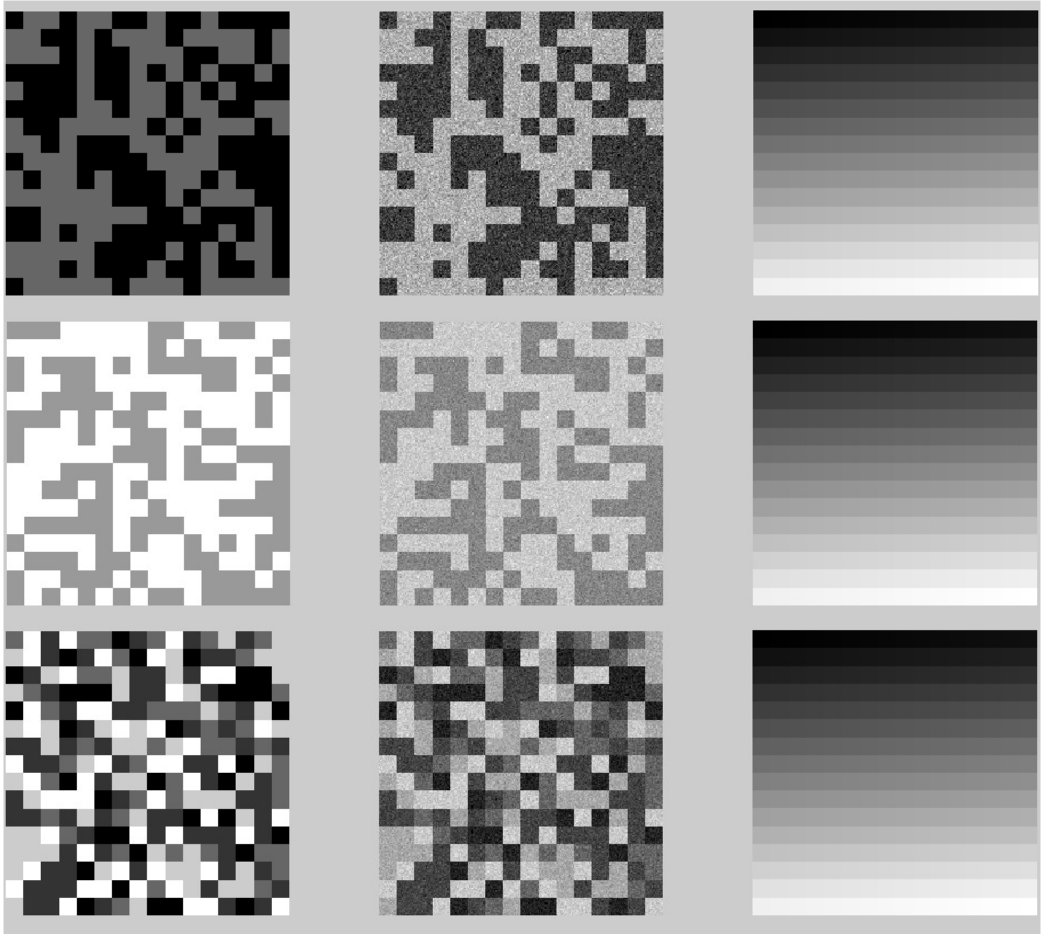


Figure 6.4: Three images from the simulated dataset. Each row is information for one image. Left column is the labelmap, middle one is the observation and right one is the superpixel map.

## 6.3 Future Directions

Research on visual question answering is at a very early stage, and is important to be explored in future. It is a multi-discipline area which involves at least two main research: image understanding and question answering in natural language processing. In this part, I point out several promising research topics for future research.

First, in the area of image understanding, one might combine the existing semantic segmentation results with natural language techniques, so that one intelligent audio description system can be constructed. Visual-impaired people can use this system to “see” colorful world. Moreover, one can apply the semantic segmentation to videos. Then combined with the natural language techniques, a wearable device with camera can be constructed to the blind.

