MEDINFO 2015: eHealth-enabled Health
I.N. Sarkar et al. (Eds.)

doi:10.3233/978-1-61499-564-7-300

A Decision Fusion Framework for Treatment Recommendation Systems

Jing Mei[a], Haifeng Liu[a], Xiang Li[a], Guotong Xie[a], Yiqin Yu[a]

[a] *IBM Research, Beijing, China*

Abstract

Treatment recommendation is a nontrivial task – it requires not only domain knowledge from evidence-based medicine, but also data insights from descriptive, predictive and prescriptive analysis. A single treatment recommendation system is usually trained or modeled with a limited (size or quality) source. This paper proposes a decision fusion framework, combining both knowledge-driven and data-driven decision engines for treatment recommendation. End users (e.g. using the clinician workstation or mobile apps) could have a comprehensive view of various engines' opinions, as well as the final decision after fusion. For implementation, we leverage several well-known fusion algorithms, such as decision templates and meta classifiers (of logistic and SVM, etc.). Using an outcome-driven evaluation metric, we compare the fusion engine with base engines, and our experimental results show that decision fusion is a promising way towards a more valuable treatment recommendation.

Keywords:

Clinical decision support system, Decision fusion, Treatment recommendation.

Introduction

Clinical Decision Support (CDS) provides recommendations from organized medical knowledge and patient information to improve healthcare delivery. A variety of CDS systems have been proposed and implemented. From the perspective of methodology, some CDS systems are knowledge-driven, such as the computerization of clinical practice guidelines, and some are data-driven, such as the discovery of unknown patterns and trends in a large volume of patient data. From the perspective of analytics, some CDS systems are descriptive, such as getting the most frequently used drugs from similar patients, some are predictive, such as the estimation of clinical outcome for taking a certain drug, and some are prescriptive, such as a long-term planning for the optimal intervention at every possible patient state.

Considering that each CDS system has its own strength and weakness, we might not only leverage a single CDS system, but also apply decision fusion technologies to combine multiple CDS systems' results. Actually, in real life, it is natural to consult "several experts" before making a final decision. As it si said, two heads are better than one. The extensive benefits of decision fusion have been shown up in the Netflix Grand Prize in 2009 [1], which was an open competition to predict user ratings for films, and the winner combined the previous three teams' results to achieve a 10.09% improvement. In addition, the Heritage Health Prize in 2012 [2] was an open competition to predict how many days a patient would spend in a hospital in the next year. Again, fusion methods were widely used by both milestone winners and final winners in this competition.

In literature, the research field of "decision fusion" is known under various names, such as multiple classifier systems, mixture of experts and ensemble learning [3]. A general solution of prior arts is the fusion of different (data-driven) learning algorithms with different parameter settings, e.g., trying various features and datasets. However, for clinical decision support, esp. in evidence-based medcine, not only the data-driven learning algorithms are useful, but also the knowledge-driven modeling techniques are greatly helpful. In this paper, we propose a decision fusion framework for a treatment recommendation system, which combines both knowledge-driven and data-driven approaches.

To be convinced of the benefits of decision fusion, we need evaluation metrics, comparing the results from base decision engines with the results from our fusion engine. However, we observe that state-of-the-art evaluation metrics [4], such as precision, recall and RMSE (Root Mean Square Error), are not applicable for evaluating treatment recommendation systems, due to the *partially observed ground truth*. In machine learning, the term "ground truth" refers to the known facts about the training data set in terms of the learning tasks. Taking the Netflix Grand Prize as an example, the ground truth is the actual user ratings for films, while taking the Heritage Health Prize as another example, the ground truth is the actual days a patient would spend in a hospital in the next year. However, as far as for treatment recommendation, what's the ground truth? A naïve answer might be the actual prescription in real data. But, is that right? Suppose that drug A was recommended by a decision engine, and the physician did choose drug A as the prescription, but unfortunately, the patient outcome of using drug A was bad. In this respect, could we mark the recommendation of drug A as correct? Another story is that drug B was recommended by an analysis module, but the physician chose drug C as the prescription, and the patient outcome of using drug C appeared good. Thus, could we mark the recommendation of drug B as incorrect? What if the patient outcome of using drug B becomes better than using drug C? Therefore, we call it the *partially observed ground truth* (i.e., not all decision options are completely observed with outcomes), and in this paper, we propose an outcome-driven measure for evaluating treatment recommendation systems. Here, we remark that the fusion itself has no impact on the "partialness" of the ground truth. Actaully, it is the "partialness" that brings challenges for evalution of treatment recommendation systems, including the base and fusioned ones.

