



**3D МОДЕЛИРОВАНИЕ И
ПРОГРАММИРОВАНИЕ**



приоритет2030⁺

лидерами становятся

ТЕМА:

Классификация систем обучения

КОНСПЕКТ ЛЕКЦИИ

Преподаватель:

**Шарипов Ильдар Курбангалиевич
к.т.н., доцент, заведующий кафедрой
электроснабжения и эксплуатации
электрооборудования.**





Тема 1. Классификация систем обучения.

Вопрос 1. Обучение с учителем.

Вопрос 2. Обучение без учителя.

Вопрос 3. Обучение с подкреплением.



Вопрос 1. Обучение с учителем.

Обучающие данные включают желательные решения, называемые метками (label). Можно выделить самые важные алгоритмы обучения с учителем:

- линейная регрессия;
- логистическая регрессия;
- метод k-ближайших соседей;
- метод опорных векторов;
- деревья принятия решений и случайные леса;
- нейронные сети.

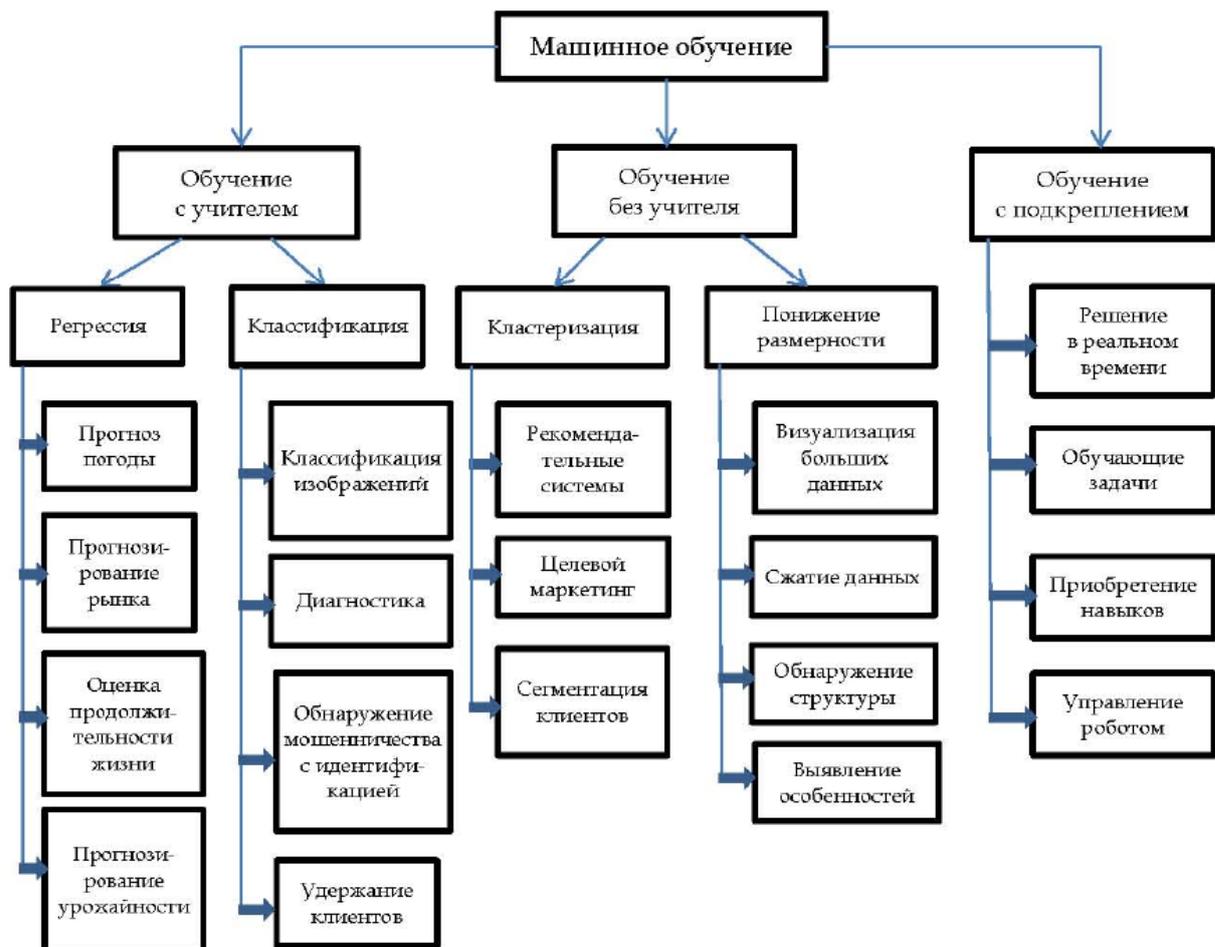


Рисунок 1 – Классификация систем машинного обучения



Обучение с учителем (или контролируемое обучение) может быть использовано для решения задач регрессии, чтобы прогнозировать целевое числовое значение переменной, располагая набором характеристик или признаков. Регрессионные проблемы, для которых входы имеют временную разницу, иногда называют анализом временных рядов. Обнаружение аномалий — это слой поверх регрессии: он относится к проблеме определения, когда наблюдаемое значение существенно отличается от прогнозируемого значения.

