Линейные модели классификации

Владимир Анатольевич Судаков

2025

Что такое «линейная» классификация?

- Классификация по своей сути нелинейна
  - Он помещает неидентичные вещи в один и тот же класс, разница во входном векторе иногда приводит к нулевому изменению ответа (к чему это приводит?)
- «Линейная классификация» означает, что часть, которая адаптируется, является линейной.
  - За адаптивной частью следует фиксированная нелинейность.
  - Ему также может предшествовать фиксированная нелинейность (например, нелинейные базисные функции).

$$y(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + w_0, \qquad Decision = f(y(\mathbf{x}))$$

адаптивная линейная функция

фиксированная нелинейная функция

Представление целевых значений для классификации

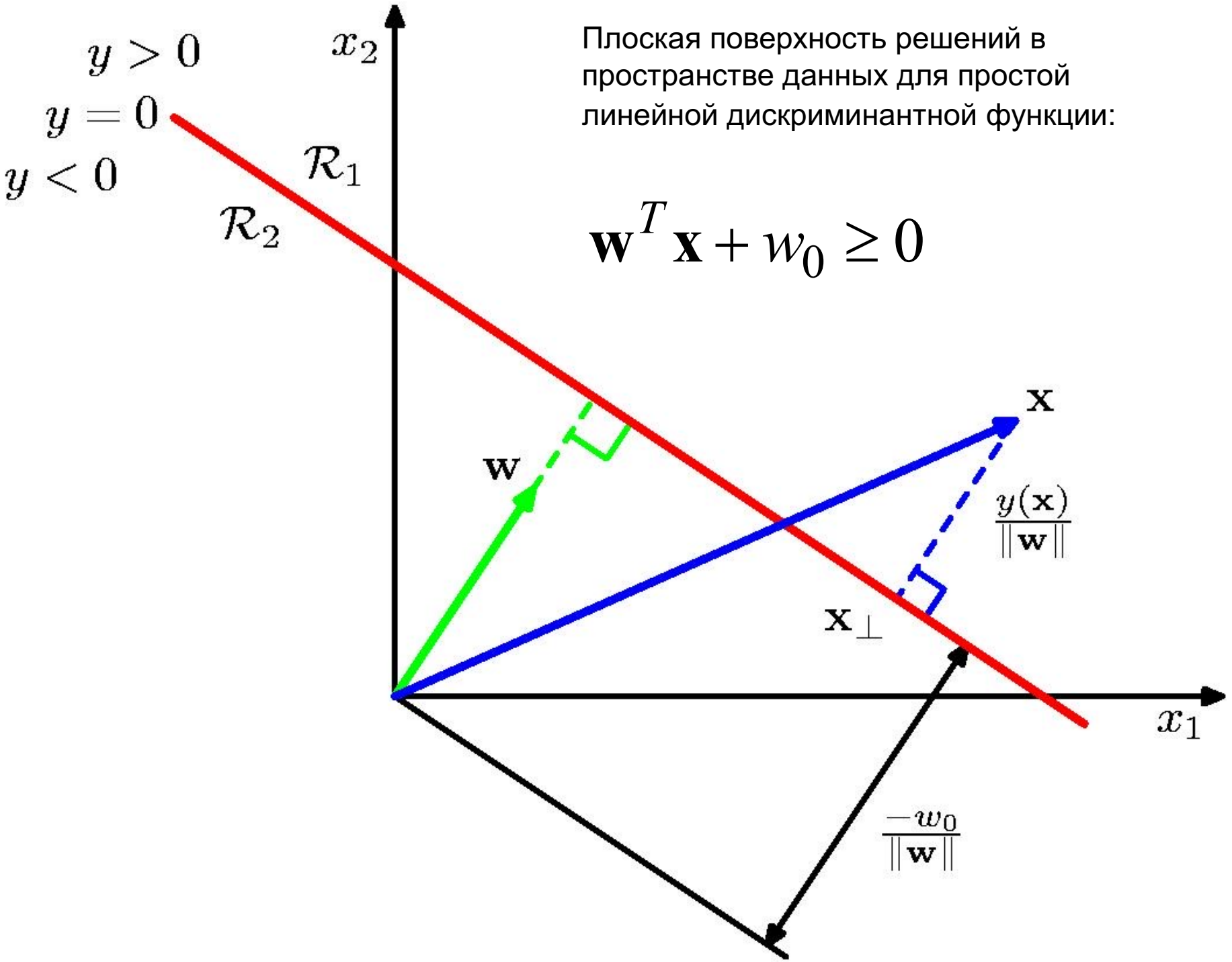
- Если есть только два класса, мы обычно используем один действительный выходной сигнал, который имеет целевые значения 1 для «положительного» класса и 0 (или иногда -1) для другого класса.
  - Для вероятностных меток классов целевым значением может быть вероятность положительного класса, а выход модели представляет собой вероятность, которую модель назначает положительному классу.
- Если имеется N классов, мы часто используем вектор из N целевых значений, содержащий одну 1 для правильного класса и нули в остальных местах.
  - Для вероятностных меток мы можем затем использовать вектор вероятностей классов в качестве целевого вектора.

Три подхода к классификации

- Используйте дискриминантные функции напрямую, без вероятностей:
  - Преобразовать входной вектор в одно или несколько действительных значений, чтобы можно было применить простую операцию (например, определение порога) для получения класса.
    - Действительные значения следует выбирать так, чтобы максимально использовать полезную информацию о метке класса, содержащуюся в действительном значении.
- Вывести условные вероятности классов: $p(class = C_k \mid \mathbf{x})$
  - Вычислите условную вероятность каждого класса.
    - Затем примите решение, которое минимизирует некоторую функцию потерь.
- Сравните вероятность ввода в отдельных, специфичных для класса, генеративных моделях.
  - Например, обучите многомерную гауссову функцию по входным векторам каждого класса и посмотрите, какая гауссова функция сделает вектор тестовых данных наиболее вероятным. (Это лучший вариант?)

Плоская поверхность решений в пространстве данных для простой линейной дискриминантной функции:

$$\mathbf{w}^T\mathbf{x} + w_0 \geq 0$$



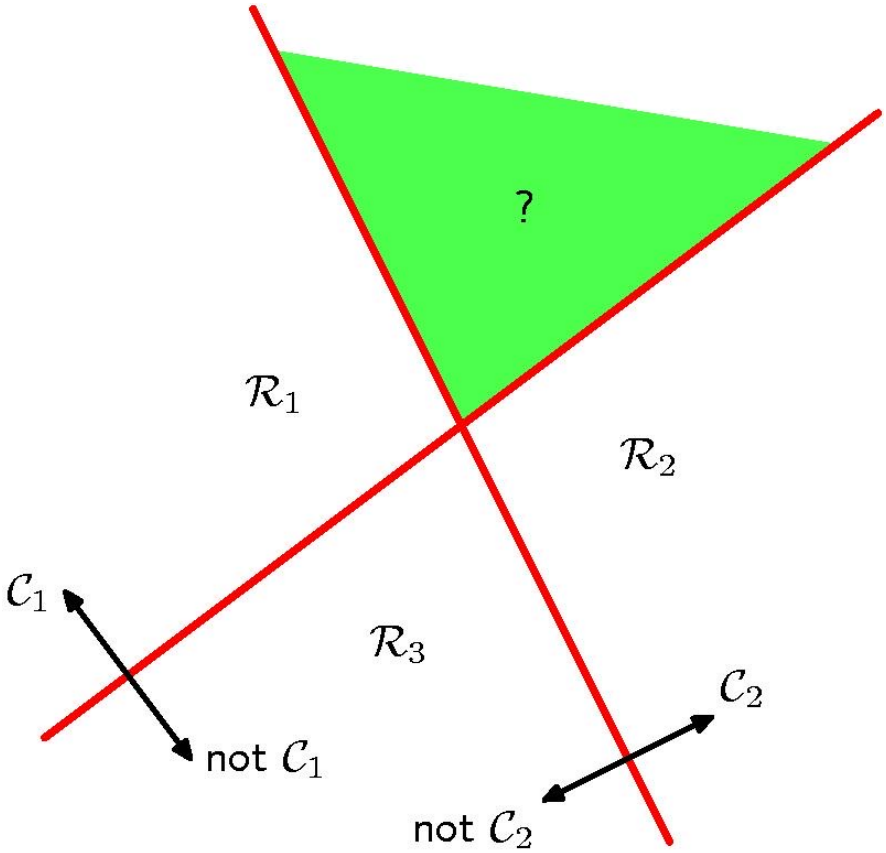
Напоминание: три разных пространства, которые легко спутать

