Федеральное государственное бюджетное образовательное
учреждение высшего образования «Московский государственный
университет имени М. В. Ломоносова»
кафедра математического моделировния и информатики

Курсовая работа
на тему

«Нечеткая логика и машинное обучения в задаче постановки диагноза»

Выполнил
студент 2 курса 205 академической группы
Новиков Дмитрий Валерьевич
Научный руководитель:
к. ф.-м. н. доц. А. В. Зубюк

Москва - 2023

Содержание

1 Введение

В настоящее время компьютерные технологии получили бурное развитие, поэтому в последние десятилетия всё больше внимания уделяется возможности автоматизации процесса выполнения различных задач, обусловленных интеллектуальными возможностями человека. Пример такой деятельности - решение различных биомедицинских задач, в частности — выявление диабета у пациентов на основании результатов анализов.

Иногда, ввиду человеческого фактора, пациенту может быть поставлен неверный диагноз из-за чего тот не получит необходимое лечение, при этом проверка каждого результата по несколько раз или перепроверка другим экспертом требует много временных и человеческих ресурсов. Поэтому разработка моделей, основанных на экспертном мнении, позволяющих быстро и с большой степенью надежности поставить диагноз, является важной и крайне актуальной задачей в медицине и биологии. Благодаря им можно убедиться в достоверности поставленного диагноза или необходимости его перепроверки.

Раньше было сложно быстро и эффективно выделить из всех пациентов тех, кому в первую очередь нужна медицинская помощь, так как врач не может осмотреть сразу всех. Теперь, благодаря описанным алгоритмам, появилась возможность предварительно оценить степень заболевания и выделить пациентов, срочно нуждающихся в лечении.

На настоящий момент существует множество работ, посвященных данной тематике. В основном эти работы опираются на применении нечеткой логики для построения предсказывающей модели.

Первые работы в этой области предлагались различные экспертные системы, созданные для помощи врачам в сложных случаях. В этих работах рассматривалась задача постановки диагноза как для конкретной болезни [4, 5], так и для нескольких болезней [3]. Во всех вышеуказанных работах системы строились на основании экспертного мнения. Ввиду всё нарастающей популярности нейронных сетей и искусственного интеллекта в последних работах всё чаще стали использоваться алгоритмы машинного обучения. В статьях [1, 2] показаны примеры успешного применения сочетаний нечеткой логики и алгоритмов машинного обучения для решения задачи постановки диагноза.

Настоящая работа посвящена задаче определения диагноза с использованием нечеткой логики и методов машинного обучения.

1.1 Цели и задачи

1) Изучение методов нечеткой логики
2) Реализация алгоритма машинного обучения для решения задачи о постановке диагноза с использованием теории вероятностей
3) Сравнение получившегося алгоритма с общеизвестными алгоритмами машинного обучения

2 Методы решения задачи

2.1 Набор данных

Изначально данные представляли собой таблицу, содержащую различные биомедицинские показатели пациенток в возрасте от 21 года такие как: количество беременностей, уровень глюкозы в крови, кровяное давление, толщина кожи, уровень инсулина в крови, индекс массы тела, функция родословной диабета, возраст. Анализы были собраны Национальным институтом диабета, болезней органов пищеварения и почек, являющегося частью Национальных институтов США. Всего таблица содержит информацию о 768 пациентах. Набор данных содержит большое количества пропущенных значений, в связи с этим перед

Out[191]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148.0	72.0	35.0	NaN	33.6	0.627	50	1
1	1	85.0	66.0	29.0	NaN	26.6	0.351	31	0
2	8	183.0	64.0	NaN	NaN	23.3	0.672	32	1
3	1	89.0	66.0	23.0	94.0	28.1	0.167	21	0
4	0	137.0	40.0	35.0	168.0	43.1	2.288	33	1

Рис. 1: Пример данных

применением алгоритма машинного обучения необходимо заполнить как можно больше данных так, чтобы они остались информативными. Диаграмма на рис. 2 наглядно показывает, что количество здоровых

людей в данном наборе данных сильно больше количества больных людей. В связи с этим необходимо правильно подобрать метрику качества работы алгоритма, которая впоследствии будет использована.

