

Использование иерархического поиска ближайших соседей для заполнения пропущенных значений в табличных данных

С. А. Артамонов*, А. М. Миронов[‡]

Пропущенные значения являются распространённой проблемой в анализе данных, для решения которой разработано множество методов. Одним из наиболее популярных подходов является заполнение пропущенных значений с использованием метода k-ближайших соседей (KNN), основанного на информации от схожих наблюдений. Данный подход широко применяется в анализе больших данных, где объёмы выборок могут достигать миллиардов объектов, что делает задачу ускорения KNN особенно актуальной из-за его линейной сложности. В данной работе предлагается алгоритм ускоренного заполнения пропущенных значений на основе методов приближённого поиска ближайших соседей (ANN) и анализируется соотношение между скоростью работы и качеством заполнения. Экспериментальные результаты показывают, что использование ANN позволяет существенно сократить время вычислений при сохранении высокого качества заполнения пропущенных значений.

Ключевые слова: заполнение пропущенных значений в табличных данных, приближенный поиск ближайших соседей.

1. Введение

Проблема пропущенных значений при обработке табличных данных широко распространена, и обычно приводит к снижению качества обработки

* *Артамонов Сергей Александрович* — аспирант каф. математической теории интеллектуальных систем мех.-мат. факультета МГУ, e-mail: sergei.artamonov@math.msu.ru, ORCID: 0009-0001-6368-8266.

Artamonov Sergei Aleksandrovich — PhD student, Department of Mathematical Theory of Intelligent Systems, Faculty of Mechanics and Mathematics, Lomonosov Moscow State University.

[‡] *Миронов Андрей Михайлович* — к.ф.-м.н., доцент каф. математической теории интеллектуальных систем мех.-мат. факультета МГУ, e-mail: andrew.mironov@math.msu.ru, ORCID: 0000-0002-9132-7804.

Mironov Andrei Mikhailovich — PhD, Associate Professor, Department of Mathematical Theory of Intelligent Systems, Faculty of Mechanics and Mathematics, Lomonosov Moscow State University.

такого рода данных при использовании алгоритмов классического машинного обучения. Наличие пропущенных значений в данных создаёт серьёзные ограничения для применения ряда алгоритмов машинного обучения, особенно тех, которые основаны на функциях потерь, допускающих аналитическое представление, и требуют полного числового представления признаков. Пропущенные значения способны приводить к систематическому искажению распределений признаков и, как следствие, к смещению оценок параметров модели. Это происходит в тех случаях, когда механизм возникновения пропусков не является полностью случайным. Искажения в распределении признаков описаний приводят к переобучению алгоритмов классического машинного обучения, поэтому пропущенные значения могут существенно влиять на точность и обобщающую способность моделей машинного обучения [1].

На сегодняшний день существует широкий спектр алгоритмов заполнения пропущенных значений в табличных данных, однако не существует единой зарекомендовавшей себя методики выбора наиболее надёжного алгоритма с учетом заданных ограничений по вычислительным ресурсам и времени. Иногда для достижения приемлемого качества достаточно использовать базовые методы — такие как заполнение пропущенных значений постоянной величиной, нулём, средним или медианой соответствующего признака.

Причины возникновения пропущенных данных разнообразны и включают ошибки на этапе сбора информации (например, при проведении опросов или измерений), неточности при ручном вводе данных, а также сбои при передаче информации. В некоторых случаях значения теряются или искажаются в процессе предобработки, либо вовсе не собираются. Пропущенные значения могут существенно влиять на точность и обобщающую способность моделей машинного обучения [1]. Хотя некоторые современные алгоритмы способны обрабатывать пропущенные значения без дополнительной предобработки, такие подходы нередко сопровождаются снижением качества модели. Один из известных примеров — алгоритм *eXtreme Gradient Boosting* [2], обладающий встроенными механизмами обработки пропущенных значений. В работе Aydin и соавт. [3] показано, что модели *XGBoost* сами по себе зачастую демонстрируют худшие результаты по сравнению с подходами, использующими специализированные методы заполнения пропущенных значений. Подобные подходы отличаются вычислительной эффективностью и простотой реализации. Однако более сложные методы могут обеспечить более высокую точность восстановления пропущенных значений, что сопровождается увеличением вычислительных затрат. Одним из наиболее широко используемых подходов является метод *KNN Imputer* [4], основанный на

алгоритме к ближайших соседей. Данный метод в настоящее время применяется в самых различных предметных областях. Несмотря на свою эффективность, этот подход имеет существенный недостаток — высокую вычислительную сложность. Как отмечают Murti и соавт. [5], Аси́ña и Rodríguez [6], а также многие другие исследователи, метод требует анализа всего набора данных для поиска схожих объектов, что делает его особенно ресурсоёмким при работе с большими объёмами данных.

В то же время существует класс алгоритмов приближённого поиска ближайших соседей (Approximate Nearest Neighbor, ANN), разработанных с целью ускорения стандартного метода к ближайших соседей путём разбиения пространства поиска и ограничения области подбора соседей для каждого запроса только релевантным подмножеством этого пространства [7–9]. В настоящее время известен широкий спектр методов и программных пакетов для реализации ANN-подходов. В работе Aumüller и соавт. [10] проведено сравнение наиболее распространённых алгоритмов приближённого поиска ближайших соседей.

В этой работе мы проводим анализ возможных рисков, связанных с применением алгоритма Hierarchical Navigable Small World для ускорения процесса заполнения пропущенных значений, а также проводим оценку этого ускорения на специально собранном с этой целью наборе данных.

2. Обзор литературы

2.1. Пропущенные значения

Проблема пропущенных значений исследуется с середины XX века; одними из первых значимых работ в этой области стали исследования Beale и Little [35], в которых проводилась сравнительная оценка ранних алгоритмов заполнения пропусков. Эти методы, как правило, интерпретировали заполнение как задачу регрессии на основе полных данных. Современные подходы сохраняют эту концептуальную основу, но используют более совершенные регрессионные алгоритмы. Одним из наиболее распространённых на практике методов остаётся KNN Imputer [4], реализующий регрессионный подход посредством алгоритма к ближайших соседей.

