Деревья решений и смеси экспертов

На основе CSC2515

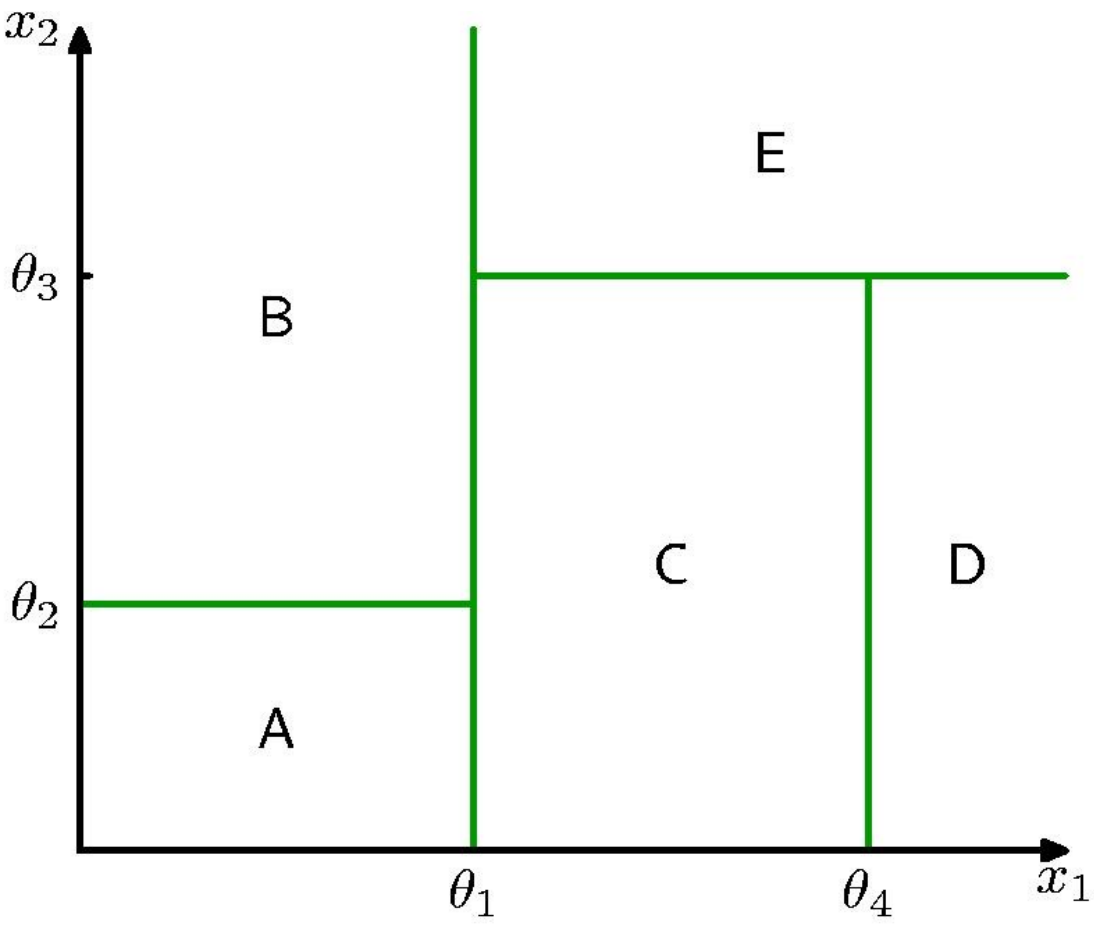
Владимир Судаков

Деревья решений: нелинейная регрессия или классификация с минимальным объемом вычислений