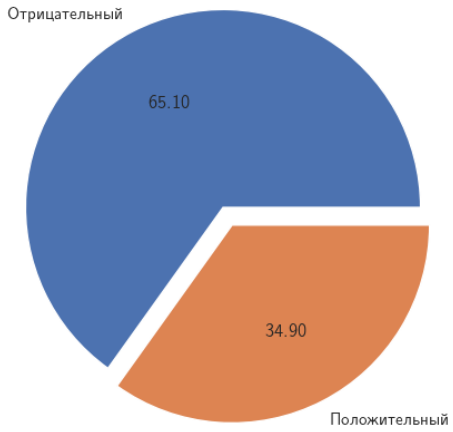

Рис. 2: Диаграмма количества больных и здоровых пациентов

2.2 Постановка задачи

Как было указано выше, данные содержат большое количество пропущенных значений. Это особенно заметно на примере столбца с информацией о содержании инсулина в крови. Поскольку инсулин является одним из самых главных факторов при постановке диагноза, необходимо аккуратно заполнить пропуски. Введём метрики качества *полнота* (в англоязычной литературе — recall) и *точность* (в англоязычной литературе — precision). Обозначим полноту как re, а точность как pr. По определению:

$$re = \frac{TP}{TP + FP} \tag{1}$$

$$pr = \frac{TP}{TP + FN} \tag{2}$$

Где TP — вероятность алгоритма верно предсказать положительный исход, FP — вероятность неверно предсказанных положительный исход, FN — вероятность неправильно предсказать отрицательный исход. В данной задаче положительным результатом считается наличие диабета, а отрицательным - его отсутствие.

Данные метрики лежат в диапазоне от 0 до 1. Чем лучше алгоритм выявляет случаи с положительным результатом, тем выше полнота. Точность, в свою очередь, тем выше, чем лучше алгоритм выявляет отрицательные результаты. Поскольку метрика «полнота» не даёт никакой информации о том, насколько хорошо алгоритм выявляет отрицательные результаты, а «точность» не даёт никакой информации о положительных результатах, часто используется метрика $f1$, которая является объединением этих двух метрик:

$$f1 = \frac{2repr}{pr + re} \tag{3}$$

Задача состоит в максимализации значения метрики $f1$. Для этой задачи, а также для задачи заполнения

Категория i		Экспертная оценка	
		Положительная	Отрицательная
Оценка системы	Положительная	TP	FP
	Отрицательная	FN	TN

Рис. 3: Матрица ошибок

пропущенных данных будут использоваться алгоритмы, основанные на теории вероятностей [6].

Помимо вышеуказанных метрик, будет использована метрика *аккуратность:* (в англоязычной литературе — accuracy)[1] . Обозначим аккуратность как ac.

$$ac = \frac{TP + TN}{TP + TN + FP + FN} \tag{4}$$

Где TP — вероятность алгоритма верно предсказать положительный исход, TN — вероятность верно предсказать отрицательный исход, FP — вероятность неверно предсказанных положительный исход, FN — вероятность неправильно предсказать отрицательный исход.

2.3 Математическое обоснование модели

2.3.1 Теория вероятностей

Теория вероятностей [6] - математический аппарат, разработанный для работы со случайностью. Эту случайность называют вероятностной. В качестве примера случайного события можно примести число, выпадающее на грани кубика при его броске, или сторона, на которую приземлится монетка. Случайность также свойственна и некоторым физическим процессам (например, взаимодействие частицы с веществом или процесс распада).
Эксперимент со случайным исходом называют стохастическим. Элементарные исходы стохастического эксперимента - такие исходы, которые реализуются только один раз в каждом испытании. Множество, образованное элементарными исходами, обозначается Ω. Пусть α - такая система подмножеств Ω, что:

1) $\emptyset \in \alpha, \Omega \in \alpha$

2) $\bigcup_{i=1}^{\infty} A_i \in \alpha \forall A_i, i = 1, 2, ...$

3) $\bigcap_{i=1}^{\infty} A_i \in \alpha \forall A_i, i = 1, 2, ...$

Эта система будет интерпретироваться как множество всех событий.
Вероятностью будем называть такую функцию $P : \alpha \mapsto [0, 1]$, удовлетворяющую аксиомам:

1) $P(\emptyset) = 0$

2) $P(\Omega) = 1$

3) $P(\bigcup_{i=1}^{\infty} A_i) = /sum_{i=1}^{\infty} P(A_i) \forall A_i, i = 1, 2, ... : A_i \bigcap A_j = \emptyset i \neq j$