Некоторые алгоритмы регрессии могут применяться также для классификации. Например, логистическая регрессия, так как она способна получать вероятность принадлежности к заданному классу.

Так, метод k -ближайших соседей (*k-nearest neighbors* — KNN) — один из простых и часто используемых непараметрических методов классификации. Он состоит из следующих шагов:

- выбрать число k и метрику расстояния;
- найти k -ближайших соседей образца, который необходимо классифицировать;
- присвоить метку класса мажоритарным голосованием.

Метрики расстояния для определения того, как «ближние» точки расположены друг к другу в пространстве признаков, обычно представляют собой евклидово расстояние для непрерывных переменных и расстояние Хэмминга — для дискретных переменных. Основным недостатком алгоритма KNN заключается в создании моделей больших размеров, которые должны хранить все обучающие данные, содержащие векторы признаков и метки.

Идея ближайшего соседа расширяется и на другие задачи. Например, в задаче восстановления плотности вероятности с применением обобщенных оценок типа k -ближайших узлов.



Задача разбиения экспериментального материала на обучающую и контрольную группы в данном случае имеет тот же смысл и порождает те же проблемы, что и в задачах классификации и построения регрессионных зависимостей. Таким способом формирования функции правдоподобия приходится пользоваться при сравнительно небольших объемах выборки. Подчеркнем содержательно наиболее важный момент: то или иное разбиение наблюдений на две группы необходимо для проверки качества оценок — возможности их использования для интерполяции восстанавливаемой зависимости.

Нейронные сети. Другим способом прогнозирования данных временных рядов является использование искусственных нейронных сетей, в частности рекуррентных нейронных сетей (*recurrent neural network* — RNN), которые уникально спроектированы для изучения тенденций и моделей при вводе временных рядов в целях классификации или прогнозирования. Многослойный перцептрон (*multilayer perceptron* — MLP) относится к классу простых нейронных сетей прямого распространения, которые могут создавать нелинейные границы принятия решений. Для обучения MLP применяют алгоритм обратного распространения ошибки. Его можно описать как градиентный спуск, использующий автоматическое дифференцирование в обратном режиме. Для каждого обучающего образца алгоритм сначала вырабатывает прогноз (прямой проход), измеряет ошибку, затем проходит через каждый слой в обратном направлении, чтобы измерить вклад в ошибку каждой связи (обратный проход), и, наконец, немного подстраивает веса связей с целью уменьшения ошибки (шаг градиентного спуска). Чтобы этот алгоритм работал правильно, разработчики внесли ключевое изменение в архитектуру MLP: заменили ступенчатую функцию активации логистической функцией $f(x) = 1/(1 + \exp(-z))$.



Нейронные сети применяются для построения очень сложных моделей, особенно в случае больших наборов данных. Однако, во-первых, они чувствительны к масштабированию данных, а во-вторых, требуется много времени для обучения таких моделей. Гибкость нейронных сетей также является одним из их главных недостатков: существует много гиперпараметров для подстройки. Можно не только применять любую воображаемую топологию сети (способ связывания нейронов друг с другом), но даже в простом MLP допускается изменять число слоев, количество нейронов на слой, тип функции активации, используемой в каждом слое, логику инициализации весов и др.

Для контролируемого обучения необходимо разделить набор данных на обучающие и тестовые наборы. По сути, алгоритм машинного обучения берет в себя набор учебных материалов и выводит модель. Модель представляет собой алгоритм, который принимает новые точки данных в том же виде, что и данные обучения, и выводит предсказание.

Все алгоритмы машинного обучения определяются тремя взаимозависимыми компонентами:

- 1) семейство моделей, в котором описывается множество моделей, из которых можно выбирать;
- 2) функция издержек, которая позволяет количественно сравнивать разные модели;
- 3) процедура оптимизации, выбирающая лучшую модель из множества.



Вопрос 2. Обучение без учителя.

Если обучающие данные не помечены, система пытается обучаться без учителя. Если в контролируемом обучении система на маркированных тренировочных данных пытается извлечь модель, которая позволяет делать прогнозы о ранее не встречавшихся или будущих данных, то в неконтролируемом обучении система старается самостоятельно найти шаблоны непосредственно из набора данных. Основные алгоритмы обучения без учителя:

а) кластеризация:

- метод k -средних;
- иерархический кластерный анализ;
- максимизация ожиданий;
- нейронная сеть Кохонена;

б) визуализация и понижение размерности:

- анализ главных компонентов;
- ядерный анализ главных компонентов;
- локальное линейное вложение;
- стохастическое вложение соседей с t -распределением;

в) обучение ассоциативным правилам.