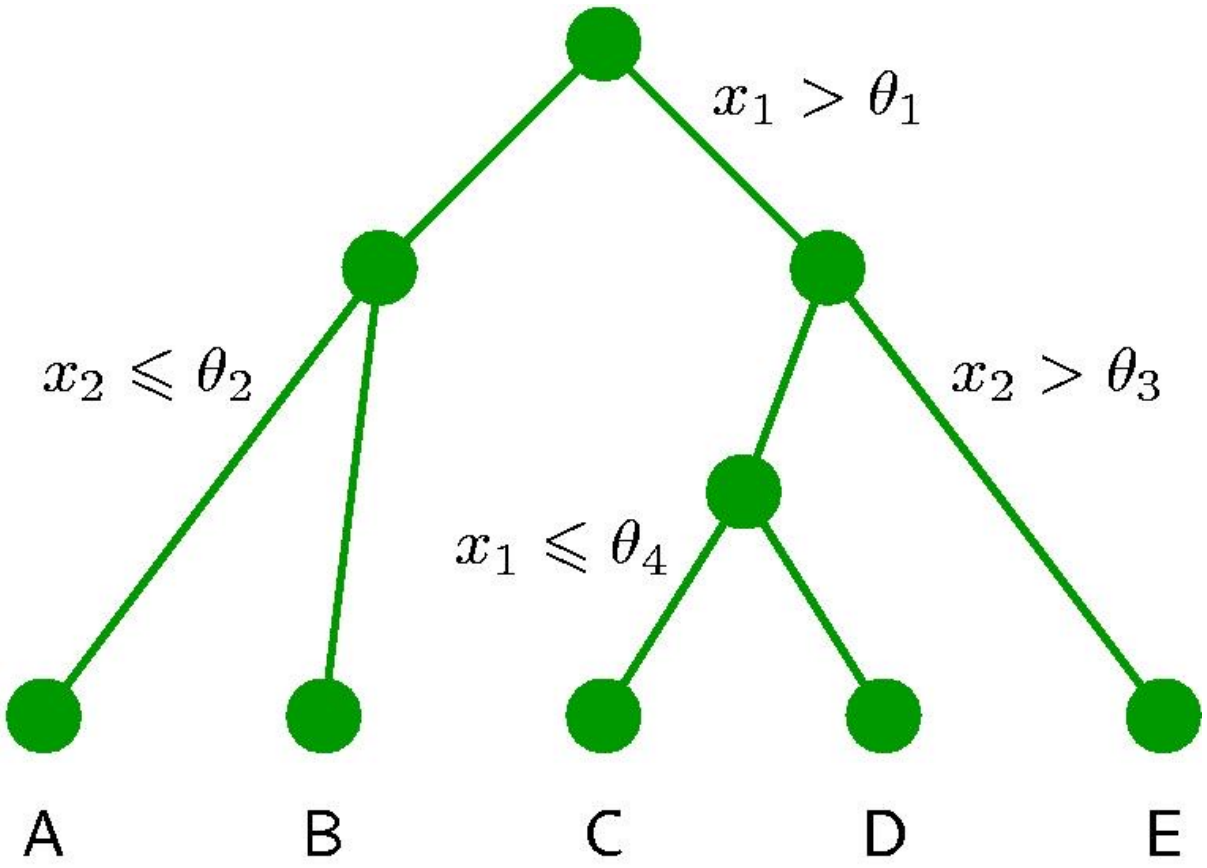
- Идея состоит в том, чтобы разделить входное пространство на непересекающийся набор областей и использовать очень простую оценку выходных данных для каждой области.
  - Для регрессии прогнозируемый результат — это просто среднее значение обучающих данных в этом регионе.
    - Но мы могли бы подогнать линейную функцию под каждую область
  - Для классификации предсказанный класс — это просто наиболее часто встречающийся класс в обучающих данных в этом регионе.
    - Мы могли бы оценить вероятности классов по частотам в обучающих данных в этом регионе.

Очень быстрый способ определить, находится ли точка данных в области

- Границы решений мы делаем ортогональными одной оси пространства и параллельными всем остальным осям.
  - Это легко проиллюстрировать в двумерном пространстве.
- Затем мы можем определить область, в которой находится точка данных, используя ряд очень простых тестов, сложность –логариф от числа областей.



Дерево решений, выровненное по осям



Как мы решаем, какие тесты использовать?

- Существует множество вариаций (CART, ID3, C4.5), но основная идея заключается в рекурсивном разбиении пространства путем жадного выбора наилучшего теста из фиксированного набора возможных тестов на каждом этапе.
  - Нам нужно определиться с набором возможных тестов.
    - Рассмотрите разбиения в каждой координате каждой точки в обучающих данных (т.е. все выровненные по осям гиперплоскости, которые касаются точек данных)
    - Мы могли бы рассмотреть все возможные гиперплоскости, но это обычно слишком затратно с точки зрения времени подгонки и сложности модели.
  - Нам нужна мера того, насколько хорош тест («чистота»).
    - Для регрессии вычислите результирующую сумму квадратов ошибок по всем разделам.
    - С классификацией все немного сложнее.

Дерево решений и алгоритм ID3

$IG(A)$— это мера разницы в энтропии до и после разделения набора S по атрибуту A. Другими словами, насколько S снижает неопределенность после разделения набора S по атрибуту A.

$$IG(S,A) = \mathrm{H}(S) - \sum_{t \in T} p\,(t)\mathrm{H}(t) = \mathrm{H}(S) - \mathrm{H}(S|A),$$

где

- $\mathrm{H}(S)$– Энтропия множества S
- T– Подмножества, созданные путем разделения набора Sпо атрибуту Aтаким образом, что $S = \mathrm{U}_{t \in T} t$
- $p(t)$– Отношение числа элементов к t числу элементов в множестве S
- $\mathrm{H}(t)$– Энтропия подмножества t

В ID3 для каждого оставшегося атрибута можно рассчитать прирост информации (вместо энтропии). Атрибут с **наибольшим** приростом информации используется для разделения набора S на этой итерации.

Когда следует прекратить добавление узлов?

- Иногда ошибка остается постоянной некоторое время, пока добавляются узлы, а затем она уменьшается.
  - Поэтому мы не можем прекратить добавление узлов, как только ошибка перестанет падать.
- Обычно лучше всего подогнать дерево под слишком большой размер, а затем отсечь наименее полезные узлы, чтобы сбалансировать сложность с ошибками.
  - Для выполнения обрезки мы могли бы использовать проверочный набор.

