

Predicting Prescription Patterns

By

Ívar S. Helgason
M.D. University of Iceland 1999

ARCHIVES

Submitted to the Division of Health Science and Technology
in Partial Fulfillment of the Requirements for the Degree of

Master of Science in Medical Informatics
at the
Massachusetts Institute of Technology

[June 2008]
May 2008

©2008 Ívar S. Helgason

All rights reserved

The author hereby grants to MIT permission to reproduce and to distribute publicly paper and electronic copies of this thesis document in whole or in part in any medium now known or hereafter created.

Signature of author: _____

Harvard-MIT Division of Health Sciences and Technology

Certified by: _____

Staal A. Vinterbo, Ph.D.
Assistant Professor
Harvard Medical School
Thesis Supervisor

Accepted by:

Martha L. Gray, Ph.D.
Edward Hood Taplin Professor of Medical and Electrical Engineering
Director, Harvard-MIT Division of Health Sciences and Technology

Predicting Prescription Patterns

by

Ívar S. Helgason

Submitted to the Division of Health Science and Technology
May 2008 in Partial Fulfillment of the Requirements
for the Degree of Master of Science in Medical Informatics

Abstract

Electronic prescription software is replacing traditional handwritten medication orders. This development however doesn't come without a cost and speed has been one of the most complained about issues. It is important to address this problem and develop methods to reduce the time spent entering medication orders into computerized prescription software.

The objective of this study was to understand the structure of prescription patterns and explore the possibility of designing a method that will predict prescription patterns with only the knowledge of past prescription history. Various machine-learning methods were used and their performance measured by the accuracy of prediction as well as their ability to produce desirable results, within practical time limits.

This paper presents a method to transform prescription data into a stochastic time series for prediction. The paper also presents a new nonlinear local algorithm based on nearest neighbor search.

In analyzing the database the drug patterns were found to be diverse and over 30% of the patients were unique, in the sense that no other patient had been prescribed the same set of active ingredients. In spite of this diversity, it was possible to create a list of 20 drugs that contained the drug to be prescribed next for 70.2% of patients. This suggests that probabilistically created pick lists, tailored specifically for one patient at the time of prescription, might be used to ease the prescription process. However, further research is needed to evaluate the impact of such lists on prescription habits.

Thesis Supervisor: Staal A. Vinterbo Ph.D.

Title: Assistant Professor, Harvard Medical School.

Table of Contents

Abstract	2
Introduction	5
Background	5
Objectives	10
Significance.....	11
Scope of the study	12
Methods	12
Data	12
Software used.....	15
Preprocessing	16
Patient clusters	17
Choosing learning algorithms	18
Baseline.....	18
K-nearest neighbor baseline algorithm	19
Markov chain	20
Improvements on basic algorithms	22
Longest Common Subsequence	23
Weighted frequency based sequence (WFBS).....	24
Support Vector Machines	28
Sensitivity analysis for WFBS algorithm	29
Experimental protocol.....	30
Results	31
K-nearest neighbor (KNN)	31
1 st order Markov chain	32
Longest common subsequence (LCS)	33
Weighted frequency based sequence (WFBS).....	34
Grouping drugs by function	35
Sensitivity analysis for WFBS algorithm	35
Support Vector Machines	38
Discussion	38

Limitations of the study	42
Conclusions	42
References	43

Introduction

Background

Computerized physicians order entry systems

The 98,000 deaths and the numerous injuries believed to occur annually from medical errors have made patient safety a priority in health care ⁽¹⁾. Medication errors are the most common type of preventable medical errors ⁽²⁾ and can be largely prevented with computerized physician order entry systems ⁽³⁾. Despite the effectiveness of computerized physician order entry systems (CPOE) to prevent serious medication errors, only 10 to 15 percent of hospitals have implemented such systems ⁽⁴⁾. Medication management software can be deployed as a standalone system or as a part of a bigger computerized physicians order entry system.

A recent review of the impact of computerized order entry systems, show on average a positive effect over a variety of outcome measurements, including cost, medication safety and organizational efficiency. However, the speed of entry has been negatively affected, especially with regard to the physician's time ⁽⁵⁾.

Computerized prescriptions initially promised to increase patient safety and streamline the prescription process. When the time comes to prescribe a new drug through a computerized prescription system it can be assumed that the care provider has already decided which drug to prescribe. Even so, the prescriber still has to search for the drug, typically receiving the search

results as a long list in alphabetical order. When compared to traditional paper based methods a two- to fourfold increase in time spent prescribing medications for inpatients has been observed ⁽⁶⁾⁽⁷⁾⁽⁸⁾. However, considering that medication management is a substantial part of the physician's workload, even a small increase in time can have big overall impact ⁽⁵⁾.

To combat these problems, various methods have been devised. Some of these methods include:

- Using prescription templates (order sets) which facilitate the prescription of the most frequently used drugs for common conditions. An order set is a static collection of medications that are appropriate in a given clinical situation. The use of order sets have been implied as a key factor in achieving acceptance of computerized order entry, especially among physicians ⁽⁹⁾⁽¹⁰⁾.
- Creating a “favorites list” for providers; where each user of a medication management system can select from a list of favorite medications offered as a quick selection or to simplify the search. A favorites list is essentially a personalized order set.
- Creating a list for ward specific medications which is sometimes based on the specialty of the primary attendant.
- A natural language-base command line CPOE interface has been shown to efficiently decrease input time, but the same study implies that order sets are superior with regard to speed ⁽¹¹⁾.

Simple methods such as using a favorites list can be effective, but they tend to scale poorly and present a significant maintenance problem ⁽¹²⁾. Personalized order sets seem to exhibit much variability between users, but are nonetheless popular among end users since they save time ⁽¹³⁾ ⁽¹⁴⁾. Other solutions use centrally maintained prescription templates (order sets) to aid in medication prescription, with the goal of both implementing best practice guidelines and to simplify the prescription process. Such order sets, tailored for specific clinical situations, can be viewed as a form of ruled based logic. The downside of centrally maintained order sets seems to be that they are rarely used and difficult to maintain ⁽¹²⁾.

Prescription templates are most effective in the initiation of therapy, but are not as effective in representing the many possible changes during the course of treatment, such as management of side effects or intolerance to treatment.

New methods are needed to assist in the prescription process, especially during drug selection when care plans and order sets fail to represent the course of treatment adequately or are absent.

Time series analyses

Prescription patterns can be viewed as a stochastic time series. Time series forecasting is a challenge in many fields. In finance, time series analysis is used to forecast stock and currency prizes ⁽¹⁵⁾. Producers of electricity and other forms of power, use time series predictions to predict the load of the following day.

So how can one analyze and use the past to predict the future? In 1927, Yule⁽¹⁶⁾ introduced Auto regressive models (AR) to predict sunspots. The term “autoregressive” is used to emphasize that future points of time series are modeled by past values of the same time series.

Technically an AR model is a linear map that maps a given number of past values into the future value. The parameters of the map are chosen to minimize the prediction error inside the given time series. There exists a vast literature about the use and optimization of AR models⁽¹⁷⁾⁽¹⁸⁾⁽¹⁹⁾⁽²⁰⁾⁽²¹⁾ in particular ARMA models that assume that the time series is a linear stochastic process. The whole class of AR models is not considered in this study since drug prescriptions cannot be viewed as a linear process.

Nonlinear techniques of time series analyses and prediction have been developed in addition to the linear approach. A distinction is made between global and local methods. Global methods start with a predetermined mapping, whose parameters are adapted to reproduce the time series as well as possible. Neural networks are an example of a global method.⁽²²⁾⁽²³⁾⁽²⁴⁾ The time complexity of neural networks training is relatively high compared with other popular methods and dependent on the model used. Initial experiments with neural networks on the data presented in this study had an unacceptable time complexity and were not pursued further. Support Vector Machines are another example of nonlinear global approaches and their use for prescription prediction is explored in this study.

The class of local nonlinear approaches was introduced by Lorenz⁽²⁵⁾ for weather forecasting. This class of prediction models is based on neighbor searches. Numerous introductions to next

neighbor predictions exists ^{(26) (27) (28) (29) (30)}. These methods have been applied to many non-artificial applications such as prediction of water marks of rivers after rainfalls ⁽³¹⁾.

The feature extraction and transformation of the prescription data described later in this manuscript, enables us to view prescription history as a discrete stochastic time series. Such time series are seen in many real world applications. Customer preference analysis is a form of discrete time series analysis. With customer preference analysis, your history of past purchases, are compared to a database of other customers to create a prediction of your future purchases. The recommendation systems that suggest books on Amazon, movies at Netflix and news at Google News are all examples of such systems. These kinds of algorithms are often referred to as collaborative or social filtering because they use the preferences of others, like minded people to filter and prioritize your experience ⁽³²⁾. There are three common approaches to solving the recommendation problem; traditional collaborative filtering, cluster models and search based models. Recently item-to-item collaborative filtering has gained attention for its ability to scale for large datasets ⁽³³⁾. Most recommendation algorithms start by finding a set of customers whose purchased and rated items overlap the user's purchased and rated items. The algorithm aggregates items from these similar customers, eliminating items the user has already purchased, and recommends the remaining items to the user ⁽³⁴⁾. The item-to-item algorithm focuses on finding similar items rather than customers, then aggregates the similar items and recommends them ⁽³³⁾. Collaborative filtering is in many aspects related to prescription predictions.

