Benchmarking Machine Learning Methods for Expert Recommendation

Seyedeh Mina Mousavifar
sem311@mail.usask.ca
University of Saskatchewan
Saskatoon, SK, Canada

Amirabbas Jalali
amj301@mail.usask.ca
University of Saskatchewan
Saskatoon, SK, Canada

Fidelia Orji
fao583@mail.usask.ca
University of Saskatchewan
Saskatoon, SK, Canada

ABSTRACT

The community question and answer (CQA) platforms, such as Stack Overflow(SO), leverage the knowledge and expertise of users to answer questions posted by fellow users. Throughout time these websites transform into information repositories. The acquisition and sharing of knowledge are generally valuable but exorbitant for both businesses and individuals, particularly technical knowledge that covers a variety of topics. CQA platforms offer an open door for sharing information requiring little to no effort, where group members, many of whom are domain experts, can possibly give top-notch answers for a given problem. Nonetheless, in the time of data blast, a huge volume of questions are posted each day, and it is challenging to distinguish the relevant qualified experts to respond to these inquiries. In this project, we aim to recommend the experts who are most likely going to answer a given question in the Stack Overflow community. Our goal is to benchmark various machine learning methods performance in expert recommendations using 2019 SO dataset. Finally, we will also empirically analyze the importance of questions' and experts' features on different methods.

KEYWORDS

community question answering, expert recommendation, naive bayes, support vector machine, linear regression, random forest

ACM Reference Format:
Seyedeh Mina Mousavifar, Amirabbas Jalali, and Fidelia Orji. 2020. Benchmarking Machine Learning Methods for Expert Recommendation. In *CMPT820: Machine Learning course, Winter, 2020, University of Saskatchewan, SK.* , 5 pages.

1 INTRODUCTION

The main concern in SO platform is the absence of successful coordinating among questions and the potential great answerers, which unfavourably influences effective knowledge acquisition. From one viewpoint, a question may experience several low-quality answers without receiving any proper answer in a limited timeframe. Differently, an expert might encounter various new inquiries without being able to distinguish their question of interest rapidly. Under this circumstance, expert recommendation is a promising approach. According to previous studies[1], most answers and knowledge in the communities originate from only a minority of users. A recent study on Stack Overflow and Quora[2] demonstrates that these platforms incorporate profoundly committed domain experts, who target not only the requester question but also a long-lasting answer to a broader crowd. Consequently, expert recommendation would bring experts' attention to relevant questions effectively and instantly, leading to stronger communities in CQA. The purpose of the expert recommender system is to suggest relevant experts for a given question. Ranking experts based on experts' features, questions' features and their interaction require effectively adjusting the importance of these features and discovering proper ranking. However, due to the complexity of this problem, we cast our recommendation problem into a classification problem, in which the recommender would predict whether a question will be answered by an expert or not. Classification methods can easily utilize multiple aspects of features from the expert, question, and answer compared to "language-based" and "network-based" models[21]. One critical task in our recommendation is to determine the attributes that define a user, question, and answer. These features consist of expert features(e.g., community badges, reputation, upvote), question and answer features(e.g., community score, text similarity, tags, view count). Further, we investigate the performance of methods such as Naive Bayes[9], Support Vector Machine[16], Logistic Regression[12], and Random Forest[8] based on metrics such as accuracy and F1-score using 2019 SO dataset.

We summarize this project contributions as follows:

- We proposed a traditional machine learning pipeline for expert recommendation in online social collaborative platform. As part of the proposed pipeline, we investigated and designed features that are representative of users and questions in SO for expert recommendations.
- We explored and benchmarked multiple traditional machine learning models for the expert recommendation task.
- We evaluated our proposed pipeline by conducting extensive experiments on a large real-world dataset.