For experiments, we implement a decision fusion framework, which combines three base decision engines. The first engine is a knowledge-driven engine based on clinical practice

guidelines (CPG [5]). The other two are data-driven engines, of which one is a descriptive analytics based on patient similarity (PSA [6]), and the other engine is a predictive analytics based on outcome prediction (PRE). Using different fusion algorithms (such as decision templates and meta classifiers), we compare the evaluation results and conclude that fusion does help to provide treatment recommedations in a more valuable way.

Methods

We first present a decision fusion framework, followed by the introduction of three base decision engines. Next, we describe the fusion engine with a variety of fusion algorithms. Finally, an outcome-driven evaluation metric is defined, which will be used to compare the fusion engine with base decision engines.

Decision fusion framework

As shown in Fig 1, we propose an open framework for decision fusion. A number of decision engines can contribute to the fusion framework. We implement a fusion engine that gets input from base decision engines. On the client side (such as the clinical workstation or mobile apps), we display the treatment recommendations from each of the component, as well as the final decision from the fusion engine. End users have the privilege to see the outcome from all engines.

The fusion engine has two phases. First is the training phase, where all the results from base decision engines will be fed into a fusion engine to learn a fusion model. Second is the testing phase, where an instance that uses the output of base decision engines as features is created, and the trained fusion model predicts the final outcome. Every base decision engine has been either trained well by its own data source (such as PSA and PRE) or modeled well by its own knowledge source (such as CPG). The training and testing phases are meant for the fusion, rather than any base decision engine.

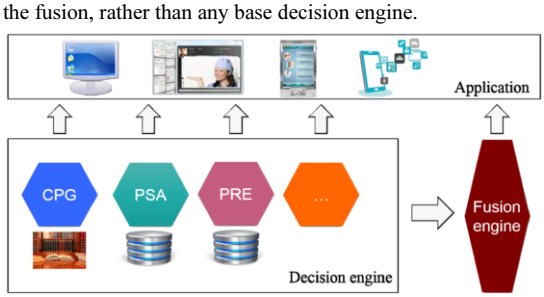

Figure 1 - The decision fusion framework

Decision engine

A decision engine itself provides treatment recommendations, but its methodology and analytical principles may differ from each other. Table 1 shows three different perspectives. The differentiation of decision engines results in the requirement to consult "several engines" before making a final decision.

Technically, CPG [5] is a knowledge-driven decision engine, which computerizes the NICE clinical guideline for Type 2 diabetes. At design time, guidelines are defined as standard (XPDL)-based business processes, where clinical conditions are represented using GELLO expressions. At run-time, a process engine would invoke a query adaptor to retrieve clinical data and a GELLO engine to evaluate clinical conditions whenever a decision-making is needed during the care process. Consequently, clinical recommendations are generated for lifestyle intervention and drug therapy, etc. For example, given a patient who is overweight and whose blood glucose is inadequately controlled by lifestyle interventions alone, CPG would recommend to start Metformin.