Метод k -средних применяется к вещественным векторам, когда точно известно количество кластеров. Цель алгоритма — назначить каждую точку набора данных кластеру таким образом, чтобы сумма расстояний от каждой точки до центра кластера была минимальной.

Здесь понятие «расстояние» является обычным евклидовым расстоянием в векторном пространстве:



$$d(x, y) = \sqrt{\sum (x_i - y_i)^2} = \|X - Y\|$$

Иерархический кластерный анализ используется для группировки непомеченных точек данных, имеющих сходные характеристики. Существует два основных подхода к иерархической кластеризации: дивизивный (разделяющий) и агломеративный (объединяющий). Дивизивный алгоритм начинается с единственного кластера, который охватывает все образцы и итеративно расщепляет кластер на более мелкие, пока каждый кластер не будет содержать всего один образец. В агломеративном алгоритме принят противоположный подход: каждый образец рассматривается как отдельный кластер, далее алгоритм объединяет ближайшие пары кластеров.

Агломеративная иерархическая кластеризация представлена двумя стандартными алгоритмами: методом одиночной связи (single linkage) и методом полной связи (complete linkage). Используя метод одиночной связи для каждой пары кластеров вычисляются расстояния между самыми похожими членами и объединяются два кластера, для которых расстояние между самыми похожими членами наименьшее. Подход на основе полной связи подобен методу одиночной связи, но в каждой паре кластеров сравниваются наиболее различающиеся члены для объединения.

Результаты работы иерархических алгоритмов обычно представляются в виде дендрограммы, построенной по сжатой матрице расстояний.

Анализ главных компонент (principal component analysis — PCA) — это метод линейного преобразования, который широко используется в самых разных областях, чаще всего для снижения размерности. Алгоритм PCA определяет ось, на долю которой приходится самая крупная величина дисперсии в обучающем наборе. Он также находит вторую ось, ортогональную к первой, на долю которой приходится самая крупная



величина оставшейся дисперсии, и т.д. При проекции на такие оси (снижении размерности) сохраняется наибольшее количество информации. Ортогональные оси (главные компоненты) нового подпространства можно интерпретировать как направления максимальной дисперсии при условии, что оси новых признаков ортогональны друг другу. Класс PCA из библиотеки Scikit-Learn реализует метод PCA с использованием сингулярного разложения SVD. Следующий код применяет PCA для уменьшения размерности набора данных до двух измерений (алгоритм автоматически выполняет центрирование данных):

```
from sklearn.decomposition import PCA
pca = PCA(n_components = 2)
X2D = pca.fit_transform(X)
```

После подгонки преобразователя PCA к набору данных можно обращаться к главным компонентам с помощью переменной `components_`, которая хранит их в виде горизонтальных векторов, поэтому, например, первым главным компонентом будет `pca.components_.T[:, 0]`.

Более того, ядерный трюк может быть применен к PCA, что позволяет выполнять сложные нелинейные проекции для понижения размерности. Этот алгоритм называется ядерным PCA (kPCA). Например, следующий код использует класс `KernelPCA` из библиотеки Scikit-Learn для выполнения kPCA с ядром RBF:

```
from sklearn.decomposition import KernelPCA
rbf_pca = KernelPCA(n_components = 2, kernel='rbf', gamma=0.04)
X_reduced = rbf_pca.fit_transform(X)
```



Обучение на базе многообразий (manifold learning) — это класс алгоритмов без учителя, нацеленных на описание наборов данных как низкоразмерных многообразий, вложенных в пространство большей размерности.

1. Метод *t-SNE* (*t-distributed stochastic neighbor embedding* — распределенное стохастическое соседнее вложение) используется для визуализации отображения пространства высокой размерности в пространство меньшей размерности. Алгоритм *t-SNE* начинается с преобразования многомерного евклидова расстояния между точками в условные вероятности, отражающие сходство точек.
2. Многомерное шкалирование (*MDS*) понижает размерность, одновременно пытаясь сохранить расстояния между образцами.
3. Изометрическое отображение (*Isomap*) создает граф, соединяя каждый образец с его ближайшими соседями, а затем понижает размерность, пытаясь сохранить геодезические расстояния между образцами (геодезическое расстояние между двумя узлами в графе представляет собой количество узлов на кратчайшем пути между этими узлами).

Локально-линейные методы снижения размерности базируются в основном на сохранении свойств в малых окрестностях точек.

Обучение ассоциативным правилам. Выявление закономерностей между связанными событиями может быть использовано для понимания природы анализируемых данных. Основные меры поиска ассоциативных правил: поддержка (*support*) — частота появления правила, достоверность (*confidence*) — показатель, характеризующий уверенность в том, что правило на самом деле верное (оценка условной вероятности).



