

УДК 511.75

БАЙЕСОВСКИЙ ПОДХОД В СОВРЕМЕННОМ АНАЛИЗЕ ДАННЫХ: ОТ ТЕОРЕМЫ БАЙЕСА К ПРАКТИЧЕСКИМ ПРИЛОЖЕНИЯМ

Якушев С.А., студент гр. ПИБ–242, II курс

Сахаров В.В., студент гр. ПИБ–242, II курс

Ильин П.Е., студент гр. ПИБ-242, II курс

Научный руководитель: Подкур П.Н., к.ф-м.н., доцент
Кузбасский государственный технический университет
имени Т.Ф. Горбачева, г. Кемерово

В эпоху больших данных, искусственного интеллекта и постоянной неопределённости байесовский подход стал одним из самых мощных и востребованных инструментов в анализе данных. Он позволяет не просто считать статистику, а осмысленно обновлять наши убеждения по мере поступления новой информации, естественно учитывать экспертные знания и количественно оценивать степень неопределённости результатов.

По состоянию на 2026 год байесовские методы активно используются в клинических испытаниях, маркетинговых экспериментах, машинном обучении и промышленном прогнозировании. В этой статье мы пройдем весь путь - от классической теоремы XVIII века до современных библиотек и реальных практических кейсов.

Теорема Байеса, опубликованная посмертно в 1763 году английским математиком Томасом Байесом и позже развитая Пьером-Симоном Лапласом, лежит в основе всего подхода. Её фундаментальная формула в общем виде выглядит так [1]:

$$P(H_i | A) = \frac{P(A | H_i) \cdot P(H_i)}{P(A)}$$

где:

- $P(H_i | A)$ - апостериорная вероятность (насколько мы верим в гипотезу H_i после получения данных A),
- $P(A | H_i)$ - правдоподобие,
- $P(H_i)$ - априорная вероятность,
- $P(A)$ - полная вероятность (нормализующая константа).

Классический пример хорошо иллюстрирует силу этого подхода. Представим редкую болезнь, которая встречается у 0,1 % населения. Тест на неё даёт положительный результат в 99 % случаев при наличии болезни и в 5 % случаев - при её отсутствии. Если тест оказался положительным, какова реальная вероятность того, что человек болен?

Для большей наглядности приведём полный пошаговый расчёт. Обозначим:

- H_1 - гипотеза наличия болезни,

- A - положительный результат теста.

Дано:

1. $P(H_1) = 0,001$
2. $P(H_2) = 0,999$
3. $P(A | H_1) = 0,99$
4. $P(A | H_2) = 0,05$

Сначала вычисляем полную вероятность положительного результата теста:

$$P(A) = P(A | H_1) \cdot P(H_1) + P(A | H_2) \cdot P(H_2) = 0,99 \times 0,001 + 0,05 \times 0,999 = 0,00099 + 0,04995 = 0,05094$$

Теперь находим апостериорную вероятность:

$$P(H_1 | A) = \frac{P(A | H_1) \cdot P(H_1)}{P(A)} = \frac{0,99 \times 0,001}{0,05094} \approx 0,01943$$

Таким образом, реальная вероятность болезни составляет всего около **1,94 %**. Даже при положительном результате теста вероятность болезни остаётся очень низкой. Этот пример наглядно демонстрирует, почему байесовский подход критически важен в медицинской диагностике и принятии решений в условиях неопределённости.

В байесовском анализе центральную роль играют три ключевых понятия. Априорное распределение отражает наши знания или предположения о параметрах модели до получения новых данных. Правдоподобие показывает, насколько хорошо наблюдаемые данные согласуются с той или иной моделью. А апостериорное распределение представляет собой обновлённое знание после объединения априорной информации и данных. Когда аналитически получить апостериорное распределение невозможно, используют численные методы, среди которых наиболее популярны Markov Chain Monte Carlo (MCMC) [2], вариационный вывод и другие современные алгоритмы.