Objectives

The main goal of this study is to explore the feasibility of developing a method to predict drug selection based on probabilistic reasoning using only the prescription history and simple demographic data such as age and gender. The method has to be implementable in the sense that it can perform predictions in real time and performs adequately on large databases. The method should also readily incorporate new data, and adapt to new prescription patterns and drugs. It should not be necessary to retrain the whole system, each time a new drug is introduced to the market. This task can be viewed 1) as a classification problem, which for the data at hand for this study has at least 1000 input features and a 1000 classes, or 2) as a discrete time series analysis. For these reasons, traditional classification methods such as Support Vector Machines and Artificial Neural Networks are infeasible due to computational complexity and the need for retraining. Methods that require much more than linear running time are not considered due to large amount of data. Even so, the use of Support Vector Machines and neural networks were explored, but the use of neural networks had to be abandoned due to high computational space and time requirements.

The second goal is to understand the structure and consistency of prescription patterns to be better able to assess the expected accuracy of prediction algorithms.

Realistically, the output from any method to predict prescription patterns is a list of possible drugs, ordered by likelihood, rather than one specific drug. The quality of the prediction is proportional to how long the list has to be to contain the drug being predicted.

The more accurate the prediction algorithm is, the shorter the resulting list can become without missing the sought after drug. For this study, the goal was to develop an algorithm that can predict any drug for any patient with a list of no more than 20 drugs in at least 50% of cases. Being able to prepare a relatively short list that contains the drug to be prescribed next, every other time a new drug is entered into a CPOE system, was deemed a goal worth pursuing.

Significance

Methods aimed at decreasing the time burden of CPOE systems have been identified as key factors in their acceptance by physicians ⁽¹⁰⁾(9). Computerized medication management systems are complex and tend to be cumbersome in usage. Providing on-time and accurate support for prescribers can significantly reduce time spent prescribing medications, and thus increase user acceptance of such systems. Using probabilistic methods has some advantages, such as providing the prescriber with some sense of frequency of the actions he is taking. Practicing within the probabilistic recommendations of the system provides an assurance that the treatment being offered is not an outlier. If the prescriber's next action is not predicted by the system, the prescriber has to fall back on more traditional drug selection methods, and is given a chance to reconsider the drug selection.

The primary purpose of methods such as those proposed by this paper are not to directly influence the drug selection, but rather to speed up the documentation process.

Scope of the study

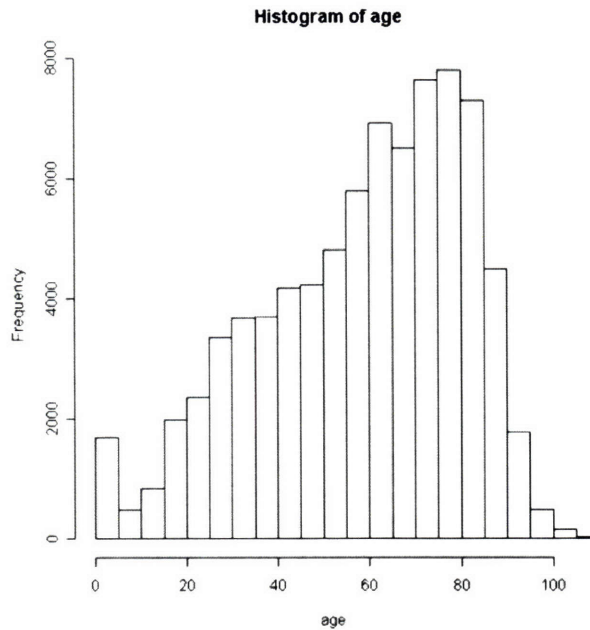
This study is limited to developing algorithms to predict prescription patterns, based only on the data commonly available in medication databases. Various algorithms were explored to develop a method to predict prescription patterns. A database of more than a million prescriptions for 75,000 admissions was used to train and test the prediction algorithms. The database contained only information about the medications, wards, rudimentary demographic information, and the primary specialty of the prescriber.

Methods

Data

The database used was from Region Nordjylland in Denmark, which covers approximately 10% of Denmark with a population of about 600,000. Electronic prescriptions have been used in Region Nordjylland since 2004. Implementation is planned to be finished in all hospitals in Q2 2008. The data used is only from hospitals (including outpatient wards).

Before preprocessing, the database contained 1,059,750 prescriptions among 80,177 admissions.



**Fig 1: Histogram of the age of patients at the time of admission.
The mean age of the patients was 58.7 years.**

The mean age of patients at the time of admission was 58.7 years (Figure 1).

Gender composition: 21,940 males and 27,018 females for a total of 48,958 patients. Of those, 33,657 patients had only one admission and 15,301 patients had more than one admission. Of the patients with more than one admission, 185 had been admitted more than 10 times.

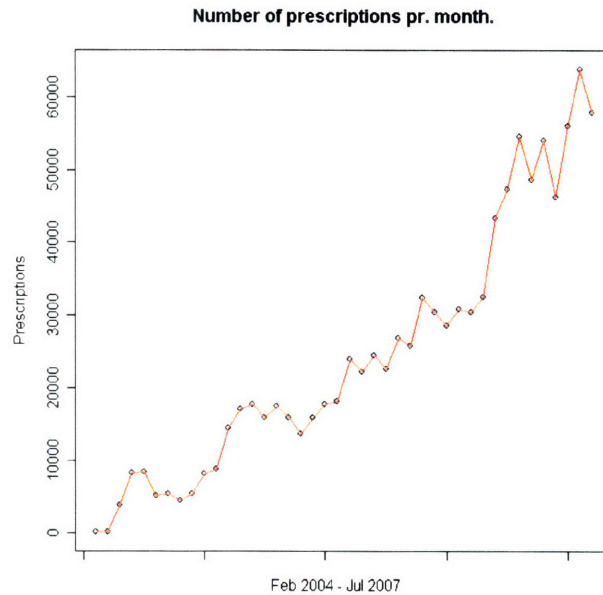


Fig 2: Electronic prescriptions were started in February 2004. At the time the database was acquired for this study, the number of prescriptions was 60,000 per month.

The medication management program in use in Region Nordjylland is called Theriak Medication Management (TMM), provided by CIS Healthcare (www.cis-healthcare.com). Implementation started in February 2004 and continued through the time the database was acquired for this study in July 2007. At the time the snapshot was taken, the number of prescriptions per month was approximately 60,000 (Figure 2). Number of drugs per patient, over the course of one admission, can be seen in Figure 2. There are approximately 8,000 active prescribers on 187 wards.

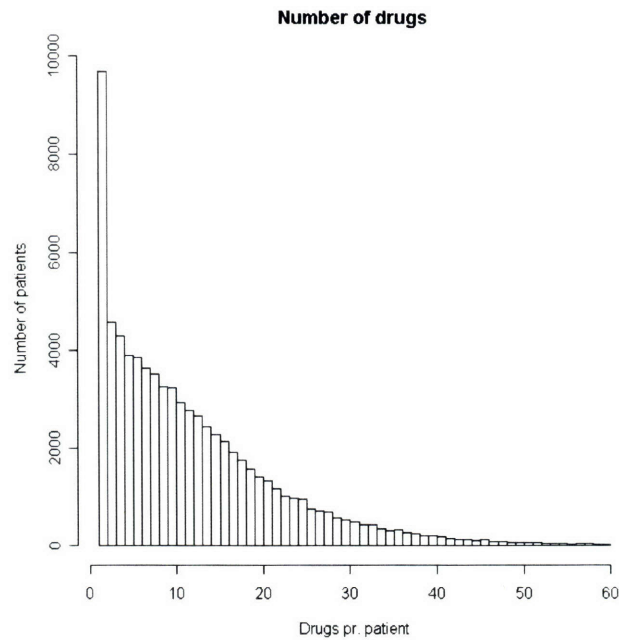


Fig 3: The number of drugs within each admission. Patients with more than 60 drugs within a single admission were filtered out of the training population.

Software used

All algorithms, except Support Vector Machines were implemented in Python v2.5.1 (<http://www.python.org/>). NumPy v1.0.4 (<http://numpy.scipy.org/>) was used to speed up array and matrix operations. The database used was Microsoft SQL Server 2005. Support Vector Machines, statistical analysis and graphics was done using R v2.5.1 (<http://www.r-project.org/>)

Preprocessing

Raw prescription data is stochastic in nature and contains an abundance of redundant information. During the course of treatment, doses and frequency of drugs get adjusted. By definition, every such change results in a new prescription. Since changing an existing prescription or restarting a temporarily halted one does not include a step where the prescriber has to search for the drug to change, such prescriptions are of no importance to us. Therefore subsequent prescriptions involving a previously prescribed drug, within one admission, were ignored.

