Содержание

1 Введение

Целью данной работы является рассмотрение роли доверия в применении систем искусственного интеллекта в повседневной жизни; способах повышения интерпретируемости моделей в общем и возможность использования байесовских сетей для создания более прозрачных моделей машинного обучения, тем самым повышая доверие к ним. Также необходимо провести сравнительный анализ существующих программных средств интерпретации байесовских сетей, выявить для них зоны роста.

В современных условиях модели машинного обучения работают с большим количеством данных и параметров. Для эксперта или аналитика тяжело взглядом оценить качество большой модели. Мы рассмотрим популярные алгоритмы, позволяющие сократить объем предоставляемых данных без потери информации для более удобного их восприятия.

Байесовские сети представляют собой вероятностные модели на основе направленных ациклических графов, где вершины отображают состояние набора переменных, а ребра - вероятность перехода системы в другое состояние. Таким образом с помощью байесовской сети возможно эмулировать работу любой обученной модели машинного обучения, и изобразить их, разработав алгоритм визуализации лишь для БС.

В работе рассматривается возможность использования байесовских сетей для развития концепции объяснимого искусственного интеллекта(XAI). XAI – это подход, направленный на создание систем ИИ, результаты которых можно объяснить в терминах, понятных человеку. Основная цель объяснимого искусственного интеллекта – сделать принятие решений в системах ИИ прозрачным и доступным.

Несмотря на то, что байесовскую сеть достаточно просто представить в виде графа, даже для этой визуализации уже существуют множество различных программ. В последней главе разобраны несколько из их представителей – преимущества, отличительные качества, и подход к интерпретации.

2 Проблемы в развитии доверия к искусственному интеллекту

Доверие лежит в основе принятия любой технологии. Его можно определить так: 1) убеждение в том, что объект создан во благо человечества 2) готовность полагаться на технологию в ситуациях, связанных с риском для здоровья 3) совокупностью этих факторов[1].

Одно из основных препятствий в развитии доверия к искусственному интеллекту заложено в том, что процесс работы непрозрачный и случайный, своего рода “черный ящик”[2]. Это можно преодолеть с помощью создания моделей, способных объяснить принятие решений, однако возникает эффект снижения точности их вычислений. Более прозрачные модели вычисляют результат медленнее и менее точно[3].

Объяснение принятых решений лежит в основе доверия пользователя к верной работе системы, особенно в ситуации когда пользователь не обладает экспертными знаниями в области машинного обучения[4]. Однако для тех же неопытных пользователей прозрачная система может показаться очевидной и правой, в связи с чем слепое доверие превысит гарантию работоспособности системы, и даже неверная работа не будет поддаваться критической оценке. Такой феномен описан в опыте с использованием “Копировальной машины”[5].

Многие люди с опасением относятся к развитию ИИ[18], потому что считают, что в будущем ручной труд исчезнет, и людям останется лишь наблюдать за работой машин, а наладить контакт с ними невозможно. К тому же возможны сбои, в результате которых роботы под руководством ИИ захватят планету. В данном случае необходимо продемонстрировать, что искусственный интеллект не настроен “думать”, а умеет лишь решать набор задач в определенной области.

3 Способы улучшения интерпретируемости моделей

При обработке данных часто приходится работать не только с большим количеством записей, но и количество атрибутов у каждой записи тоже неуклонно растет. Проблема большой размерности данных в том, что сложно адекватно составить геометрическое представление данных в виде графика или кластеров. Для уменьшения размерности данных рассмотрим два подхода: отбор признаков[6](feature selection) и извлечение признаков[7](feature extraction).

Стоит отметить, что в реальной работе применяются простые алгоритмы. Один из самых распространенных алгоритмов для извлечения признаков это метод главных компонент[8](Primary Component Analysis, PCA). Преимущество данного алгоритма в том, что с его помощью можно преобразовать исходный набор данных во множество собственных векторов ковариационной матрицы, таким образом выделяются наиболее значимые атрибуты данных, и из-за меньшей размерности их легче проецировать на плоскость.

Интересный способ интерпретации модели называется LIME[9]. Идея заключается в том, чтобы, основываясь на сопоставлении значений ввода-вывода, подобрать похожую существующую модель вместо имеющейся, но к которой можно представить в понятном виде. LIME популярен как раз из-за своей гибкости и возможности применения в любых сферах приложения машинного обучения. В роли описательной модели может использовать байесовская сеть. Недостаток этого подхода в том, что работа модели не полностью соответствует оригиналу, поскольку процесс обучения аппроксимируется с определенной точностью из соображений времени и производительности. Более того, при использовании на специфических оригинальных моделях может выдавать недетерминированный результат.