2 LITERATURE REVIEW

Recommender systems are systems that use specific algorithms in suggesting relevant information to its users based on inferred needs of the users. Expert recommendation on CQA platforms is a challenging problem due to its various aspects for many years. Previous research has shown that one of the factors that motivate people to join these platforms is the promptness in providing quality-answers for any given question posted on the platforms[4]. Recommendation on these platforms involves linking questions to users who are knowledgeable in a question domain so that they can provide quality-answers to the question. Several research works have tried to find experts for questions in different CQA platforms through different methods. Riahi et al [18] compared the performance of two statistical topic models, the "Segmented Topic" model and the

Machine Learning Course'20, Winter, 2020, University of Saskatchewan, SK

"Latent Dirichlet Allocation" model for expert recommendation using the SO dataset. They found that the "Segmented Topic" model outperformed the "Latent Dirichlet Allocation" model. Xu et al [22] explored the "Dual Role models" (DRM) approach for question recommendation in CQA using the Yahoo dataset. Their results show that the DRM model approach improved recommendations for questions. There are other existing research works on expert finding for CQA such as the use of "ranking metric network learning" [23], identifying experts using entity ranking on graphs [19], expert finding with reputation and link analysis [13], and expert recommendation through data-driven techniques [5].

The existing research works on experts finding achieved improved performance through various methods. For instance, Xu et al[22] modelled users in CQA using two roles. The first role modelled users that ask questions and the second modelled those that provide answers to questions. Using a probabilistic framework, they showed that the second role is more effective for recommending questions to users. Boeva et al[5] used a weighing method to classify experts based on domain-specific topics. They modelled expert profiles based on different topics and levels (weighted) of knowledge an expert has on each topic. Another research study[23] investigated the use of graph-based algorithms on a social network in recommending experts. In these recommendation systems, the ability to determine experts' domain knowledge is very crucial. As questions will be recommended to experts based on their knowledge in a specific area or topic. Quantifying the knowledge level of experts and questions is often very difficult. Hence, some of the research works combined either social network analysis or graph-based ranking with content analysis to obtain features for their models.

Most of the current research work on CQA was based on either a collaborative approach or a content-based approach. The collaborative approach uses user-question interaction while the content-based adds information about users, questions or answers. Another important approach is the use of a hybrid approach in finding experts for a question in CQA platforms. This approach was explored by Liu et al[13] using Yahoo! Answer Taiwan dataset. In their research, they consider a user's reputation, domain knowledge, and authority. Reputations of users were provided by question-answering history and authority was obtained through link analysis. Their results show that the most relevant features were knowledge profile and reputation, which contributed more than authority. Also, the researchers reported that consideration of past question relevance to the target question led to better performance. Liu et al[13] research is closely related to our research because our research used a hybrid approach. We used reputation and other features in modelling expert users of SO. We built experts' domain-knowledge based on previous answers to questions and our questions' features, then check the relevance of an expert's domain-knowledge to a newly posted question. We used this hybrid approach to evaluate the effectiveness of five current machine learning algorithms for recommendations in the SO platform. However, our research is different from Liu et al.'s work because they use link analysis and authority in constructing their features and they did not explore any of the machine learning algorithms that we are investigating in this research.

3 ALGORITHM DESCRIPTION

We first present the feature engineering process, in which we extract features from questions and answers textual data. Then, we elaborate on different classification methods that would predict whether a question will be answered by an expert or not.

3.1 Feature Engineering

Evaluation of similarity in questions and experts' answers, which is a textual data is one of the most critical tasks for expert recommendation in SO. Question textual features are extracted based on the question title. The user textual features are extracted based on the titles of questions that the user has previously answered. The raw data, as a sequence of symbols with variable length and sequential dependency, cannot be fed directly into the ML methods. On one hand, if we treat this text as nominal data, many methods would expect numerical feature vectors with a fixed size. On the other hand, if we manage this data as categorical input, we would have numerous features without any measure of importance between them. Hence, we use methods from information retrieval domain. The most popular technique for text feature extraction is frequency-inverse document frequency (TF-IDF)[20], which is a numerical statistic that reflects how important a word is to a document in a corpus. This model, convert the textual representation of information into the sparse features. In this project, we extract textual features from "question titles". To obtain a more meaningful corpus, we first tokenize question titles, then remove stop words and finally stem the words to their root using Porter Stemmer algorithm[17].