For each patient, all medications were reduced to the active ingredients or blends thereof using the Anatomical Therapeutic Chemical (ATC) classification system⁽³⁵⁾, essentially reducing the list of prescribed drugs from 5420 brand names + strength to 978 active ingredients or unique blends of active ingredients. The day since admission was calculated for first prescription of a drug within each admission. For each admission the feature vector is thus a set of active ingredient or blends thereof, in the order they were first prescribed. This way the prescription history can be viewed as a discrete stochastic time series.

Outliers were removed by omitting patients with more than 60 drugs prescribed per admission and patients with admission longer than 60 days. Care was taken to remove all test patients and teaching patients. The distribution of the number of drugs per admission can be seen in Figure 3.

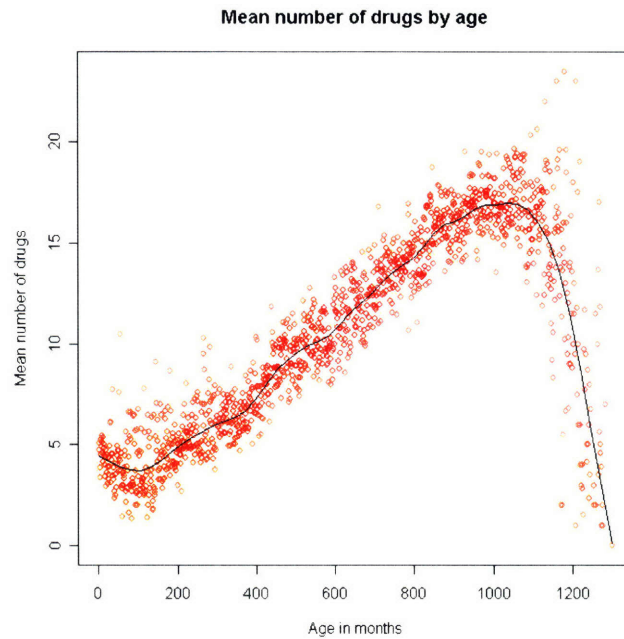


Fig 4: Mean number of drugs per admission viewed by age in months.

Patient clusters

The predictive power of the data can be viewed as a function of the variance of the prescription patterns. The simplest way to measure this is to examine clusters of medication combinations. The number of neighbors, where the neighborhood is defined by the similarity of drugs, has an inverse relationship to the uniqueness of the combination of medicines for each patient. It is interesting to see how variable the drug patterns are. 31.2% of admissions had a unique drug set in the sense that no other patient had been prescribed the exact same set of active ingredients. Examining the neighbors of patients X by defining a neighbor to be any patient with at least 80% of their drugs in the set of drugs of patient X, shows that most patients have a neighborhood of patients that is of size 0.1% - 2% of the total population.

Choosing learning algorithms

The main concern when choosing learning algorithms was the time and space complexity caused by the number of inputs and multiple classes. As an example, a method such as artificial neural networks failed to be applicable due to space and time complexity required when applied to very large data sets. No literature was found that deals specifically with the prediction of prescription patterns. This suggests that investigations into this are warranted. Unfortunately this also means that there are no established standards to compare methods against.

Other methods, such as K-nearest neighbors and Markov chains were chosen for their well known characteristics and ease of use. Both methods have been use extensively in the domain of financial forecasting and natural language processing⁽³⁶⁾.

Support Vector Machines (SVM) have received much attention recently in the domain of time series predictions. SVMs have for example, successfully been applied in the art of spam filtering⁽³⁷⁾, which is a classification problem with a high dimensional feature space and sparse data for individual sample, not unlike prescription patterns. Spam filtering has, unlike prescription patterns, a binary outcome that is well adapted for SVMs. This study on the other hand deals with a high dimensional feature space and unusually many classes. Not much literature exists on the use of Support Vector Machines for this domain.

Baseline

To create baseline prediction algorithms to be used as benchmarks for development, two methods were used: K-nearest neighbors and a 1st order Markov chain. The feature vectors were kept as

simple as possible to better understand the impact of each feature as it was added to the algorithms. Here these methods are used, primarily to gain insight into the structure and nature of prescription patterns. Therefore K-nearest neighbor was chosen to explore the importance of patient clustering and Markov chain to measure the effects of sequential ordering within prescriptions.

K-nearest neighbor baseline algorithm

K-nearest neighbor (Knn) is a method for classifying objects based on closest training examples in feature space. Knn is a type of instance based learning, or lazy learning function, where computation is deferred until classification ⁽³⁸⁾. It is among the simplest of machine learning algorithms. The key idea is that the properties of any given input point are likely to be similar to those points in its neighborhood.

The definition of the neighborhood of X requires a distance measure $D(X, Y)$. For discrete unordered features, such as medications, the Hamming distance, which defines $D(X, Y)$ as the number of features on which X and Y differ, can be used. K-nearest neighbor is computationally intensive, since for each patient in $X = (x_1, x_2 \dots x_n)$ to be classified, the distances to every other patient $Y = (y_1, y_2 \dots y_m)$ has to be calculated. Therefore the classification time in Knn algorithm is proportional to the number of features and the number of instances.

For binary feature vectors, the Hamming distance is:

$$X_n = (x_{n1}, x_{n2}, \dots, x_{nm}) \quad x_{n1\dots nm} \in \{1,0\}$$

$$D(X_1, X_2) = \sum_{i=0}^m x_{1i} \text{ XOR } x_{2i}$$

To predict the next drug for a patient X, the K nearest neighbor's of X are found. We denote the set containing the K nearest neighbors of X as $n(X, K)$. This neighborhood consists of K points for which there are at most K-1 points that lie closer to X than the point furthest away among those K. A list of the neighbors' drugs L is created and sorted in decreasing order by how many neighbors have the drug in their drug list. Drugs already prescribed for x are omitted from L. Finally, if the drug to be predicted for x is found in L, its location is reported.

Markov chain

The second algorithm used to create a baseline prediction is a 1st order Markov process. The assumption made is that each individual drug at times t-1 is an independent predictor of drugs at time t⁽³⁶⁾. This is intuitively plausible in some instances. To name a few examples, some drugs are routinely prescribed together, and other drugs have a high incidence of complications that in turn are treated with drugs. In most cases however, it is the combined state of the patient that dictates the next step in treatment.

A process $X(t)$ is called a 1st order Markov chain, if for $t_1 < t_2 < \dots < t_n$, the distribution of $X(t_n)$ given $(X(t_1), \dots, X(t_{n-1}))$ is the same as the distribution of $X(t_n)$ given $X(t_{n-1})$ ⁽³⁶⁾. This implies that

$$P \{X(t_n) \leq x | X(t_{n-1}, \dots, X(t_1))\} = P \{X(t_n) \leq x | X(t_{n-1})\}$$

If the set of all drugs being considered are $Q = (q_1 \dots q_n)$, then the 1st order Markov chain is described with a n-dimensional matrix M , where n is the number of drugs considered and the probability for drug q_2 to be prescribed after q_1 is q_{21}

$$P(q_2 | q_1) = q_{21}$$

$$M = \begin{bmatrix} q_{11} & \dots & q_{n1} \\ \vdots & \ddots & \vdots \\ q_{1n} & \dots & q_{nn} \end{bmatrix}$$

Changing the algorithm to take into account all drugs at times $t_0 \dots t_{n-1}$ and calculating the relative likelihood of drug q_x being the next drug in drug sequence L , $R(q_x | L)$, where L is a sequence of m drugs, $L = (i_1 \dots i_m)$, ($L \subseteq Q$) as the product of $P(q_x | L)$ gave better results.

$$R(q_x | L) = \prod_{i \in L} P(q_x | i)$$

The most likely drug to be next in the sequence L is therefore

$$\max_{q \in Q}(R(q|L))$$

where Q are all drugs considered.

Improvements on basic algorithms

Initial experiments failed to yield results that met the stated goal of producing predictions that could be used to produce a short list containing the sought after drug. Adding features such as gender ward or specialty of the prescribing doctor, to the feature vectors of KNN and the Markov method did little to increase the accuracy. It is also complicated to weight features such as the ward the patient is being treated in. Scaling of distance measures also became a problem. When features such as wards are treated as binary features, the computational complexity increases substantially. However, changing the distance measurement did help.