Table 1 – Decision engines

	Methodology	Analytics	Source
PSA	Data-driven	Descriptive	An EHR dataset
PRE	Data-driven	Predictive	An EHR dataset*
CPG	Knowldege-driven	Prescriptive	The NICE clinical guideline for Type 2 diabetes

Whereas, PSA [6] is a data-driven decision engine based on the patient similarity analytics that uses an EHR dataset from one of the largest hospitals in China and its affiliated community centers in order to manage type 2 diabetic patients. For example, given a patient encounter, the descriptive analytics of PSA reports that 75% similar patients take Metformin, while 15% similar patients take Insulin, and the rest take a combination of Insulin with oral anti-diabetic drugs. We employ various feature selection algorithms to identify the factors that affect physicians' prescription decisions. Given a patient encounter, his/her clinical conditions are represented using a vector of selected features, and we would find out the K most similar prescription instances, where the similarity is measured by the Euclidean distance between the representing feature vectors. Finally, PSA outputs a list of frequently presented medication options (among the K most similar prescription instances), and each option is attached with its occurrence percentage.

PRE is also a data-driven decision engine based on the outcome prediction analytics. The engine uses the same EHR dataset used by PSA but different features. For example, given a patient encounter, the predictive analytics of PRE reports that taking Insulin would get good outcome with support degree of 0.78, while taking Metformin would get good outcome with support degree of 0.54. Given a decision option, we collect the instances whose prescription is the same as the given decision option, and label its outcome as good or bad by comparing the next HbA1c test result after treatment with the current one. Thus, instances are grouped by different decision options, and we perform the feature selection (correlation based) and model training (logistic regression) per group separately. Then, given a patient encounter, for each decision option, his/her clinical conditions are represented using a vector of selected features, and we test it with the corresponding trained model. Finally, PRE outputs a list of decision options, and each option is attached with its support degree of good outcome.

Fusion engine

In the book [3], numerous methods for decision fusion have been presented. Based on the output of base decision engine, the fusion types are categorized into two: one is the fusion of label outputs, and the other is the fusion of value outputs. As described earlier, our base decision engines, CPG outputs labels, while PSA and PRE output values. It's easy to transform the CPG outputs as values, by assigning the label (which CPG recommended) with value of 1, and others (which CPG did not recommended) with value of 0, given a

patient at an encounter. Also, we can transform the PSA and PRE outputs as labels, but unavoidably, would have information loss. For instance, we could choose the most frequently presented medication option as the label for recommendation in PSA, and we could choose the decision option with the highest support degree as the label for recommendation in PRE. In this way we only keep the top one recommendation and ignore the lower ranked decision options – such a transformation from values to labels is not desirable. Consequently, our decision fusion framework is unified as a fusion of value outputs (and we will do a transformation from labels to values, if any base engine outputs labels).

Next, we will present a formal definition for our fusion engine. Suppose $E = \{e_1, ..., e_n\}$ be the set of base decision engines, $O = \{o_1, ..., o_m\}$ be the set of decision options, $v_{ij}(x)$ is the value that a decision engine e_j gives to the decision option o_i, for an instance x, $1<=i<=m$ and $1<=j<=n$. The fusion engine will take the following matrix $v(x)$ as an input, for each instance x.

$$v(x) = \begin{pmatrix} v_{1,1}(x) & ... & v_{1,n}(x) \\ ... & v_{i,j}(x) & ... \\ v_{m,1}(x) & ... & v_{m,n}(x) \end{pmatrix}$$

For fusion algorithm, we first apply the approach of decision templates [7]. At training phase, we calculate a decision template DT_i (as defined below) for each decision option o_i, where S_i is the set of training instances whose presecription is the same as the decision option o_i with good outcome, and N_i is the number of S_i. That is, DT_i is the mean of values of all training instances who take the decision option and get good outcome. Here, we highlight that S_i consists of training instances with good outcome, because our fusion is outcome-driven, instead of just learning the physicians' decisions (i.e. prescriptions, which do not always result in good outcomes).

$$DT_i = \frac{1}{N_i} \sum_{x \in S_i} v(x)$$

At testing phase, we calculate the squared Euclidean distance $d_i(x)$ between a decision template DT_i and $v(x)$ for an instance x, where $DT_i(k, j)$ is the $(k, j)^{th}$ entry in DT_i.

$$d_i(x) = \frac{1}{m \times n} \sum_{k=1}^{m} \sum_{j=1}^{n} (DT_i(k, j) - v_{k,j}(x))^2$$

The fusion engine outputs a list of decision options, and each option is attached with its distance value.