- Пространство весов
  - Каждая ось соответствует весу
  - Точка — это вектор весов
  - Размерность = #веса +1 дополнительное измерение для потери
- Пространство данных
  - Каждая ось соответствует входному значению
  - Точка — это вектор данных. Поверхность решения — это плоскость.
  - Размерность = размерность вектора данных
- «Пространство случая»
  - Каждая ось соответствует учебному случаю
  - Точка присваивает скалярное значение каждому обучающему случаю.
    - Таким образом, он может представлять одномерные цели или может представлять значение одного входного компонента по всем обучающим данным.
  - Размерность = #кейсы обучения

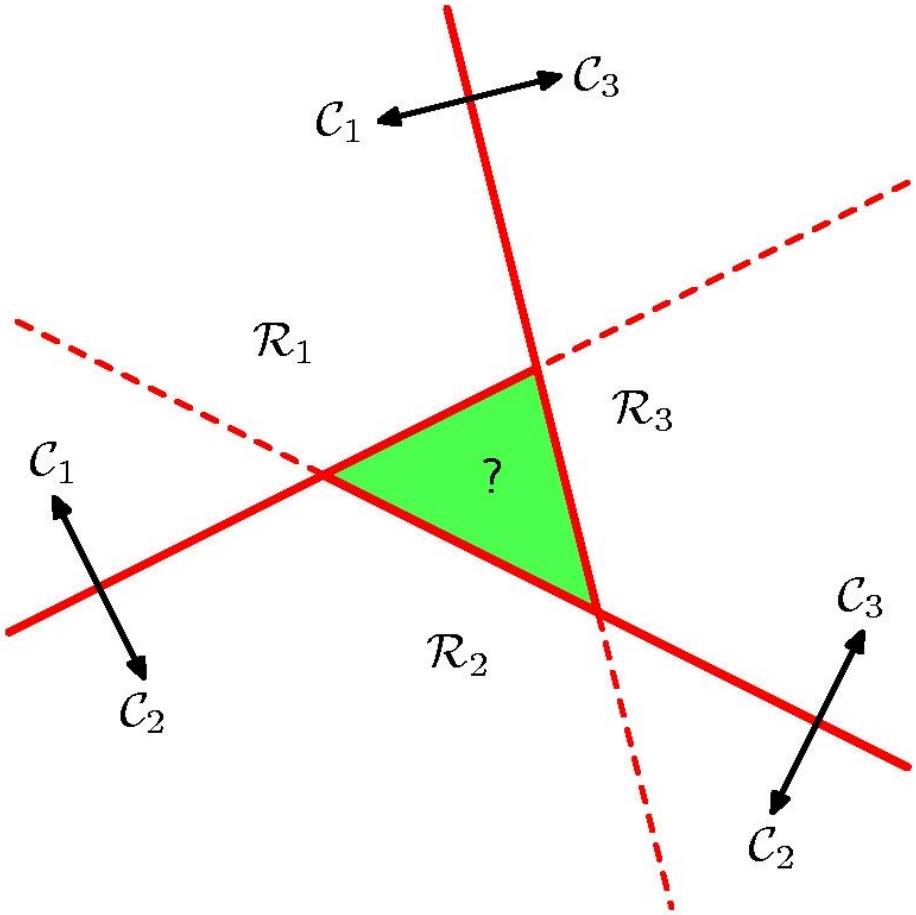
Дискриминантные функции для N>2 классов

- Одной из возможностей является использование N двусторонних дискриминантных функций.
  - Каждая функция отличает один класс от остальных.
- Другая возможность — использовать N(N-1)/2 двусторонних дискриминантных функций.
  - Каждая функция различает два конкретных класса.
- Оба эти метода имеют проблемы

Проблемы с многоклассовыми дискриминантными функциями



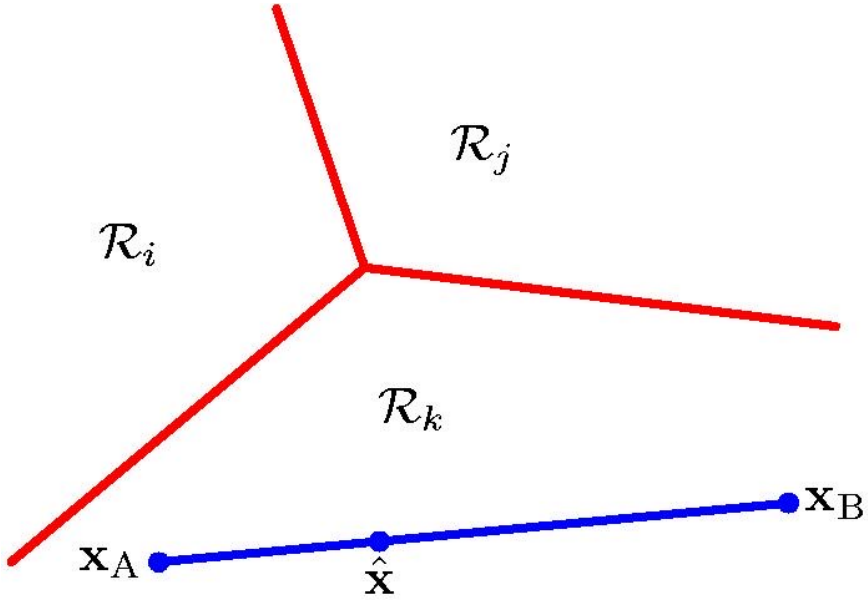
Более одного хорошего ответа



Двусторонние предпочтения не обязательно должны быть транзитивными!

Простое решение

- Используйте N $y_i, y_j, y_k \cdots$ дискриминантных функций и выберите максимальную.
  - Это гарантированно даст последовательные и выпуклые области решений, если *y* линейна.



$$y_k(\mathbf{x}_A) > y_j(\mathbf{x}_A) \;\; and \;\; y_k(\mathbf{x}_B) > y_j(\mathbf{x}_B)$$

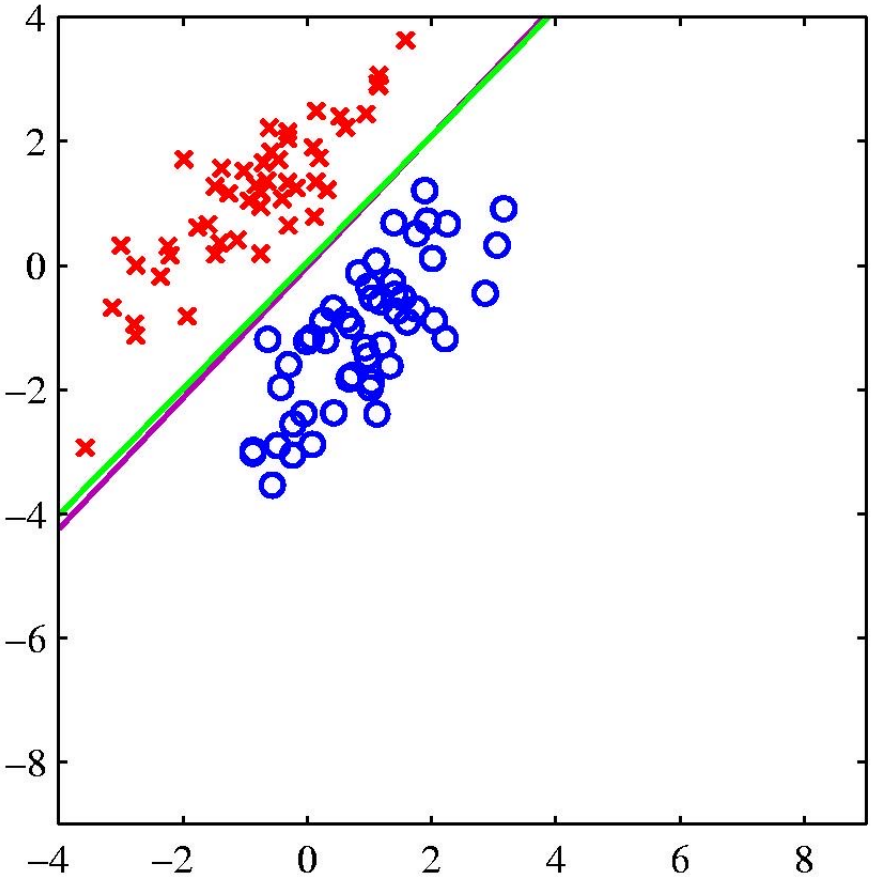
$$implies \; (for \;\; positive \; \alpha) \;\; that$$