Преимущества и недостатки деревьев решений

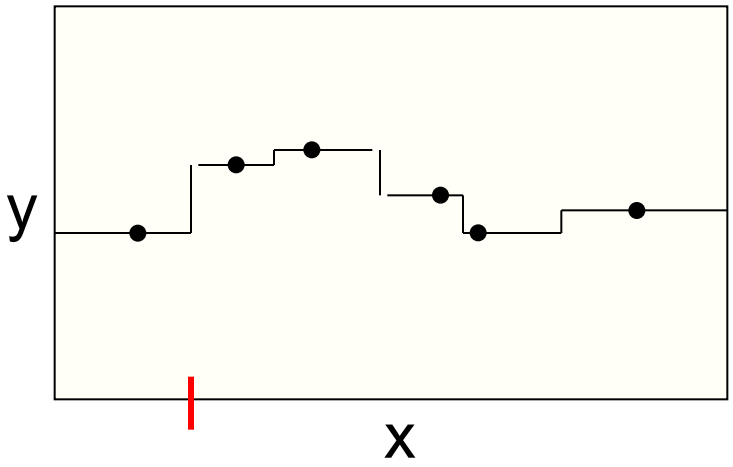
- Их легко устанавливать, использовать и интерпретировать как фиксированную последовательность простых тестов. (Врачи их любят.)
- Они нелинейны, поэтому работают гораздо лучше линейных моделей для сильно нелинейных функций.
- Они обычно обобщают хуже, чем нелинейные модели, использующие адаптивные базисные функции, но их легко улучшить, усредняя прогнозы многих деревьев.
  - Каждое дерево подгоняется под обучающий набор, созданный путем выборки набора данных с заменой («бэггинг»).
  - Вот вам и интерпретируемость!

Спектр моделей

Очень локальные модели

- например, Ближайшие соседи

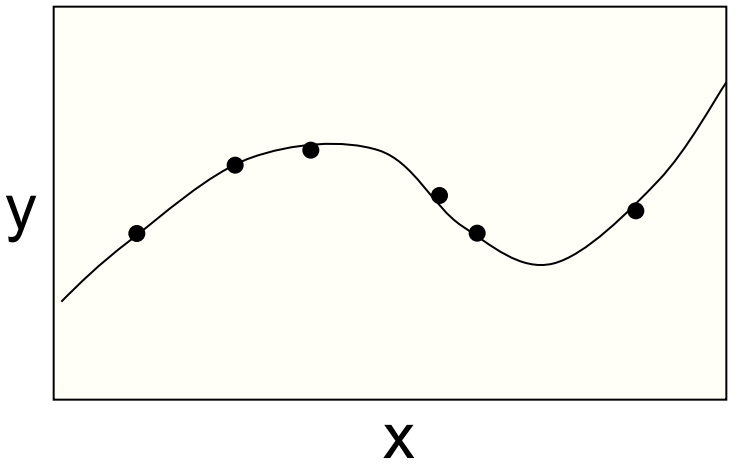
- Очень быстро устанавливается
  - Просто храните учебные кейсы
- Локальное сглаживание, очевидно, улучшает ситуацию.



Полностью глобальные модели

- например, многочлен

- Может быть медленно вписывается
  - Каждый параметр зависит от всех данных

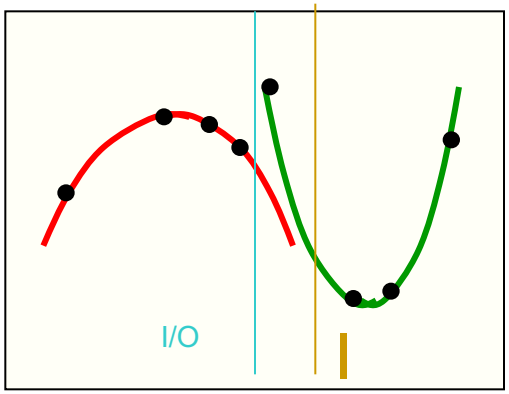


Несколько локальных моделей

- Вместо использования одной глобальной модели или множества локальных моделей используйте несколько моделей промежуточной сложности.
  - Хорошо, если набор данных содержит несколько различных режимов, которые имеют разные соотношения между входными и выходными данными.
  - Но как разделить набор данных на подмножества для каждого эксперта?

Разделение, основанное только на входе, в сравнении с разделением, основанным на отношении вход-выход

- Нам необходимо сгруппировать обучающие случаи в подмножества, по одному для каждой локальной модели.
  - Целью кластеризации НЕ является поиск кластеров схожих входных векторов.
  - Мы хотим, чтобы каждый кластер имел взаимосвязь между входом и выходом, которую можно было бы хорошо смоделировать с помощью одной локальной модели.