Second, for the question answering research, my work focuses on solving how to determine the best answer where there is one question and multiple received answers. Then my approach can obtain which answer is the best one. However in the real life, there can be a large number of other situations, for example, users may want to search for subjective questions immediately, instead of waiting for a long time period for answer choice. In these cases, research needs to focus on generating best answers for one given question.

# REFERENCES

AARTS/KORST., *Simulated annealing and boltzmann machines. A stochastic approach to combinatorial optimization and neural computing* (John Wiley., 1990).

Abuse, S. and M. H. S. Administration, “Results from the 2013 national survey on drug use and health: Summary of national findings”, NSDUH Series H-48 **14**, 4863 (2014).

Achanta, R., A. Shaji, K. Smith, A. Lucchi, P. Fua and S. Susstrunk, “Slic superpixels compared to state-of-the-art superpixel methods”, Pattern Analysis and Machine Intelligence, IEEE Transactions on **34**, 11, 2274–2282 (2012).

Adamic, L., J. Zhang, E. Bakshy and M. Ackerman, “Knowledge sharing and yahoo answers: everyone knows something”, in “Proceedings of the 17th international conference on World Wide Web”, pp. 665–674 (ACM, 2008).

Agarwal, A., H. Raghavan, K. Subbian, P. Melville, R. D. Lawrence, D. C. Gondek and J. Fan, “Learning to rank for robust question answering”, in “Proceedings of the 21st ACM international conference on Information and knowledge management”, pp. 833–842 (ACM, 2012).

Agichtein, E., C. Castillo, D. Donato, A. Gionis and G. Mishne, “Finding high-quality content in social media”, in “Proceedings of the international conference on Web search and web data mining”, pp. 183–194 (ACM, 2008).

Anderson, A., D. Huttenlocher, J. Kleinberg and J. Leskovec, “Discovering value from community activity on focused question answering sites: a case study of stack overflow”, in “Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining”, pp. 850–858 (ACM, 2012).

Antol, S., A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. Lawrence Zitnick and D. Parikh, “Vqa: Visual question answering”, in “Proceedings of the IEEE International Conference on Computer Vision”, pp. 2425–2433 (2015).

Arbeláez, P., B. Hariharan, C. Gu, S. Gupta, L. Bourdev and J. Malik, “Semantic segmentation using regions and parts”, in “Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on”, pp. 3378–3385 (IEEE, 2012).

Bachman, J. G., J. M. Wallace Jr, P. M. O’Malley, L. D. Johnston, C. L. Kurth and H. W. Neighbors, “Racial/ethnic differences in smoking, drinking, and illicit drug use among american high school seniors, 1976-89.”, American Journal of Public Health **81**, 3, 372–377 (1991).

Baker, C. M., L. R. Milne, R. Drapeau, J. Scofield, C. L. Bennett and R. E. Ladner, “Tactile graphics with a voice”, ACM Transactions on Accessible Computing (TACCESS) **8**, 1, 3 (2016).

Barzilai, J. and J. M. Borwein, “Two-point step size gradient methods”, IMA Journal of Numerical Analysis **8**, 1, 141–148 (1988).

Beck, A. and M. Teboulle, “A fast iterative shrinkage-thresholding algorithm for linear inverse problems”, SIAM journal on imaging sciences **2**, 1, 183–202 (2009).

Besag, J., “Statistical analysis of non-lattice data”, The statistician pp. 179–195 (1975).

Bien, J., J. Taylor, R. Tibshirani *et al.*, “A lasso for hierarchical interactions”, The Annals of Statistics **41**, 3, 1111–1141 (2013).

Bigham, J. P., C. Jayant, H. Ji, G. Little, A. Miller, R. C. Miller, R. Miller, A. Tatarowicz, B. White, S. White *et al.*, “Vizwiz: nearly real-time answers to visual questions”, in “Proceedings of the 23nd annual ACM symposium on User interface software and technology”, pp. 333–342 (ACM, 2010).