Using the Hamming distance for binary features had one significant drawback. It treated additional drugs in the training sample the same as additional drugs in the test sample. In fact, it increased the distance, between similar patients, if one is farther along in treatment. This behavior is counterintuitive for our purpose. If we want to learn what the most likely drug to be added to any given patient's regimen, it stands to reason that the most valuable lessons can be learned from patients who have had similar drugs at a given point in their treatment and then received additional drugs. These additional drugs are the one of interest. The Hamming distance treats those additional, information rich drugs, the same as mismatches between patients. For example, patient A has drugs (1, 2, 3, 4) and we want to predict the next drug. Consider patient B that has

drugs (1, 2, 3, 5). The Hamming distance would assign a distance of 2 between A and B since they have 3 drugs in common and do not share 2 drugs. Now consider patient C with drugs (1, 2, 3, 4, 6, 7). The Hamming distance between A and C is also 2. However it seems likely that patient C has more predictive value for patient A than B has. As a rule of thumb, we can learn most from patients with more drugs than the one we are trying to predict the next drug for.

Longest Common Subsequence

The Markov chain, in its simplest form, contains information solely based on the order in which drugs are prescribed. Even if the Markov chain is not accurate enough as a useful predictor by itself, the results hint at the importance of the order of drugs within each patient. To capitalize on this information and to overcome the limitations imposed by the Hamming distance, a distance measurement based on the longest common subsequence (LCS) could potentially increase the accuracy of K-nearest neighbors.

The longest common subsequence is NP-hard for the general case of arbitrary number of sequences. If the numbers of sequences are two, the problem is solvable in polynomial time.

Given the sequences $X_{1\dots i}$ and $Y_{1\dots y}$

we define

$$LCS(X_{1\dots i}, Y_{1\dots y}) = \begin{cases} \emptyset, & i = 0 \text{ or } j = 0 \\ LCS(X_{1\dots i-1}, Y_{1\dots j-1}) + x_i, & X_i = Y_j \\ \text{MAX} \left(LCS(X_{1\dots i}, Y_{1\dots j-1}), LCS(X_{1\dots i-1}, Y_{1\dots j}) \right) & \end{cases}$$

where MAX returns the longest sequence.

For K-nearest neighbor, the LCS algorithm becomes a similarity measure or can also be thought of as an inverse distance measurement. The sum of the length of all subsequences found between two patients was used as the similarity measurement for K-nearest neighbor.

Weighted frequency based sequence (WFBS)

The LCS algorithm treats all drugs the same, regardless of how near in time they are to the drugs that are similar between the patients. A better approach would be to locate similarities (drugs in common) between patients, and then pay more attention to the drugs that are prescribed subsequent to the similarity. I therefore created an algorithm that takes into account the degree of similarity between two drug lists and the sequential distance of drugs from last location of similarity. The algorithm assigns a weight to every drug that is a product of the degree of similarity and the sequential distance from the similarity. The relationship between both the degree of similarity and sequential distance, to the weight can be linear or exponential.

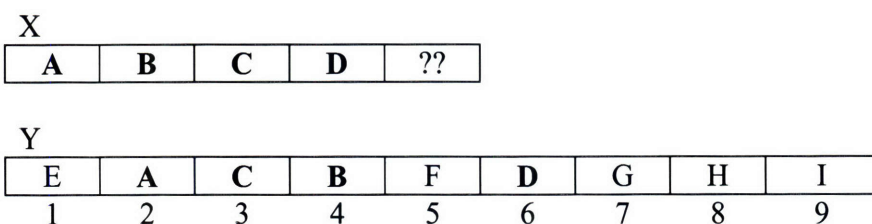


Fig 5:

Consider Figure 5 above. Patient X is the patient to be predicted for. A is already taking four drugs A, B, C and D. The fifth drug, marked as “??” is the one to be predicted. To create a list of

candidate drugs, patient X is in turn compared to all patients in a chosen set. Consider the case where patient X is currently being compared to patient Y. Patient Y is taking nine drugs (or possible was taking 9 drugs since she could be discharged long ago). As can be seen in Figure 5, X and Y have four drugs in common (bold type in Figure 5), note that all drugs in X are also in Y, making Y an ideal candidate for prediction. In patient Y, drugs G is of highest interest and F, H and I to a lesser degree. Drug E is on the other hand of no interest since it is prescribed before the region of similarity and is therefore unlikely to be the drug we are looking for. The WFBS algorithm compares every drug in Y sequentially to the set of drugs in X, beginning with drug 1 in Y which is E and up to I. It assigns a weight to every drug in Y, that is not in X if it is found after some region of similarity. The greater the similarity (more drugs in common), the greater the weight, and like vice the closer to the similarity (subsequent from the last common drug) the greater the weight. Following is a formal description of the WFBS algorithm.

We want to predict drug x_{n+1} for patient X having the set of drugs $\{x_1, x_2, \dots, x_n\}$ the first step is to retrieve all patients $H(X)$ that meet the criteria we use to limit the search space. In our case, the retrieval step might include filtering on age, gender or ward, for example. Then for every patient $Y \in H(X)$ having the set of drugs $\{y_1, y_2, \dots, y_m\}$, where drug y_i is prescribed before y_{i+1} we calculate the weight W_{y_i} for every drug in Y sequentially, beginning at drug $i=1$ and continue until drug $i=m$. The weight W_{y_i} is the product of factors A_i and B_i and is an indicator of drug y_i being the drug to be predicted for X. A represents the intersection of X and Y at drug i . B represents the sequential distance between drug y_i and the last found region of similarity between X and Y. For drug y_i :

$$\begin{aligned}
Y_i &= \{y_1, y_2, \dots, y_{i-1}\} \\
S_i &= X \cap Y_i \\
lm &= \min j \leq i \rightarrow S_j = S_i \\
K &= i - lm
\end{aligned}$$

$$A_i = \left(\frac{|S_i|}{|X|}\right)^\alpha \quad B_i = \left(\frac{\tau - K + |K - \tau|}{2\tau}\right)^\beta$$

$$W_{y_i} = A_i B_i$$

We also let τ be the tail constant that dictates how many drugs (sequentially after drug lm) are considered likely candidates. α and β are constants that dictate the shape of the exponential functions A and B .

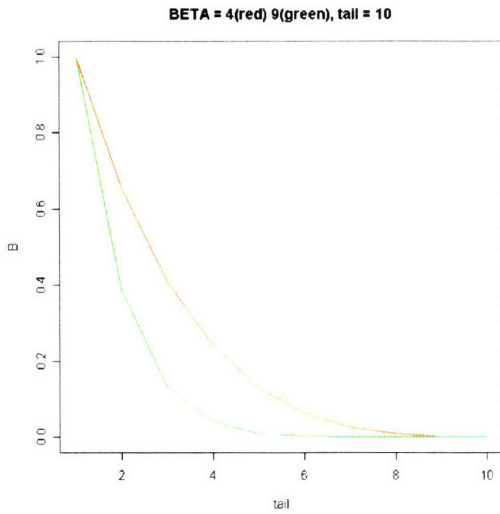


Fig 6: B can be linear or exponential depending on the value of β . B assigns a decreasing weight to drugs by their sequential distance from common areas between two drug lists.

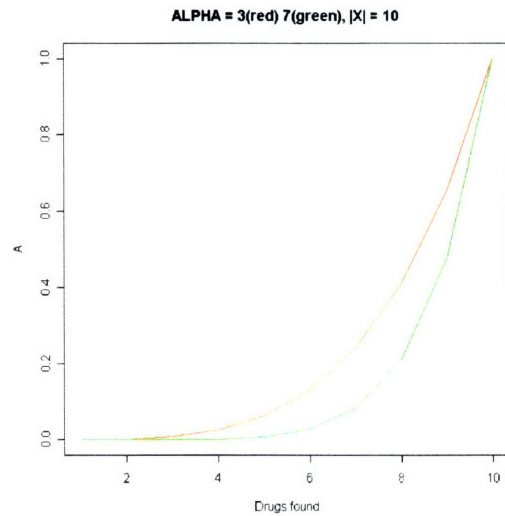


Fig 7: A can be linear or exponential depending on the value of α . A assigns an increasing weight to drugs by the degree of similarity between two drug lists

The rationale behind this algorithm is that the weight W for a drug in Y at time t is the product of how many drugs Y has in common with X up until time t , and how many drugs have been searched in Y since a common drug was last found. As can be seen in Figure 6 and Figure 7, A grows exponentially as X and Y have more drugs in common and B decreases exponentially as more drugs pass from last common drug in Y and X .

Candidate drugs are then ordered in descending order by the sum of their weights. The location in the list of candidate drugs of the drug to be predicted is then reported.