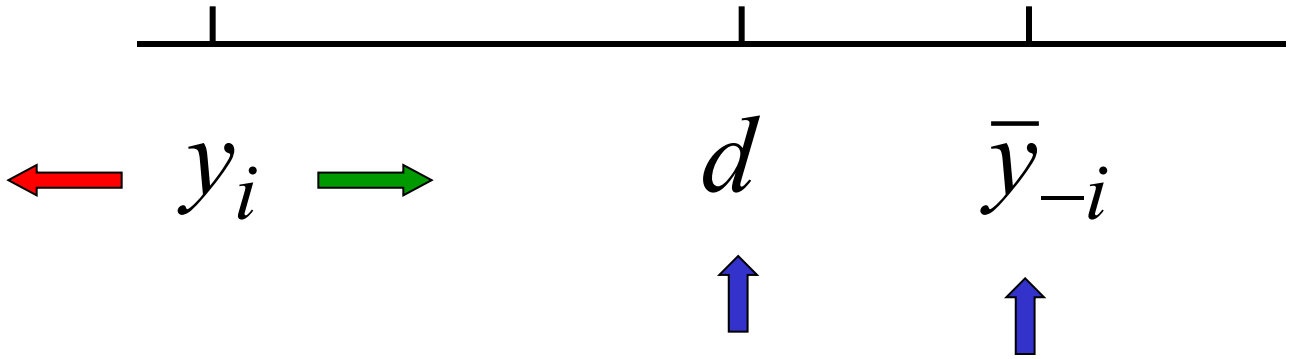


Какое разделение лучше: I = только вход или I/O = отображение входных данных →выходных данных ?

Смеси экспертов

- Можем ли мы добиться большего, чем просто усреднение предикторов, не завися от конкретного случая обучения?
  - Возможно, мы сможем рассмотреть входные данные для конкретного случая, чтобы решить, на какую модель положиться.
    - Это может позволить определенным моделям специализироваться на подмножестве обучающих случаев. Они не обучаются на случаях, для которых они не были выбраны. Поэтому они могут игнорировать то, что они плохо моделируют.
- Основная идея заключается в том, чтобы заставить каждого эксперта сосредоточиться на прогнозировании правильного ответа в тех случаях, когда он уже справляется лучше других экспертов.
  - Это приводит к специализации.
  - Если мы всегда усредняем все предикторы, каждая модель пытается компенсировать объединенную ошибку, допущенную всеми остальными моделями.

Иллюстрация того, почему усреднение — это плохо



Действительно ли мы хотим отодвинуть выход предиктора i от целевого значения?

Создание функции ошибки, которая поощряет специализацию вместо сотрудничества

- Если мы хотим поощрить сотрудничество, мы сравниваем среднее значение всех предикторов с целевым значением и обучаем, чтобы уменьшить расхождение.
  - Это может привести к серьёзному переобучению. Это делает модель гораздо более мощной, чем обучение каждого предиктора по отдельности.

Среднее значение всех предикторов

$$E = (d - < y_i >_i)^2$$

- Если мы хотим стимулировать специализацию, мы сравниваем каждый предиктор отдельно с целевым значением и обучаем, чтобы уменьшить среднее значение всех этих несоответствий.
  - Лучше всего использовать средневзвешенное значение, где веса p — это вероятности выбора данного «эксперта» для конкретного обучающего случая.

$$E = < p_i (d - y_i)^2 >_i$$

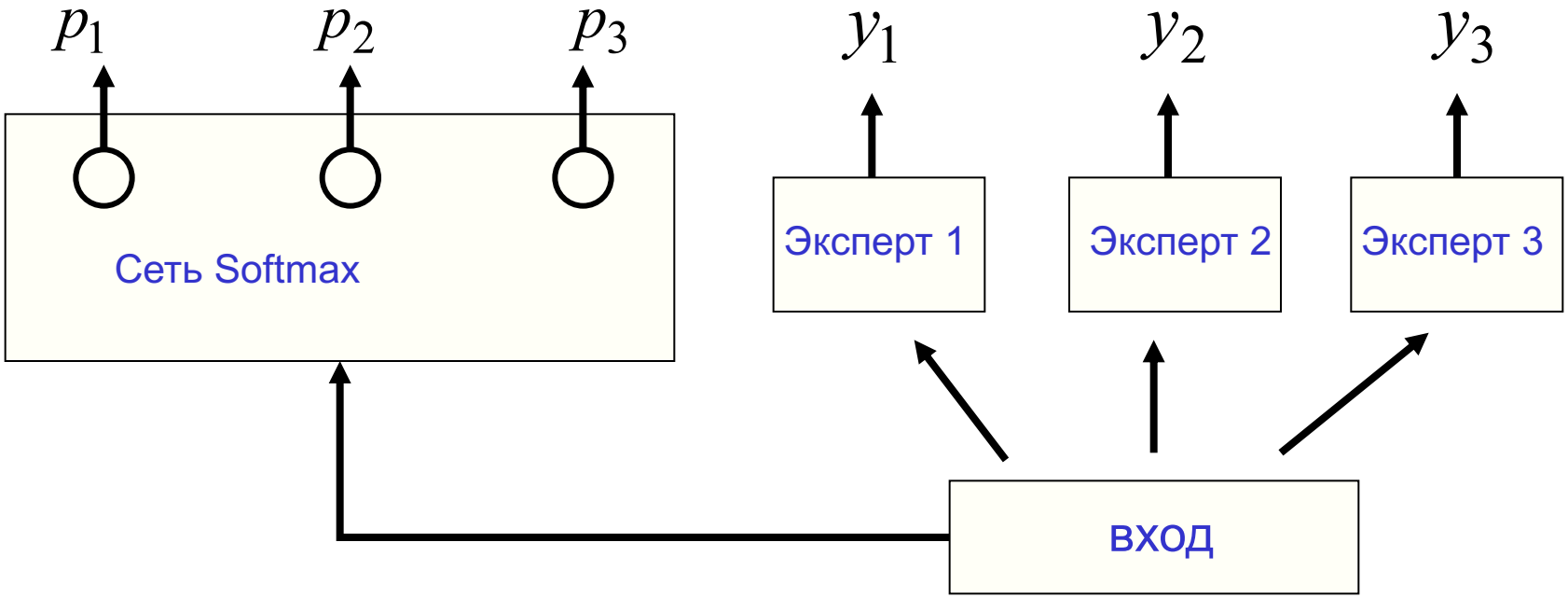
вероятность выбора эксперта i для этого случая