Частотную вероятность можно определить как следующий предел:

$$P(A_i) = \lim_{N \to \infty} \frac{n_i}{N} \tag{5}$$

Где N - общее количество проведенных испытаний, n_i - количество испытаний, результатом которых стало событие A_i

2.3.2 Байесовский подход. Нахождение неизвестной величины

Рассмотрим байесовский подход [7]. Пусть есть некоторый случайный параметр θ. Байесовский подход заключается в том, что θ рассматривается как случайная величина, имеющая плотность распределения $q(t), t \in \Theta$ (которая может быть неизвестна). $q(t)$ называют априорной, так как она фактически дана до эксперимента. Байесовский подход рассматривает θ как случайно выбранный из распределения $q(t)$ параметр. Введём функцию правдоподобия $f_t(\mathbf{x}), t \in \Theta, \mathbf{x} \in X^n$ аналогично тому, как она была введена в [7]. $f_t(\mathbf{x})$ при некотором определенном t - плотность распределения в X^n, поэтому следующая функция:

$$f(\mathbf{x}, t) = f_t(\mathbf{x}) q(t) \tag{7}$$

может быть рассмотрена как плотность совместного распределения X и Θ. Поэтому $f_t(\mathbf{x}), t \in \Theta, \mathbf{x} \in X^n$ - плотность X при условии $\Theta = t$:

$$f_t(\mathbf{x}) = f(\mathbf{x}|t) \tag{8}$$

[1] С английского языка слова «precision» и «accuracy» переводятся как «точность». Поэтому, в данной работе, будем называть метрику «accuracy» аккуратностью, а не точностью

Для условной плотности $q(t|\mathbf{x})$ величины Θ при некотором $X = (x)$ справедливо следующее соотношение:

$$q(t|\mathbf{x}) = \frac{f_t(\mathbf{x})q(t)}{f(x)} \tag{9}$$

Выражение (9) - формула Байеса или теорема Байеса.Теорема Байеса играет существенную роль во всем байесовском подходе. Она позволяет определить апостериорное распределение для θ.
Путем интегрирования формулы (9) по t, можно получить следующее выражение:

$$f(\mathbf{x}) = \int_{\Theta} f_t(\mathbf{x})q(t)dt \tag{10}$$

Пусть имеется некоторое количество случайных параметров, некоторые из которых зависят друг от друга. $x_i \in X_i$ — величины, точные значения которых нам известны в рамках данной задачи, $y_i \in Y_i$ — величины с неизвестными точными значениями, $z \in Z$ - величина, значение которой необходимо предсказать. Для этого надо посчитать вероятность $P(z = a_0|x_1 = a_1, ..., x_n = a_n)$. Где a_0 - некоторое значение из Z, a_i — некоторое значение из X_i. По определению условной вероятности $P(x|y) = \frac{P(x,y)}{P(y)}$. Применяя эту формулу, приходим к соотношению:

$$P(z = a_0|x_1 = a_1, ..., x_n = a_n) = \frac{P(z = a_0, x_1 = a_1, ..., x_n = a_n)}{P(x_1 = a_1, ..., x_n = a_n)} \tag{11}$$

Применяя соотношение (9) к (11), придём к следующей формуле:

$$P(z = a_0|x_1 = a_1, ..., x_n = a_n) = \frac{\int_Y P(z = a_0, x_1 = a_1, ..., x_n = a_n, y_1, ..., y_m)dy_1...dy_m}{\int_Y P(z, x_1 = a_1, ..., x_n = a_n, y_1, ..., y_m)dzdy_1...dy_m} \tag{12}$$

Известно, что некоторые величины зависят друг от друга. Во спользуемся этим фактом для упрощения формулы (8). В общем случае:

$$P(x_1, ..., x_n) = P(x_n|x_1, ..., x_{n-1})P(x_{n-1}|x_1, ..., x_{n-2})...P(x_1) \tag{13}$$

Если какие-либо величины не зависят друг от друга, то эта формула упрощается. Например, пусть x_n зависит только от x_{n-1}, x_{n-1} только от x_{n-2} и так далее. Тогда справедливо: $P(x_1, ..., x_n) = P(x_n|x_{n-1})$ $P(x_{n-1}|x_{n-2})...P(x_1)$.