TF-IDF consists of two statistics, term frequency and inverse document frequency. Term frequency(tf) measures frequency of a word in a document, which is the number of occurrences of a term in a document divided by the whole number of terms in the document for normalization to remove the dependency of frequency on document length. Inverse document frequency(idf) measures the importance of the document in the corpus. Document frequency is the number of occurrences of a term in the documents in the corpus divided by the whole number of documents for normalization. The inverse of document frequency shows the rareness of a term in the corpus. For instance, if the word is very common and appears in many documents, this number will approach 0. Otherwise, it will approach 1.

$$tf(t,d) = \frac{f_{t,d}}{\sum_{t'} f_{t',d}}$$

$$df(t) = \sum_{d'} f(t,d')$$

$$idf(t,D) = \log(\frac{N}{1+df(t)})$$

$$tfidf(t,d,D) = tf(t,d) \cdot idf(t,D)$$

Where t and $d$denotes term and document respectively, N is the total number of documents in the corpus, and D shows the corpus. At last, these TF-IDF vectors are normalized by *"Euclicidian Norm"*.

3.2 Machine Learning Models

The input to our machine learning models is the feature vector, that we denote by $\mathbf{x}_i$, representing vector of question features $\mathbf{q}_m$, the vector of expert features $\mathbf{u}_n$, and the label indicating whether an expert has answered the question r_i.

$$\mathbf{x}_i = (\mathbf{q}_m, \mathbf{u}_n, r_i)$$

3.2.1 Naive Bayes. This method is a simple probabilistic model applying *"Bayes Theorem"* with the strong assumption of "conditional independence" between features. This model fits our problem for its high scalability and linear time complexity because it requires a number of parameters linear in the number of features. We only have nominal features, so we use Gaussian Naive Bayes for our binary classification.

$$P(y_j|\mathbf{x}_i) = \frac{P(\mathbf{x}_i|y_j)P(w_j)}{P(\mathbf{x}_i)}$$

Where $\mathbf{x}_i$ is the feature vector of samples i, $i \in 1, 2, ..., n$, y_j is the notation of class j, $j \in 0, 1$. Classes are binarized 1 or 0 which can be seen as 1 means expert will answer question , and 0 otherwise. $P(\mathbf{x}_i|y_j)$ is the probability of observing sample $\mathbf{x}_i$ given that it belongs to class y_j.

Further, the class condition probabilities of individual features d are as follows because of conditional independence assumption.

$$P(\mathbf{x}|y_j) = P(x_1|y_j)...P(x_d|y_j) = \Pi_{k=1}^{d} P(x_k|y_j)$$

Where d is the total number of features and x_k represent each feature.

Finally, the decision rule is:

$$predicted\ class\ label \leftarrow arg\ max\ P(y_j|\mathbf{x}_i) \quad for\ j = 0, 1$$

3.2.2 Random Forest[7]. This approach consists of a large number of individual decision trees that operate as an ensemble, which uses multiple learning algorithms to obtain better predictive performance compared to each learning algorithm separately. This algorithm aims to utilize the wisdom of crowds in which each individual tree in the random forest outputs a class prediction and the class with the most votes becomes the model's prediction. Random forest allows each tree to randomly sample from the dataset with replacement, which results in different trees. These different trees would use diverse sets of features for prediction and add variability to this model. The key characteristic in the random forest method is obtaining trees with low correlation with each other, this leads to better performance. While the algorithm via feature randomness tries to establish these low correlations, the selection of features and the hyper-parameters such as depth of tree also impact this correlation.

3.2.3 Logistic Regression[14]. This method is a supervised binary classifier, that uses a logistic function to model a binary dependent variable based on features. Logistic regression assumes a linear relationship between the features and the log-odds of each class. Logistic Regression has equations similar to it"Linear Regression" model but it limits the cost function between 0 and 1, so it uses *"Sigmoid function"* to map predicted values to probabilities.

$$Z = \beta_0 + \beta_1 x_1 + \cdots + \beta_n x_d$$

$$P(Y = 1|\mathbf{x}_i) = \frac{1}{1 + e^{-Z}} = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_1 + \cdots + \beta_n x_d)}}$$

Where β_i is weight parameter for each feature, which are tuned during learning. We should note that logistic regression outputs the probability of the class, so we should use a threshold(e.g. 0.5) to map probabilities higher than this threshold to the current class and lower probabilities to the other class.

$$P(Y = 1|\mathbf{x}_i) = 0.5 \longrightarrow \text{expert will answer the question}$$
$$P(Y = 1|\mathbf{x}_i) \leq 0.5 \longrightarrow \text{expert won't answer the question}$$