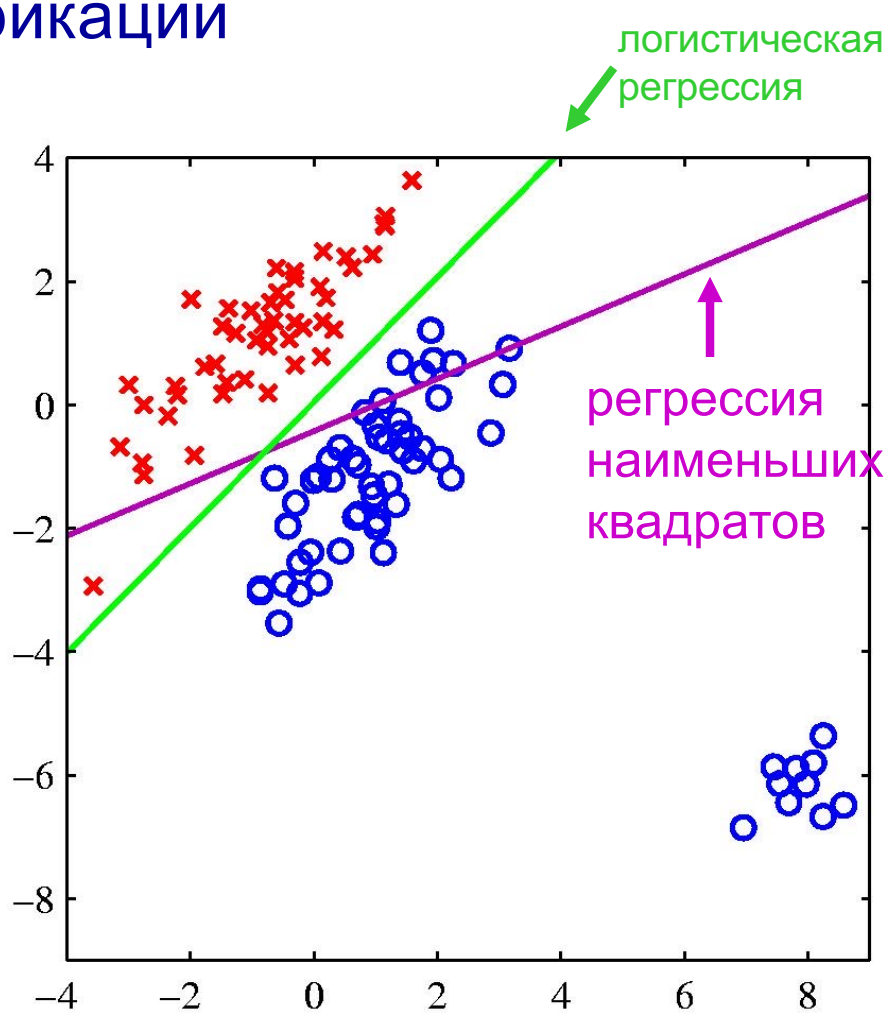
$$y_k\left(\alpha\,\mathbf{x}_A + (1-\alpha)\,\mathbf{x}_B\right) > y_j\left(\alpha\,\mathbf{x}_A + (1-\alpha)\,\mathbf{x}_B\right)$$

Использование метода наименьших квадратов для классификации

- Не работает так хорошо, как лучшие методы, но он прост:
  - Он сводит классификацию к регрессии наименьших квадратов.
  - Мы уже умеем строить регрессию. Остаётся только найти оптимальные веса, используя матричную алгебру (см. ML 2).
- Используем цели, которые равны условной вероятности класса с учетом входных данных.
  - Когда имеется более двух классов, мы рассматриваем каждый класс как отдельную задачу (мы не можем избежать этого, если используем функцию принятия решения “max”).

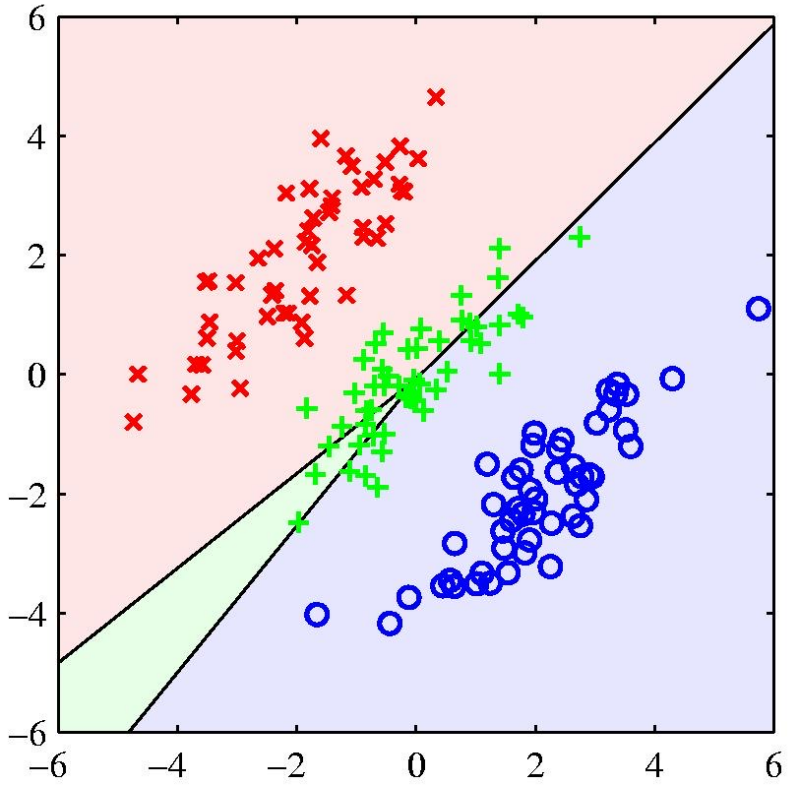
Проблемы с использованием наименьших квадратов для классификации

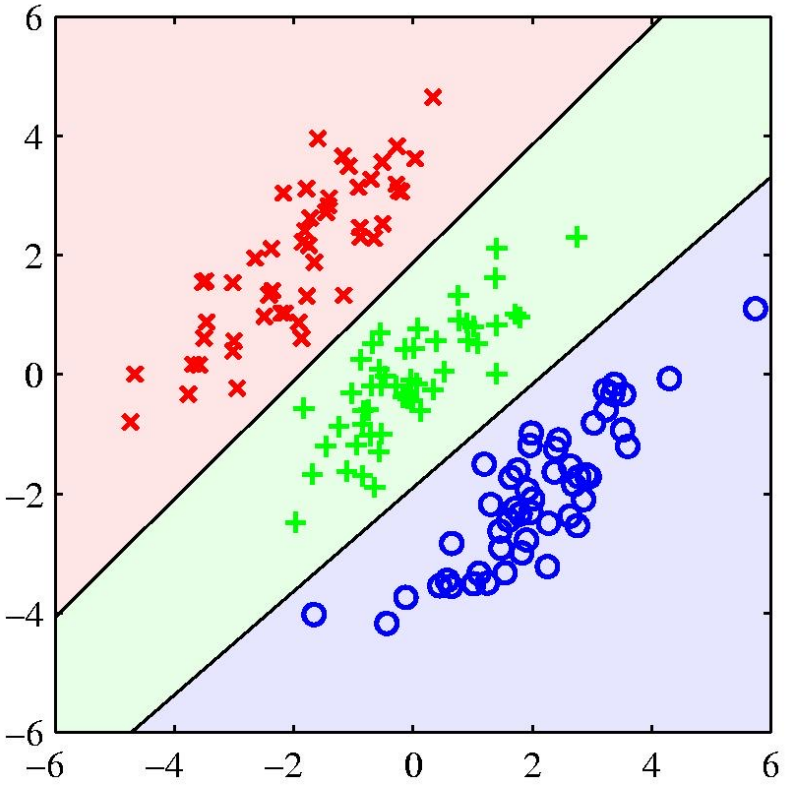




Если правильный ответ — 1, а модель говорит 1,5, она проигрывает, поэтому она меняет границу, чтобы избежать «слишком правильной» оценки.