A variant of the WFBS algorithm that assigns a negative weight for drugs sequentially farther from the last common drug between X and Y than τ (the case when $K > \tau$) was also tried. Assigning a negative weight to a drug that is far, in time, from the similarity between X and Y could potentially increase accuracy by compensating for drugs that are frequently used and therefore could be assigned an inappropriately high combined weight. For this variant A is unchanged and B is (also see Figure 8):

$$B_i = \left(\frac{\tau - K}{\tau}\right)^\beta \quad (\beta \text{ is odd})$$

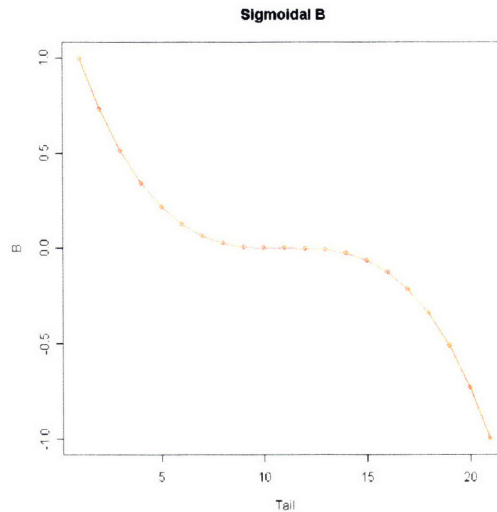


Fig 8: The Sigmoid version of B assigns negative weights drugs that are sequentially farther than τ from the last drug that was found to be common with X and Y

Functional grouping of drugs

Lastly, to try to increase the accuracy of prediction further, the drugs were grouped into functional groups using the ATC coding system. Examples of groups are “Selective beta-2-adrenoreceptor agonists” and “Renin-inhibitors”. This reduces the drugs from 978 generic drugs to 456 groups of therapeutically equivalent drugs. The rationale behind this approach is that once the drug group is known, it is trivial for the prescriber to choose the right drug within a drug group. That extra step might be acceptable if the increase in accuracy is enough.

Support Vector Machines

The theory of Support Vector Machines can be used both for classification and regression problems. A Support Vector Machine (SVM) is a classifier derived from statistical learning theory and was first introduced by Vapnik et al ⁽³⁹⁾. The Support Vector Machines used in this study were trained using the e1071 (v1.5-18) package of the R environment. For each experiment

978 Support Vector Machines were trained, one for each prediction class or drug. For prediction, the relative probabilities from each SVM were ordered by decreasing degree and the prediction list created from the 20 highest ranked drugs.

Training and testing the SVMs with the same protocol as the other algorithms resulted in poor performance. The reason for this bad performance is that the frequencies of individual classes in the database are unevenly distributed, with a relatively small minority of drugs responsible for majority of prescriptions. Therefore enriched sub sampling of the training set was used to ensure that the ratio of positive versus negative samples was at least 1:5 and each Support Vector Machine trained with 8000 samples. This was not possible for the less frequent drugs so some compromises had to be made.

In addition, a separate subsample of patients was created that was enriched with the 200 most frequently prescribed drugs to estimate how the performance would be on a larger database.

The SVMs were trained using both the linear and radial bases kernels and their performance compared. Only prescription history was used, and no adjustment for age or gender was done.

Sensitivity analysis for WFBS algorithm

To find the best possible parameters for the WFBS method, repeated runs with incremental changes in one parameter while holding the others constant, were done.

The parameters being adjusted were α , β , τ and the age-gap used to limit the search space. An age gap of 10% means that the absolute age difference between the patient being predicted for and

the patient database he was being compared to is within 10% of the age of the patient to be predicted for.

Experimental protocol

For testing the all the algorithms except two of the Support Vector Machines, a randomly selected set of 20,000 admissions was generated for each run, and then a 5-fold cross validation used. In a 5-fold cross validation, the sample is divided in 5 equally sized parts and in turn 4 are used for training and 1 for testing until all parts have been used for testing.

Patients with one drug in their drug list were filtered out. For each patient in the test set, a random time point was selected between the second drug from admission and the third drug before discharge, and only the drugs before that time point were passed to the algorithm to create a list of candidate drugs, sorted by decreasing relevance. The location in the list of candidate drugs, of the drug to be predicted, was then reported.

As described above, the SVMs were trained and tested on three separate datasets. The first set of SVMs were trained and tested the same way as the other algorithms. (Std. dataset)

The second set of SVMs was trained with specific datasets for each SVM. They were then tested with 10,000 patients to simulate the conditions the other algorithms were tested in. (Sub sampling)

The third set of SVMs was tested with 1000 patients that were specifically chosen for their high prevalence of drugs in the frequency enriched dataset. (Enriched dataset)

Results

Algorithm	Accuracy (10)	Accuracy (20)	Recall
KNN with Hamming distance K=120	29%	36%	70%
KNN with LCS	44.8%	55%	95%
1 st order Markov Chain	<1%	<1%	100%
1 st order Markov Chain (Markov approximation)	6.5%	12.2%	100%
WFBS	60.1%	70.2%	92%
WFBS with sigmoid B	54.5%	63.6%	90%
WFBS with drug groups	71.7%	79.4%	98%
SVM linear kernel (std dataset)	3%	5%	100%
SVM linear kernel (sub sampling)	43%	56%	100%
SVM linear kernel (enriched dataset)	62.5%	65%	100%
SVM radial base (std dataset)	5.1%	6,9%	100%7
SVM radial base kernel (sub sampling)	39.1%	48.2%	100%
SVM radial base kernel (enriched dataset)	74.3%	85.7%	100%

Table 1. Except for SVMs, accuracy is the mean of 5-fold cross validation on 20,000 patients.

K-nearest neighbor (KNN)

Running K-nearest neighbors with increasing value for K, finds an optimal value for K to be around 120. The algorithm could produce a list containing the drug to be predicted for 70% of patients and of those, 41% had the drug in the top 10 list and 51% in top 20. This means that the K-nearest neighbor algorithm using the Hamming distance, with a K of 120, can produce a list of 10 drugs that contains the drug to be predicted in 29% of all patients and a list of 20 drugs in 36% of cases. A graph of KNN performance can be seen in Figure 9.

Stratifying the patients by age or gender did not increase the accuracy of KNN using the Hamming distance.

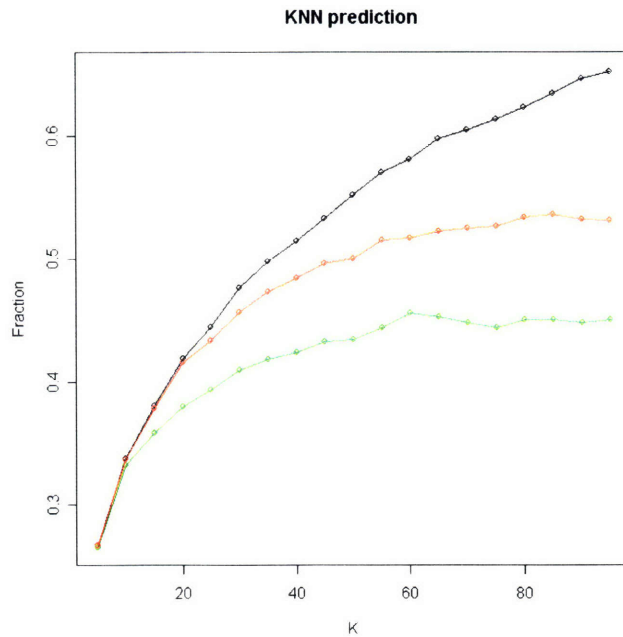


Fig 9: Accuracy of prediction using KNN with Hamming distance metric. (Black) Recall fraction. The fraction of patients the algorithm could produce a list containing the drug to be predicted. (Red) Fraction of patients with a list length 20. (Green) The fraction of patients with a list length 10.

1st order Markov chain

As expected, the standard 1st order Markov chain was not useful as a prediction algorithm since it only takes into account the last drug in the drug sequence of the patient to be predicted for. The algorithm could produce a list containing the drug to be predicted for all patients. The mean length of those drug lists was around 200 and no patient had a list longer than 500 drugs, so these results are not due to chance alone.

Using Markov approximation (combining the probabilities for all drugs the patient to predict for was taking) yielded marginally better result, but not useful results (5.6% of patients with prediction drug in top 10 lists).

Longest common subsequence (LCS)

The LCS algorithm is significantly more accurate than the Hamming distance and can make correct predictions in 44.8% of cases with a drug list length of 10 and 55% of cases with drug list length of 20 drugs. This is close to what I hypothesized to be practical for use in clinical systems. On the other hand the time complexity of this algorithm is very high and is unlikely to be feasible to calculate predictions in real time. The recall rate for the Hamming distance was 70%. The LCS algorithm on the other hand had a recall rate of 95%, meaning that it could create a list for 95% of patients that contained the drug to be predicted. The LCS algorithm was just not accurate enough in ordering the resulting list of candidate drugs. The main difference between LCS and Hamming distance is that LCS disregards the difference between patients and puts emphasis on elements they have in common. LCS is however sensitive to the order of drugs and therefore ignores common elements, even if they are close in time, if the order is not the same. A graphic illustrating the sensitivity to the choice of the neighborhood size K for the LCS variant of KNN can be seen in Figure 10.