Blooma, M. J., A.-K. Chua and D.-L. Goh, “Selection of the best answer in cqa services”, in “Information Technology: New Generations (ITNG), 2010 Seventh International Conference on”, pp. 534–539 (IEEE, 2010).

Boyd, S. and L. Vandenberghe, “Convex optimization”, in “Cambridge University Press”, (2004).

Breiman, L., “Random forests”, Machine learning **45**, 1, 5–32 (2001).

Burel, G., Y. He and H. Alani, “Automatic identification of best answers in online enquiry communities”, in “The Semantic Web: Research and Applications”, pp. 514–529 (Springer, 2012).

Cai, Y. and S. Chakravarthy, “Answer quality prediction in Q/A social networks by leveraging temporal features”, Proceedings of International Journal of Next-Generation Computing **4**, 1 (2013).

Cavazos-Rehg, P., M. Krauss, R. Grucza and L. Bierut, “Characterizing the followers and tweets of a marijuana-focused twitter handle”, Journal of medical Internet research **16**, 6 (2014).

Center, U. D. o. J. N. D. I., “The economic impact of illicit drug use on american society”, Product No. 2011-Q0317-002 (2011).

Chang, F.-J., Y.-Y. Lin and K.-J. Hsu, “Multiple structured-instance learning for semantic segmentation with uncertain training data”, in “Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on”, pp. 360–367 (IEEE, 2014).

Chapelle, O. and S. S. Keerthi, “Efficient algorithms for ranking with svms”, Information Retrieval **13**, 3, 201–215 (2010).

Chen, M., A. Zheng and K. Weinberger, “Fast image tagging”, in “Proceedings of the 30th International Conference on Machine Learning”, pp. 1274–1282 (2013).

Dalip, D. H., M. A. Gonçalves, M. Cristo and P. Calado, “Exploiting user feedback to learn to rank answers in q&a forums: a case study with stack overflow”, in “Proceedings of the 36th international ACM SIGIR conference on research and development in information retrieval”, pp. 543–552 (ACM, 2013).

Delong, A., A. Osokin, H. N. Isack and Y. Boykov, “Fast approximate energy minimization with label costs”, International journal of computer vision **96**, 1, 1–27 (2012).

Deluca, P., Z. Davey, O. Corazza, L. Di Furia, M. Farre, L. H. Flesland, M. Mannonen, A. Majava, T. Peltoniemi, M. Pasinetti *et al.*, “Identifying emerging trends in recreational drug use; outcomes from the psychonaut web mapping project”, Progress in Neuro-Psychopharmacology and Biological Psychiatry **39**, 2, 221–226 (2012).

Fichman, P., “A comparative assessment of answer quality on four question answering sites”, Journal of Information Science **37**, 5, 476–486 (2011).

Fusco, G. and V. S. Morash, “The tactile graphics helper: Providing audio clarification for tactile graphics using machine vision”, in “Proceedings of the 17th International ACM SIGACCESS Conference on Computers & Accessibility”, pp. 97–106 (ACM, 2015).

Geman, S. and D. Geman, “Stochastic relaxation, gibbs distributions, and the bayesian restoration of images”, IEEE Transactions on pattern analysis and machine intelligence **PAMI-6**, 6, 721–741 (1984).

Gong, P., C. Zhang, Z. Lu, J. Z. Huang and J. Ye, “A general iterative shrinkage and thresholding algorithm for non-convex regularized optimization problems”, in “Machine learning: proceedings of the International Conference. International Conference on Machine Learning”, vol. 28, p. 37 (NIH Public Access, 2013).

Gu, Q., Z. Li and J. Han, “Linear discriminant dimensionality reduction”, in “Machine Learning and Knowledge Discovery in Databases”, pp. 549–564 (Springer, 2011).