Смесь экспертов архитектуры

Комбинированный предиктор: $y = \sum_i p_i \, y_i$

Простая функция ошибки для обучения : $E = \sum_i p_i (d - y_i)^2$

(Эта функция ошибки лучше)



Производные простой функции стоимости

- Если мы продифференцируем по выходам экспертов, то получим сигнал для обучения каждого эксперта.

- Если мы дифференцируем по выходам сети Softmax, то получим сигнал для обучения сети.
  - Мы хотим увеличить p для всех экспертов, которые дают значение меньше, чем средняя квадратическая ошибка всех экспертов (взвешенная по p).

$$E = \sum_i p_i (d - y_i)^2$$

$$\frac{\partial E}{\partial y_i} = p_i (d - y_i)$$

$$\frac{\partial E}{\partial x_i} = p_i [(d - y_i)^2 - E]$$

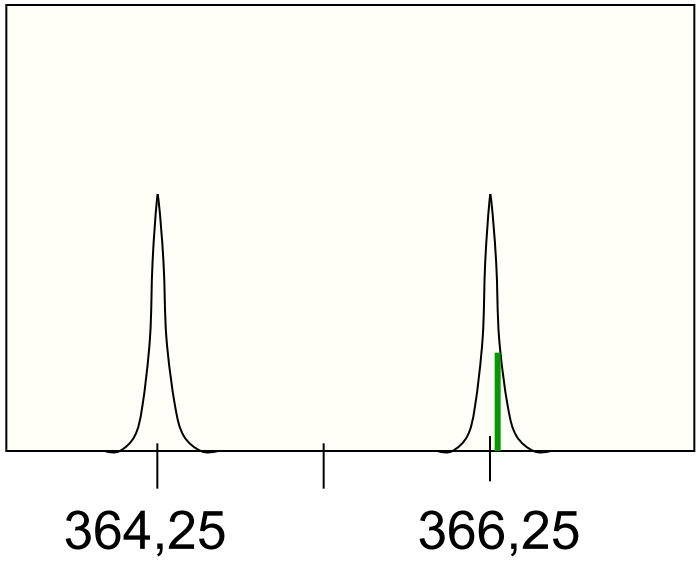
$$where \quad p_i = \frac{e^{x_i}}{\sum_j e^{x_j}}$$

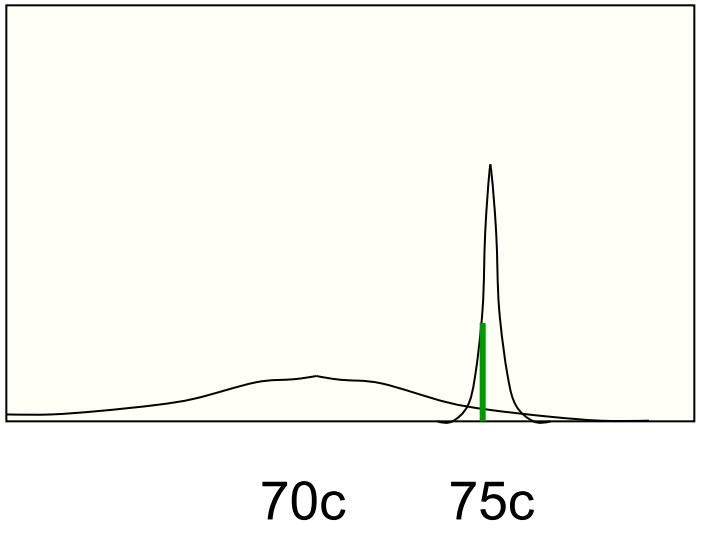
Другой взгляд на смеси экспертов

- Одним из способов объединения результатов работы экспертов является вычисление средневзвешенного значения с использованием сети Softmax для определения веса каждого эксперта.
- Но есть и другой способ объединить экспертов.
  - Сколько раз Земля вращается вокруг своей оси в год?
  - Каким будет обменный курс канадского доллара на следующий день после референдума в Квебеке?