Другой пример, когда регрессия наименьших квадратов дает плохие поверхности решений

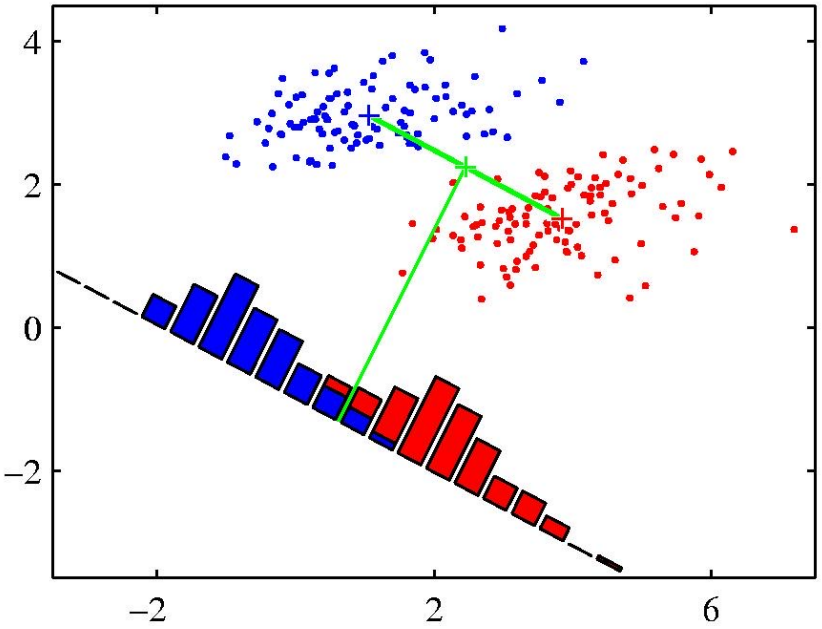




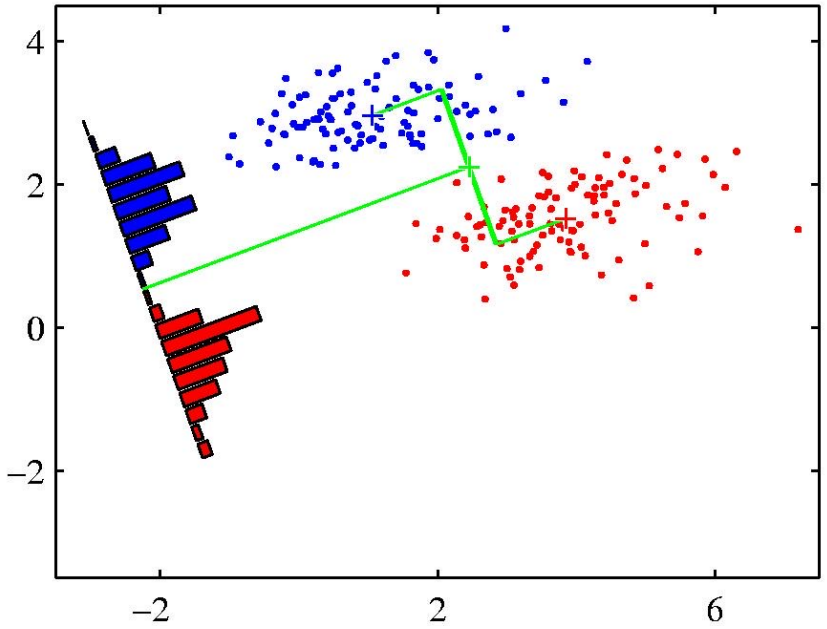
Линейный дискриминант Фишера

- Простая линейная дискриминантная функция представляет собой проекцию данных на одномерный уровень.
  - Итак, выберите проекцию, которая обеспечивает наилучшее разделение классов. Что мы подразумеваем под «наилучшим разделением»?
- Очевидным направлением выбора является направление линии, соединяющей средние значения класса.
  - Однако если основное направление дисперсии в каждом классе не ортогонально этой линии, то это не даст хорошего разделения (см. следующий рисунок).
- Метод Фишера выбирает направление, которое максимизирует отношение межклассовой дисперсии к внутриклассовой дисперсии.
  - Это направление, в котором спроецированные точки содержат наибольшую информацию о принадлежности к классу (в соответствии с гауссовыми предположениями).

Рисунок, демонстрирующий преимущество линейного дискриминанта Фишера



При проецировании на линию, соединяющую классы, классы не очень хорошо разделены.



Фишер выбирает направление, которое делает прогнозируемые классы гораздо плотнее, даже если их прогнозируемые средние значения не так далеко друг от друга.

Математика линейных дискриминантов Фишера

- Какое линейное преобразование лучше всего подходит для дискриминации?
- Проекция на вектор, разделяющий класс - кажется разумным

$$y = \mathbf{w}^T \mathbf{x}$$

$$\mathbf{w} \propto \mathbf{m}_2 - \mathbf{m}_1$$

- Нам также нужна небольшая дисперсия внутри каждого класса:

$$s_1^2 = \sum_{n \varepsilon C_1} (y_n - m_1)^2$$

$$\mathbf{m}_1 = \frac{1}{N_1} \sum_{n \in \mathcal{C}_1} \mathbf{x}_n, \qquad \mathbf{m}_2 = \frac{1}{N_2} \sum_{n \in \mathcal{C}_2} \mathbf{x}_n$$

$$s_2^2 = \sum_{n \varepsilon C_2} (y_n - m_2)^2$$

- Целевая функция Фишера:

$$J(\mathbf{w}) = \frac{(m_2 - m_1)^2}{s_1^2 + s_2^2}$$

← между

← в пределах

Использование линейных дискриминантов Фишера

$$J(\mathbf{w}) = \frac{(m_2 - m_1)^2}{s_1^2 + s_2^2} = \frac{\mathbf{w}^T \mathbf{S}_B \mathbf{w}}{\mathbf{w}^T \mathbf{S}_W \mathbf{w}}$$

$$\mathbf{S}_B = (\mathbf{m}_2 - \mathbf{m}_1)(\mathbf{m}_2 - \mathbf{m}_1)^T$$

$$\mathbf{S}_W = \sum_{n \in C_1} (\mathbf{x}_n - \mathbf{m}_1)(\mathbf{x}_n - \mathbf{m}_1)^T + \sum_{n \in C_2} (\mathbf{x}_n - \mathbf{m}_2)(\mathbf{x}_n - \mathbf{m}_2)^T$$

$$Optimal \;\; solution: \;\; \mathbf{w} \propto \mathbf{S}_W^{-1} (\mathbf{m}_2 - \mathbf{m}_1)$$

Персептрон

- «Персептроны» представляют собой целое семейство обучающихся машин, но стандартный тип состоит из слоя фиксированных нелинейных базисных функций, за которым следует простая линейная дискриминантная функция.
  - Они были введены в конце 1950-х годов и имели простую процедуру онлайн-обучения.
  - Об их способностях были сделаны громкие заявления. Это вызвало множество споров.
  - Исследователи символического ИИ подчеркивали его ограничения (в рамках идеологической кампании против действительных чисел, вероятностей и обучения)
- Машины опорных векторов (Support Vector Machines)— это просто персептроны с умным способом выбора неадаптивных, нелинейных базисных функций и лучшей процедурой обучения.
  - Они имеют те же ограничения, что и персептроны, в отношении того, какие типы функций они могут изучить.

Персептрон

представляет собой модель бинарной классификации, в которой входной вектор $\mathbf{x}$ сначала преобразуется с помощью нелинейного преобразования для получения вектора признаков $\boldsymbol{\phi}(\mathbf{x})$, а затем используется для построения обобщенной линейной модели вида

$$y(\mathbf{x}) = f(\mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}))$$

Предполагаем, что вектор $\boldsymbol{\phi}(\mathbf{x})$ включает компонент смещения ϕ_0. Нелинейная функция активации $f(\cdot)$ задаётся ступенчатой функцией вида

$$f(\alpha) = \begin{cases} +1, & \alpha \geq 0 \\ -1, & \alpha < 0 \end{cases}$$

Это связано с тем, что для персептрона удобнее использовать целевые значения

$t = +1$ для класса $\mathcal{C}_1$

$t = -1$ для класса $\mathcal{C}_2$, вместо $t \in \{0,1\}$.