Hanson, C. L., S. H. Burton, C. Giraud-Carrier, J. H. West, M. D. Barnes and B. Hansen, “Tweaking and tweeting: exploring twitter for nonmedical use of a psychostimulant drug (adderall) among college students”, Journal of medical Internet research **15**, 4 (2013).

Harper, F. M., D. Raban, S. Rafaeli and J. A. Konstan, “Predictors of answer quality in online q&a sites”, in “Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems”, pp. 865–874 (ACM, 2008).

He, X., X. Li, G. Yang, J. Xu and Q. Jin, “Adaptive tag selection for image annotation”, in “Advances in Multimedia Information Processing–PCM 2014”, pp. 11–21 (Springer, 2014).

He, X., R. S. Zemel and M. Á. Carreira-Perpiñán, “Multiscale conditional random fields for image labeling”, in “Computer vision and pattern recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE computer society conference on”, vol. 2, pp. II–II (IEEE, 2004).

Hieber, F. and S. Riezler, “Improved answer ranking in social question-answering portals”, in “Proceedings of the 3rd international workshop on Search and mining user-generated contents”, pp. 19–26 (ACM, 2011).

Hosmer Jr, D. W., S. Lemeshow and R. X. Sturdivant, *Applied logistic regression*, vol. 398 (John Wiley & Sons, 2013).

Jackson, N. J., J. D. Isen, R. Khoddam, D. Irons, C. Tuvblad, W. G. Iacono, M. McGue, A. Raine and L. A. Baker, “Impact of adolescent marijuana use on intelligence: Results from two longitudinal twin studies”, Proceedings of the National Academy of Sciences p. 201516648 (2016).

Jeon, G. Y., Y.-M. Kim and Y. Chen, “Re-examining price as a predictor of answer quality in an online q&a site”, in “Proceedings of the 28th international conference on Human factors in computing systems”, pp. 325–328 (ACM, 2010).

Jeon, J., W. B. Croft, J. H. Lee and S. Park, “A framework to predict the quality of answers with non-textual features”, in “Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval”, pp. 228–235 (ACM, 2006).

Joachims, T., “Optimizing search engines using clickthrough data”, in “Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining”, pp. 133–142 (ACM, 2002).

Johnston, L. D., *Monitoring the Future: National Survey Results on Drug Use, 1975-2008: Volume II: College Students and Adults Ages 19-50* (DIANe Publishing, 2010).

Katsuki, T., T. K. Mackey and R. Cuomo, “Establishing a link between prescription drug abuse and illicit online pharmacies: Analysis of twitter data”, Journal of medical Internet research **17**, 12 (2015).

Kittler, J. and J. Föglein, “Contextual classification of multispectral pixel data”, Image and Vision Computing **2**, 1, 13–29 (1984).

Kohli, P., P. H. Torr *et al.*, “Robust higher order potentials for enforcing label consistency”, International Journal of Computer Vision **82**, 3, 302–324 (2009).

Krauss, M. J., S. J. Sowles, M. Moreno, K. Zewdie, R. A. Grucza, L. J. Bierut and P. A. Cavazos-Rehg, “Peer reviewed: Hookah-related twitter chatter: A content analysis”, Preventing chronic disease **12** (2015).

Kwak, S., S. Hong and B. Han, “Weakly supervised semantic segmentation using superpixel pooling network.”, in “AAAI”, pp. 4111–4117 (2017).

Lacson, J. C. A., J. D. Carroll, E. Tuazon, E. J. Castelao, L. Bernstein and V. K. Cortessis, “Population-based case-control study of recreational drug use and testis cancer risk confirms an association between marijuana use and nonseminoma risk”, Cancer **118**, 21, 5374–5383 (2012).

Lafferty, J., A. McCallum, F. Pereira *et al.*, “Conditional random fields: Probabilistic models for segmenting and labeling sequence data”, in “Proceedings of the eighteenth international conference on machine learning, ICML”, vol. 1, pp. 282–289 (2001).