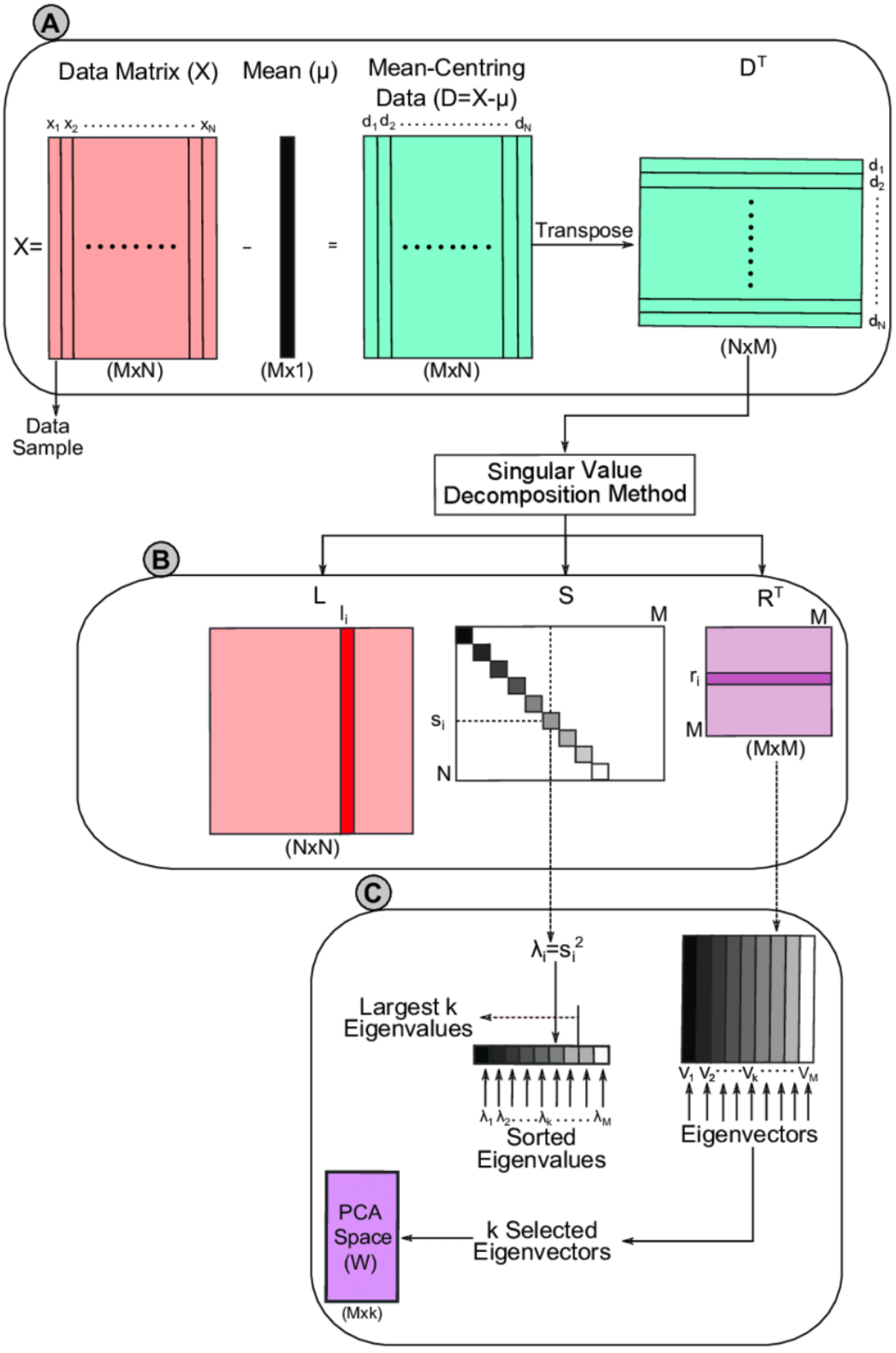

Рисунок 1 – Вычисление главных компонент с помощью сингулярного разложения

В опыте применения машинного обучения для классификации МРТ головного мозга[10] рассматривается применение алгоритма отбора признаков(обратный отбор), с помощью чего удалось достичь приемлемой точности и интерпретируемости модели. В таких случаях использование алгоритмов уменьшения размерности данных практически необходимо, поскольку экспертам нужен не только лишь результат работы, но и качественные, на которых скану был присвоен определенный класс. В данном случае удалось уменьшить размерность данных с 200 параметров до 3.

Input: Set of features $X = \{x_0, x_1, \ldots, x_n\}$, size of feature set n, size of target feature subset d
Output: Suboptimum feature subset Y_{subopt} of size d
1: $Y_{subopt} \leftarrow X$
2: **for** $j = 1 \rightarrow n - d$ **do**
3: $x \leftarrow \min(J(Y_{subopt} - \{x_i\})) \mid x_i \in X \, and \, x_i \in Y_{subopt}$
4: $Y_{subopt} \leftarrow Y_{subopt} - \{x\}$
5: **end for**

Рисунок 2 – Псевдокод алгоритма обратного отбора

Так же существуют алгоритмы уменьшения размерности, результатом работы которых является нелинейная функция. Они слабо распространены из-за сложности проекции данных на плоскость, однако на конференции ESANN регулярно проводятся сессии с докладами по этой теме[11]. Первым был разработан алгоритм Sammon mapping[12], широко применяемый в разведочном анализе данных. Нелинейное разложение применяется в случаях, когда нельзя радикально уменьшить количество параметров данных без потери информации.

4 Примеры визуализации моделей машинного обучения

Для представления публике результатов исследований или анализа данных нет лучшего решения чем визуализация. Возможность начертить трендовые линии, обозначить классификацию в цветовой палитре, интуитивно понятное обозначение результатов работы модели позволяет ускорить процесс обмена знаниями и командной работы. В качестве примера рассмотрим два варианта представления моделей.

Дерево решений довольно просто визуализировать в виде графа бинарного дерева. С помощью визуализации удобно определить какие пороговые значения определяют принадлежность элементов к классу. Количество цветов соответствует количеству классов, а их оттенок обозначает количество классифицированных элементов на развилке. Прозрачные листья появляются когда набор параметров слишком разнообразен для формирования класса(node impurity). Это может быть полезно для отделения выбросов или нежелательных классов. На рисунке 3 представлено дерево решений из пакета sklearn, график plot_tree.