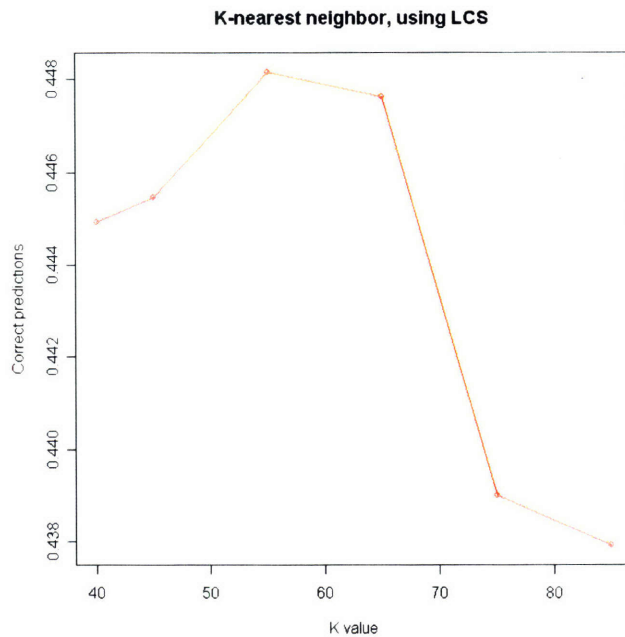


Fig 10: Finding the optimal K-value for KNN using the LCS algorithm. Correct predictions are based on a list length of 10.

Weighted frequency based sequence (WFBS)

The WFBS algorithm was the most accurate of the all the algorithms examined. By stratifying the patient population by age and gender, the input could be reduced significantly without decreasing the accuracy of the predictions, thus decreasing execution time. This property was not observed with the other algorithms.

The best accuracy archived for the whole database was 60.1% for a drug list length of 10 and 70.2% for a drug list of length 20. This is somewhat better than the initial criteria of success. This means that for any given prescription, a list of 20 drugs can be generated that contains the drug to be prescribed in majority of cases. The recall rate was 95%

Using the sigmoid version of B for the WFBS decreased the accuracy of the algorithm by 5% on average. The reason for this is probably that when comparing to patients with long drug lists with a low degree of similarity, a negative weight is assigned to drugs in an arbitrary manner.

Grouping drugs by function

Consolidation of the prediction classes, from active substances into drug groups, increased the accuracy considerably. Using the best possible parameters (Figures 15-18) for this approach, the WFBS algorithm could produce a drug list of length 20 for 79.4% of random prescriptions for random patients; in 71.1% of cases the drug to be predicted was among the top 10 drugs of that list.

Sensitivity analysis for WFBS algorithm

The best parameters to predict any randomly chosen drug for any randomly chosen patient were:

- τ : 7 (Figure 12)
- Age gap: 0.1. If the patients were stratified further by wards and/or prescribing physicians, the accuracy can be increased by increasing the age-gap (Figure 11)
- α : 3 (Figure 14)
- β : 4 (Figure 13)

The best parameters for the grouped drugs were similar from ungrouped drugs. The age-gap and alpha were unchanged, but the additional benefit was gained by increasing the length of τ (10) but was partially compensated by a large increase in beta (7) (Figures 15-18).

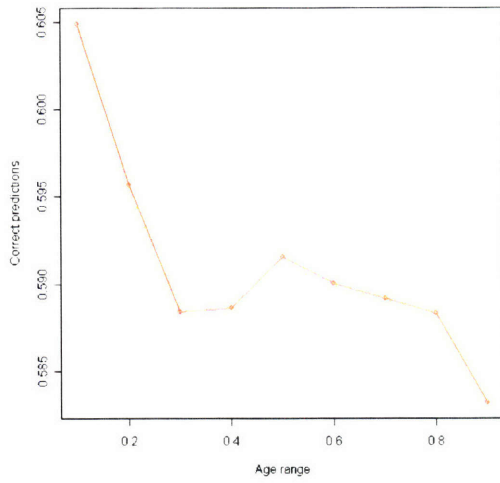


Fig 11: The effect of age-gap on the accuracy of the WFBS algorithm

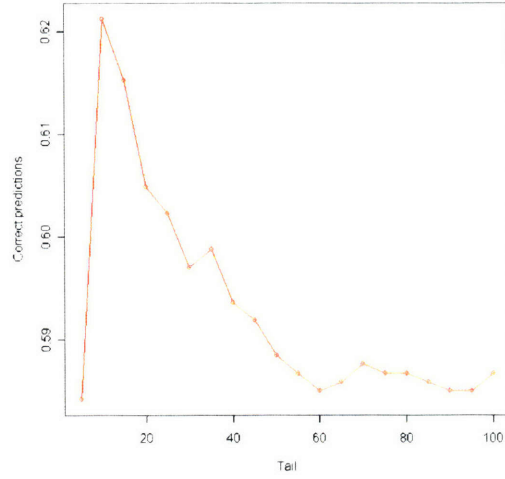


Fig 12: The effect of τ on the accuracy of the WFBS algorithm

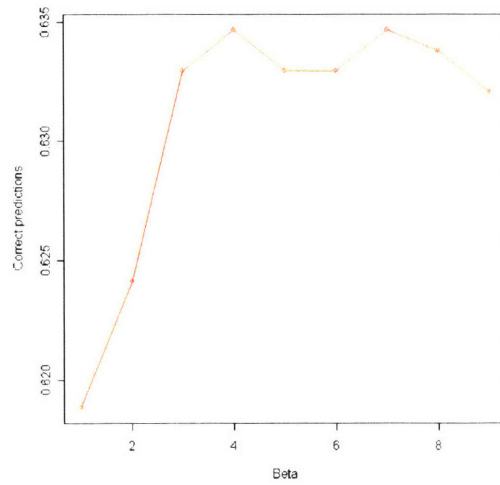


Fig 13: The effect of β on the accuracy of the WFBS algorithm

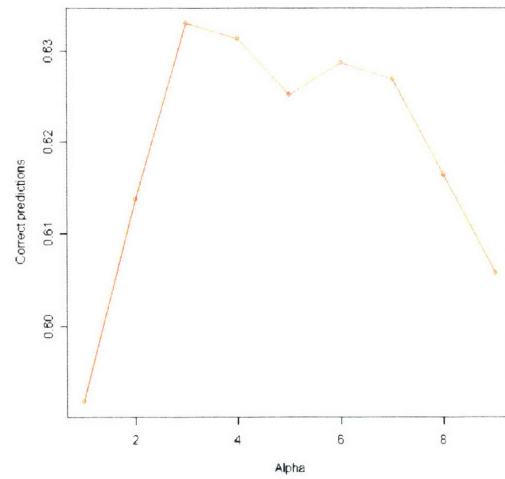


Fig 14: The effect of α on the accuracy of the WFBS algorithm

Figures 11 through 14 show the effect of incrementally changing one parameter for the WFBS algorithm while holding all others constant.

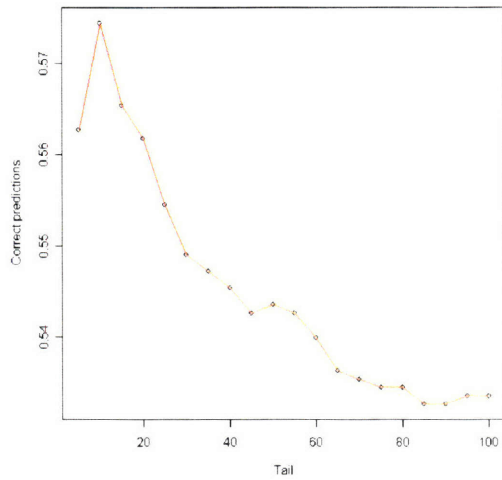


Fig 15: τ

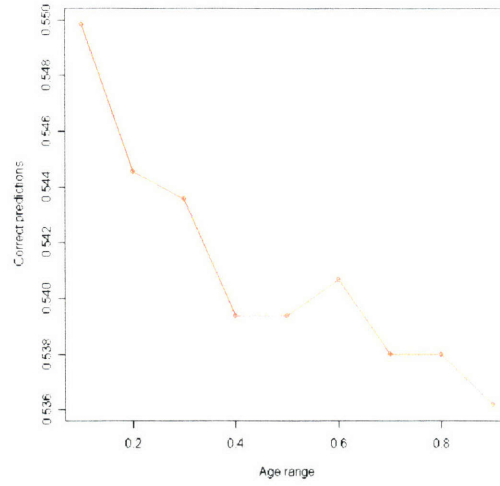


Fig 16: Age-gap

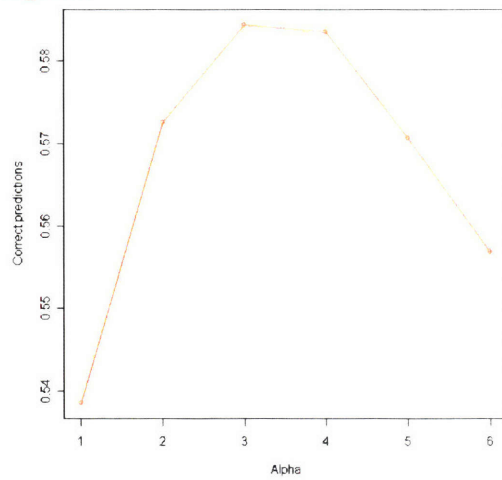


Fig 17: α

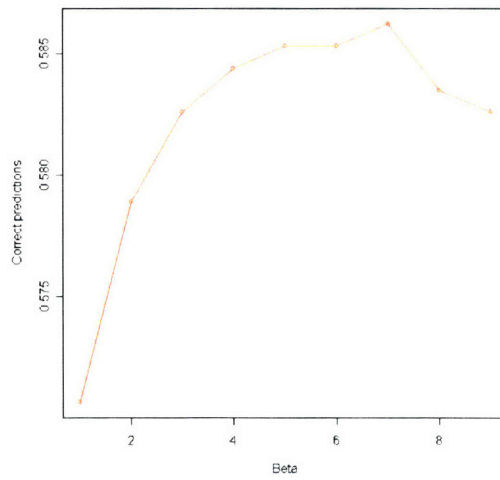


Fig 18: β

Figures 15 through 18 show the effect of incrementally changing one parameter for the WFBS algorithm while holding all others constant.