3.2.4 Support Vector Machine[6]. SVM is a non-probabilistic supervised classifier, with the objective of finding a hyperplane in D-dimensional space(D — the number of features) that clearly classifies the data points. SVM is capable of both linear and non-linear classification. This model fits our problem because it can utilize kernels to implicitly map inputs into high-dimensional feature spaces. In this algorithm, we aim to maximize the margin between the data points and the hyperplane. So our objective is minimizing the following function by taking partial derivatives with respect to the weights to find the gradients.

$$Z = [\frac{1}{n}\sum_{i=1}^{n} max(0, 1 - y_i(\mathbf{w} \odot \mathbf{x}_i - b))] + \lambda \left\|\mathbf{w}\right\|^2$$

$$\frac{\partial}{\partial w_k} \lambda \left\|\mathbf{w}\right\|^2 = 2\lambda w_k$$

$$\frac{\partial}{\partial w_k} max(0, 1 - y_i(\mathbf{w} \odot \mathbf{x}_i - b)) = \{0, -\mathbf{y_i} \odot \mathbf{x_i}\}$$

So the weights would be updated based on the gradients.

$$No\ misclassification \longrightarrow \mathbf{w} = \mathbf{w} - \alpha \cdot 2\lambda\mathbf{w}$$
$$Misclassification \longrightarrow \mathbf{w} = \mathbf{w} - \alpha \cdot (2\lambda\mathbf{w} - \mathbf{y_i} \odot \mathbf{x_i})$$

SVM has α and λ as hyperparameters. λ is the regularization parameter for preventing overfitting by penalizing for additional and higher-order weights, and *alpha* is the learning rate. Furthermore, we should also choose our preferred kernel.

4 DATA DESCRIPTION

We use Stack Overflow dataset[1], which has approximately 1 million users and 19 million posts. Due to limitations on computational power, we only used 2019 posts, and consider users who have answered more than 15 questions during last year as experts. Furthermore, we only consider the accepted answer for a proposed question, because most of the unaccepted answers have a low quality, which doesn't contribute to expert identification. We use 80% of our records for the training set, which will be further used for

[1] https://archive.org/details/stackexchange

Table 1: Summary of Active experts 2019 SO dataset

Feature	Min	Mean	Max
Experts Answers	21	72	4507
Reputation	1	25,278	1,166,685
Views	23	3,551	1,911,062
Upvote	0	1,218	53,163
Downvote	0	716	77,537

5-fold cross-validation[11] for parameter tuning, and the remaining 20% as the test set.

4.1 Negative Sampling

Negative sampling[15] is a method used for machine learning models that have numerous negative observations than positive ones. Furthermore, we don't have records of experts who didn't answer a question directly. Consequently, we apply negative sampling to obtain negative interactions, such that our machine learning methods would learn both positive and negative interactions. For each question, we randomly sample 4 experts who didn't answer the question as negative sample.

4.2 Feature Summary

- 392,798 Questions
  - Vote: Voting is how the community indicates which questions and answers are most useful and appropriate, which can be either affirmative or negative(as upvote and downvote).
  - Score: The difference between question upvotes and downvotes, between -25 to 491.
  - View Count: Number of times this question has been viewed on SO, between 6 to 94,473.
  - Title: String of question title, which is further described as tf-idf weights of most 5000 frequent terms.
- 5,448 Experts, and table 1 provides summary of experts' features.
  - Reputation: a user is awarded 10 reputation points for receiving an "up" vote on an answer given to a question and 10 points for the "up" vote of a question.
  - Views: Number of times this user has been viewed on SO.
  - Upvote: Number of affirmative votes the user has obtained in previous questions and answers.
  - Downvote: Number of negative votes the user has obtained in previous questions and answers.
  - Mean of answers' score: Average of user provided answers' score, which is the difference between answer upvotes and downvotes.
  - Previously answered questions titles: Previously answered questions titles which is described as tf-idf weights of most 5000 frequent terms.
- 1,963,990 Question and Expert Interactions
  - Interaction Matrix: The info about a question with mentioned features, accompanied by user mentioned features who has answered the question, a total of 10,006 features.