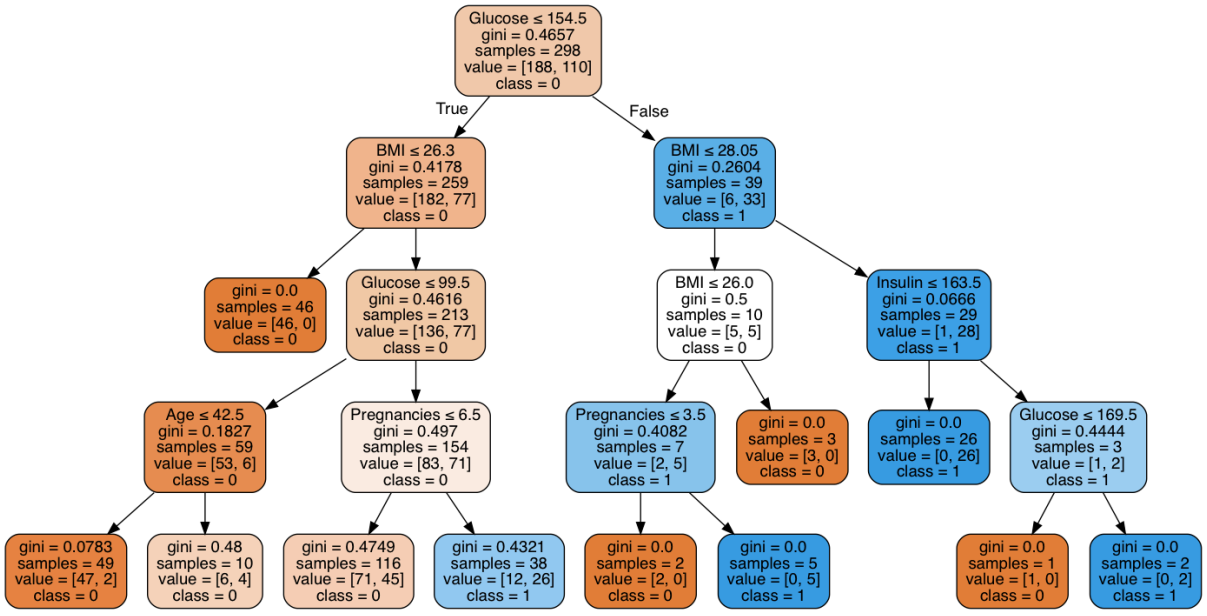

Рисунок 3 – Визуализации модели Decision Tree Classifier

Одной из самых удобных визуализаций данных можно считать Scatter Plot из библиотеки matplotlib. На графике возможно пометить классы, отобразить легенду, даже линейно неразделимые данные, классифицированные с помощью алгоритма DBSCAN, как представлено на рисунке 4.1. Трехмерное отображение поддерживается в том числе.

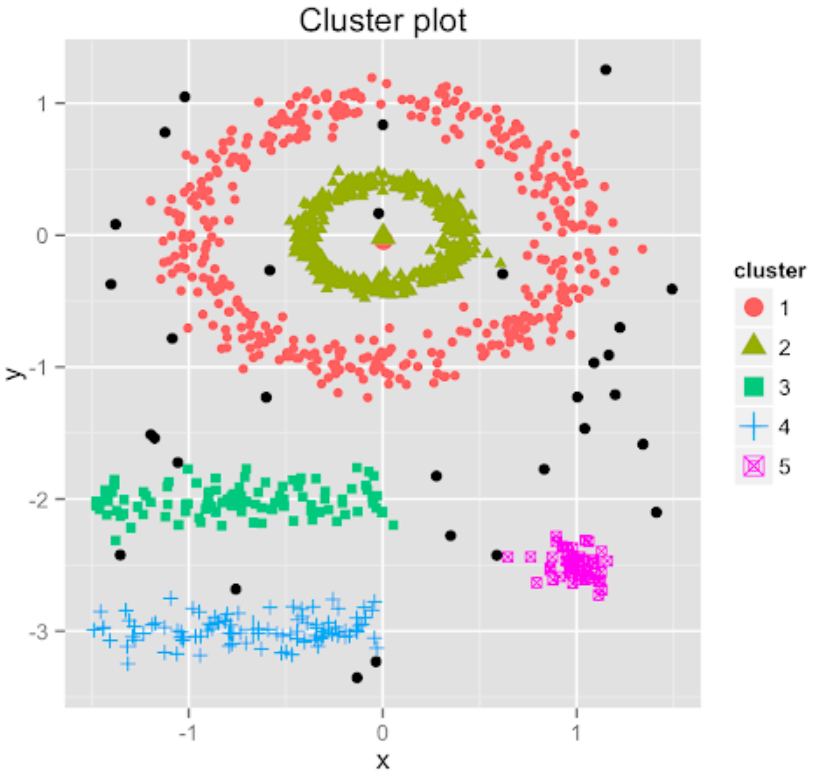

Рисунок 4.1 – Визуализация работы DBSCAN на Scatter plot графике

Ансамблевые модели состоят из нескольких простых(базовых) моделей, соединенных в большую. Например, случайный лес состоит из нескольких деревьев решений. В данном случае необходимо увидеть вклад каждой из базовых моделей в общий результат. На точечной диаграмме можно закрасить область по границам класса и найти пересечения с наложив изображения друг на друга.