Вопрос 3. Обучение с подкреплением.

Обучающая система или программный агент может наблюдать, выбирать и выполнять действия внутри среды (среда — это замкнутый физический или виртуальный объект), получая в ответ награды или штрафы в форме отрицательных наград. Затем агент должен определить наилучшую стратегию, называемую политикой, для получения максимальной награды. Например, политикой может быть нейронная сеть, которая на входе принимает наблюдения и выдает действие, подлежащее выполнению, по оценочным вероятностям для каждого действия. Другие способы исследования пространства политик связаны с использованием генетических алгоритмов и градиентных методов для определения оптимальной политики.

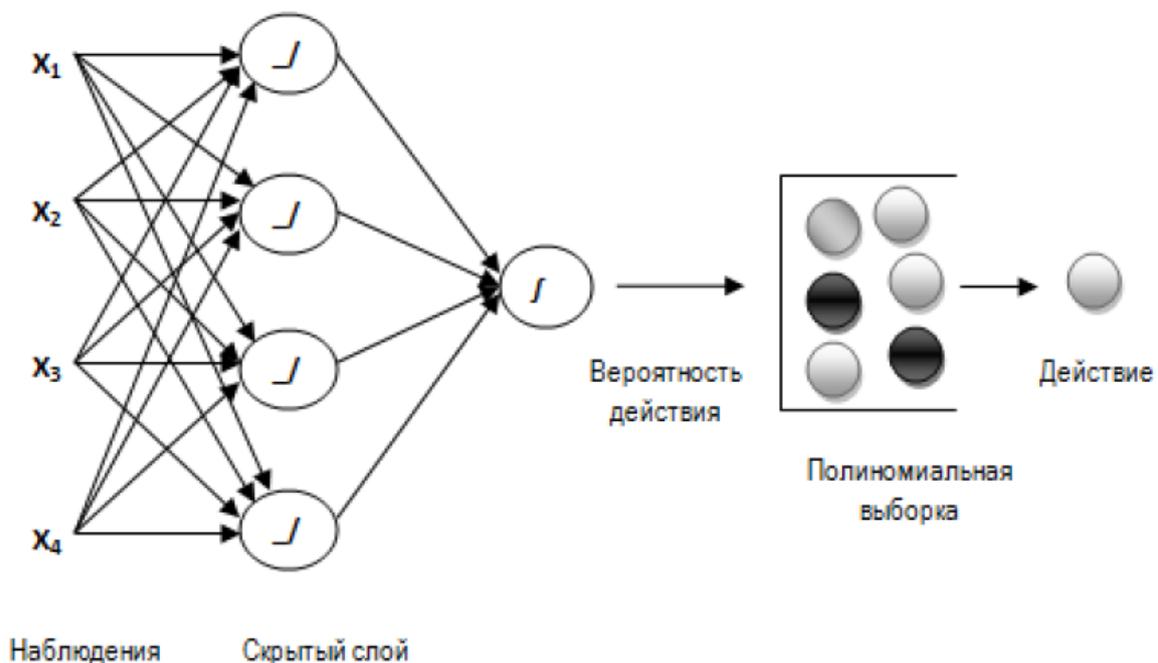


Рисунок 2 – Политика в форме нейронной сети

Марковские системы принятия решений. Марковский процесс принятия решений состоит из множества состояний S , множества действий



A , функции перехода $T(s,a,s')$ из состояния s в состояние s' при условии, что агент выбрал действие $a \in A$ и функции вознаграждения $R(s,a,s')$ для этого перехода. Для марковского процесса справедливо утверждение: вероятности переходов между состояниями не зависят от истории предыдущих переходов.

Данный подход был адаптирован для поиска оптимального маршрута передвижения мобильного агента в условиях дорожного движения и сохранения конфиденциальности его местоположения.

Коммуникационные возможности мобильных устройств позволяют отслеживать местоположение мобильных агентов по их уникальному идентификатору путем прослушивания сети.

Предположим, что злоумышленник имеет ограниченное число прослушивающих станций для расположения в сети с целью обнаружения мобильных агентов на перекрестках. Ограничение на количество прослушивающих станций связано с затратами на установку каждой станции.

Для обеспечения своей конфиденциальности мобильные агенты могут использовать несколько уникальных идентификаторов. Смена идентификаторов происходит в зонах смешивания, которые требуют определенных затрат — стоимости смены псевдонимов и пребывания в состоянии молчания для мобильного агента.

В моделируемой конфликтной ситуации, основанной на теоретико-игровом взаимодействии мобильных агентов и злоумышленника, при расчете ценности состояний вводятся некоторые определенные правила в качестве исходных данных для выбора оптимальной стратегии.