Выдача всего распределения в качестве выходного сигнала

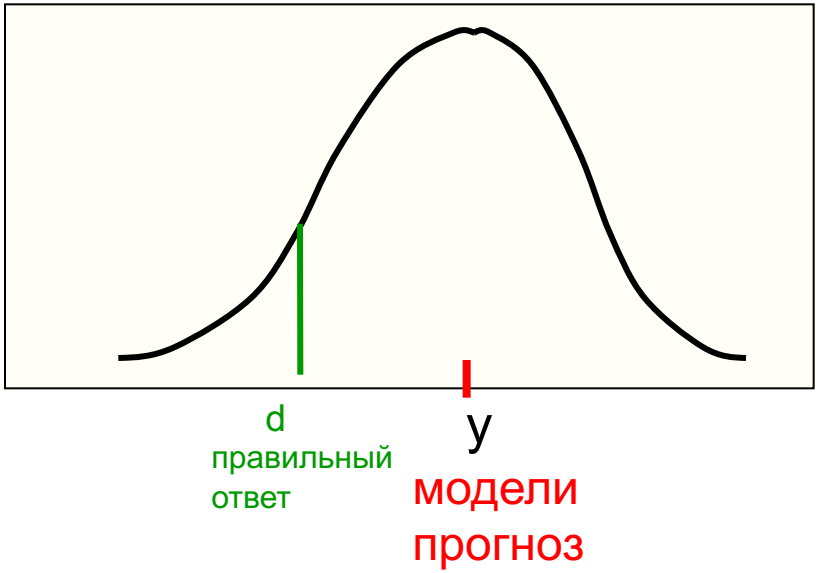
- Если существует несколько возможных режимов и мы не уверены, в каком из них находимся, лучше вывести все распределение.
  - Ошибка — это отрицательный логарифм вероятности правильного ответа.





Распределение вероятностей, которое неявно предполагается при использовании квадрата ошибки

- Минимизация квадратов остатков эквивалентна максимизации логарифмической вероятности правильных ответов при гауссовом распределении, центрированном на предположении модели.
  - Если предположить, что дисперсия гауссовой функции одинакова для всех случаев, то нам не важно ее значение.



$$p(d) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(d-y)^2}{2\sigma^2}}$$

$$-\log p(d) = k + \frac{(d-y)^2}{2\sigma^2}$$

Вероятность правильного ответа при смешении гауссианов

Пропорция смешивания, назначенная эксперту i для случая с сетью управления

$$p(d^c \mid MoG) = \sum_i p_i^c \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}\|d^c - o_i^c\|^2}$$

Вероятность желаемого результата в случае c с учетом смеси

Нормировочный член для гауссова распределения c

$$\sigma^2 = 1$$

вывод эксперта i

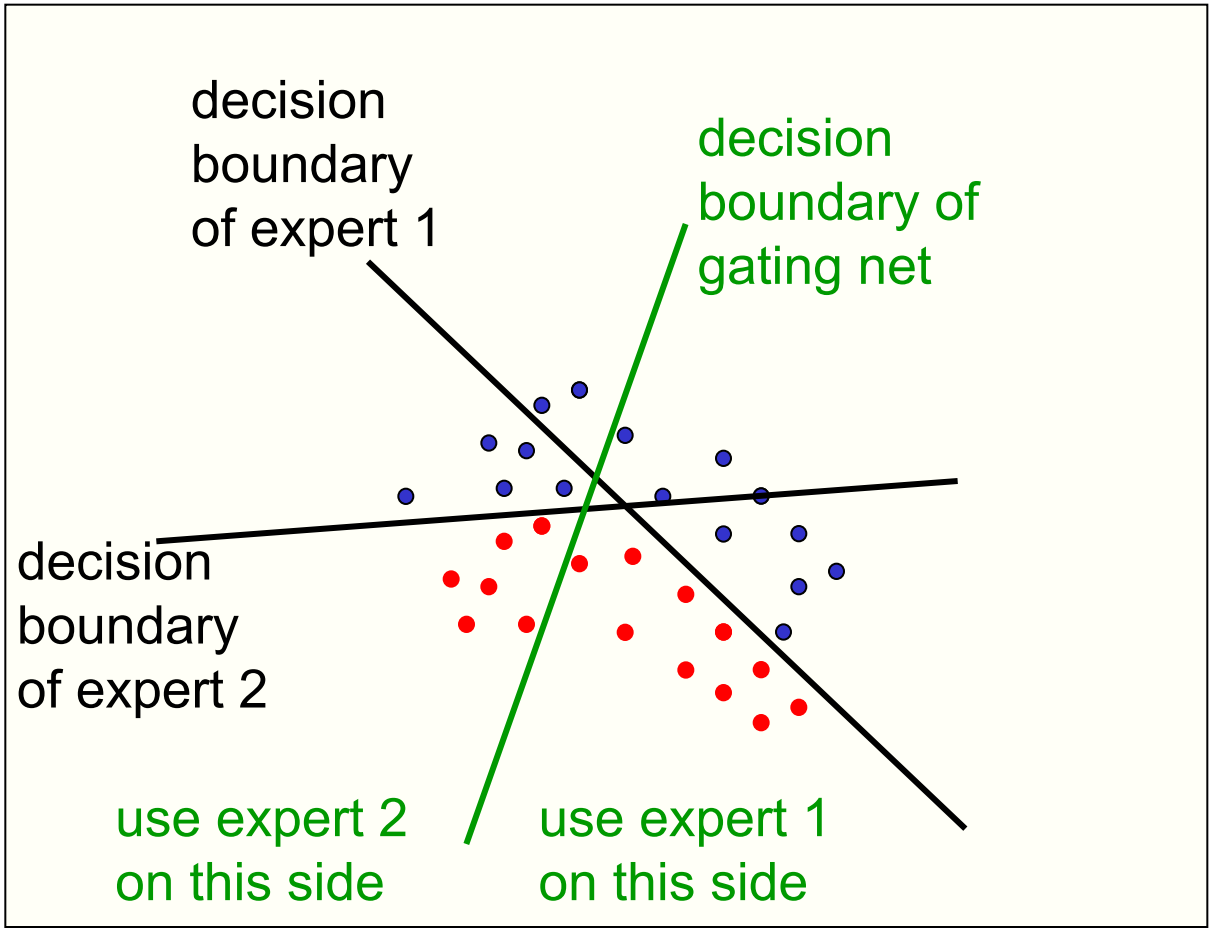
Естественная мера погрешности для смеси экспертов