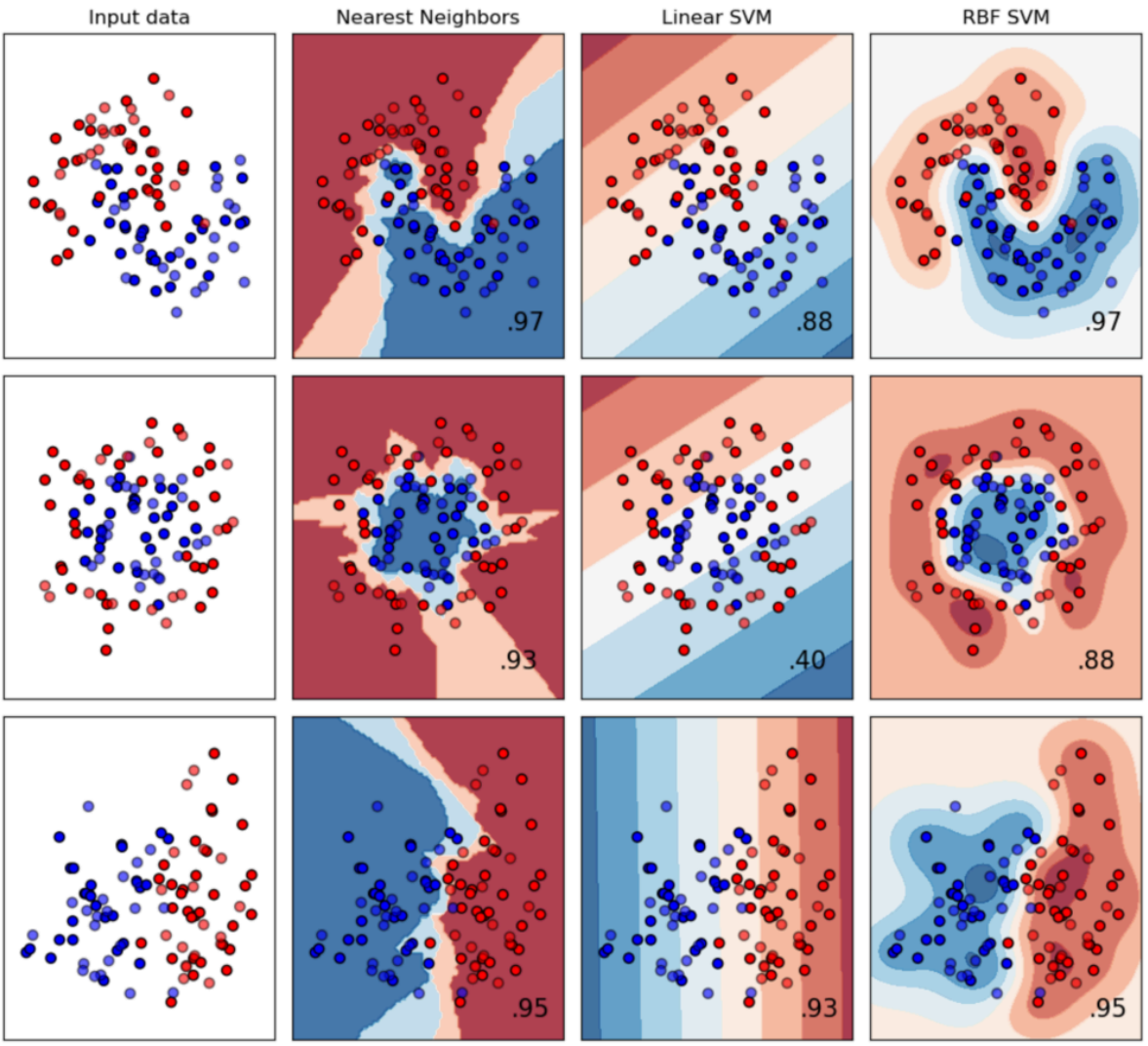

Рисунок 4.2 – Точечная диаграмма классов присвоенных базовыми моделями в ансамбле. Более темный оттенок означает большую вероятность принадлежности к классу

5 Использование байесовских сетей для повышения интерпретируемости моделей ИИ

Байесовскую сеть можно представить несколькими способами: с уклоном на саму модель, процесс принятия решений или демонстрацию свидетельств. Представление самой модели в виде графа и представление шагов принятия решений удобно для пользователя, представление свидетельств помогает экспертам исследовать непосредственно моделируемое событие.

Интерпретация модели в широком смысле заключаются в ее изображении или словесном объяснении. Когда график слишком большой и не помещается на экран, некоторые программы для отображения БС свертывают подграфы в похожие на вершины элементы, которые можно развернуть при необходимости[13].

Байесовские сети широко используются для моделирования вероятностный событий так как с помощью них можно компактно изобразить совместное распределение многих случайный событий $p(X_1, \ldots, X_n)$. При знании совместного распределения возможно вычислить вероятность любого события зная значения для других. Однако на практике совместное распределение вычислить сложно в больших сетях по двум причинам: для его вычисления требуется слишком много параметров (например, 2^n-1 для двоичных значений), и такое количество параметров сложно понять эксперту для оценки надежности модели. На рисунке 5 представлен ациклический граф простой байесовской сети.