Lapin, M., B. Schiele and M. Hein, “Scalable multitask representation learning for scene classification”, in “Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on”, pp. 1434–1441 (IEEE, 2014).

Lee, C., “Recruitment through social networking sites: Are substance use patterns comparable to traditional recruitment methods?”, in “Medicine 2.0 Conference”, (JMIR Publications Inc., Toronto, Canada, 2014).

Lenhart, A., K. Purcell, A. Smith and K. Zickuhr, “Social media & mobile internet use among teens and young adults. millennials.”, Pew Internet & American Life Project (2010).

Li, B., T. Jin, M. R. Lyu, I. King and B. Mak, “Analyzing and predicting question quality in community question answering services”, in “Proceedings of the 21st international conference companion on World Wide Web”, pp. 775–782 (ACM, 2012).

Liaw, A. and M. Wiener, “Classification and regression by randomforest”, R news **2**, 3, 18–22 (2002).

Liu, C., J. Yuen and A. Torralba, “Nonparametric scene parsing: Label transfer via dense scene alignment”, in “Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on”, pp. 1972–1979 (IEEE, 2009).

Liu, Y., J. Bian and E. Agichtein, “Predicting information seeker satisfaction in community question answering”, in “Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval”, pp. 483–490 (ACM, 2008).

Liu, Y., J. Liu, Z. Li, J. Tang and H. Lu, “Weakly-supervised dual clustering for image semantic segmentation”, in “Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on”, pp. 2075–2082 (IEEE, 2013).

Liu, Y., J. Wang and J. Ye, “An efficient algorithm for weak hierarchical lasso”, in “Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining”, pp. 283–292 (ACM, 2014).

Murphy, K. P., Y. Weiss and M. I. Jordan, “Loopy belief propagation for approximate inference: An empirical study”, in “Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence”, pp. 467–475 (Morgan Kaufmann Publishers Inc., 1999).

Myeong, H. and K. M. Lee, “Tensor-based high-order semantic relation transfer for semantic scene segmentation”, in “Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on”, pp. 3073–3080 (IEEE, 2013).

Myslín, M., S.-H. Zhu, W. Chapman and M. Conway, “Using twitter to examine smoking behavior and perceptions of emerging tobacco products”, Journal of medical Internet research **15**, 8 (2013).

Noble, W. S., “What is a support vector machine?”, Nature biotechnology **24**, 12, 1565–1567 (2006).

Pinheiro, P. O. and R. Collobert, “From image-level to pixel-level labeling with convolutional networks”, in “Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition”, pp. 1713–1721 (2015).

Plath, N., M. Toussaint and S. Nakajima, “Multi-class image segmentation using conditional random fields and global classification”, in “Proceedings of the 26th Annual International Conference on Machine Learning”, pp. 817–824 (ACM, 2009).

Platt, J. *et al.*, “Sequential minimal optimization: A fast algorithm for training support vector machines”, (1998).

Press, S. J. and S. Wilson, “Choosing between logistic regression and discriminant analysis”, Journal of the American Statistical Association **73**, 364, 699–705 (1978).

Qin, T. and T.-Y. Liu, “Introducing letor 4.0 datasets”, arXiv preprint arXiv:1306.2597 (2013).

Quattoni, A., M. Collins and T. Darrell, “Conditional random fields for object recognition”, in “Advances in neural information processing systems”, pp. 1097–1104 (2005).

Quintelier, K. J., K. Ishii, J. Weeden, R. Kurzban and J. Braeckman, “Individual differences in reproductive strategy are related to views about recreational drug use in belgium, the netherlands, and japan”, Human Nature **24**, 2, 196–217 (2013).

Ramo, D. E. and J. J. Prochaska, “Broad reach and targeted recruitment using facebook for an online survey of young adult substance use”, Journal of Medical Internet Research **14**, 1, e28 (2012).

Ravi, S., B. Pang, V. Rastogi and R. Kumar, “Great question! question quality in community q&a”, in “Eighth International AAAI Conference on Weblogs and Social Media”, (2014).