$$-\log p(d^c \mid MoE) = -\log \sum_i p_i^c \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}\|d^c - o_i^c\|^2}$$

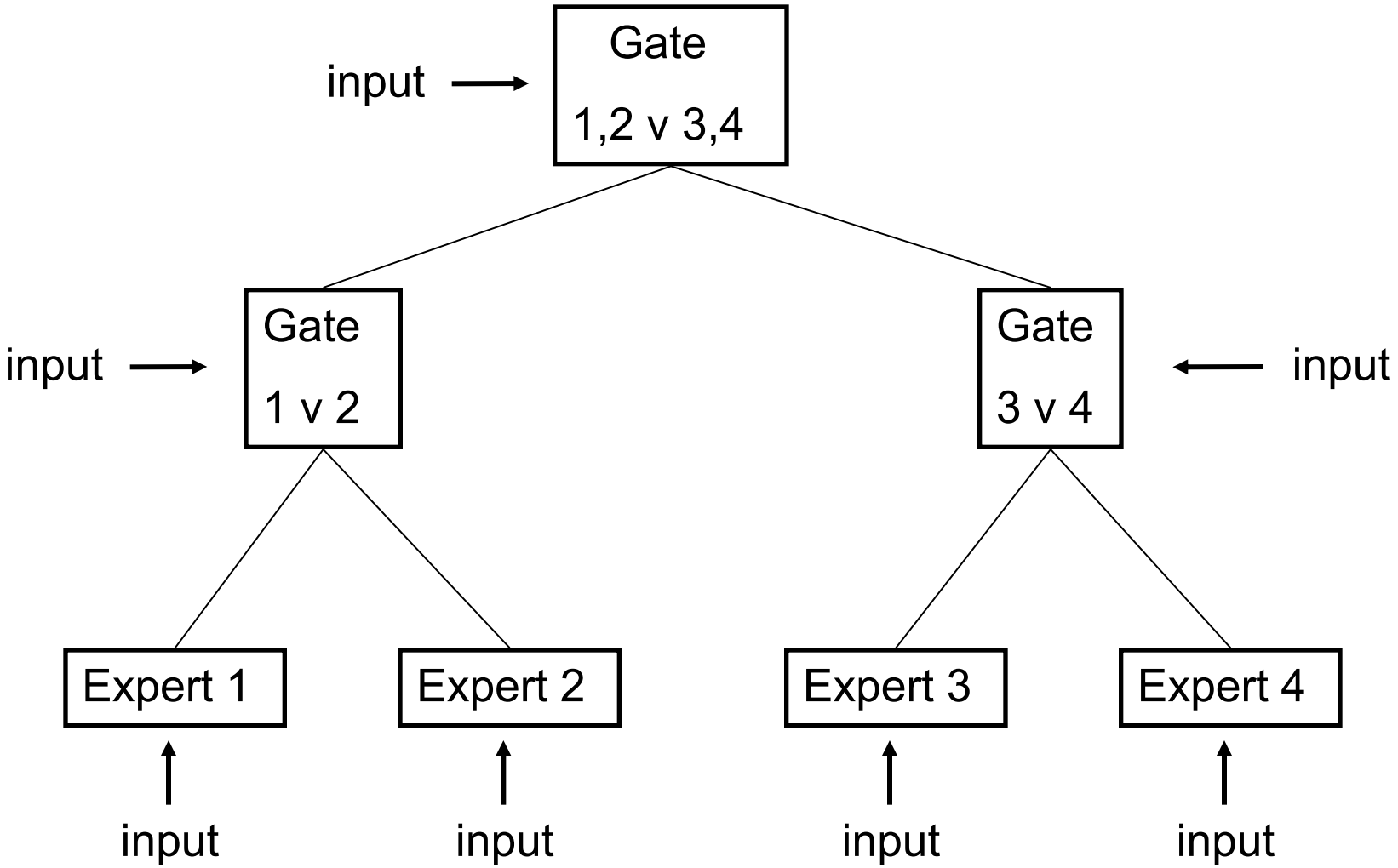
$$\frac{\partial E^c}{\partial o_i^c} = -2 \frac{p_i^c e^{-\frac{1}{2}\|d^c - o_i^c\|^2}}{\sum_j p_j^c e^{-\frac{1}{2}\|d^c - o_j^c\|^2}} (d^c - o_i^c)$$