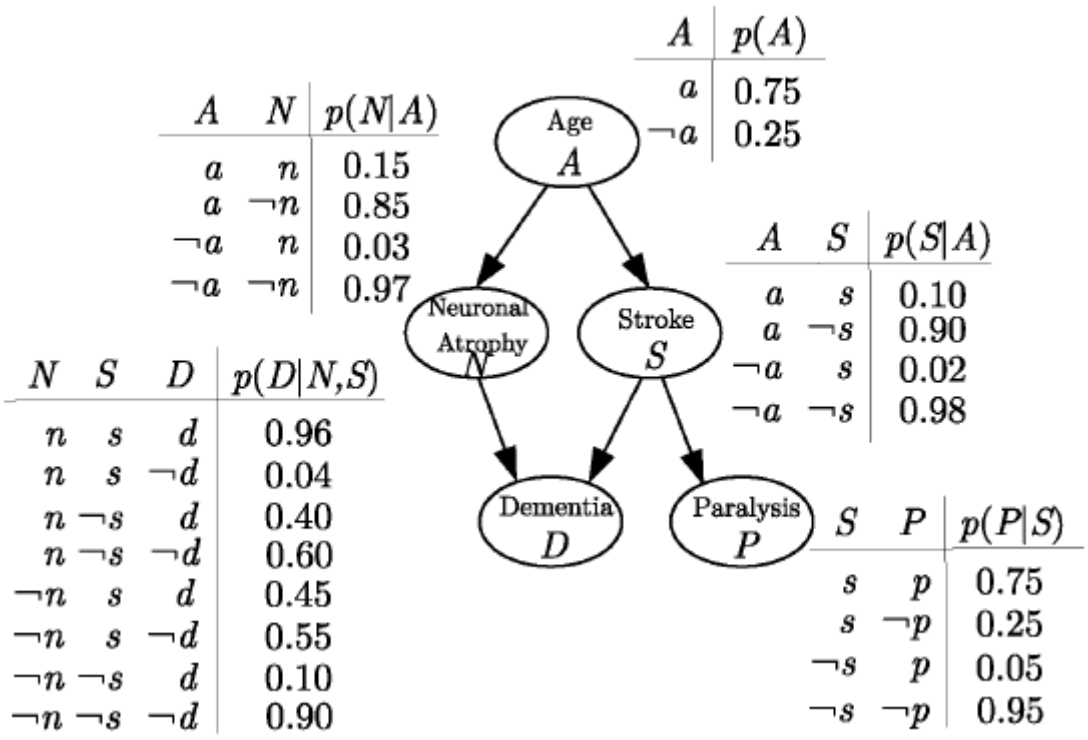

Рисунок 5 – Гипотетическая БС для моделирования риска развитии деменции. Факторами риска выступают возраст, пережитый инсульт, энцефалопатия

Помимо визуализации связи между переменными и подтверждения вероятностной зависимости событий, при работе с байесовскими сетями можно применять прогнозирование, диагностирование и обратный вывод. Это позволяет вычислять апостериорную маргинальную вероятность событий или апостериорный максимум с помощью вероятностных запросов.

Нахождение апостериорной вероятности(событие x произошло на основании свидетельства e) производится вычислением p(x|e). В системе также могут присутствовать не наблюдаемые события Y, но они не участвуют в вычислении. К примеру, на рисунке 6 показано, как вероятность паралича у пациента поднимается 8% до 75% после пережитого инсульта. В данном случае наличие энцефалопатии не влияет на наличие паралича.

Вычисление апостериорного максимума производится через нахождения набора переменных, лучше всех объясняющих некое свидетельство. Апостериорный максимум равняется $\max_y p(y|e)$, а решение называется наиболее вероятностным объяснением, которое для пациента с параличом выглядит так: возраст 65 лет, пережил инсульт, не имеет деменции или энцефалопатии.

Для вычисления этих вероятностей требуется нахождение y для каждой из переменной системы, поэтому сложность алгоритма растет экспоненциально, поэтому существуют специальные алгоритмы для расчета точного и приближенного вывода[14].

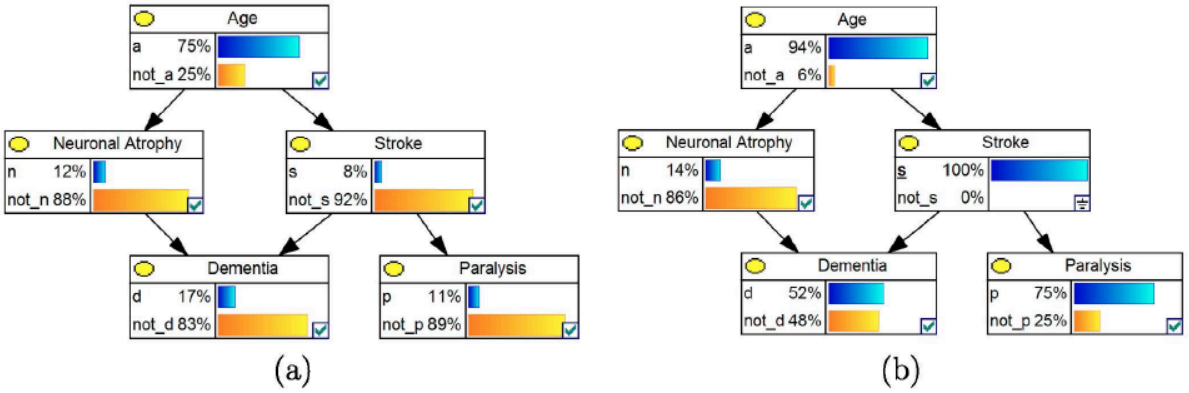

Рисунок 6 – Динамика состояний байесовской сети

Вычисление точного вывода является NP-сложной задачей, это значит что решения задачи за полиномиальное время скорее всего не существует. Метод грубой силы не подходит для больших моделей, поэтому в общем случае применяются алгоритмы устранения переменных и пропагации сообщений между узлами[15].

Для сложных сетей или нестандартных распределений приходится рассчитывать приближенный вывод, что тоже является NP-сложной задачей. Алгоритмы базируются на основе частичного вывода. БС используется для сэмплирования некоторого количества частиц (исходов) из совместного распределения, а затем искомая вероятность вычисляется на выборке. Самый простой алгоритм это формирование выборок с

исключением, в котором исходя из топологической сортировки вершин производится сэмплирование нужных вершин и их родителей. Проблема алгоритма в том, что если некое свидетельство маловероятно, выборка вершин будет разрознена и граф получится несвязным, в этом случае не удастся исследовать все свидетельства, поскольку они будут недостижимы из некоторых состояний системы. Алгоритм оценки выборок с учетом правдоподобия решает эту проблему путем выборки свидетельств для сэмплирования с учетом вероятности происхождения их родителя, которая должна быть выше заданного значения, таким образом выбросы, мешавшие построить связный граф, пропадают из выборки[16].

6 Обзор существующих способов визуализации БС

Исходя из прошлого раздела, выделим основные критерии качественного представления графа БС: отображение БС в виде графа с возможностью развернуть каждую вершину и увидеть состояние, возможность моделирования процесса обучения, возможность масштабирования, отображение описательной статистики датасета.