Подведём итог. Для того, чтобы найти вероятность того, что интересующая нас величина z принимает значение a_0 необходимо воспользоваться формулами (12) и (13). Чтобы получить вид (13), необходимо знать, как зависят между собой все входящие в рассмотрение величины x_i, y_i. Таким образом можно найти значение необходимой переменной, которое имеет наибольшую вероятность. Этот метод используется, так как является оптимальным для программирования на компьютере.
Именно на этом будет построен алгоритм постановки диагноза. Зависимость между величинами будет искаться вручную, на основании экспертных суждений, а также исходя из анализа имеющегося набора данных. Совместные вероятности, входящие в формулу (13) будут найдены в ходе работы алгоритма также исходя из анализа таблицы с данными.

2.4 Некоторые известные алгоритмы машинного обучения

2.4.1 Дерево решений

Дерево принятия решений — один из наиболее распространенных средств поддержки принятия решений, которое широко используется в анализе даных и машинном обучении. Структурными элементами дерева являются *узлы* и *листья*. В узлах находятся решающие правила. В результате проверки на соответствие этому правилу множество примеров, попавших в узел, разбивается на два: соответствующее и не соответствующее этому правилу. Затем эти множества вновь разбиваются на несколько последующими правилами. В результате в последнем узле проверка не производится и он объявляется листом. Лист, в свою очередь, определяет решение. В листах содержится не правило, а подмножество объектов, удовлетворяющих всем предшествующим правилам. Пример дерева решений можно увидеть на рис. 4.

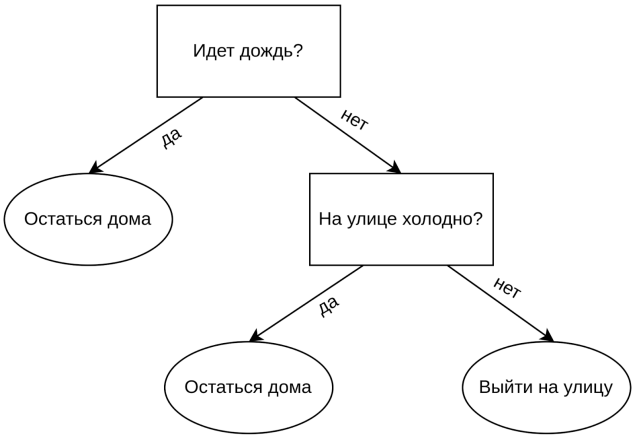

Рис. 4: Пример дерева решений

Как было сказано выше, дерево решений часто используется в алгоритмах машинного обучения. Алгоритм сам находит правила, по которым в последствии будут работать узлы и составляет из них само дерево (методы, при помощи которых это происходит могут быть разными и обсуждаться в данной работе не будут). Далее интересующее множество признаков проходит через это дерево и на выходе получается множество решений.

2.4.2 Градиентный бустинг

Градиентный бустинг — более продвинутый алгоритм по сравнению с деревьями решений. Такие алгоритмы как градиентный бустинг называют ансамблевыми, поскольку они содержат несколько других, более слабых алгоритмов внутри. Этот алгоритм является одним из самых эффективных инструментов для решения различных задач, связанных с машинным обучением. Он строит предсказание в виде ансамбля слабых предсказывающих моделей (чаще всего это деревья решений). Таким образом, из нескольких слабых моделей получается одна сильная. Основная идея этого алгоритма состоит в последовательном применении слабого алгоритма таким образом, что тот предсказывает ошибки предыдущего, сводя их к минимуму таким образом.

2.4.3 Метод опорных векторов

Метод опорных векторов также является довольно известным алгоритмом машинного обучения, применяющимся для решения различных задач. Данный алгортим строит гиперплоскость в пространстве признаков, разделяющую объекты оптимальным способом. Объекты по одну сторону относятся к одному классу, объекты по другую сторону относятся к другому классу. Алгоритм построен на предположении, что чем больше зазор между гиперплоскостью и ближайшим к ней объектом, тем более правильной будет классификация. Ближайшие к гиперплоскости объекты называются опорными объектами, а прямые проведенные через них — опорными векторами. На рис. 5 можно видеть пример подобного алгоритма.