Ren, X. and J. Malik, “Learning a classification model for segmentation”, in “Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on”, pp. 10–17 (IEEE, 2003).

Robertson, S. E., S. Walker, S. Jones, M. M. Hancock-Beaulieu, M. Gatford *et al.*, “Okapi at trec-3”, NIST SPECIAL PUBLICATION SP pp. 109–109 (1995).

Roig, G., X. Boix, R. D. Nijs, S. Ramos, K. Kuhnlenz and L. V. Gool, “Active map inference in crfs for efficient semantic segmentation”, in “Computer Vision (ICCV), 2013 IEEE International Conference on”, pp. 2312–2319 (IEEE, 2013).

Russell, B. C., A. Torralba, K. P. Murphy and W. T. Freeman, “Labelme: a database and web-based tool for image annotation”, International journal of computer vision **77**, 1-3, 157–173 (2008).

Saito, P., P. J. de Rezende, A. X. Falcão, C. T. Suzuki and J. F. Gomes, “A data reduction and organization approach for efficient image annotation”, in “Proceedings of the 28th Annual ACM Symposium on Applied Computing”, pp. 53–57 (ACM, 2013).

Schuster, R. M., R. Mermelstein and L. Wakschlag, “Gender-specific relationships between depressive symptoms, marijuana use, parental communication and risky sexual behavior in adolescence”, Journal of youth and adolescence **42**, 8, 1194–1209 (2013).

Shah, C. and J. Pomerantz, "Evaluating and predicting answer quality in community qa", in "Proceeding of the 33rd international ACM SIGIR conference on Research and development in information retrieval", pp. 411–418 (Citeseer, 2010).

Sharma, A., O. Tuzel and D. W. Jacobs, "Deep hierarchical parsing for semantic segmentation", arXiv preprint arXiv:1503.02725 (2015).

Shotton, J., J. Winn, C. Rother and A. Criminisi, "Textonboost for image understanding: Multi-class object recognition and segmentation by jointly modeling texture, layout, and context", International Journal of Computer Vision **81**, 1 (2009).

Shtok, A., G. Dror, Y. Maarek and I. Szpektor, "Learning from the past: answering new questions with past answers", in "Proceedings of the 21st international conference on World Wide Web", pp. 759–768 (ACM, 2012).

Shu, G., A. Dehghan and M. Shah, "Improving an object detector and extracting regions using superpixels", in "Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition", pp. 3721–3727 (2013).

Stoddard, S. A., J. A. Bauermeister, D. Gordon-Messer, M. Johns and M. A. Zimmerman, "Permissive norms and young adults alcohol and marijuana use: The role of online communities", Journal of Studies on Alcohol and Drugs **73**, 6, 968–975 (2012).

Surdeanu, M., M. Ciaramita and H. Zaragoza, "Learning to rank answers on large online qa collections.", in "ACL", pp. 719–727 (2008).

Surdeanu, M., M. Ciaramita and H. Zaragoza, "Learning to rank answers to non-factoid questions from web collections", Computational Linguistics **37**, 2, 351–383 (2011).

Suykens, J. A. and J. Vandewalle, "Least squares support vector machine classifiers", Neural processing letters **9**, 3, 293–300 (1999).

Thompson, L., F. P. Rivara and J. M. Whitehill, "Prevalence of marijuana-related traffic on twitter, 2012–2013: a content analysis", Cyberpsychology, Behavior, and Social Networking **18**, 6, 311–319 (2015).

Tian, Q., P. Zhang and B. Li, "Towards predicting the best answers in community-based question-answering services", in "Seventh International AAAI Conference on Weblogs and Social Media", (2013).

Tighe, J. and S. Lazebnik, "Finding things: Image parsing with regions and per-exemplar detectors", in "Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on", pp. 3001–3008 (IEEE, 2013a).