Функция ошибки

рассматриваем функцию ошибки, называемую *критерием персептрона* .

Ищется весовой вектор $\mathbf{w}$, такой, что входные данные $\mathbf{x}_n$, принадлежащие классу $\mathcal{C}_1$, имеют $\mathbf{w}^T\boldsymbol{\phi}(\mathbf{x}_n) > 0$, тогда как входные данные , принадлежащие классу $\mathcal{C}_2$, имеют $\mathbf{w}^T\boldsymbol{\phi}(\mathbf{x}_n) < 0$.

Учитывая схему кодирования $t \in \{-1, +1\}$, следует, что все входные данные должны удовлетворять:

$$\mathbf{w}^T\boldsymbol{\phi}(\mathbf{x}_n)t_n > 0$$

Таким образом, критерий персептрона пытается минимизировать величину $-\mathbf{w}^T\boldsymbol{\phi}(\mathbf{x}_n)t_n$, для всех неправильно классифицированных входных данных.

$$E_P(\mathbf{w}) = -\sum_{n\in\mathcal{M}} \mathbf{w}^T\,\boldsymbol{\phi}(\mathbf{x}_n)t_n$$

где $\mathcal{M}$ обозначает набор неправильно классифицированных шаблонов.

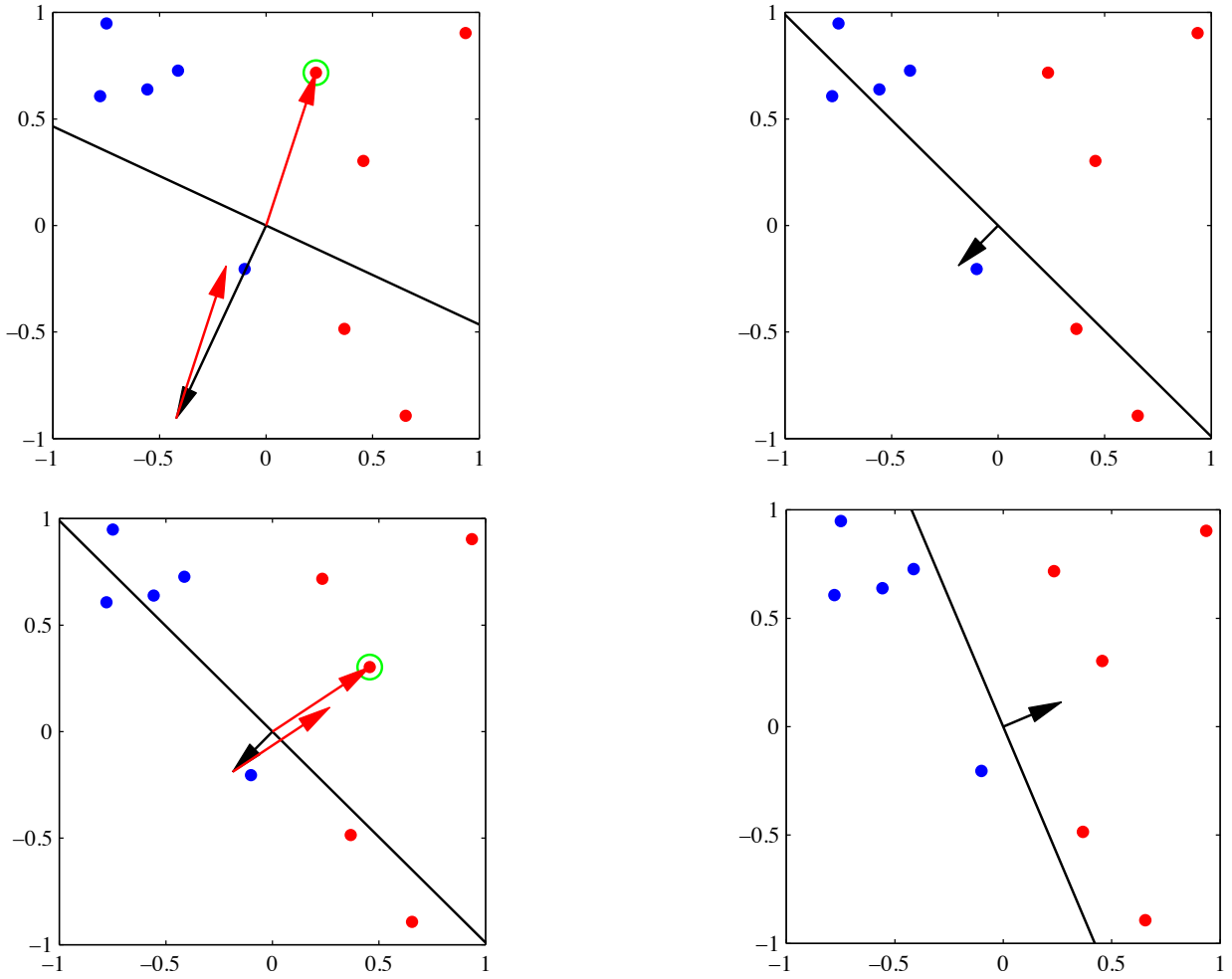
Процедура сходимости персептрона

- Добавьте к каждому вектору признаков дополнительный компонент со значением 1. Вес «смещения» этого компонента равен минус пороговому значению. Теперь о пороговом значении можно забыть.
- Выбирайте учебные случаи, используя любую политику, которая гарантирует, что каждый учебный случай будет выбран.
  - Если вывод правильный, оставьте его веса в покое.
  - Если выход равен 0, а должен быть 1, добавьте вектор признаков к вектору весов.
  - Если выход равен 1, а должен быть 0, вычтите вектор признаков из вектора веса.
- Это гарантирует нахождение набора весов, который даст правильный ответ на всем обучающем наборе, если такой набор существует.
- Нет необходимости выбирать скорость обучения.

Естественный способ попытаться доказать сходимость

- Очевидный подход — записать функцию ошибки и попытаться показать, что каждый шаг процедуры обучения уменьшает ошибку.
  - Для стохастического онлайн-обучения мы хотели бы показать, что каждый шаг уменьшает ожидаемую ошибку, где ожидание касается выбора обучающих случаев.
  - Это не может быть квадратичная ошибка, поскольку размер обновления не зависит от размера ошибки.
- В учебнике в качестве меры погрешности пытаются использовать сумму расстояний на неправильной стороне поверхности принятия решений.
  - В результате был сделан вывод о том, что процедура сходимости персептрона не гарантирует снижения общей ошибки *на каждом шаге*.
    - Это справедливо для данной функции ошибки, даже если существует набор весов, дающий правильный ответ для каждого случая обучения.

Вес и пространство данных



Сходимость алгоритма обучения персептрона, на которой показаны точки из двух классов {красного и синего) в двухмерном пространстве признаков $(\phi 1, \phi 2)$. Слева вверху показан начальный вектор параметров представленный в виде черной стрелки вместе с соответствующей границей решения (черная линия), где стрелка указывает на область принятия решения, которая классифицируется как принадлежащая к красному классу. Точка, обведенная зеленым кружком, классифицирована ошибочно, поэтому ее вектор-функция добавляется к текущему вектору весов, что дает новую границу решения, показанную справа вверху. Слева внизу показана следующая ошибочная точка, обозначенная зеленым кружком, которую следует учесть, а ее вектор-функция снова добавляется к весовому вектору, давая границу решения, показанную справа внизу, где все точки классифицированы правильно

Лучший способ доказать сходимость
(используя выпуклость решений в весовом пространстве)

- Очевидный тип функции ошибок измеряет расхождение между целевыми значениями и выходными данными модели.
- Другой тип функции стоимости использует квадрат расстояния между текущими весами и допустимым набором весов.
  - Используя эту функцию стоимости, можно показать, что каждый шаг процедуры уменьшает ошибку.
    - При условии, что существует набор допустимых весов.
- Используя этот тип функции стоимости, процедуру можно легко обобщить на более чем два класса, используя правило принятия решений MAX.