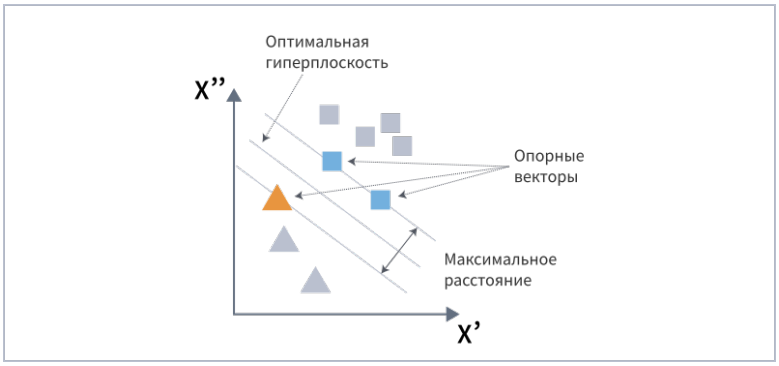

Рис. 5: Метод опорных векторов

Однако этот алгоритм сталкивается с серьёзной проблемой. Для его корректной работы пространство признаков должно быть *линейно разделимым*. Линейно разделимой называется та выборка, в которой

возможно провести некоторую гиперплоскость разделяющую объекты класса 1 и класса 2 без каких либо ошибок. В противном случае выборка называется линейно неразделимой. Пример линейно разделимой выборки можно видеть на рис. 5, пример линейно неразделимой выборки представлен на рис. 6.

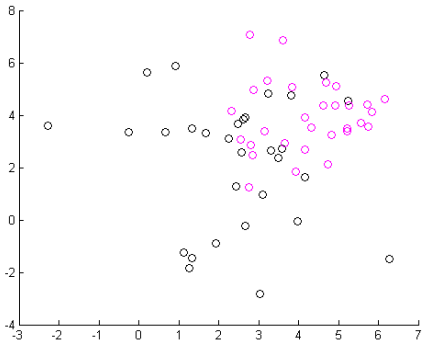

Рис. 6: Линейно неразделимая выборка

В реальных задачах почти никогда не встречаются линейно разделимые данные. Почти всегда границы классов нечеткие ввиду наличия погрешностей или по каким-либо ещё причинам. По этим причинам метод опорных векторов зачастую проигрывает в точности классификации другим методам.

3 Заключение

Подводя итоги, в ходе проделанной работы удалось решить задачу постановки диагноза с использованием нечеткой логики и машинного обучения. Для этого пропущенные данные в исходном наборе данных были заполнены и использованием аппарата теории вероятностей. Удалось сохранить информативность строк с пропущенными значениями. Результат заполнения представлен на рис. 7 в виде зависимости уровня инсулина в крови от уровня глюкозы в крови.
Был построен алгоритм, основанный на нечеткой логике и теории вероятностей, решающий задачу постановки диагноза. Результат работы алгоритма представлен на рис. 8. Алгоритм даёт хорошие результаты в сравнении с некоторыми другими популярными алгоритмами такими как «Градиентный бустинг», «Дерево решений» , «Метод опорных векторов». Такой результат даёт в дальнейшем возможность постановки диагноза пациенту на основании некоторых биомедицинских данных.