Эта дробь представляет собой апостериорную вероятность эксперта i

Смесь двух линейных экспертов после обучения



Иерархические смеси экспертов (HMoE)



Генеративная модель для HMoE

- Сначала пусть верхний уровень шлюза выбирает одну ветвь дерева шлюзования, используя входные данные для определения относительных вероятностей
- Затем используйте следующую сеть вентилей в выбранной ветви и т. д.
- Наконец, сгенерируйте вывод эксперта в выбранном конечном узле.

Составление прогнозов после изучения дерева

- Если мы проводим регрессию и наша функция потерь представляет собой квадрат ошибки, то усредняем результаты всех экспертов, используя вероятности путей через дерево вентилей в качестве весов.
- Это позволяет избежать разрывов на границах между регионами, поскольку вероятности являются мягкими.
  - Это похоже на использование сигмоидальных единиц для обеспечения градиентного спуска в обучающих сетях прямого распространения.

Изучение упрощенного HMoE

- Существует очень эффективный способ обучения HMoE, если мы сделаем два предположения:
- Линейные эксперты: заставьте каждого эксперта выдавать выходной сигнал, являющийся линейной функцией входного сигнала, и используйте квадратичную ошибку.
  - Это позволяет подбирать эксперта без итеративного подхода, если мы знаем, какую долю ответственности он несет за каждый учебный случай.
- Обобщенные линейные сети управления: сделать каждого эксперта софтмаксом, примененным к линейному преобразованию входного вектора.
  - Это позволяет быстро настроить каждую сеть вентилей, если мы знаем, какие вероятности она должна выдавать для каждого случая. Функция стоимости выпуклая.
  - Для подгонки используется метод IRLS (итеративный рекурсивный метод наименьших квадратов).

Использование EM для установки HMoE

- Шаг E: вычислить выходные данные каждого эксперта и априорные вероятности, полученные каждой сетью фильтрации. Затем объединить априорные вероятности с вероятностью, которую каждый эксперт присваивает правильному ответу. Это даст апостериорные вероятности для каждого эксперта и каждой сети фильтрации.
- Шаг M: перенастроить каждого эксперта на данные, взвешенные по апостериорной вероятности того, что каждая точка данных получена от этого эксперта. Перенастроить каждую сеть стробирования, чтобы минимизировать перекрёстную энтропию между «априорным» распределением, которое она предоставляет, и апостериорным распределением, вычисленным на шаге E.
  - Для этого требуется IRLS, который является итеративным, но быстро сходится к глобальному оптимуму (этой подзадачи).

Лучше ли HMoE, чем фиксированный MoE?

- Если мы используем простые сети стробирования, которые можно быстро настроить с помощью IRLS, действительно ли HMoE мощнее плоского MoE?
  - Обоснованый ответ не найден.
  - HMoE, использующий бинарное дерево, имеет такое же число степеней свободы в вероятностях пути, как и один плоский softmax по всем экспертам.
  - Но делает ли зависимость от входного вектора два способа выполнения стробирования различными?

Другой (возможно лучший) тип иерархии для смешанного состава экспертов

- Вместо использования только иерархии сетей используйте также иерархию экспертов.
- Изучите всю систему методом жадного «разделяй и властвуй».
  - Начните с обучения одного эксперта.
  - Затем сделайте две немного отличающиеся копии эксперта и используйте EM для быстрой подгонки MoE с одной сетью стробирования и двумя экспертами.
  - Теперь разделите каждого из этих двух экспертов. Используйте предыдущую сеть вентилирования в качестве начальной сети верхнего уровня и добавьте две новые сети вентилирования (с нулевыми весами) на следующем уровне ниже.

Преимущество «видообразования»

- Знания, которыми обмениваются разные эксперты, не требуют отдельного изучения каждым из них, как это происходит в HMoE.
  - Подумайте, насколько неэффективно было бы людям и шимпанзе по отдельности изобретать глаза.

Работает ли видообразование лучше, чем стандартный HMoE?

- Несмотря на то, что алгоритм видообразования был изобретен (Хинтоном и Ноуланом) до HMoE, неизветсны его сравнения с HMoE.
  - Может быть, это сработает лучше.