В последние годы было предложено множество более сложных методов восстановления пропущенных данных. Так, в 2018 году Yoon и соавт. представили подход GAIN, основанный на генеративно-сопоставительных сетях (GAN) [36]. В том же году Ma и соавт. предложили алгоритм импутации на основе байесовского вывода [37]. Среди других популярных методов можно отметить MICE [38], MissForest [39], а также методы, основанные на автоэнкодерах, такие как MIDA [40].

Несмотря на достижения в области обработки пропущенных данных, метод KNN Imputer остаётся широко используемым при работе с большими наборами данных — например, в банковской и медицинской отраслях — благодаря рациональному соотношению вычислительной эффективности, простоты реализации и масштабируемости. В данной работе мы не стремимся превзойти современные подходы к заполнению пропусков по качеству восстановления данных. Цель исследования состоит в том, чтобы предложить усовершенствование одного из наиболее распространённых методов и оценить его влияние на вычислительные затраты и точность заполнения пропусков.

Важно отметить, что не все пропущенные значения имеют одинаковую природу. Широко используемая классификация пропущенных данных основана на связи между отсутствующими значениями и наблюдаемыми или ненаблюдаемыми величинами в наборе данных, а также на факторах, вызывающих пропуски [17, 18]. Выделяют три основных типа:

- **Missing completely at random (MCAR):** В этом случае пропуски возникают полностью случайно и не зависят ни от наблюдаемых, ни от ненаблюдаемых значений в наборе данных. Вероятность отсутствия значения одинакова для всех наблюдений. MCAR могут возникать из-за случайных технических проблем или внешних факторов, не связанных с данными.
- **Missing at random (MAR):** Здесь вероятность пропуска связана с другими наблюдаемыми переменными, но не зависит от самого отсутствующего значения. Например, данные могут чаще отсутствовать для людей определённого возраста или профессии, то есть пропуски объяснимы известными признаками.
- **Missing not at random (MNAR):** В данном случае вероятность пропуска зависит от значения самого отсутствующего элемента. Например, если респондент не указывает свой доход в налоговой декларации, чтобы избежать раскрытия информации, вероятность пропуска напрямую зависит от значения величины. Работа с MNAR-данными особенно сложна, так как механизм возникновения пропусков не наблюдаем.

2.2. Приближенный поиск ближайших соседей

Методы приближённого поиска ближайших соседей (Approximate Nearest Neighbors, ANN) представляют собой эффективный способ ускорения поиска ближайших соседей и находят широкое применение в различных областях науки. В частности, в компьютерном зрении поиск ближайших

соседей активно используется для идентификации схожих изображений в больших коллекциях [11][12]. Аналогично, в задачах обработки естественного языка эти методы применяются для извлечения релевантной информации из крупных текстовых корпусов, что характерно для систем типа RAG [13]. В связи с этим методы приближённого поиска ближайших соседей стали популярными как средство значительного повышения вычислительной эффективности систем, основанных на поиске ближайших соседей [14].

В данной работе мы используем один из хорошо известных методов приближенного поиска ближайших соседей — алгоритм Hierarchical Navigable Small World (HNSW), предложенный Malkov и Yashunin в 2016 году [19]. Метод основан на построении иерархической структуры графов Small World [20], где элементы набора данных представлены вершинами графа. На сегодняшний день HNSW является одним из наиболее широко используемых методов для работы с наборами данных среднего размера [26]. В нашем исследовании применяется наиболее распространённая реализация алгоритма, доступная через программный пакет HNSWlib.

Выбор рассматриваемого метода был основан на соображениях, изложенных в работе [26], где предлагается стратегия подбора индексных структур в зависимости от ключевых характеристик данных. В рамках данного подхода методы индексирования классифицируются на плоские индексы, графовые индексы и индексы на основе инвертированных файлов (IVF). Метод HNSW является наиболее распространённым и практически значимым представителем графовых индексных структур для приближённого поиска ближайших соседей и рассматривается как основной метод для задачи заполнения пропусков в датасетах умеренного размера, не относящихся ни к малым, ни к крайне крупным выборкам.

3. Предлагаемый метод

В данной работе предлагается новый метод заполнения пропущенных значений, основанный на общей идее метода k ближайших соседей (KNN). Основной целью является устранение одного из наиболее существенных недостатков классического подхода KNN при обработке больших массивов данных — длительного времени работы при применении к большим наборам данных, обусловленного линейной сложностью данного алгоритма. Необходимость вычисления расстояний между всеми парами объектов в наборе данных становится серьёзным ограничением, особенно при работе с промышленными датасетами умеренного и крупного размера.

Для преодоления данного ограничения в работе применяется приближённый поиск ближайших соседей, позволяющий существенно снизить вычислительные затраты по сравнению с точным KNN-поиском. Стандартная процедура поиска k ближайших соседей заменяется приближённым поиском на основе индексной структуры Hierarchical Navigable Small World.

Схематическое описание предлагаемого метода представлено в виде следующего алгоритма:

Входные данные: неполная матрица данных $X \in \mathbb{R}^{N \times d}$.

Выходные данные: матрица с заполненными пропущенными значениями $\tilde{X} \in \mathbb{R}^{N \times d}$.