Байесовский подход сильно отличается от традиционного фреквентистского [3] по своей философии. Если фреквентисты понимают вероятность как долгосрочную частоту событий при многократном повторении эксперимента, то байесианцы трактуют её как меру личной уверенности. В байесовской статистике параметры модели считаются случайными величинами, что позволяет получать не точечную оценку, а полное распределение вероятностей. Это особенно ценно при работе с небольшим количеством данных, при необходимости учитывать экспертное мнение и в задачах, где данные поступают последовательно.

Сегодня для практической работы с байесовскими моделями чаще всего используют библиотеку PyMC[4] на языке Python, высокопроизводительный Stan[5], а также Pyro[6], которые хорошо интегрируются с глубоким обучением. Для анализа и визуализации результатов байесовских вычислений удобно применять пакет ArviZ[7].

Байесовские методы нашли широкое применение в самых разных областях. В маркетинге и A/B-тестировании они позволяют в любой момент времени рассчитывать вероятность превосходства одного варианта над другим и

гибко принимать решение об остановке теста. В медицине байесовский подход особенно актуален после выхода в январе 2026 года руководства FDA[8], которое рекомендует его использование для ускорения клинических исследований, анализа подгрупп и заимствования исторических данных.

Особенно ярко байесовский подход проявляет себя в связке с искусственным интеллектом. Один из самых наглядных примеров — байесовские нейронные сети. В обычной нейронной сети веса - это фиксированные числа, которые модель находит во время обучения. В байесовской версии каждый вес становится распределением вероятностей. Когда сеть делает предсказание, она не выдаёт одно число, а сразу показывает диапазон возможных ответов вместе с мерой уверенности.

Представьте медицинскую систему распознавания опухолей на МРТ. Обычная нейросеть скажет: «Это злокачественно». Байесовская нейросеть добавит: «Вероятность злокачественности 92 % с интервалом неопределённости ± 8 %». Врач сразу понимает, когда стоит доверять модели, а когда нужно дополнительное обследование. Такой подход уже активно используется в 2026 году в системах IBM Watson Health[9] и Google DeepMind для задач медицинской визуализации.

Пример расчёта в байесовской нейронной сети (классификация тональности отзыва)

Для примера мы написали простую программу на Python с использованием библиотеки **scikit-learn**. Мы применили два модуля из этой библиотеки: **CountVectorizer** - чтобы превращать текст отзывов в набор слов, **MultinomialNB** - это и есть наивный байесовский классификатор.

Программа читает отзывы и сама определяет: хороший отзыв или плохой. А в конце показывает, сколько процентов отзывов хороших и сколько плохих.

Как программа учится в файле обучение.txt мы даём ей примеры. Например: «Долго выбирал между разными моделями и не прогадал, этот аппарат превзошел все ожидания, всем рекомендую|positive»

Программа разбивает длинный отзыв на отдельные слова и считает, сколько раз каждое слово встретилось в хороших отзывах и сколько раз - в плохих.

Сглаживание Лапласа. Если какое-то слово вообще не встретилось в плохих отзывах, программа всё равно не поставит вероятность 0 %. Она добавляет маленькую «подушку безопасности» (+1 к каждому слову). Это и называется сглаживание Лапласа. Благодаря этому программа работает даже если данных мало.

Что такое «+1 к каждому слову» - это просто добавка к количеству раз, сколько слово встретилось.

Маленький пример: Представь, что в хороших отзывах всего 100 слов. Слово «супер» встретилось 5 раз. Слово «ужасный» вообще не встретилось (0 раз).

Без +1 было бы:

- «супер» = 5 раз

- «ужасный» = 0 раз
- А с +1 становится:
- «супер» = 5 + 1 = **6 раз**
- «ужасный» = 0 + 1 = **1 раз**

Теперь программа считает вероятность как: (новое количество раз) ÷ (общее количество всех слов + количество разных слов). Это называется сглаживание Лапласа. Благодаря ему ничего не обнуляется.