Besides decision templates, we also leverage various meta classifiers provided by Weka (a Java libarary for data mining [9]) to implement fusion algorithms. The main idea is to consider the values generated by base decision engines as new features, and feed them into a classifier for classification. Specifically, for each decision option o_i, we learn a model Mi, and the training data consists of instances x whose prescription is the same as the decision option o_i. The feature vector of x is $<v_{i1}(x), ..., v_{in}(x)>$, with label of 1 for good outcome and label of 0 for bad outcome, where $v_{ij}(x)$ is the value that a decision engine e_j gives to the decision option o_i, for 1<=j<=n. Thus, a variety of classifiers such as logistic and SVM (support vector machine) could be used for this meta-classification problem to decide whether the decision option o_i is a recommendation for an instance x.

Evaluation metrics

The motivation of decision fusion is "to do better" than base decision engines. This wish of "to do better" needs some evaluation metrics. As mentioned above, the often used metrics [4] such as precision, recall and RMSE are not applicable, when evaluating treatment recommendation systems, because we have only the partially observed ground truth – i.e., not all decision options are completely observed with outcomes. Formally, we denote $q(x) \in O$ as the prescription of an instance x, i.e., $q(x)$ is one of the decision option in O. Next, we denote $t(x, q(x))$ as the outcome of an instance x taking the prescription $q(x)$, where $t(x, q(x))=1$ means good outcome and $t(x, q(x))=0$ means bad outcome.

Back to the output of engines. All engines ouput values, but the values have different meanings. For example, PSA outputs the percentage of similar patients who takes the same decision option, and PRE outputs the support degree of good outcome which takes the given decision option, while the fusion engine outputs the distance between the decision template and the given decision option. Therefore, a value itself contributes little for evaluation, but a ranked list ordered by values does mean a lot. In particular, a rank score is calculated according to the position of prescription in a ranked list, which avoids to be overlooked when it's lowerly ranked.

For a formal representation, we denote $r_j(x) = <p_1, ..., p_m>$ as the ranked list of an instance x recommended by an engine $e_j \in E \cup \{e_0\}$ where each $p_k \in O$ is a decision option in O, and the subscript k means its position at the ranked list. Here, the base decision engine set E is union of the fusion engine e_0. We note that both PSA and PRE approach gets the recommended ranked list in descending order (because the more similar patients or the more supports, the better for recommendation), while our fusion algorithm of decision templates would get the recommended ranked list in ascending order (because the shorter distance, the better for recommendation).

Next, we denote $g_j(x)$ as the rank score of an instance x, given an engine $e_j \in E \cup \{e_0\}$. The following calculation means that, given a ranked list $r_j(x) = <p_1, ..., p_m>$ as recommended by an engine e_j, for any $0<=j<=n$, the prescription $q(x) = p_k$ is located at the position k, if an instance x taking the prescription $p(x)$ has good outcome, then $g_j(x)=(m-k)/(m-1)$, else $g_j(x)=(k-1)/(m-1)$. In particular, if the ranked list for recommendation does not contain the prescription, then its position is set as $k=m$. We note that, in spite of the partially observed ground truth (i.e. prescription with its outcome in real data), this calculation takes all recommended decision options into account, with information loss as little as possible.

$$g_j(x) = \begin{cases} \frac{m-k}{m-1} & if\ t(x, q(x)) = 1 \\ \frac{k-1}{m-1} & if\ t(x, q(x)) = 0 \end{cases}$$

$$where\ r_j(x) =< p_1, \ldots, p_m > and\ p_k = q(x)$$

The score of an engine e_j is calculated as $g_j = \frac{\sum_x g_j(x)}{\sum_x 1}$ for any $0<=j<=n$. Using this outcome-driven evaluation metric, we can directly compare the score g_0 of our fusion engine e_0 with other scores g_j of base decision engines $e_j \in E$.