1. Инициализировать результирующую матрицу $\tilde{X}$ на основе исходной матрицы X .
2. Заменить все пропущенные значения (NaN) в матрице X нулевыми значениями, получив матрицу $\bar{X}$.
3. Построить ANN-индекс I в соответствии с выбранной стратегией приближённого поиска ближайших соседей.
4. Для каждого объекта $i = 1, 2, \dots, N$ выполнить следующие действия:
 - а) Обозначить через x_i i -ю строку матрицы $\bar{X}$.
 - б) С использованием ANN-индекса I найти k приближённых ближайших соседей для вектора x_i :

$$X_{\text{nearest}} = I(x_i, k).$$

- в) Для каждого пропущенного значения в i -й строке матрицы $\tilde{X}$:
 - i. Исключить из множества X_{nearest} элементы, содержащие пропуск в искомом столбце и заменить эти элементы на следующих соседей, в соответствии с ANN-индексом I , пока не будет набрано нужное количество приближённых ближайших соседей
При практической реализации удобно выполнять этот пункт, беря небольшой запас по k на шаге 4.б
 - ii. вычислить среднее значение соответствующего признака по множеству X_{nearest} , исключая пропущенные значения (NaN);
 - iii. заменить пропущенное значение полученным средним.
5. Вернуть матрицу $\tilde{X}$ как результат работы алгоритма.

На первом этапе все пропущенные значения в исходном наборе данных заменяются нулевыми значениями. Данный шаг предварительной обработки соответствует подходу, используемому в наиболее распространённой реализации метода заполнения пропущенных значений на основе KNN, представленной в библиотеке `scikit-learn` [15]. Такое заполнение необходимо для обеспечения корректного вычисления расстояний между объектами.

Далее на основе полученной матрицы данных строится индекс HNSW, который используется для выполнения приближённого поиска ближайших соседей. После построения индекса для каждой строки матрицы X определяется множество из K ближайших соседей. Следует отметить, что данный этап допускает наличие некоторой погрешности, что является характерной особенностью приближённых методов поиска ближайших соседей.

На заключительном этапе пропущенные значения в каждой строке восстанавливаются путём замены их на среднее значение соответствующего признака, вычисленное по найденному множеству соседей. При этом значения NaN, присутствующие у самих соседей, исключаются из расчёта среднего.

4. Данные

4.1. Искусственное создание пропущенных значений

Для оценки качества заполнения пропущенных значений в экспериментах использовались наборы данных, не содержащие исходных пропусков. В связи с этим пропущенные значения вводились искусственно, что потребовало корректного моделирования различных механизмов возникновения значений NaN. В соответствии с общепринятыми определениями были реализованы различные стратегии внесения пропусков, соответствующие каждому типу пропущенных значений.

MCAR (Missing Completely At Random). В случае MCAR пропущенные значения возникают полностью случайным образом и не зависят ни от наблюдаемых, ни от ненаблюдаемых переменных. Соответственно, стратегия внесения NaN заключается в случайном выборе индексов в каждом столбце с последующей заменой соответствующих значений на NaN. Каждое значение в наборе данных имеет одинаковую вероятность быть удалённым, при этом общая доля пропусков определяется параметрами эксперимента. Формально данный процесс может быть описан

следующим образом:

$$\text{NaN}_{i,j} \sim \text{Bern}(p), \quad (1)$$

где p — фиксированный параметр, задающий вероятность замены значения на NaN, i и j — индексы строки и столбца соответственно, а $\text{NaN}_{i,j}$ — бинарная переменная, указывающая на наличие пропущенного значения.

MAR (Missing At Random). В сценарии MAR вероятность появления пропущенного значения зависит от значений других признаков в наборе данных. Для моделирования данного типа пропусков предполагается, что значения NaN вводятся в некоторый столбец j , а вероятность пропуска определяется функцией от остальных признаков в той же строке. В частности, вычисляется нормированная сумма всех признаков строки, за исключением целевого столбца j , которая используется для задания вероятности внесения пропуска:

$$\begin{cases} \text{NaN}_{i,j} \sim \text{Bern}(p), \\ p \propto X_{i,1} + X_{i,2} + \dots + X_{i,j-1} + X_{i,j+1} + \dots \end{cases} \quad (2)$$

Здесь i и j обозначают индексы строки и столбца, а $X_{i,k}$ — значение признака k в строке i . В рамках данной модели вероятность появления пропущенного значения в столбце j возрастает с увеличением суммы остальных признаков в соответствующей строке.

MNAR (Missing Not At Random). В случае MNAR вероятность пропуска зависит от самого значения, которое становится пропущенным. Для моделирования данного механизма вероятность появления NaN задаётся как величина, пропорциональная значению соответствующего элемента:

$$\begin{cases} \text{NaN}_{i,j} \sim \text{Bern}(p), \\ p \propto X_{i,j}. \end{cases} \quad (3)$$

Данный подход отражает типичную ситуацию MNAR, например, когда экстремальные значения с большей вероятностью оказываются пропущенными вследствие эффектов цензурирования или систематических искажений при сборе данных.

Следует отметить, что в рамках настоящего исследования пропущенные значения рассматриваются исключительно в числовых признаках. Такой подход соответствует распространённой практике в смежных исследованиях и позволяет избежать дополнительной сложности, связанной с обработкой категориальных переменных.

4.2. Наборы данных

В экспериментальной части используются следующие наборы данных, к которым применяются описанные выше методы искусственного внесения пропущенных значений.

Набор данных 1 (регрессия): California Housing Dataset [16].

Набор данных содержит информацию о жилой недвижимости в штате Калифорния, полученную на основе переписи населения США 1990 года. Он включает 20 640 наблюдений с 8 числовыми признаками.

Набор данных 2 (классификация): Loan Approval Dataset [21].

Данный набор данных содержит финансовую информацию о клиентах банка и соответствующее решение о выдаче кредита. Он включает 45 000 наблюдений с 13 признаками (8 числовых и 5 категориальных) и одну бинарную целевую переменную, принимающую значение 1 в случае одобрения кредита и 0 в противном случае.

Набор данных 3 (классификация): Air Quality Dataset [22].

Набор данных состоит из 5 000 наблюдений, описывающих качество воздуха в различных регионах, и содержит 9 числовых признаков. Целевая переменная категоризирует качество воздуха на четыре класса: хорошее, умеренное, плохое и опасное.