Обозначения, которые использует программа: чтобы было понятно, введём простые буквы:

- $P(positive)$ - вероятность хорошего отзыва = **0,61** (61 %)
- $P(negative)$ - вероятность плохого отзыва = **0,39** (39 %)
- $P(\text{слова} | positive)$ - вероятность увидеть все слова отзыва в хороших отзывах = **0,00015**
- $P(\text{слова} | negative)$ - вероятность увидеть все слова отзыва в плохих отзывах = **0,000001**

Формула Байеса (как считает программа)

$$= \frac{P(positive | \text{слова}) \times P(\text{слова} | positive) \times P(positive)}{P(\text{слова} | positive) \times P(positive) + P(\text{слова} | negative) \times P(negative)}$$

Подставляем наши цифры и считаем по шагам:

1. Числитель (верхняя часть):

$$0,00015 \times 0,61 = \mathbf{0,0000915}$$

2. Нижняя часть для плохого класса:

$$0,000001 \times 0,39 = \mathbf{0,00000039}$$

3. Полная нижняя часть формулы:

$$0,0000915 + 0,00000039 = \mathbf{0,00009189}$$

4. Делим:

$$0,0000915 \div 0,00009189 = \mathbf{0,972 = 97,2\%}$$

Результат: программа выдаёт: **хороший** отзыв с вероятностью **97,2%**

Что видит пользователь в конце: после обработки всех отзывов программа выводит: Хороших отзывов: 68 (72,3 %) Плохих отзывов: 26 (27,7 %)

Вот и вся работа! Программа считает слова, добавляет +1 к каждому счётчику, подставляет числа в простую формулу и выдаёт понятный процент. Даже длинный отзыв разбирается слово за словом.

Такую программу можно поставить на сайт, чтобы быстро видеть, что думают покупатели.

В промышленности и финансах такие комбинации Байеса и ИИ помогают предсказывать отказы оборудования с указанием уровня уверенности и управлять инвестиционными портфелями, автоматически балансируя риски.

Главные преимущества байесовского подхода заключаются в отличной работе с малыми выборками, естественном учёте экспертных знаний, возможности последовательного обновления модели и получении полной картины неопределённости. Среди ограничений можно отметить субъективность выбора априорного распределения и относительно высокие вычислительные затраты. Впрочем, эти проблемы во многом решаются с помощью современных алгоритмов и тщательного анализа чувствительности.

Байесовский подход - это не просто ещё один статистический метод. Это целая философия мышления в условиях неопределённости. С развитием удобных библиотек и поддержкой регуляторов он окончательно перешёл из академической науки в промышленное производство, особенно тесно переплетаясь с искусственным интеллектом.

Список литературы:

1. Байесова теорема [Электронный ресурс] // GeeksforGeeks. – URL: <https://www.geeksforgeeks.org/maths/bayes-theorem/> (дата обращения: 18.03.2026).
2. Кузьмин А. Интуитивное использование методов Монте-Карло с цепями Маркова [Электронный ресурс] // Нетология. – URL: <https://netology.ru/blog/intuitivnoe-ispolzovanie-metodov-monte-karlo-s-cepami-markova> (дата обращения: 18.03.2026).
3. Оценивание параметров в практике статистического анализа. Тестирование статистических гипотез [Электронный ресурс] // WLM 2023 : лекции. – URL: <https://angelgardt.github.io/wlm2023/lectures/16.html> (дата обращения: 18.03.2026).
4. PyMC : probabilistic programming framework [Электронный ресурс] // PyMC project website. – URL: <https://www.pymc.io/welcome.html> (дата обращения: 18.03.2026).
5. Stan : Software for Bayesian Data Analysis [Электронный ресурс] // Stan Development Team. – URL: <https://mc-stan.org/> (дата обращения: 18.03.2026).
6. Pyro : Deep Universal Probabilistic Programming [Электронный ресурс] // Pyro. – URL: <https://pyro.ai/> (дата обращения: 18.03.2026).
7. ArviZ : Exploratory analysis of Bayesian models [Электронный ресурс] // ArviZ. – URL: <https://arviz-devs.github.io/arviz/> (дата обращения: 18.03.2026).
8. Руководство по использованию байесовской статистики в клинических испытаниях [Электронный ресурс] // U.S. Food and Drug Administration (FDA). – Январь 2026.
9. IBM Watson Health [Электронный ресурс] // IBM. – URL: <https://www.ibm.com/watson-health> (дата обращения: 18.03.2026).