Results

Our experimental dataset is an EHR dataset from one of the largest hospitals in China and its affiliated community centers that manage type 2 diabetic patients. After anonymity, it consists of 3150 encounter instances of diabetic patients. Their prescriptions are categoried as 7 types of decision options: METFORMIN (metformin alone), ARFA (either insulin secretagogues or α-glucosidase inhibitors), TZD (either thiazolidinediones or DPP-IV inhibitors), BI (two oral anti-diabetic drugs), TRI (three oral anti-diabetic drugs), INSULIN (insulin alone), and COMBINED (insulin and oral anti-diabetic drugs). For the outcome of glucose control, we use the widely adopted clinical ranges (also cited in [8]): HbA1c <= 6.4: normal; 6.5 <= HbA1c < 7: well controlled; 7 <= HbA1c<9: moderately controlled and 9 <= HbA1c: poorly controlled. Each patient's prescription outcome is labeled by comparing the next HbA1c test result after prescription with the current one. The outcome is labeled 1 (good) if the HbA1c level moves into a lower range, or remains in the well-controlled range, otherwise it is labeled as 0 (bad). In our dataset, there are 2574 instances of 3150 as labeled 1.

Given an instance, different base decision engines use different feature vectors as the input. Specially, CPG has 19 features, including overweight (BMI>24), old (age>70), etc. By feature selection, PSA identifies 39 features, such as average of glucose, value of HbA1c, and maximum value of HbA1c, etc. It's interesting that CPG considers the age status (e.g. old if age>70), while PSA considers the age (e.g. age of 72) and the age at the earliest diabetes diagnosis (e.g. age of 45). A more complicated case is PRE, which has 7 learning models for the 7 types of decision options. For example, the METFORMIN model in PRE has 13 features, while the ARFA model in PRE has 8 features.

In spite of different input features, the base decision engines have a uniform output format, i.e. v_{ij}, the value that a decision engine e_j gives to the decision option o_i, for an instance x, where e_1=CPG, e_2=PSA, e_3=PRE, and o_1=METFORMIN, o_2=ARFA, o_3=TZD, o_4=BI, o_5=TRI, o_6=INSULIN, o_7=COMBINED.

Regarding the fusion engine as e_0, it also follows the unified output format, and below is the sample output for an instance x. According to the real data, this instance x is prescribed o_2=ARFA, as having good outcome.

	CPG	*PSA*	*PRE*		*FUSION*
*o*1	1	0.4	0.6068		0.4286
*o*2	0	0.31	0.6943		0.2772
*o*3	0	0.01	0.7489		0.3196
*o*4	0	0.18	0.6580	⇒	0.5817
*o*5	0	0	0.5761		0.6454
*o*6	0	0.09	0.6674		0.4961
*o*7	0	0.01	0.5939		0.6164

The ranked list of x recommended by CPG is $r_1(x)=<o_1>$, PSA is $r_2(x)=<o_1,o_2,o_4,o_6,o_7,o_3,o_5>$ in descending order, PRE is $r_3(x)=<o_3,o_2,o_6,o_4,o_1,o_7,o_5>$ in descending order, and after fusion using algorithm of decision templates, it is $r_0(x)=<o_2,o_3,o_1,o_6,o_4,o_7,o_5>$ in ascending order.