Support Vector Machines

The SVMs were trained and tested on three different datasets. Using the same dataset as the other algorithms produced poor results with less than 3% of patients having a drug list length of 10 and 5% with 20 using the Radial Base kernel. The linear kernel performed marginally better with 5.1% of patients with drug list length of 10 and 6.9% with 20.

Using specifically enhanced datasets for each SVM was significantly better. Using a linear kernel the SVM could generate correct prediction with a drug list length of 10 in 43% of cases and 56% of cases with drug list length of 20 drugs. The Radial Base kernel performed somewhat worse with accuracy of 39.1% and 48.2% for drug lists of length 10 and 20 respectively.

Training and testing with only the 200 most frequent drugs produced better results, 62.5% and 65% with the linear kernel and 74.3% and 85.7% with the Radial Base kernel for drug list length of 10 and 20 respectively.

Discussion

This study indicates that using probabilistic methods to predict prescription patterns is a viable method to decrease the time it takes to enter medication orders into computerized physicians order entry systems. Each time a physician prescribes a medication for a patient the algorithm can create a list of 20 medications and for 70% of the patients, the correct medication is in the list. This is probably time saving for the physician since the list has been shortened from about 5,000 medications. The algorithm suggested has reasonable time complexity, making this a feasible option for real time support.

This study also suggests that accuracy could even be increased by taking into account prescriber preferences based on past prescription habits. This would mean combining the commonly used method of making a favorites list with this new algorithm. However further studies on this combination are beyond the scope of this study.

WFBS

The predictability of WFBS is especially interesting based on the fact how much variability there is in the prescription patterns. Even though the database contains 80,000 patients and more than one million prescriptions the variability is still high and patient clusters are small. This prevents us from finding well defined clusters of patients. More data is needed to increase the predictability to include the 20 - 30% that still is missing.

It is also interesting to see how the predictability of WFBS increases from 60% with a list of 10 medications to 70% by doubling the list to 20. The accuracy can even be increased up to about 80% by offering a list of 20 medication groups instead of a specific medication. This is potentially of value but changes the drug selection process from a single step to two steps. Once the relevant medication group has been chosen a new short list of medications is created.

The WFBS algorithm has three parameters: α , β and τ that dictate various aspects of how the algorithm weights candidate drugs. These parameters are sensitive to the input and have to be reevaluated if the process that selects patients for training is changed. The change in input that had the biggest impact on both time complexity and accuracy was the age difference allowed between the patient being predicted for and the patient database he was being compared to (age-gap). This shows that drug patterns are correlated with age, as expected.

The predictive accuracy of WFBS varies marginally over the course of treatment. It seems to be easier to predict the first and last medications with the exception of the last two medications for each patient. The explanation for this might be that drugs for chronic conditions, temporarily halted during the admission, are being restarted at discharge. These changes are however marginal (within 10%), but hint at the possibility of other methods to increase the accuracy of prescription prediction in the middle part of treatment where uncertainty seem to be higher, and before discharge where other factors than current treatment might be dictating the prescriptions.

When enough data is available, the accuracy can be further increased by considering well defined subsets of patients by ward and age. Even if the database contained more than 1 million prescriptions, there were not many such patient clusters. It is probable that as more prescription data is gathered, such approaches become more viable, but that is beyond the scope of this study.

SVM

The Support Vector machine shows promise for this type of prediction. It was however plagued by the infrequency of most of the drugs in the database. 40% of drugs had 20 or fewer patients. This is not enough samples to train a SVM sufficiently, while predictions based on neighborhood searches and local approximations are not as much affected by this. The radial basis kernel performed best when sample sizes were large enough but the linear kernel coped better with few samples and therefore had better overall performance. The radial basis kernel is however more likely to perform well on a larger dataset.

Support Vector Machines have however a high time complexity during training and need a richer dataset than we have at our disposal at this time. Training 978 SVMs on 20,000 random samples

from the database produced poor results. Probability estimates from SVMs are based on the distance of the testing sample from the margin. There were issues with the calibration and the SVMs overestimated probabilities positively correlated to the class prevalence. The prior probabilities of the classes were also badly represented by the SVMs causing prevalence of infrequent drugs to be overestimated. Ranking based on those probabilities can be arbitrary. Interactions between the prior probabilities and the SVM predictions are a complex matter and beyond the scope of this study.

Based on the performance of SVM in an enriched dataset of frequent drugs, it is possible that SVM based methods will outperform nonlinear local methods such as the WFBS algorithm presented in this paper. Especially considering that no adjustments were made with regards to age and gender for the SVMs. If the enriched dataset is a good approximation to a larger dataset, these results are promising. It is however difficult to compare SVMs to the other algorithms since they were trained and tested using different methods for subject sampling.

Even if reasonable accuracy is achieved using only prescription history, there is no obvious reason to omit potentially more information rich data. Future studies could aim at expanding the input parameters to include more specific clinical information such as signs, symptoms, diagnoses and laboratory values.

Limitations of the study

The main limitation of the study is the data available to train the prediction algorithms on. It is probable that in an enterprise setting, more categories of data would be available to make inference on. Information such as diagnoses, signs and symptoms, or lab values is likely to enhance the predictive abilities of any algorithm.

The performance measurements are based on random samples from one database. This might cause bias in favor of our methods and the results are not necessarily reproducible in other settings. Also since the implementation of the CPOE system is still ongoing there might be considerable sampling bias caused by the types of wards that were early in the implementation process. Further inspection reveals them to be of similar nature.

Even if the database contains more than a million prescriptions and 80,000 patients, it is still not showing clear signs of patient clustering. Therefore one might conclude that database is too small for our purposes. On the other hand that might indicate that accuracy could increase as the database grows.

Conclusions

This study suggests that using probabilistic methods to create patient specific medication pick lists at the time of prescription using only the prescription history, could be a promising method to reduce the time burden of CPOE systems. However, further research is needed to evaluate the impact of such lists on prescription habits.

References

1. *To Err Is Human: Building a Safer Health System*. **L.T. Kohn, J.M. Corrigan, and M.S. Donaldson, [ed.]**. 1999, Washington: National Academies Press.
2. *The Nature of Adverse Events in Hospitalized Patients: Results from the Harvard Medical Practice Study II*. **L.L. Leape et al.** 6, 1991, *New England Journal of Medicine*, Vol. 324, pp. 377–384.
3. *Effects of Computerized Physician Order Entry and Clinical Decision Support Systems on Medication Safety: A Systematic Review*. **R. Kaushal et al.** 12, 2003, *Archives of Internal Medicine*, Vol. 163, pp. 1409–1416.
4. *Computerized Physician Order Entry in U.S. Hospitals: Results of a 2002 Survey*. **J.S. Ash et al.** 2004, *Journal of the American Medical Informatics Association* 11, no. 2, pp. 95-99.
5. *The impact of computerized physician medication order entry in hospitalized patients—A systematic review*. **Saeid Eslami, Nicolette F. de Keizer, Ameen Abu-Hanna.** 6, 2008, *International Journal of Medical Informatics*, Vol. 77, pp. 365-376.
6. *Impact of computerized physician order entry on physician time*. **Bates DW, Boyle DL, Teich JM.** 1994, *Proc Annu Symp Comput Appl Med Care*, p. 996.
7. *Controlled trial of direct physician order entry: effects on physicians' time utilization in ambulatory primary care internal medicine practices 2001*;8:361-371. **Overhage JM, Perkins S, Tierney W, McDonald CJ.** 8, 2001, *J Am Med Inform Assoc*, pp. 361-371.