Tighe, J. and S. Lazebnik, "Superparsing", International Journal of Computer Vision **101**, 2, 329–349 (2013b).

Triggs, B. and J. J. Verbeek, "Scene segmentation with crfs learned from partially labeled images", in "Advances in neural information processing systems", pp. 1553–1560 (2008).

van Hoof, J. J., J. Bekkers and M. van Vuuren, “Son, youre smoking on facebook! college students disclosures on social networking sites as indicators of real-life risk behaviors”, Computers in human behavior **34**, 249–257 (2014).

Vemulapalli, R., O. Tuzel, M.-Y. Liu and R. Chellapa, “Gaussian conditional random field network for semantic segmentation”, in “Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition”, pp. 3224–3233 (2016).

Vezhnevets, A. and J. M. Buhmann, “Towards weakly supervised semantic segmentation by means of multiple instance and multitask learning”, in “Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on”, pp. 3249–3256 (IEEE, 2010).

Vezhnevets, A., J. M. Buhmann and V. Ferrari, “Active learning for semantic segmentation with expected change”, in “Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on”, pp. 3162–3169 (IEEE, 2012a).

Vezhnevets, A., V. Ferrari and J. M. Buhmann, “Weakly supervised semantic segmentation with a multi-image model”, in “Computer Vision (ICCV), 2011 IEEE International Conference on”, pp. 643–650 (IEEE, 2011).

Vezhnevets, A., V. Ferrari and J. M. Buhmann, “Weakly supervised structured output learning for semantic segmentation”, in “Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on”, pp. 845–852 (IEEE, 2012b).

Volkow, N. D., R. D. Baler, W. M. Compton and S. R. Weiss, “Adverse health effects of marijuana use”, New England Journal of Medicine **370**, 23, 2219–2227 (2014).

Voravuthikunchai, W., B. Crémilleux and F. Jurie, “Histograms of pattern sets for image classification and object recognition”, in “Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on”, pp. 224–231 (IEEE, 2014).

Wang, Y., K.-F. Loe and J.-K. Wu, “A dynamic conditional random field model for foreground and shadow segmentation”, IEEE transactions on pattern analysis and machine intelligence **28**, 2, 279–289 (2006).

Whitehill, J. M., M. A. Pumper and M. A. Moreno, “Emerging adults use of alcohol and social networking sites during a large street festival: A real-time interview study”, Substance abuse treatment, prevention, and policy **10**, 1, 1 (2015).

Xie, W., Y. Peng and J. Xiao, “Semantic graph construction for weakly supervised image parsing”, in “Twenty-Eighth AAAI Conference on Artificial Intelligence”, (2014a).

Xie, W., Y. Peng and J. Xiao, “Weakly-supervised image parsing via constructing semantic graphs and hypergraphs”, in “Proceedings of the ACM International Conference on Multimedia”, pp. 277–286 (ACM, 2014b).

Yang, L., S. Bao, Q. Lin, X. Wu, D. Han, Z. Su and Y. Yu, “Analyzing and predicting not-answered questions in community-based question answering services”, in “Proceedings of AAAI”, pp. 1273–1278 (2011).

Yao, Y., H. Tong, T. Xie, L. Akoglu, F. Xu and J. Lu, “Detecting high-quality posts in community question answering sites”, Information Sciences (2015).

Zeiler, M. D. and R. Fergus, “Visualizing and understanding convolutional networks”, in “European conference on computer vision”, pp. 818–833 (Springer, 2014).

Zhang, L., Y. Gao, Y. Xia, K. Lu, J. Shen and R. Ji, “Representative discovery of structure cues for weakly-supervised image segmentation”, Multimedia, IEEE Transactions on **16**, 2, 470–479 (2014a).

Zhang, L., M. Song, Z. Liu, X. Liu, J. Bu and C. Chen, “Probabilistic graphlet cut: Exploiting spatial structure cue for weakly supervised image segmentation”, in “Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on”, pp. 1908–1915 (IEEE, 2013).