The rank score of x given by CPG is $g_1(x)=(7-7)/(7-1)=0$, because $r_1(x)=<o_1>$ does not contain o_2, and its position is set as $k=m$, where $m = 7$ is the number of decision options. It makes sense, since CPG does not recommend o_2 which however results in good outcome in real data, and such a CPG recommendation gets the score of 0. For PSA, $g_2(x)=(7-2)/(7-1)=5/6$, because the position of o_2 in $r_2(x)$ is $k=2$ and $m=7$. It also makes sense, since o_1 is preferred by PSA than o_2, and such a PSA recommendation cannot get the full score of 1, but only 5/6. Similarly for PRE, $g_3(x)=(7-2)/(7-1)=5/6$. Here, we point out that, although PSA prefers o_1 while PRE prefers o_3 for the top one recommendation, the prescription o_2 is located at the same position $k=2$ in both PSA and PRE, so the rank scores $g_2(x)$ and $g_3(x)$ are the same. Finally, for fusion, $g_0(x)=(7-1)/(7-1)=1$, because the position of o_2 in $r_0(x)$ is $k=1$ and $m=7$. It matches the practice that such a fusion recommendation is appreciated.

Fig. 2 illustrates these treatment recommendation results, which could be integrated to the clinician workstation or mobile apps, and end users could have an overview about the engines' opinions in a comprehensive way.

Treatment Recommendation

Rank	CPG	PSA	PRE	Fusion
1	METFORMIN: 1.0	METFORMIN: 04	TZD: 0.7489	**ARFA**: 0.2772
2		ARFA: 0.31	ARFA: 0.6943	TZD: 0.3196
3		BI: 0.18	INSULIN: 0.6674	METFORMIN: 0.4286
4		INSULIN: 0.09	BI: 0.6580	INSULIN: 0.4961
5		COMBINED: 0.01	METFORMIN: 0.6068	BI: 0.5817
6		TZD: 0.01	COMBINED: 0.5939	COMBINED: 0.6164
7		TRI: 0.0	TRI: 0.5761	TRI: 0.6454

Figure 2 – Treatment recommendation results

To compare the evaluation results, we do 10-fold cross validation. That is, the total of 3150 instances is randomly partitioned into 10 equal size subsamples. Of the 10 subsamples, a single subsample (of 315 instances) is retained as the validation data for testing, and the remaining 9 subsamples (of 2835 instances) are used as training data.

Table 2 – Evaluation results

Treatment recommendation systems		Rank score
Decision engine	CPG	0.6771
	PSA	0.6815
	PRE	0.6062
Fusion engine	Decision templates	**0.6917**
	Logisic classifier	0.6826
	SVM classifier	0.6773
	SVM classifier –s 3	0.6823
	Naïve Bayes	0.6776
	Naïve Bayes -K	0.6824

As shown in Table 2, among base decision engines, PSA has a higher rank score (0.6815) than CPG (0.6771) and PRE (0.6062). After fusion, the engine of decision templates outforms all of the base decision engines, getting the highest rank score (0.6917). Also, the logistic classifier is promising and gets the rank score of 0.6826. Although we observe that

fusion engines using the default SVM classifier and Naïve Bayes have the rank scores lower than the base decision engine PSA, it's not a big drop since using machine learning algorithms always require careful tuning. Actually, we set an option "–K" for Naïve Bayes, which indicates to use kernel density estimator rather than normal distribution for numeric attributes, the rank score improves from 0.6776 to 0.6824. Similarly, we set an option "–s 3" for the SVM classifier, which indicates to use the SVM type of Epsilon-SVR rather than C-SVC, the rank score improves from 0.6773 to 0.6823.

Discussion

Fusion is not a new topic in machine learning and data mining, and its great success has been shown in a series of KDD Cup competitions from 2007 to 2014, as well as other open competitions such as the Neflix Grand Prize in 2009 [1] and the Heritage Health Prize in 2012 [2]. However, related work mainly focuses on the fusion of different learning algorithms with different parameter settings, and pays little attention to the knowledge sources. Furthermore, the evaluation metrics used in prior arts are based on the ground truth, but we have only the partially observed ground truth in treatment recommendation. To address these problems, we have two contributions as presented in this paper. First is a decision fusion framework for treatment recommendation systems, which combines both knowledge-driven and data-driven decision engines. Second is an outcome-driven evaluation metric, which has no information loss while facing the partially observed ground truth. As for experimental results, the fusion engine gets better performance than base decision engines.