5 RESULTS

We have constructed our final expert-question interaction matrix including both expert and question features. In the data collection phase, our main problem was the enormous XML dataset format, that we set up a database so that we could collect one-year data by an SQL query swiftly. However, many textual data types contain special characters that would interfere with text formatting, which makes exporting the dataset problematic. Subsequently, we cleaned our textual data, by removing all concerning characters and limiting question languages only to English characters.

In the feature engineering phase, we applied TF-IDF method. Our corpus consists of 52,326 terms, which leads to a gigantic sparse matrix. This large number of dimensions would be troublesome due to its huge dimension. So we limited our corpus to 5,000 most frequent terms. At this stage, we split our questions into training set and test set, to extract experts' features from provided answers. Further, we obtained experts' TF-IDF terms as the average of question titles they had answered and computed the average score of these answers.

For the evaluation of our models, we use four set-based metrics.

- Precision, the fraction of experts who will truly answer the question, among all the experts that model classifies to answer the question.
- Recall, the fraction of experts who are predicted to answer the question and will actually answer the question, among all experts who answer the question.
- F1-score, the harmonious average between Precision and Recall.
- Accuracy, the fraction of experts who are correctly classified as either answering the question or not answering the question.

There are far fewer experts who will answer the question, compared to the rest of the experts who won't. Consequently, we speculate that F1-score would be a better metric than Accuracy to our problem due to the uneven class distribution.

6 DISCUSSION

Currently, we have 10,006 dimensions which are concerning because of the *curse of dimensionality problem*[3]. We predict that Random Forest would have the best performance between our models, because of its capability to datasets with high dimensionality. Moreover, this method can identify the significance of features, that can be further used for dimensionality reduction. However, this method would have a high computational cost for deep tress and its slower than other methods in prediction.

SVM would have the next best performance, because of its ability to be effective in high dimensional spaces. Furthermore, most experts would be knowledgable in a specific domain leading to non-overlapping classes in our dataset. This margin of separation will facilite better performance.

Logistic regression would perform better than Naive Bayes because of a more useful approach in multicollinearity[10] condition and preventing overfitting with regularization. Though, it would have worse performance than SVM, because of its limitation in capturing non-linear relation in data.

Naive Bayes might suffer from the zero probability problem, where a feature conditional probability equals to zero, and perform badly in case of multicollinearity. This problem is probable in our recommendation because of our large feature set. Naive Bayes in this condition would fail to produce a valid prediction.

Questions:

- Would PCA be suitable for dimension reduction, without losing information obtained from TF-IDF?
- Should we use question tags as one hot encoding, or this will increase dimension without contributing to our results?

7 FUTURE WORK

For the following phase, we will implement the mentioned models. Mostly, we will be testing adding, removing or summarizing our features on the validation set, and compare the results on the test set.

REFERENCES

[1] Lada A. Adamic, Jun Zhang, Eytan Bakshy, and Mark S. Ackerman. 2008. Knowledge sharing and yahoo answers. *Proceeding of the 17th international conference on World Wide Web - WWW 08* (May 2008), 665–674. https://doi.org/10.1145/1367497.1367587

[2] Ashton Anderson, Daniel Huttenlocher, Jon Kleinberg, and Jure Leskovec. 2012. Discovering value from community activity on focused question answering sites. *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD 12* (Aug 2012), 850–858. https://doi.org/10.1145/2339530.2339665

[3] Richard Bellman. 1966. Dynamic programming. *Science* 153, 3731 (1966), 34–37.

[4] Vasudev Bhat, Adheesh Gokhale, Ravi Jadhav, Jagat Pudipeddi, and Leman Akoglu. 2014. Min(e)d your tags: Analysis of Question response time in StackOverflow. *2014 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2014)* (2014), 328–335. https://doi.org/10.1109/asonam.2014.6921605

[5] Veselka Boeva, Milena Angelova, and Elena Tsiporkova. 2017. Data-driven Techniques for Expert Finding. *Proceedings of the 9th International Conference on Agents and Artificial Intelligence* (2017). https://doi.org/10.5220/0006195105350542

[6] Bernhard E. Boser, Isabelle M. Guyon, and Vladimir N. Vapnik. 1992. A training algorithm for optimal margin classifiers. *Proceedings of the fifth annual workshop on Computational learning theory - COLT 92* (1992). https://doi.org/10.1145/130385.130401