Набор данных 4 (классификация): MAGIC Gamma Telescope Dataset [25].

Синтетический набор данных, содержащий 19 020 наблюдений с 10 числовыми признаками, полученными на основе характеристик гамма-событий, зарегистрированных наземным черенковским телескопом. Задача состоит в классификации событий на сигналы гамма-частиц и адронный фон; целевая переменная является бинарной.

Набор данных 5 (классификация): Mushrooms Dataset [23].

Набор данных включает 54 000 наблюдений с 8 числовыми признаками, описывающими характеристики грибов. Целевая переменная указывает на съедобность гриба: значение 0 соответствует съедобным, а значение 1 — ядовитым.

Набор данных 6 (регрессия): Paris Housing Dataset [24].

Синтетический набор данных, включающий 10 000 наблюдений, описывающих цены на жильё в городской среде, смоделированной по аналогии с Парижем.

5. Численные эксперименты

5.1. Описание численного эксперимента

Потенциальные преимущества предлагаемого метода могут быть оценены через анализ соотношения между качеством заполнения пропущенных значений и временем вычислений. Соответственно, наши эксперименты направлены на количественную оценку этих двух аспектов и сравнение их с базовым методом KNN-заполнения пропусков.

В работе проводятся два типа экспериментов:

1. Прямая оценка качества заполнения пропущенных значений
2. Оценка через метрики качества задачи регрессии или классификации (далее - **базовая задача**)

Во втором случае для каждого варианта датасета обучалась модель градиентного бустинга (XGBoost) с одинаковыми гиперпараметрами для всех версий, чтобы обеспечить сопоставимость результатов.

Численный эксперимент описывается следующей последовательностью действий:

1. Ввести $p\%$ пропущенных значений в набор данных с использованием подходящей стратегии для выбранного типа пропусков.
2. Применить метод заполнения пропущенных значений и сохранить все полученные версии датасета.
3. Для каждой версии выполнить прямую оценку качества заполнения.
4. Для каждой версии обучить модель XGBoost и оценить показатели на соответствующей базовой задаче

Такой эксперимент зависит от нескольких ключевых параметров:

- Количество ближайших соседей K , используемых при заполнении пропусков.
- Доля пропущенных значений p .
- Тип пропусков: **MCAR**, **MAR** или **MNAR**.

Мы исследовали все возможные комбинации этих параметров, выполняя описанную процедуру для каждой конфигурации. Сетка значений параметров определялась следующим образом:

- Доля пропусков (p): 5%, 10%, 15%, 20%, 30%

- Тип пропусков: **MCAR**, **MAR**, **MNAR**
- Количество ближайших соседей (K): нелинейная сетка:
 - От 3 до 30 с шагом 1
 - От 30 до 100 с шагом 5
 - От 100 до 1000 с шагом 25

Также мы зафиксировали параметры, определяющие алгоритм поиска при помощи HNSW. Основные параметры - это количество двунаправленных связей, создаваемых для каждого нового элемента при построении графа (M) и количество кандидатов, которые учитываются при построении графа ($ef_construction$). Мы зафиксировали эти значения: $M = 16$, $ef_construction = 64$.

Поскольку неизвестно, существуют ли стабильные зависимости между качеством заполнения пропусков и результатами решения соответствующих базовых задач, а также как эти зависимости связаны с типом пропусков, мы используем методологию оценки, учитывающую оба аспекта.

Для оценки решения базовых задач применяются различные метрики в зависимости от типа задачи: для регрессии используется коэффициент детерминации (R^2) [27], а для классификации — F_1 -мера [28]. Одновременно мы оцениваем качество заполнения пропущенных значений напрямую, сравнивая исходные данные с восстановленными данными после применения каждого метода.

Однако не существует общепринятого способа оценки качества заполнения пропусков. Shadbahr et al. [29] выделяют три принципиально различных подхода, ни один из которых на сегодняшний день не является общепринятым. Следуя современным подходам [30, 31], мы используем наиболее простой из методов, описанных в [29], известный как оценка расхождения по образцам (sample-wise discrepancy). Этот метод сравнивает заполненные значения с исходными на уровне каждой строки, используя коэффициент детерминации (R^2).

Также представляют интерес изменения ключевых характеристик решения при различных значениях числа ближайших соседей K . Ожидается, что время заполнения пропусков будет расти с увеличением K , однако важно оценить, как качество восстановления данных и показатели качества базовых задач реагируют на изменения K . Поскольку оптимальное значение K заранее неизвестно для каждой задачи, для каждого тестируемого значения K фиксируются время заполнения пропусков, метрики решения базовых задач и качество восстановления данных.

Численные эксперименты выполнялись на CPU без использования GPU. Расчёты запускались на вычислительном кластере на базе серверных процессоров Intel Xeon с 56 физическими ядрами на узел. Для одного независимого эксперимента выделялся один поток; внутреннее распараллеливание алгоритмов не использовалось. Параллельный запуск применялся только между независимыми экспериментами.

5.2. Результаты численных экспериментов

В каждом эксперименте мы оцениваем три ключевых показателя: **(а)** качество заполнения пропущенных значений в соответствии с ранее описанной методологией оценки; **(б)** время заполнения в секундах; и **(в)** качество решения соответствующей базовой задачи — измеряемое с помощью коэффициента детерминации R^2 для задач регрессии и метрики F_1 для задач классификации. Поскольку оптимальное число ближайших соседей (K) заранее неизвестно для конкретного набора данных, мы усредняем значение рассматриваемых характеристик по всему рассматриваемому набору значений K .