В первую очередь рассмотрим bayes server. Программа доступна онлайн или в качестве устанавливаемого пакета для популярных ОС. Можно настраивать вывод различных статистик, но структуру сети в виде графа представить нельзя. Удобного использовать для получения представления о структуре обработанных данных и дальнейшей работы над ними, о результатах решенной задачи(например, кластеризации). Анимацию работы тоже посмотреть нельзя. Возможно задавать вероятностные запросы, применять разные типы выводов. Интерфейс представлен на рисунке 7.

Рисунок 7 – Интерфейс веб-приложения bayes server

Следующая в очереди программа это BNViewer[19], распространяемая в виде R библиотеки. В отличие от bayes server предоставляет возможность отображения структуры сети и анимации.

Вывод статистики результатов работы статистик не является частью библиотеки, но это возможно сделать с помощью средств самого языка R. В общем полезная библиотека для визуализации модели “для пользователя”, а “для эксперта” не составит труда получить данные самостоятельно. Интерфейс представлен на рисунке 8. График составлен этим кодом:

```
viewer(bn.learn.hc,
   bayesianNetwork.width = "100%",
   bayesianNetwork.height = "80vh",
   bayesianNetwork.layout = "layout_with_sugiyama",
   bayesianNetwork.title = "Discrete Bayesian Network - Alarm",
   bayesianNetwork.subtitle = "Monitoring of emergency care patients",
   bayesianNetwork.footer = "Fig. 1 - Layout with Sugiyama")
```

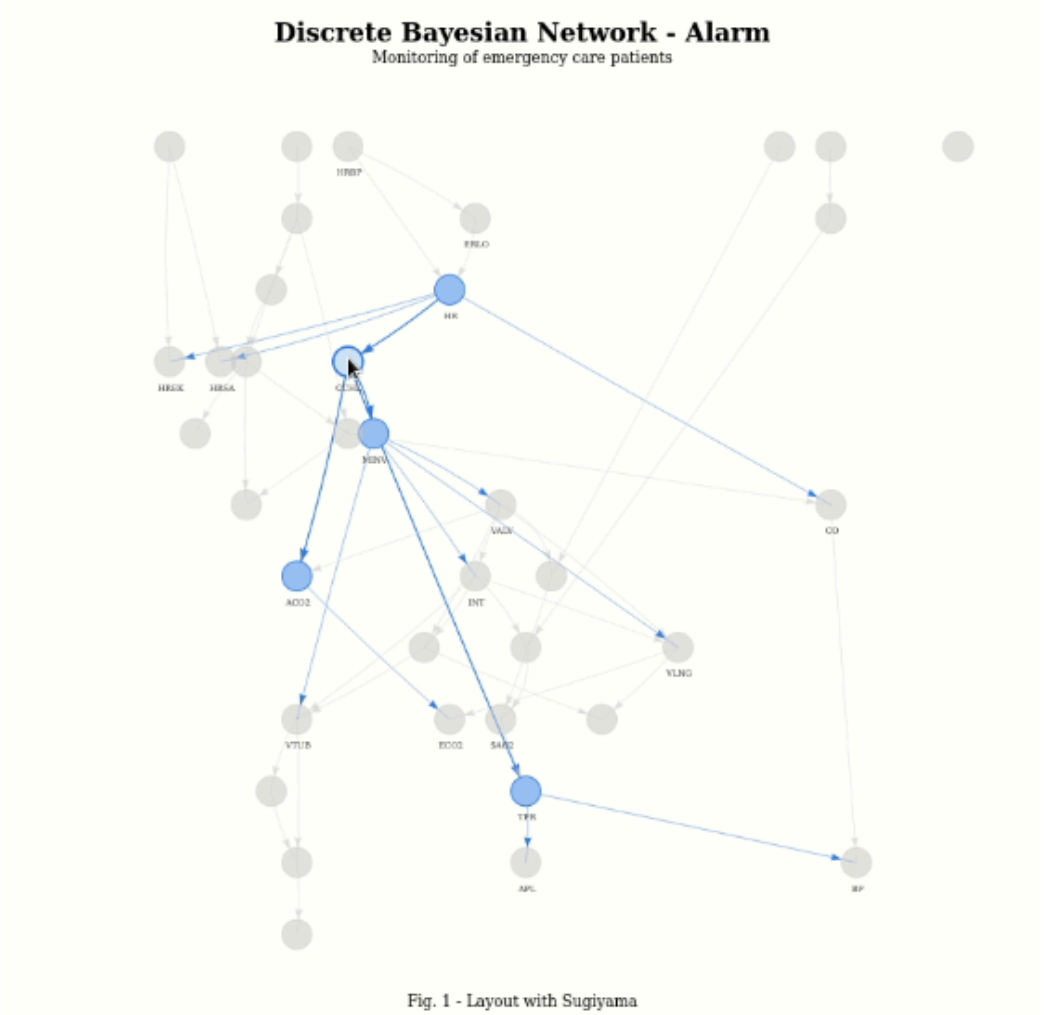

Fig. 1 - Layout with Sugiyama

Рисунок 8 – Результат визуализации модели, обученной на датасете Alarm

Следующая программа называется ShinyBN[17]. Это тоже R фреймворк, созданный на основе bnlearn, gRain, visNetwork, pROC и rmda. Отличительной особенностью является поддержка нескольких форматов ввода данных, из Excel таблицы или из R объектов. С помощью gRain поддерживается разрешение вероятностных запросов и выводов. Дизайн приложения на рисунке 9.1. Результат работы программы виде графа на рисунке 9.2.