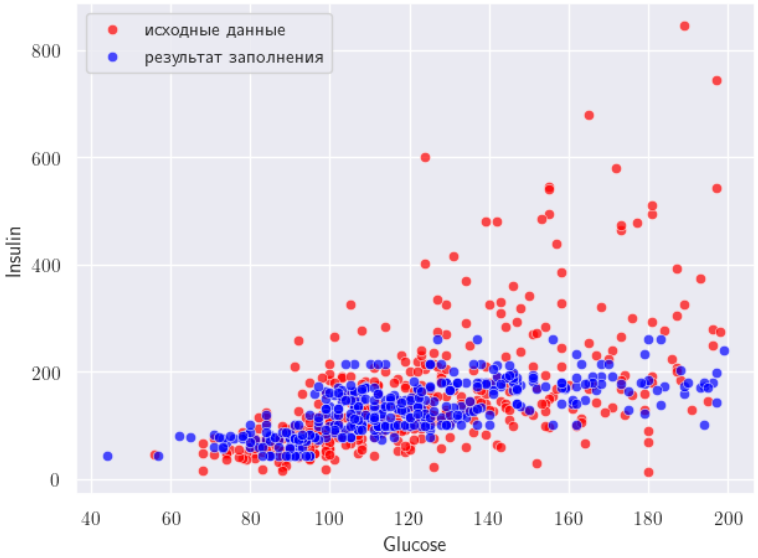

Рис. 7: Результат заполнения пропущенных данных

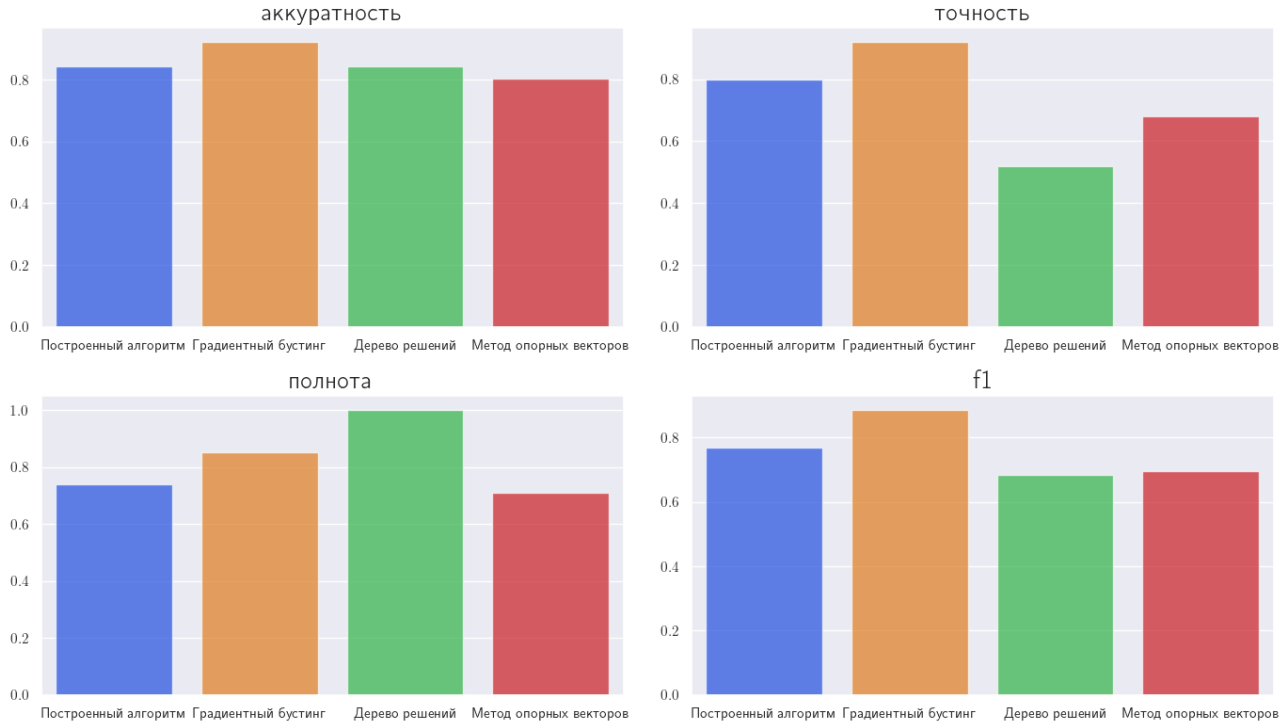

Рис. 8: Сравнение результатов работы алгоритмов

4 Список используемой литературы

⌊1⌋ Коробова Л.А., Гладких Т.В. Разработка модели принятия решения для постановки диагноза заболеваний на основе нечеткой логики // Вестник ВГУИТ. 2018. №4 (78).

⌊2⌋ Ялышев Тимур Наилевич, Филоненко Александр Васильевич Разработка экспертной системы поддержки принятия решений при постановке диагноза сахарный диабет 2 типа // Международный журнал прикладных наук и технологий «Integral». 2019. №4-2.

⌊3⌋ Орлов С. В. Использование приближенных моделей для поддержки решений по диагностике // Известия ТулГУ. Технические науки. 2009. №4.

⌊4⌋ Мельник К. В., Голоскоков А. Е. Процедура диагностирования состояния сердечно-сосудистой системы пациента на основе нечеткой логики // Вестник НТУ ХПИ. 2008. №49.

⌊5⌋ Бурилич И. Н., Уварова А. Г., Филист С. А. Автоматизированная система диагностики шизофрении на основе нечеткой логики принятия решений // ВНМТ. 2006. №2.

⌊6⌋ А. Н. Колмогоров, Теория вероятностей, Тео- рия вероятн. и ее примен., 2003, том 48, вы- пуск 2, 211–248

⌊7⌋ Боровков А. А. Математическая статистика. — Новосибирск: Наука, Изда- тельство Института математики, 1997.