8. *Comparison of time spent writing orders on paper with computerized physician order entry.* **Shu K, Boyle D, Spurr C, et al.** 10, 2001, Medinfo, pp. 1207-1211.
9. *Key attributes of a successful physician order entry system implementation in a multi-hospital environment.* **Ahmad A, Teater P, Bentley TD, Kuehn L, Kumar RR, Thomas A, et al.** 9, 1002, J Am Med Inform Assoc., pp. 16–24.
10. *A consensus statement on considerations for a successful CPOE implementation.* **Ash JS, Stavri PZ, Kuperman GJ.** . 10, 2003, J Am Med Inform Assoc., pp. 229–34.
11. *Evaluation of a command-line parser-based order entry pathway for the Department of Veterans Affairs electronic patient record.* **Lovis C, Chapko MK, Martin DP, Payne TH, Baud RH, Hoey PJ, et al.** . 2001, J Am Med Inform Assoc., Vol. 8, pp. 486–98.
12. *Preparation and use of preconstructed orders, order sets, and order menus in a computerized provider order entry system.* **Payne TH, Hoey PJ, Nichol P, Lovis C.** 10, 2003, J Am Med Inform Assoc. , pp. 322–9.
13. *The characteristics of personal order sets in a computerized physician order entry system at a community hospital.* **Thomas SM, Davis DC.** . 1031, 2003, Proc AMIA Symp. .
14. *A cross-site qualitative study of physician order entry.* . **Ash JS, Gorman PN, Lavelle M, Payne TH, Massaro TA, Frantz GL, et al.** . 10, 2003, J Am Med Inform Assoc., pp. 188–200.

15. *Determinants of Common Stock Prices: A Time Series Analysis*. **Kraft, John Kraft and Arthur.** 2, 1977, *The Journal of Finance*, Vol. 32, pp. 417-425 .
16. *On a Method of Investigating Periodicities in Disturbed Series, with Special Reference to Wolfer's Sunspot Numbers*. **G. Udny Yule.** 1927, *Philosophical Transactions of the Royal Society of London, Ser. A,* , Vol. 226, pp. 267–298.
17. **ANDERSON, O.D.** *Time Series Analysis and Forecasting: The Box-Jenkins Approach*. s.l. : Butterworth, 1976.
18. **T. W. Anderson.** *The Statistical Analysis of Time Series*. s.l. : Wiley-Interscience, 1994.
19. **Box, G. E. P., Jenkins, G. M., and Reinsel, G. C.** *Time Series Analysis, Forecasting and Control*. Englewood Clifs, NJ. : Prentice Hall, 1994.
20. **Fuller, W.A.** *Introduction to statistical time series*. s.l. : Wiley Series in Probability and Mathematical Statistics (USA), 1976.
21. **E.J. Hannan, M. Deistler.** *The statistical theory of linear systems*. s.l. : Wiley, 1988.
22. *Neural” Computation Of Decisions In Optimization Problems*. **JJ Hopfield, DW Tank.** 1985, *Biol Cybern*, Vol. 52, pp. 141—152.
23. *Learning internal representations by error propagation*. **D. E. Rumelhart, G. E. Hinton, R. J. Williams.** s.l. : MIT Press Cambridge, MA, USA , *Parallel distributed processing: explorations in the microstructure of cognition, foundations book contents*, Vol. 1, pp. 318 - 362 .

24. **Christopher M. Bishop.** *Neural Networks for Pattern Recognition.* s.l. : Oxford University Press , 1995.
25. , 1969: *Atmospheric Predictability as Revealed by Naturally Occurring Analogues.* **Lorenz, E.N.** J. Atmos. Sci., Vol. 26, pp. 636–646.
26. *Time-Series Analysis.* **D. S. Broomhead and R. Jones.** 1864, May 1989, Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences, Vol. 423.
27. **Martin Casdagli and Stephen Eubank.** *Nonlinear modeling and forecasting.* s.l. : Addison-Wesley, Reading, MA, 1992.
28. *Chaos and forecasting.* **B. Grenfell, R.M. May, H. Tong.** 1994 , Trans. Philos. R. Soc. London A , Vol. 348.
29. **Holger Kantz, Thomas Schreiber.** *Nonlinear Time Series Analysis.* s.l. : Cambridge University, 2004.
30. *Time series prediction: Forecasting the future and understanding the past.* **Weigend, Andreas S.; Gershenfeld, Neil A.** 1992. Santa Fe Institute Studies in the Sciences of Complexity, Proceedings of the NATO Advanced Research Workshop on Comparative Time Series Analysis, held in Santa Fe, New Mexico.
31. *Nearest-Neighbor methods for nonparametric rainfall-runoff forecasting.* **Karlsson, M., and S. Yakowitz.** 7, 1987a., Water Res. Res, Vol. 23, pp. 1300-1308.

32. *Amazon.com recommendations: item-to-item collaborative filtering*. **Linden, G.; Smith, B.; York, J.** 1, jan/feb 2003, Internet Computing, IEEE, Vol. 7, pp. 76 - 80.
33. *Amazon.com Recommendations: Item-to-Item Collaborative Filtering*. **Greg Linden, Brent Smith, and Jeremy York.** 1, 2003, IEEE Internet Computing, Vol. 7, pp. 76-80.
34. *GroupLens: An open architecture for collaborative filtering of netnews*. **al., P. Resnick et.** 1994, Proc. ACM 1994 conf. Computer supported cooperative work, pp. 175-185.
35. ATC. *WHO Collaborating Centre for Drug Statistics Methodology*. [Online] <http://www.whocc.no/atcddd/>.
36. **Madsen, Henrik.** *Time Series Analysis*. s.l. : Kgs. Lyngby, 2001.
37. *Support vector machines for spam categorization*. **Drucker, H. Donghui Wu Vapnik, V.N.** 5, Neural Networks, IEEE Transactions on, Vol. 10, pp. 1048-1054.
38. **Richard O. Duda, Peter E. Stork, David. Stork.** *Pattern Classification*. s.l. : Wiley-interscience, 2001.
39. *A training algorithm for optimal margin classifiers*. **Bernhard E. Boser , Isabelle M. Guyon , Vladimir N. Vapnik.** July 27-29, 1992, Proceedings of the fifth annual workshop on Computational learning theory, pp. 144-152.

40. *Physicians, information technology, and health care systems: a journey, not a destination.* McDonald CJ, Overhage JM, Mamlin BW, Dexter PD, Tierney WM. 11, 2004, J Am Med Inform Assoc. , pp. 121–4.
41. *A usability study of physicians interaction with a paper-based patient record system and a graphical-based electronic patient record system.* Rodriguez NJ, Murillo V, Borges JA, Ortiz J, Sands DZ. 2002, Proc AMIA Symp. , pp. 667–71.
42. *Primary care physician time utilization before and after implementation of an electronic health record: a time-motion study.* Pizziferri L, Kittler A, Volk L, Honour M, Gupta S, Wang S, et al. jan 2005, J Biomed Inform. .
43. *Ten commandments for effective clinical decision support: making the practice of evidence-based medicine a reality.* Bates DW, Kuperman GJ, Wang S, Gandhi T, Kittler A, Volk L, et al. 10, 2003, J Am Med Inform Assoc. , pp. 523–30.
44. *Searching for answers on a clinical information-system.* C., Safran. 34, 1995, Methods Inf Med. , pp. 79–84.
45. *A framework for analyzing the cognitive complexity of computer-assisted clinical ordering.* Horsky J, Kaufman DR, Oppenheim MI, Patel VL. 2003, J Biomed Inform. , Vol. 36, pp. 4–22.
46. *The characteristics of personal order sets in a computerized physician order entry system at a community hospital.* Thomas SM, Davis DC. 1031, 2003, Proc AMIA Symp.

47. *Information Retrieval Performance of Probabilistically Generated, Problem-Specific Computerized Provider Order Entry Pick-Lists: A Pilot Study.* **Adam S. Rothschild, Harold P. Lehmann.** 12, 2005, *J Am Med Inform Assoc.* , pp. 322-330.
48. *The impact of computerized physician medication order entry in hospitalized patients—A systematic review.* **Saeid Eslami, Nicolette F. de Keizer, Ameen Abu-Hanna.** 6, jun 2008, *International Journal of Medical Informatics*, Vol. 77, pp. 365-376.
49. *Adding insight: A qualitative cross-site study of physician order entry.* **Joan S. Ash, Dean F. Sittig, Veena Seshadri, Richard H. Dykstra, James D. Carpenter, P. Zoe Stavri.** 7-8, August 2005, *International Journal of Medical Informatics*, Vol. 74, pp. 623-628.
50. *U.S. adoption of computerized physician order entry systems.* **Cutler DM, Feldman NE, Horwitz JR.** 6, Nov-Dec 2005, *Health Aff (Millwood).*, Vol. 24, pp. 1654-63.
51. *Medication-related clinical decision support in computerized provider order entry systems: a review.* **Kuperman GJ, Bobb A, Payne TH, Avery AJ, Gandhi TK, Burns G, Classen DC, Bates DW.** 1, Jan-Feb 2007, *J Am Med Inform Assoc*, Vol. 14, pp. 29-40.
52. *Analyzing transaction workflows in an ePrescribing system.* **Virk P, Bates DW, Halamka J, Fournier GA, Rothschild JM.** 1129, 2006, *AMIA Annu Symp Proc.* .