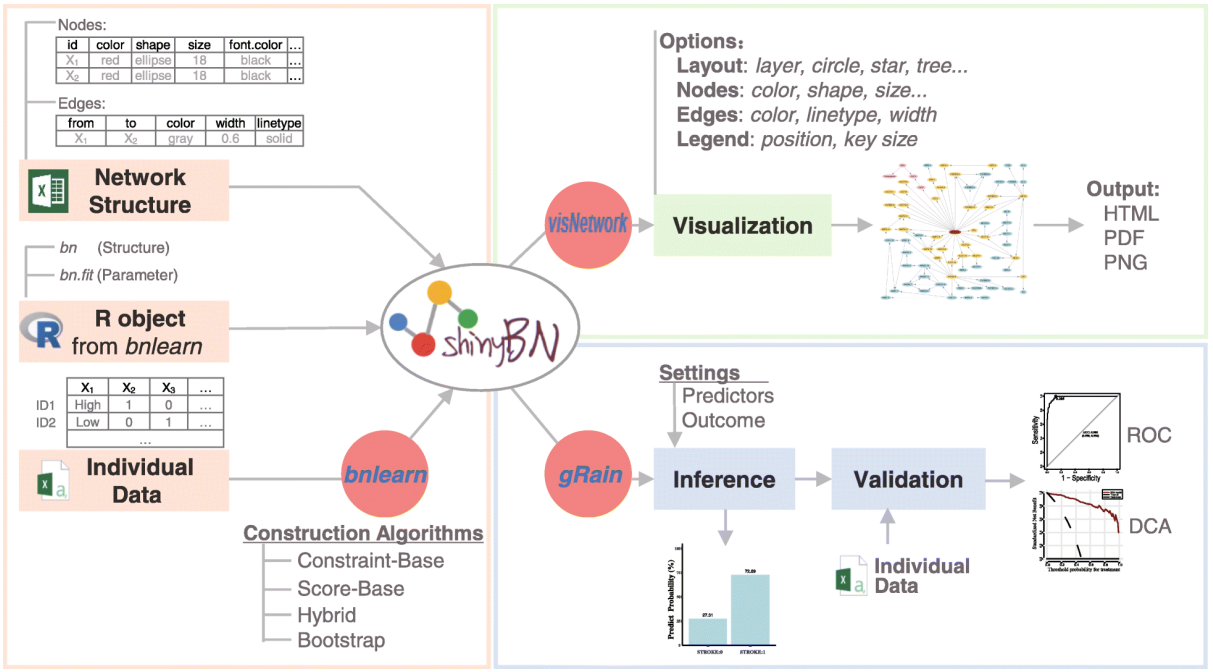

Рисунок 9.1 – Дизайн ShinyBN

Сравнение преимуществ и отличительных качеств описанных программ приведено в таблице 1.