Also, we realize our limitations. First, we assume that base decision engines output a uniform format. However, this assumption is challenged, if the labels (from different decision engines) are heterogenous. For example, suppose CPG just recommends using one of oral anti-diabetic drugs, but does not specify which one. Meanwhile, the decision options of PSA are still the 7 types: METFORMIN, ARFA, TZD, BI, TRI, INSULIN, and COMBINED. Now, one of oral anti-diabetic drugs could be METFORMIN, ARFA or TZD, how to assign values for such CPG recommendation? In our current work, CPG outputs label all the 7 types, and we do a simple transformation from these labels to the value of 1 or 0. Actually, we observe that normalization of heterogenous labels with values has not been mentioned or well addressed in previous work, because previous work is about multiple classifier systems, whose base engines are all classifiers to be trained for the homegenous labels. However, in our framework, we take both knowledge-driven and data-driven modules into account, which are heterogenous in nature. This situation would become more common, when facing the cloud-based decision services. We cannot assume each cloud-based decision service outputs a uniform format, but we still want to leverage the fusion of those services towards a better final decision. Therefore, the normalization for fusion would be our ongoing work.

Besides, in this paper, we only use three base decision engines of CPG, PSA and PRE for fusion. This is not enough, and we are planning to inclue more base decision engines involved towards an open (e.g. cloud-based) fusion platform. Moreover, we observe there is some contraindication information in the domain knowledge for treatment recommendation, e.g. you should not use statins for a woman becoming pregnant. Such contraindication information is hard to discover for learning algorithms, but it can be easily represented by knowledge modeling. Our future work will develop some novel fusion algorithm to get more benefits from both knowledge and data.

Last but not least, the so-called "partialness" of the ground truth deserves more investigation. For intervention in diabetes, we could regard the HbA1c value as partially observed, however, for a more nebulous topic, like antibiotics for fever, the partially observed ground truth may actually never have a known answer.

References

[1] The Netflix Prize. http://www.netflixprize.com/

[2] The Heritage Health Prize. https://www.heritagehealthprize.com/c/hhp

[3] Ludmila I. Kuncheva. Combining Pattern Classifiers: Methods and Algorithms. Wiley-Interscience, 2004.

[4] Asela Gunawardana, Guy Shani. A Survey of Accuracy Evaluation Metrics of Recommendation Tasks. Journal of Machine Learning Research, 2009, pp. 2935-2962.

[5] Haifeng Liu, Jing Mei, Guotong Xie. Towards Collaborative Chronic Care Using a Clinical Guideline-Based Decision Support System, Proceedings of the 24th European Federation for Medical Informatics, MIE 2012, pp. 492-496.

[6] Haifeng Liu, Guo Tong Xie, Jing Mei, Weijia Shen, Wen Sun, Xiang Li. An Efficacy Driven Approach for Medication Recommendation in Type 2 Diabetes Treatment Using Data Mining Techniques, Proceedings of the 14th World Congress on Medical and Health Informatics, MedInfo 2013, pp. 1071.

[7] Ludmila I. Kuncheva, James C. Bezdek, Robert P. W. Duin. Decision templates for multiple classifier fusion: an experimental comparison. Pattern Recognition, 2001, pp. 299-314.

[8] Hani Neuvirth, Michal Ozery-Flato, Jianying Hu, Jonathan Laserson, Martin S. Kohn, Shahram Ebadollahi, Michal Rosen-Zvi. Toward Personalized Care Management of Patients at Risk: The Diabetes Case Study. Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2011, pp. 395-403.

[9] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, Ian H. Witten. The WEKA Data Mining Software: An Update. SIGKDD Explorations, 2009, Volume 11, Issue 1.