Почему процедура обучения работает

- Рассмотрим квадрат расстояния между любым удовлетворительным весовым вектором и текущим весовым вектором.
  - Каждый раз, когда персептрон совершает ошибку, алгоритм обучения уменьшает квадрат расстояния между текущим вектором веса и любым удовлетворительным вектором веса (если только он не пересекает плоскость принятия решений).
- Поэтому рассмотрим «вполне удовлетворительные» векторы веса, которые лежат в допустимой области с запасом, по крайней мере таким же большим, как и наибольшее обновление.
  - Каждый раз, когда персептрон совершает ошибку, квадрат расстояния до всех этих весовых векторов всегда уменьшается по крайней мере на квадрат длины наименьшего вектора обновления.

Чему персептроны не могут научиться

- Адаптивная часть персептрона не может даже определить, имеют ли два однобитовых признака одинаковое значение!

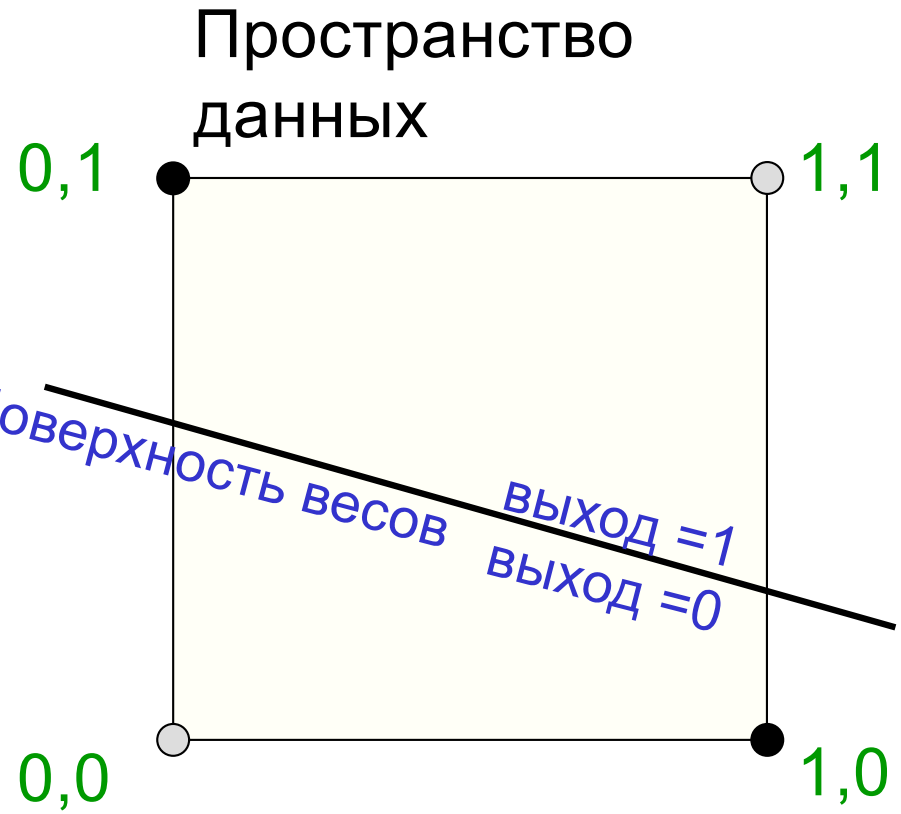
  Одинаковые : (1,1) → 1; (0,0) → 1

  Разные : (1,0) → 0; (0,1) → 0

- Четыре пары «признак-выход» дают четыре неравенства, которые невозможно удовлетворить:

$$w_1 + w_2 \geq \theta, \quad 0 \geq \theta$$

$$w_1 < \theta, \qquad w_2 < \theta$$



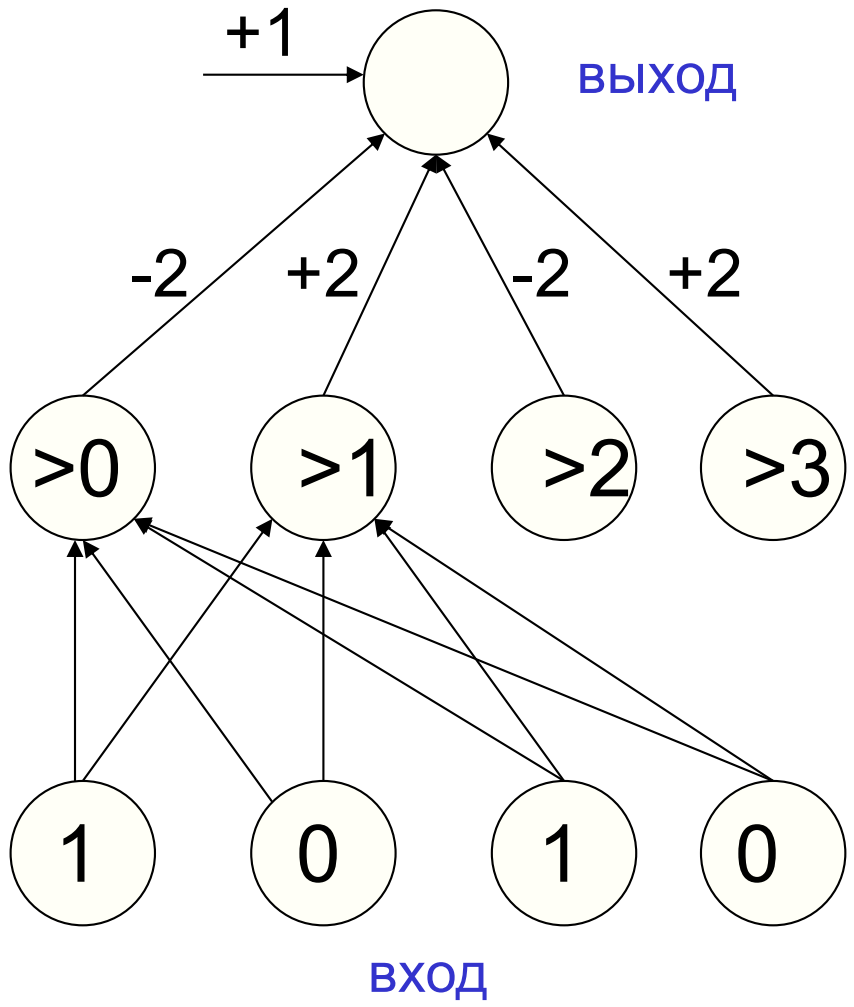
Положительные и отрицательные случаи не могут быть разделены плоскостью

Что умеют персептроны?

- Они могут решать задачи только в том случае, если закодированные вручную признаки преобразуют исходную задачу в линейно разделимую. Насколько это сложно?
- Задача проверки четности N-бит:
  - Требуется N признаков вида: Включены ли по крайней мере m бит?
  - Каждая функция должна учитывать все компоненты входных данных.
- Задача на двумерную связность
  - требуется экспоненциальное количество функций!

Задача проверки четности N-бит

- Существует простое решение, требующее N скрытых единиц.
  - Каждый скрытый блок вычисляет, включено ли более M входов.
  - Это линейно разделимая задача.
- Существует множество вариантов этого решения.
  - Этому можно научиться.
  - Он хорошо обобщает, если: $2^N >> N^2$



Вероятностные генеративные модели

Модели с линейными границами принятия решений возникают из простых предположений о распределении данных.

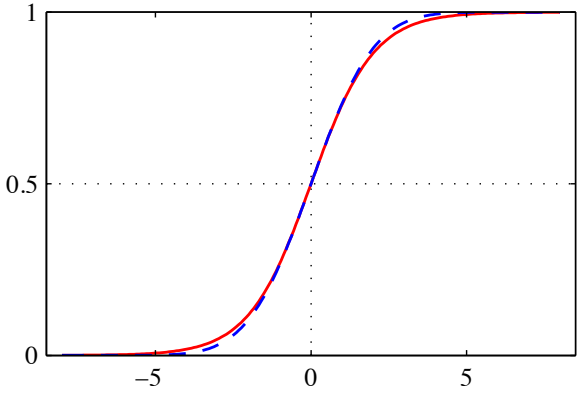
Генеративный подход моделирует условные плотности классов $p(\mathbf{x}|\mathcal{C}_k)$, а также априорные распределения классов $p(\mathcal{C}_k)$, и использует их для вычисления апостериорной вероятности $p(\mathcal{C}_k|\mathbf{x})$по *теореме Байеса* .