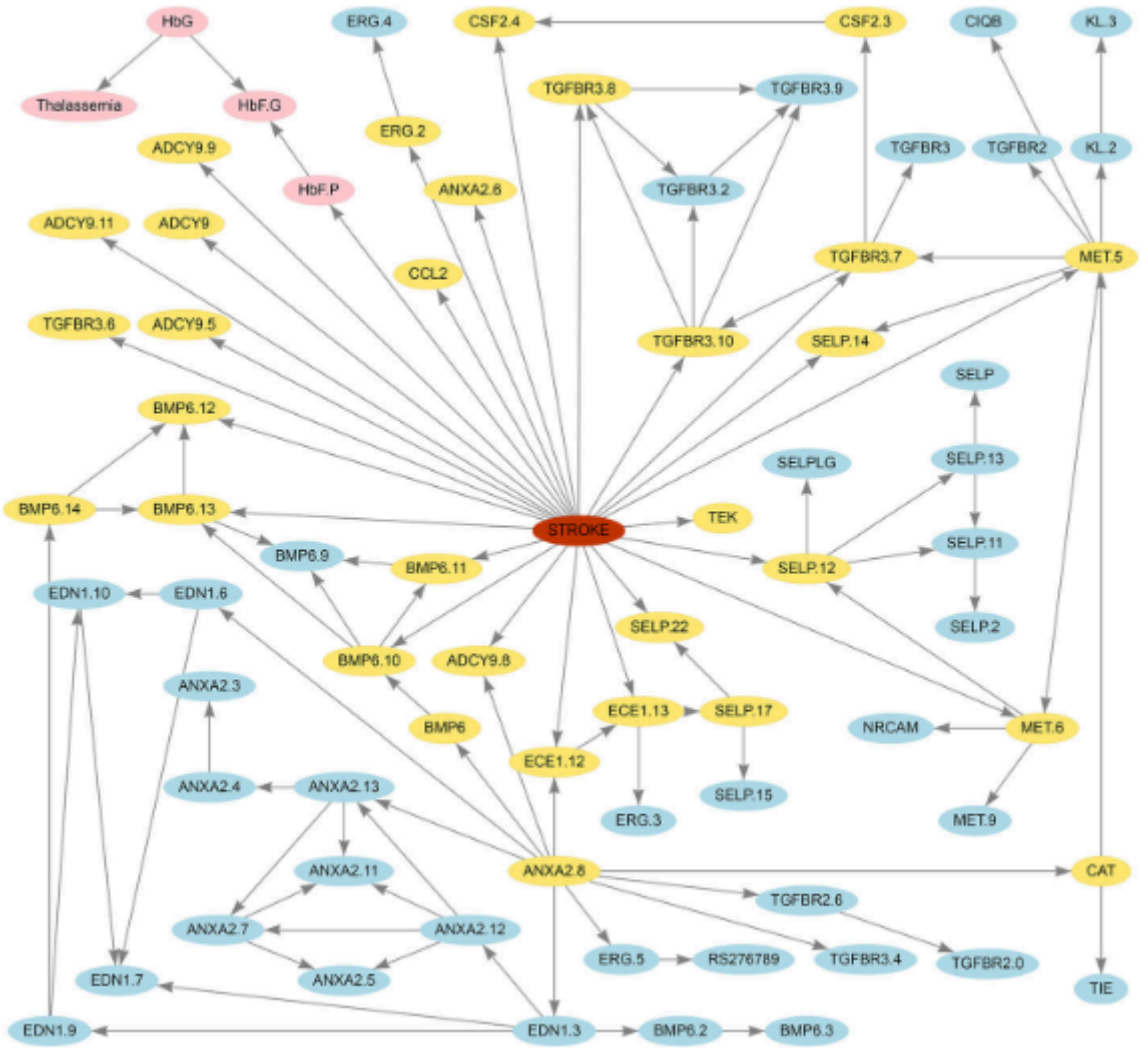

Рисунок 9.2 – Вывод графика БС

Таблица 1 – Сравнение существующих программ для визуализации БС

Название	Отображение графовой структуры	Моделирование процесса обучения	Сводка о результатах работы	Веб-приложение
Bayes server	Нет	Нет	Возможно задавать вероятностные запросы и генерировать блоки со статистикой	Да

BNViewer	Есть	Есть	Дополнительно с помощью других функций R	Нет
ShinyBN	Есть	Нет	Встроенное в фреймворк API вероятностных запросов на основе библиотеки gRain	Да

7 Заключение

В соответствии с целью литературного обзора были рассмотрены основные методы интерпретации моделей машинного обучения, проведен разбор применимости БС для развития XAI, рассмотрены существующие программы для визуализации БС.

В качестве ориентира для сравнения был выбран Bayes server, однако это приложение не позволяет рассмотреть структуру БС в виде графа, в то время как остальные реализации имеют эту функцию. В остальных качествах Bayes server, по большей части из-за того что программа не зависима от языка R, ее можно портировать на другие системы и фреймворки.

Учитывая это, есть пространство для улучшения текущей реализации. Возможно добавить визуализацию процесса обучения и структуры сети, добавить API для вероятностных запросов к дополнение к имеющемуся графическому интерфейсу.

8 Список использованной литературы

1. Gefen, David. “E-Commerce: The Role of Familiarity and Trust.” Omega, Vol. 28, No. 6, 2000 (https://www.sciencedirect.com/science/article/pii/S0305048300000219)

2. A. Adadi and M. Berrada, “Peeking inside the black-box: A survey on Explainable Artificial Intelligence(XAI)”, IEEE Access, 6, 2018, pp. 52138-52160.

3. A. Holzinger, et al., “What do we need to build explainable AI systems for the medical domain?”, arXiv preprint arXiv1712.09923, 2017.

4. Andras et al., (2018) “Trusting Intelligent Machines: Deepening Trust Within Socio-Technical Systems”. IEEE Technology and Society Magazine, 37(4), 2018, pp. 6-83.

5. E.J. Langer et al., “The mindlessness of ostensibly thoughtful action: The role of" placebic" information in interpersonal interaction”, Journal of Personality and Social Psychology, 36(6), 1978, pp. 635-642.

6. G. Dy, C.E. Brodley, Feature subset selection and order identification for unsupervised learning. In proceedings of the 17th International Conference on Machine Learning (ICML 2000), Morgan Kaufmann Publishers Inc., pages 247-254, Standford, CA (USA), 2000.

7. I. Guyon, S. Gunn, M. Nikravesh, L.A. Zadeh (Eds.), Feature Extraction: Foundations and Applications. Studies in Fuzziness and Soft Computing, Springer, 2006

8. I.T. Jolliffe, Principal Component Analysis. Springer; 2nd edition, 2002.

9. Ribeiro, M.T., Singh, S., Guestrin, C.: Why should I trust you?: explaining the predictions of any classifier. In: 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2016), pp. 1135–1144. ACM (2016)

10. A. Vellido, E. Romero, M. Juli`a-Sap´e, C. Maj´os, `A Moreno-Torres, C. Ar´us, Robust discrimination of glioblastomas from metastatic brain tumors on the basis of single-voxel proton MRS. NMR in Biomedicine. Accepted for publication. doi: 10.1002/nbm.1797.

11. A. Wismueller, M. Verleysen, M. Aupetit, J.A. Lee, Recent advances in nonlinear dimensionality reduction, manifold and topological learning. In M. Verleysen, editor, proceedings of the 18th European Symposium on Artificial Neural Networks (ESANN 2010), d-side pub., pages 71-80, Bruges (Belgium), 2010

12. Cossalter, Michele, Ole J. Mengshoel, and Ted Selker. "Visualizing and Understanding Large-Scale Bayesian Networks." *Scalable Integration of Analytics and Visualization*. 2011.

13. Henderson, Paul. "Sammon mapping." *Pattern Recognit. Lett* 18.11-13 (1997): 1307-1316.

14. Holtzen, Steven, Guy Van den Broeck, and Todd Millstein. "Scaling exact inference for discrete probabilistic programs." *Proceedings of the ACM on Programming Languages* 4.OOPSLA (2020): 1-31

15. S. Lauritzen, D. Spiegelhalter, Local computations with probabilities on graphical structures and their application to expert systems, J. R. Stat. Soc. Ser. B (MethodoL) 50 (1988) 157-224

16. R. Fung, K.-C. Chang, Weighing and integrating evidence for stochastic simulation in Bayesian networks, in: Uncertainty in Artificial Intelligence, North-Holland, 1990, pp. 209-219.

17. Chen, Jiajin, et al. "shinyBN: an online application for interactive Bayesian network inference and visualization." *BMC bioinformatics* 20 (2019): 1-5.

18. GHERHEŞ, VASILE. "Why are we afraid of artificial intelligence (AI)." *European Review Of Applied Sociology* 11.17 (2018): 6-15.

19. Fernandes, Robson, and Maintainer Robson Fernandes. "Package ‘dbnlearn’." (2020).