Zhang, L., Y. Yang, Y. Gao, Y. Yu, C. Wang and X. Li, “A probabilistic associative model for segmenting weakly-supervised images”, IEEE Transactions on Image Processing **23**, 9, 4150–4159 (2014b).

Zhang, W., S. Zeng, D. Wang and X. Xue, “Weakly supervised semantic segmentation for social images”, in “Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition”, pp. 2718–2726 (2015).

Zhang, Y., J. Wu and J. Cai, “Compact representation for image classification: To choose or to compress?”, in “CVPR 2014 IEEE Conference on”, pp. 907–914 (IEEE, 2014c).

Zheng, S., S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang and P. H. Torr, “Conditional random fields as recurrent neural networks”, in “Proceedings of the IEEE International Conference on Computer Vision”, pp. 1529–1537 (2015).

# APPENDIX A

# RELATED PATENT

Baoxin Li, Parag Shridhar Chandakkar, Qiongjie Tian, “SYSTEMS AND METHODS FOR A CONTENT-ADAPTIVE PHOTO-ENHANCEMENT RECOMMENDER”, Patent:US9576343B2, Issued Date Feb 21, 2017

# APPENDIX B

# RELATED PUBLICATIONS

- Qiongjie Tian, Baoxin Li, “Weakly Hierarchical Lasso based Learning to Rank in Best Answer Prediction”, IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), 2016.

- Qiongjie Tian, Jashmi Lagisetty, Baoxin Li, “Finding Needles of Interested Tweets in the Haystack of Twitter Network”, IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), 2016,

- Qiongjie Tian, Baoxin Li, “Simultaneous Semantic Segmentation of a Set of Partially Labeled Images”, IEEE Winter Conference on Applications of Computer Vision (WACV), 2016

- Devi Archana Paladugu, Qiongjie Tian, Hima Bindu Maguluri, Baoxin Li, “Towards Building an Automated System for Describing Indoor Floor Maps for Individuals with Visual Impairment”, doi: 10.1080/23335777.2016.1141801, Journal of Cyber-Physical Systems, 2015

- Parag Shridhar Chandakkar, Qiongjie Tian, Baoxin Li, “Relative learning from web images for content-adaptive enhancement”. IEEE International Conference on Multimedia and Expo (ICME), 2015

- Hima Bindu Maguluri, Qiongjie Tian and Baoxin Li, “Detecting text in floor maps using Histogram of Oriented Gradients”. Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on, Vancouver, 2013

- Qiongjie Tian, Peng Zhang and Baoxin Li. “Towards Predicting the Best Answers in Community-based Question-Answering”, The International AAAI Conference on Web and Social Media (ICWSM), 2013

- Qiongjie Tian, Lin Chen, Qiang Zhang and Baoxin Li, “Enhancing Fundamentals of Laparoscopic Surgery Trainer Box via Designing A Multi-Sensor Feedback System”, NextMed/MMVR20, 2013

- Lin Chen, Qiongjie Tian, Qiang Zhang and Baoxin Li, “Learning Skill-Defining Latent Space in Video-Based Analysis of Surgical Expertise - A Multi-Stream Fusion Approach”, Next/MMVR20, 2013

- Qiang Zhang, Lin Chen, Qiongjie Tian and Baoxin Li, “Video-based analysis of motion skills in simulation-based surgical training”, SPIE Multimedia Content Access: Algorithms and Systems VII, 2013

- Devi Archana Paladugu, Hima Bindu Maguluri[1], Qiongjie Tian[1], Baoxin Li, “Automated description generation for indoor floor maps”, Proceedings of the 14th international ACM SIGACCESS conference on Computers and accessibility. 2012

- Yu Wang, Bin Li, Thomas Weise, Jianyu Wang, Bo Yuan, Qiongjie Tian. “Self-adaptive learning based particle swarm optimization”, Information Sciences 181(20): 4515-4538 (2011)

[1] Equally contributed.