В связи с этим апостериорная вероятность для класса $\mathcal{C}_1$в задаче бинарной классификации определяется следующим образом:

$$p(\mathcal{C}_1|\mathbf{x}) = \frac{p(\mathbf{x}|\mathcal{C}_1)p(\mathcal{C}_1)}{p(\mathbf{x})} = \frac{p(\mathbf{x}|\mathcal{C}_1)p(\mathcal{C}_1)}{p(\mathbf{x}|\mathcal{C}_1)p(\mathcal{C}_1) + p(\mathbf{x}|\mathcal{C}_2)p(\mathcal{C}_2)} = \frac{1}{1+\exp(-\alpha)} = \sigma(\alpha),$$

где

$$\alpha = \ln \frac{p(\mathbf{x}|\mathcal{C}_1)p(\mathcal{C}_1)}{p(\mathbf{x}|\mathcal{C}_2)p(\mathcal{C}_2)}$$



Вероятностные генеративные модели

Для $K > 2$классов апостериорная вероятность для класса $\mathcal{C}_k$выглядит следующим образом:

$$p(\mathcal{C}_k|\mathbf{x}) = \frac{p(\mathbf{x}|\mathcal{C}_k)p(\mathcal{C}_k)}{p(\mathbf{x})} = \frac{p(\mathbf{x}|\mathcal{C}_k)p(\mathcal{C}_k)}{\sum_i p\,(\mathbf{x}|\mathcal{C}_i)} = \frac{\exp(\alpha_k)}{\sum_i \exp\,(\alpha_i)}$$

которая известна как *нормализованная экспонента* и может рассматриваться как многоклассовое обобщение логистической сигмоидальной функции. Величины α_kопределяются следующим образом:

$$\alpha_k = \ln(p(\mathbf{x}|\mathcal{C}_k)p(\mathcal{C}_k))$$

Нормализованная экспонента также известна как *функция softmax* , поскольку она представляет собой сглаженную версию функции max, потому что если $\alpha_k \gg \alpha_i \;\; \forall i \neq k$, то $p(\mathcal{C}_k|\mathbf{x}) \approx 1$и $p(\mathcal{C}_i|\mathbf{x}) \approx 0$.

Вероятностная модель для нормального распределения

Гауссовские распределения могут быть использованы для моделирования непрерывных переменных. Предполагая, что все классы имеют одинаковую ковариационную матрицу, плотность распределения для класса $\mathcal{C}_k$ определяется как

$$p(\mathbf{x}|\mathcal{C}_k) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \Sigma)$$

$$\begin{aligned}
\alpha &= \ln \frac{p(\mathbf{x}|\mathcal{C}_1)p(\mathcal{C}_1)}{p(\mathbf{x}|\mathcal{C}_2)p(\mathcal{C}_2)} = \ln \frac{\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_1, \Sigma)p(\mathcal{C}_1)}{\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_2, \Sigma)p(\mathcal{C}_2)} \\
&= \ln \frac{\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_1, \Sigma)}{\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_2, \Sigma)} + \ln \frac{p(\mathcal{C}_1)}{p(\mathcal{C}_2)} \\
&= \ln \frac{\exp\left\{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_1)^T\Sigma^{-1}(\mathbf{x}-\boldsymbol{\mu}_1)\right\}}{\exp\left\{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_2)^T\Sigma^{-1}(\mathbf{x}-\boldsymbol{\mu}_2)\right\}} + \ln \frac{p(\mathcal{C}_1)}{p(\mathcal{C}_2)} \\
&= -\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_1)^T\Sigma^{-1}(\mathbf{x}-\boldsymbol{\mu}_1) + \frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_2)^T\Sigma^{-1}(\mathbf{x}-\boldsymbol{\mu}_2) + \ln \frac{p(\mathcal{C}_1)}{p(\mathcal{C}_2)} \\
&= -\frac{1}{2}\mathbf{x}^T\Sigma^{-1}\mathbf{x} + \boldsymbol{\mu}_1^T\Sigma^{-1}\mathbf{x} - \frac{1}{2}\boldsymbol{\mu}_1^T\Sigma^{-1}\boldsymbol{\mu}_1 + \frac{1}{2}\mathbf{x}^T\Sigma^{-1}\mathbf{x} - \boldsymbol{\mu}_2^T\Sigma^{-1}\mathbf{x} + \frac{1}{2}\boldsymbol{\mu}_2^T\Sigma^{-1}\boldsymbol{\mu}_2 + \ln \frac{p(\mathcal{C}_1)}{p(\mathcal{C}_2)} \\
&= \boldsymbol{\mu}_1^T\Sigma^{-1}\mathbf{x} - \boldsymbol{\mu}_2^T\Sigma^{-1}\mathbf{x} - \frac{1}{2}\boldsymbol{\mu}_1^T\Sigma^{-1}\boldsymbol{\mu}_1 + \frac{1}{2}\boldsymbol{\mu}_2^T\Sigma^{-1}\boldsymbol{\mu}_2 + \ln \frac{p(\mathcal{C}_1)}{p(\mathcal{C}_2)} \\
&= (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^T\Sigma^{-1}\mathbf{x} - \frac{1}{2}\boldsymbol{\mu}_1^T\Sigma^{-1}\boldsymbol{\mu}_1 + \frac{1}{2}\boldsymbol{\mu}_2^T\Sigma^{-1}\boldsymbol{\mu}_2 + \ln \frac{p(\mathcal{C}_1)}{p(\mathcal{C}_2)}
\end{aligned}$$

Вероятностная модель для нормального распределения

$$p(\mathcal{C}_1|\mathbf{x}) = \sigma(\mathbf{w}^T\mathbf{x} + w_0),$$

где

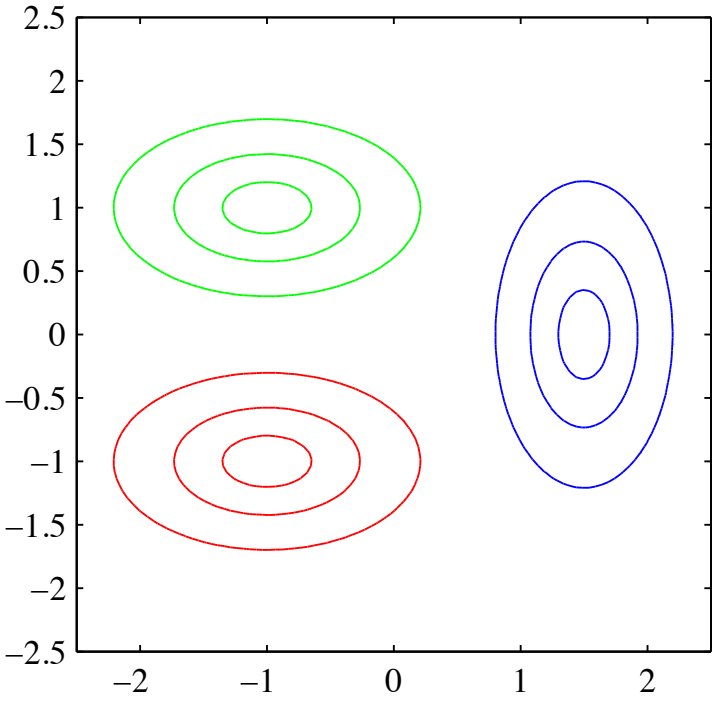
$$\mathbf{w} = \Sigma^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)$$