[7] Leo Breiman. 2001. Random Forests. *Machine Learning* 45, 1 (2001), 5–32. https://doi.org/10.1023/A:1010933404324

[8] Morakot Choetkiertikul, Daniel Avery, Hoa Khanh Dam, Truyen Tran, and Aditya Ghose. 2015. Who Will Answer My Question on Stack Overflow? *2015 24th Australasian Software Engineering Conference* (Sep 2015), 155–164. https://doi.org/10.1109/aswec.2015.28

[9] David Van Dijk, Manos Tsagkias, and Maarten De Rijke. 2015. Early Detection of Topical Expertise in Community Question Answering. *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval - SIGIR 15* (Apr 2015), 995–998. https://doi.org/10.1145/2766462.2767840

[10] Donald Eugene. Farrar. 1967. *Multicollinearity in regression analysis: the problem revisited.* Review of Economics.

[11] R. Kohavi. 1995. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *International joint Conference on artificial intelligence*, Vol. 14. Citeseer, 1137–1145.

[12] Long T. Le and Chirag Shah. 2016. Retrieving Rising Stars in Focused Community Question-Answering. *Intelligent Information and Database Systems Lecture Notes in Computer Science* (Mar 2016), 25–36. https://doi.org/10.1007/978-3-662-49390-8_3

[13] Duen-Ren Liu, Yu-Hsuan Chen, Wei-Chen Kao, and Hsiu-Wen Wang. 2013. Integrating expert profile, reputation and link analysis for expert finding in question-answering websites. *Information Processing & Management* 49, 1 (2013), 312–329. https://doi.org/10.1016/j.ipm.2012.07.002

[14] P. Mccullagh and J. A. Nelder. 1989. An outline of generalized linear models. *Generalized Linear Models* (1989), 21–47. https://doi.org/10.1007/978-1-4899-3242-6_2

[15] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed Representations of Words and Phrases and their Compositionality. In *NIPS*. Curran Associates, Inc., 3111–3119. http://papers.nips.cc/paper/5021-distributed-representations-of-words-and-phrases-and-their-compositionality.pdf

[16] Aditya Pal and Joseph A. Konstan. 2010. Expert identification in community question answering. *Proceedings of the 19th ACM international conference on Information and knowledge management - CIKM 10* (2010), 1505–1508. https://doi.org/10.1145/1871437.1871658

[17] Martin F Porter. 1980. An algorithm for suffix stripping. *Program* 14, 3 (1980), 130–137.

[18] Fatemeh Riahi, Zainab Zolaktaf, Mahdi Shafiei, and Evangelos Milios. 2012. Finding expert users in community question answering. *Proceedings of the 21st international conference companion on World Wide Web - WWW 12 Companion* (2012). https://doi.org/10.1145/2187980.2188202

[19] H. Rode, Pavel Serdyukov, Djoerd Hiemstra, and H. Zaragoza. 2007. *Entity Ranking on Graphs: Studies on Expert Finding.* Number FS-07-05/TR-CTIT-07-81 in CTIT Technical Report Series. Centre for Telematics and Information Technology (CTIT), Netherlands.

[20] Gerard Salton and Michael J. McGill. 1987. *Introduction to modern information retrieval.* McGraw-Hill Intern.

[21] Xianzhi Wang, Chaoran Huang, Lina Yao, Boualem Benatallah, and Manqing Dong. 2018. A Survey on Expert Recommendation in Community Question Answering. *Journal of Computer Science and Technology* 33, 4 (2018), 625–653. https://doi.org/10.1007/s11390-018-1845-0

[22] Fei Xu, Zongcheng Ji, and Bin Wang. 2012. Dual role model for question recommendation in community question answering. *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval - SIGIR 12* (2012). https://doi.org/10.1145/2348283.2348387

[23] Zhou Zhao, Furu Wei, Ming Zhou, and Wilfred Ng. 2015. Cold-Start Expert Finding in Community Question Answering via Graph Regularization. *Database Systems for Advanced Applications Lecture Notes in Computer Science* (2015), 21–38. https://doi.org/10.1007/978-3-319-18120-2_2