Таблица 1: Среднее время заполнения пропусков для различных наборов данных

P	Датасет	T_{KNN}	T_{HNSW}
5	CH	10.1	0.72
5	Loan	44.06	1.52
5	Air	10.17	0.69
5	Mag	3.92	0.49
5	PH	59.13	1.58
5	Mus	0.63	0.16
10	CH	18.49	0.94
10	Loan	89.72	2.92
10	Air	18.12	0.92
10	Mag	6.81	0.67
10	PH	109.79	2.13
10	Mus	1.13	0.22
15	CH	25.35	1.11
15	Loan	122.24	3.16
15	Air	24.12	1.14
15	Mag	9.16	0.84
15	PH	150.73	2.64
15	Mus	1.56	0.27
20	CH	31.23	1.34
20	Loan	146.95	3.99
20	Air	29.53	1.36
20	Mag	11.59	1.01
20	PH	190.47	3.24
20	Mus	1.87	0.33
30	CH	47.34	1.75
30	Loan	194.96	4.48
30	Air	39.32	1.80
30	Mag	15.88	1.34
30	PH	253.50	4.27
30	Mus	2.50	0.43

Таблица 2: Среднее изменение качества заполнения пропущенных значений на уровне отдельных наблюдений при замене KNN на ANN на основе HNSW (%)

P%	Dataset	MAR	MNAR	MCAR
5	CH	-1.28	-0.54	-1.02
5	Loan	0.17	0.28	0.08
5	Air	-1.05	-0.90	-0.50
5	Mag	-0.92	-1.08	-1.06
5	PH	0.16	0.12	0.18
5	Mus	-0.99	-1.28	-1.40
10	CH	-1.45	-0.19	-1.23
10	Loan	0.19	0.81	0.32
10	Air	-1.75	-1.02	-1.93
10	Mag	-2.04	-2.38	-1.89
10	PH	0.27	-0.23	0.37
10	Mus	-1.84	-2.23	-2.13
15	CH	-0.09	1.48	-0.01
15	Loan	0.60	1.03	0.69
15	Air	-3.62	-2.86	-2.18
15	Mag	-2.89	-3.47	-2.38
15	PH	0.39	-0.28	0.39
15	Mus	-1.48	-2.02	-2.53
20	CH	2.98	4.24	1.18
20	Loan	1.36	1.32	1.17
20	Air	-4.74	-5.69	-3.75
20	Mag	-3.32	-4.20	-3.24
20	PH	1.10	-0.41	0.59
20	Mus	-0.94	-1.02	-1.25
30	CH	6.98	15.04	7.36
30	Loan	2.42	3.87	2.15
30	Air	-9.26	-13.7	-4.07
30	Mag	-4.61	-5.61	-3.44
30	PH	1.94	-4.18	1.85
30	Mus	2.35	2.76	1.06

Таблица 3: Среднее изменение качества решения базовой задачи при замене KNN на ANN на основе HNSW (%)

P%	Dataset	MAR	MNAR	MCAR
5	CH	-6.32	-4.80	-6.53
5	Loan	0.14	-0.25	0.10
5	Air	-0.47	-0.72	-0.40
5	Mag	-2.13	-1.10	-1.83
5	PH	0.18	0.31	0.55
5	Mus	2.02	2.20	3.00
10	CH	-3.13	-4.57	-6.07
10	Loan	0.33	0.14	-0.05
10	Air	-0.82	-1.15	-0.58
10	Mag	-3.22	-3.06	-3.33
10	PH	0.54	0.34	0.24
10	Mus	2.32	0.38	1.79
15	CH	-3.47	-3.58	-7.40
15	Loan	0.69	1.02	0.63
15	Air	-0.61	-0.87	-0.50
15	Mag	-4.26	-4.36	-4.06
15	PH	0.80	0.91	1.22
15	Mus	2.60	1.73	3.70
20	CH	-1.08	-6.49	-7.27
20	Loan	1.66	1.22	0.91
20	Air	-1.43	-2.28	-0.88
20	Mag	-3.66	-3.64	-3.84
20	PH	1.00	1.20	0.50
20	Mus	0.45	-0.17	0.91
30	CH	12.44	-5.57	2.42
30	Loan	3.34	2.31	3.00
30	Air	-0.34	-0.48	-0.62
30	Mag	1.04	-1.86	-1.37
30	PH	1.18	1.49	1.09
30	Mus	-0.60	0.40	-0.55

В таблице 1 приведено среднее время заполнения пропущенных значений для каждого из рассматриваемых наборов данных и каждого значения параметра p , агрегированное по всем типам механизмов пропусков. Основная цель предложенного метода заключается в ускорении традиционного процесса заполнения пропусков при помощи KNN Imputer. Существенных различий во времени заполнения между случаями MAR, MNAR и MCAR выявлено не было.

Следует также отметить, что время построения индекса является относительно небольшим по сравнению со временем поиска. В случае HNSW медианное время построения индекса составляет приблизительно 8% от времени поиска в среднем по всем наборам данных.

В таблице 1 показано ускорение, достигаемое за счёт замены точного поиска ближайших соседей на заполнение на основе ANN. В среднем заполнение пропусков с использованием HNSW оказывается в **17.9** раза быстрее по сравнению с классическим KNN Imputer при доле пропущенных значений 5%, в **23.8** раза быстрее при 10%, в **26.5** раза при 15%, в **26.5** раза при 20% и в **28.4** раза быстрее при 30% пропущенных значений.

В таблицах 2 и 3 представлены результаты, характеризующие влияние замены классического KNN на ANN-подход на основе HNSW на качество заполнения пропущенных значений и на качество решения базовой задачи. Таблица 2 отражает средний относительный прирост или потерю качества заполнения пропущенных значений на уровне отдельных наблюдений для различных механизмов пропусков (MAR, MNAR и MCAR), а таблица 3 показывает соответствующие изменения показателей качества решения базовых задач. Ускорение процедуры заполнения пропущенных значений достигается ценой неоднозначных изменений качества: для одних наборов данных и сценариев пропусков наблюдаются умеренные потери, тогда как в других случаях фиксируется сопоставимый или даже положительный эффект, особенно при высоких долях пропусков. Таким образом, результаты, представленные в таблицах 2 и 3, важны в контексте интерпретации полученного ускорения, поскольку они позволяют количественно оценить, к каким именно изменениям качества приводит использование ANN-методов. Для удобства анализа, результаты, представленные в этих таблицах также отображены на графиках 1, 2 и 3 (с усреднением по всем p). Хорошо видно, что методы измерения качества заполнения пропусков рассогласованы - в ряде случаев знаки изменения качества разнонаправлены. Наиболее важным критерием является изменение в оценке качества решения базовой задачи.

В целом наблюдается тенденция к увеличению относительных различий между алгоритмами по мере роста доли пропущенных данных (p). В частности (по таблице 2):

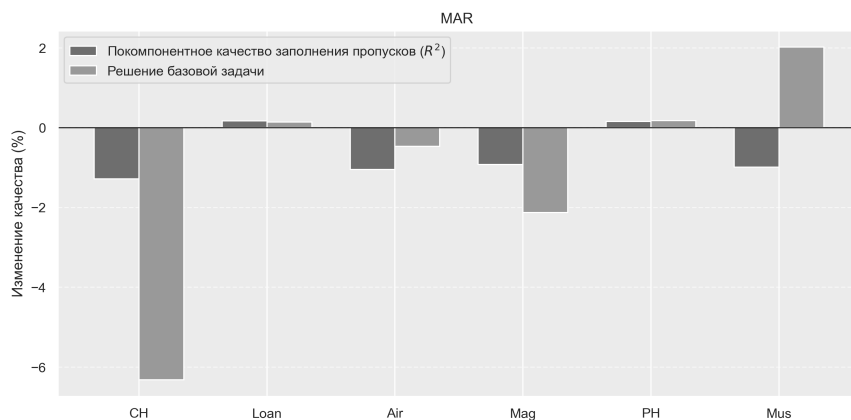


Рис. 1: Изменение качества заполнения пропущенных значений и решения соответствующей базовой задачи при переходе от KNN к ANN (MAR)

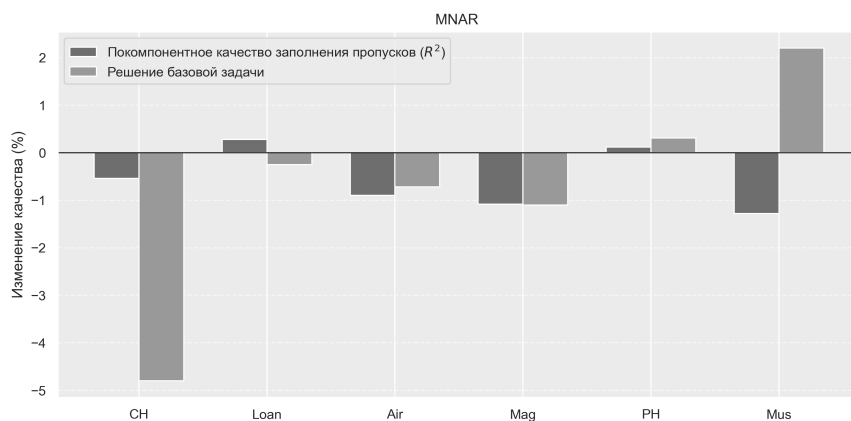


Рис. 2: Изменение качества заполнения пропущенных значений и решения соответствующей базовой задачи при переходе от KNN к ANN (MNAR)

- при $p = 5\%$ отклонение варьируется от -1.28% до $+0.29\%$;
- при $p = 10\%$ — от -1.89% до $+0.84\%$;
- при $p = 15\%$ — от -3.77% до $+1.48\%$;
- при $p = 20\%$ — от -5.69% до $+4.24\%$;
- при $p = 30\%$ — от -13.7% до $+15.04\%$.

С точки зрения изменения качества решения базовых задач, наибольшая зафиксированная потеря качества составляет -7.4% , тогда как максимальный прирост достигает $+12.44\%$. Эти эффекты существенно

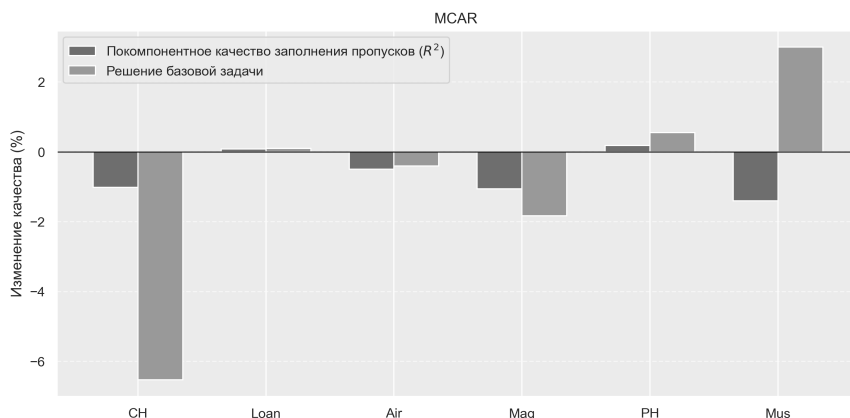


Рис. 3: Изменение качества заполнения пропущенных значений и решения соответствующей базовой задачи при переходе от KNN к ANN (MCAR)

зависят как от конкретного набора данных, так и от доли пропущенных значений. Так, для наборов данных Loan Classification, Mushrooms и Paris Housing в большинстве случаев наблюдаются положительные изменения показателей, тогда как для набора California Housing преимущественно фиксируются отрицательные изменения значения метрики F_1 . В прикладных задачах допустимость различий в качестве порядка 5–6% может существенно варьироваться в зависимости от контекста применения: в одних случаях такие отклонения являются критичными, тогда как в других приоритет отдается вычислительной эффективности. Если основным ограничением является время обработки, ANN-подход к заполнению пропущенных значений, в частности метод на основе HNSW, позволяет добиться значительного ускорения. Однако для задач, чувствительных даже к небольшим изменениям качества данных, целесообразно проводить дополнительную эмпирическую проверку перед заменой классических KNN-методов на ANN-решения.

6. Заключение

Эксперименты показали, что методы на основе HNSW позволяют существенно ускорить процесс заполнения пропущенных значений на основании алгоритма KNN IMputer, достигая прироста скорости до 28× при сохранении сопоставимого уровня точности практически во всех случаях. Вопреки ожиданиям, использование ANN-подхода не всегда сопровождается снижением качества заполнения пропущенных значений: изменения

в показателях могут быть как положительными, так и отрицательными и в значительной степени зависят от конкретного набора данных. Это указывает на то, что отдельные наборы данных более или менее подходят для применения методов на основе ANN, независимо от доли и типа пропусков. В то же время ключевые особенности данных, определяющие оптимальный подход к заполнению пропусков, пока остаются неясными и представляют собой перспективное направление для дальнейших исследований.

Кроме того, мы отметили, что с ростом доли пропущенных значений в данных увеличивается относительная вариативность разницы в качестве между ANN- и KNN-методами. Эта разница может быть как положительной, так и отрицательной. При этом тип пропусков, по-видимому, не оказывает значимого влияния на такую зависимость. Несмотря на возможные риски, предложенный ANN-подход хорошо подходит для задач, требующих быстрого заполнения пропущенных значений в больших наборах данных при ограничениях по времени. В случаях, когда критична максимальная точность модели, рекомендуется провести дополнительную валидацию или использовать классический KNN-подход.

Список литературы

- [1] Acock, Alan, “What to do about missing Values”, *APA Handbook of Research Methods in Psychology*, 2012, 27–50.
- [2] Chen, Tianqi and Guestrin, Carlos, “XGBoost: A Scalable Tree Boosting System”, 2016, 785–794. DOI: 10.1145/2939672.2939785.
- [3] Ergul Aydin, Zeliha and Kamisli Ozturk, Zehra, “Performance Analysis of XGBoost Classifier with Missing Data”, 2021.
- [4] Troyanskaya, Olga and Cantor, Mike and Hastie, Trevor and Tibshirani, Rob and Botstein, David and Altman, Russ, “Missing Value Estimation Methods for DNA Microarrays”, *Bioinformatics*, **17** (2001), 520–525. DOI: 10.1093/bioinformatics/17.6.520.
- [5] Murti, Della and Pujianto, Utomo and Wibawa, Aji and Akbar, Muhammad, “K-Nearest Neighbor (K-NN) based filling of missing values”, 2019, 83–88. DOI: 10.1109/ICSITech46713.2019.8987530.
- [6] Acuna, Edgar and Rodriguez, Caroline, “The Treatment of Missing Values and its Effect on Classifier Accuracy”, *Journal of Classification*, 2004, 639–647. DOI: 10.1007/978-3-642-17103-1_60.
- [7] Liu, Ting and Moore, Andrew and Gray, Alexander and Yang, Ke, “An Investigation of Practical Approximate Nearest Neighbor Algorithms”, 2004.
- [8] Arya, Sunil and Mount, David M. and Netanyahu, Nathan S. and Silverman, Ruth and Wu, Angela Y., “An optimal algorithm for approximate nearest neighbor searching fixed dimensions”, *J. ACM*, **45** (1998), 891–923.

-
- [9] Andoni, Alexandr and Indyk, Piotr and Razenshteyn, Ilya, “Approximate Nearest Neighbor Search in High Dimensions”, 2019, 3287–3318. DOI: 10.1142/9789813272880_0182.
 - [10] Aumüller, Martin and Bernhardsson, Erik and Faithfull, Alexander John, “ANN-Benchmarks: A Benchmarking Tool for Approximate Nearest Neighbor Algorithms”, *Inf. Syst.*, **87** (2018).
 - [11] Datta, Ritendra and Joshi, Dhiraj and Li, Jia and Wang, James Ze, “Image retrieval: Ideas, influences, and trends of the new age”, *ACM Comput. Surv.*, **40** (2008), 5:1–5:60.
 - [12] Hays, James and Efros, Alexei A., “Scene Completion Using Millions of Photographs”, *Seminal Graphics Papers: Pushing the Boundaries, Volume 2*, 2007.
 - [13] Şakar, Tolga and Emekci, Hakan, “Maximizing RAG efficiency: A comparative analysis of RAG methods”, *Natural Language Processing*, **31** (2024), 1–25. DOI: 10.1017/nlp.2024.53.
 - [14] Chen, Wei and Chen, Jincai and Zou, Fuhao and Li, Yuan-Fang and Lu, Ping and Zhao, Wei, *Proceedings of the 2019 on International Conference on Multimedia Retrieval*, 2019.
 - [15] Pedregosa, F. and Varoquaux, G. and Gramfort, A. and Michel, V. and Thirion, B. and Grisel, O. and Blondel, M. and Prettenhofer, P. and Weiss, R. and Dubourg, V. and Vanderplas, J. and Passos, A. and Cournapeau, D. and Brucher, M. and Perrot, M. and Duchesnay, E., “Scikit-learn: Machine Learning in Python”, *Journal of Machine Learning Research*, **12** (2011), 2825–2830.
 - [16] Pace, R. Kelley and Barry, Ronald P., “Sparse spatial autoregressions”, *Statistics & Probability Letters*, **33** (1997), 291–297.
 - [17] Schafer, Joseph and Graham, John, “Missing Data: Our View of the State of the Art”, *Psychological Methods*, **7** (2002), 147–177. DOI: 10.1037/1082-989X.7.2.147.
 - [18] Du, Jinghan and Hu, Minghua and Zhang, Weining, “Missing Data Problem in the Monitoring System: A Review”, *IEEE Sensors Journal*, **20**:23 (2020), 13984–13998. DOI: 10.1109/JSEN.2020.3009265.
 - [19] Malkov, Yury and Yashunin, Dmitry A., “Efficient and Robust Approximate Nearest Neighbor Search Using Hierarchical Navigable Small World Graphs”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **42** (2016), 824–836.
 - [20] Lietz, Haiko, “Watts, Duncan J./Strogatz, Steven H. (1998). Collective Dynamics of » Small-World « Networks. Nature 393, S. 440–442.”, 2019, 551–553. DOI: 10.1007/978-3-658-21742-6_130.
 - [21] Sharma, Rohit, “Loan Approval Dataset”, 2024. DOI: 10.34740/KAGGLE/DSV/8173310.
 - [22] Mateen, Mujtaba, “Air Quality and Pollution Assessment”, 2024. DOI: 10.34740/KAGGLE/DS/6197184.
 - [23] Sawhney, Prisha, “Mushroom Dataset”, 2024.
 - [24] “Paris Housing Price Prediction Dataset”, 2024.
 - [25] Bock, R., “MAGIC Gamma Telescope”, 2004. DOI: 10.24432/C52C8B.
 - [26] Douze, Matthijs and Guzhva, Alexandr and Deng, Chengqi and Johnson, Jeff and Szilvasy, Gergely and Mazaré, Pierre-Emmanuel and Lomeli, Maria and

- Hosseini, Lucas and Jégou, Hervé, “The faiss library”, *arXiv preprint*, **240** (2024).
- [27] Skovgaard, Lene, “Applied regression analysis. 3rd edn. N. R. Draper and H. Smith, Wiley, New York, 1998. No. of pages: xvii+706. Price: £45. ISBN 0-471-17082-8”, *Statistics in Medicine - STAT MED*, **19** (2000), 3136–3139.
- [28] Sasaki, Yutaka, “The truth of the F-measure”, *Teach Tutor Mater*, 2007.
- [29] Shadbahr, Tolou and Roberts, Michael and Stanczuk, Jan and Gilbey, Julian and Teare, Philip and Dittmer, Sören and Thorpe, Matthew and Viñas Torné, Ramon and Sala, Evis and Lio, Pietro and Patel, Mishal and Preller, Jacobus and Rudd, James and Mirtti, Tuomas and Rannikko, Antti and Aston, John and Tang, Jing and Schönlieb, Carola-Bibiane, “The impact of imputation quality on machine learning classifiers for datasets with missing values”, *Communications medicine*, **3** (2023), 139. DOI: 10.1038/s43856-023-00356-z.
- [30] Jadhav, Anil and Pramod, Dhanya and Ramanathan, Krishnan, “Comparison of Performance of Data Imputation Methods for Numeric Dataset”, *Applied Artificial Intelligence*, **33** (2019), 1–21. DOI: 10.1080/08839514.2019.1637138.
- [31] Thurow, Maria and Dumpert, Florian and Ramosaj, Burim and Pauly, Markus, “Imputing missings in official statistics for general tasks – our vote for distributional accuracy”, *Statistical Journal of the IAOS*, **37** (2021), 1–12. DOI: 10.3233/SJI-210798.
- [32] Rivolli, Adriano and Garcia, Luís P.F. and Soares, Carlos and Vanschoren, Joaquin and de Carvalho, André C.P.L.F., “Meta-features for meta-learning”, *Knowledge-Based Systems*, **240** (2022), 108101. DOI: 10.1016/j.knosys.2021.108101.
- [33] Akiba, Takuya and Sano, Shotaro and Yanase, Toshihiko and Ohta, Takeru and Koyama, Masanori, “Optuna: A next-generation hyperparameter optimization framework”, 2019, 2623–2631.
- [34] Altmann, Andre and Tolosi, Laura and Sander, Oliver and Lengauer, Thomas, “Permutation importance: A corrected feature importance measure”, *Bioinformatics (Oxford, England)*, **26** (2010), 1340–1347. DOI: 10.1093/bioinformatics/btq134.
- [35] Beale, E. M. L. and Little, R. J. A., “Missing Values in Multivariate Analysis”, *Journal of the Royal Statistical Society: Series B (Methodological)*, **37** (1975), 129–145. DOI: 10.1111/j.2517-6161.1975.tb01037.x.
- [36] Yoon, Jinsung and Jordon, James and Schaar, Mihaela, “Gain: Missing data imputation using generative adversarial nets”, 2018, 5689–5698.
- [37] Ma, Zhihua and Chen, Guanghui, “Bayesian methods for dealing with missing data problems”, *Journal of the Korean Statistical Society*, **47** (2018), 297–313.
- [38] Zhang, Zhongheng, “Multiple imputation with multivariate imputation by chained equation (MICE) package”, *Annals of translational medicine*, **4:2** (2016), 30.
- [39] Stekhoven, Daniel J and Bühlmann, Peter, “MissForest—non-parametric filling of missing values for mixed-type data”, *Bioinformatics*, **28:1** (2012), 112–118.
- [40] Gondara, Lovedeep and Wang, Ke, “Mida: Multiple imputation using denoising autoencoders”, 2018, 260–272.

Hierarchical Nearest Neighbor Search for Missing Value Imputation in Tabular Data

S. A. Artamonov, A. M. Mironov

Missing data represent one of the most frequent difficulties encountered in data science. A wide range of effective techniques for dealing with missing values has been developed to date. One commonly adopted approach relies on nearest neighbor methods, which perform imputation by leveraging information from the most similar observations. Such techniques are particularly prevalent in big data analysis, where datasets can comprise billions of instances. Under these conditions, improving the efficiency of k -Nearest Neighbors (KNN) algorithms becomes crucial, as their computational cost can be substantial. In this study, we investigate approaches for accelerating missing value imputation based on KNN. We introduce a novel algorithm that employs approximate nearest neighbor (ANN) techniques and examine the relationship between computational efficiency and imputation accuracy. The experimental results demonstrate that widely used ANN algorithms are capable of significantly reducing runtime while preserving a high level of imputation quality.

Keywords: missing values filling, approximate nearest neighbors.

Received on March 21, 2026