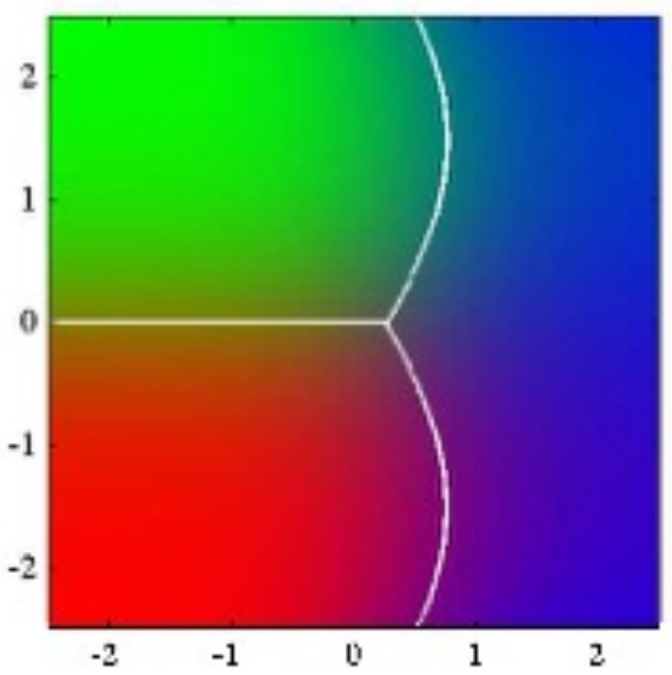
$$w_0 = -\frac{1}{2}\boldsymbol{\mu}_1^T\Sigma^{-1}\boldsymbol{\mu}_1 + \frac{1}{2}\boldsymbol{\mu}_2^T\Sigma^{-1}\boldsymbol{\mu}_2 + \ln\frac{p(\mathcal{C}_1)}{p(\mathcal{C}_2)}$$

Обратите внимание, что априорные вероятности $p(\mathcal{C}_k)$входят через параметр смещения w_0, тем самым создавая параллельные сдвиги границы решения.

Вероятностная модель для нормального распределения

Ослабляя предположение об общей ковариационной матрице для классов, позволяя каждому классу иметь собственную ковариационную матрицу Σ_k, мы получаем квадратичные функции от $\mathbf{x}$, что приводит к *квадратичному дискриминанту*.





Вероятностная модель для нормального распределения для нескольких классов

$$\begin{aligned}\alpha_k &= \ln\left(\frac{1}{(2\pi)^{D/2}}\right) + \ln\left(\frac{1}{|\Sigma|^{1/2}}\right) - \frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_k)^T\Sigma^{-1}(\mathbf{x}-\boldsymbol{\mu}_k) + \ln p(\mathcal{C}_k)\\ &= \ln\left(\frac{1}{(2\pi)^{D/2}}\right) + \ln\left(\frac{1}{|\Sigma|^{1/2}}\right) - \frac{1}{2}\mathbf{x}^T\Sigma^{-1}\mathbf{x} + \boldsymbol{\mu}_k^T\Sigma^{-1}\mathbf{x} - \frac{1}{2}\boldsymbol{\mu}_k^T\Sigma^{-1}\boldsymbol{\mu}_k + \ln p(\mathcal{C}_k)\\ &= \ln A + \ln B + Q + \mathbf{w}_k^T\mathbf{x} + w_{k0},\end{aligned}$$

где $A = \ln\left(\frac{1}{(2\pi)^{D/2}}\right), B = \ln\left(\frac{1}{|\Sigma|^{1/2}}\right), Q = -\frac{1}{2}\mathbf{x}^T\Sigma^{-1}\mathbf{x}, \mathbf{w}_k = \Sigma^{-1}\mu_k, \mathbf{w}_{k0} = -\frac{1}{2}\boldsymbol{\mu}_k^T\Sigma^{-1}\boldsymbol{\mu}_k + \ln p(\mathcal{C}_k)$

$$p(\mathcal{C}_k|\mathbf{x}) = \frac{\exp(\alpha_k)}{\sum_j \exp(\alpha_j)} = \frac{\exp(A+B+Q)\exp(\mathbf{w}_k^T\mathbf{x} + w_{k0})}{\exp(A+B+Q)\sum_j \exp(\mathbf{w}_j^T\mathbf{x} + w_{j0})}$$

α_k определяется следующим образом:

$$a_k(\mathbf{x}) = \mathbf{w}_k^T\mathbf{x} + w_{k0}$$

Следовательно, мы видим, что для $K > 2$ классов , α_k являются линейными функциями от $\mathbf{x}$ поскольку квадратичные члены взаимно компенсируют друг друга из-за общих ковариаций.

Решение методом максимального правдоподобия

Имея набор данных, включающий наблюдения $\mathbf{x}$ и соответствующие метки классов, мы можем определить параметры условных плотностей классов и априорных вероятностей классов, используя метод максимального правдоподобия. Предположим, что нам дан набор данных $\{\mathbf{x}, t_n\}$, где $t_n = 1$обозначает класс $\mathcal{C}_1$, а $t_n = 0$ обозначает $\mathcal{C}_2$. Тогда для точки данных, $\mathbf{x}_n$принадлежащей классу $\mathcal{C}_1(\, t_n = 1)$, имеем:

$$p(\mathbf{x}_n, \mathcal{C}_1) = p(\mathcal{C}_1)p(\mathbf{x}_n|\mathcal{C}_1) = \pi\mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_1, \Sigma)$$

Аналогично, для класса $\mathcal{C}_2(\, t_n = 0)$,

$$p(\mathbf{x}_n, \mathcal{C}_2) = p(\mathcal{C}_2)p(\mathbf{x}_n|\mathcal{C}_2) = (1-\pi)\mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_2, \Sigma)$$

где $p(\mathcal{C}_1) = \pi$ и $p(\mathcal{C}_2) = 1 - \pi$.

Таким образом, функция правдоподобия определяется как:

$$p(\mathbf{t}, \mathbf{X}|\pi, \boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \Sigma) = \prod_{n=1}^{N} [\,\pi\mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_1, \Sigma)]^{t_n}[(1-\pi)\mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_2, \Sigma)]^{1-t_n}$$

Решение методом максимального правдоподобия

Логарифм правдоподобия выглядит следующим образом:

$$\ln p(\mathbf{t}, \mathbf{X}|\pi, \boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \Sigma) = \sum_{n=1}^{N} t_n \left(\ln\pi + \ln\mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_1, \Sigma)\right) + (1 - t_n)(\ln(1-\pi) + \ln\mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_2, \Sigma))$$

Приравняв производную π к нулю, получим,

$$\frac{d}{d\pi}\ln p(\mathbf{t}, \mathbf{X}|\pi, \boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \Sigma) = 0 \Leftrightarrow \frac{d}{d\pi}\sum_{n=1}^{N}\{ t_n \ln\pi + (1 - t_n)\ln(1-\pi)\} = 0 \Leftrightarrow$$

$$\frac{1}{\pi}\sum_{n=1}^{N} t_n - \frac{1}{1-\pi}\sum_{n=1}^{N}(1 - t_n) = 0 \Leftrightarrow \frac{1}{\pi}\sum_{n=1}^{N} t_n - \frac{1}{1-\pi}\left(N - \sum_{n=1}^{N} t_n\right) = 0 \Leftrightarrow$$

$$\frac{1}{\pi}\sum_{n=1}^{N} t_n = \frac{1}{1-\pi}\left(N - \sum_{n=1}^{N} t_n\right) \Leftrightarrow \frac{1-\pi}{\pi}\sum_{n=1}^{N} t_n = N - \sum_{n=1}^{N} t_n \Leftrightarrow$$

$$\frac{1}{\pi}\sum_{n=1}^{N} t_n - \sum_{n=1}^{N} t_n = N - \sum_{n=1}^{N} t_n \Leftrightarrow \pi = \frac{1}{N}\sum_{n=1}^{N} t_n$$

Как и ожидалось, максимальная оценка правдоподобия для π— это просто доля точек в классе $\mathcal{C}_1$.

Решение методом максимального правдоподобия

Приравняв производную $\boldsymbol{\mu}_1$ к нулю, получим,

$$\frac{d}{d\boldsymbol{\mu}_1}\ln p(\mathbf{t},\mathbf{X}|\pi,\boldsymbol{\mu}_1,\boldsymbol{\mu}_2,\Sigma)=0 \Leftrightarrow \frac{d}{d\boldsymbol{\mu}_1}\sum_{n=1}^{N} t_n \ln \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_1,\Sigma)=0 \Leftrightarrow$$

$$\frac{d}{d\boldsymbol{\mu}_1}\left[-\frac{1}{2}\sum_{n=1}^{N} t_n\,(\mathbf{x}_n-\boldsymbol{\mu}_1)^T\Sigma^{-1}(\mathbf{x}_n-\boldsymbol{\mu}_1)\right]=0 \Leftrightarrow -\frac{1}{2}\sum_{n=1}^{N}-2t_n\Sigma^{-1}(\mathbf{x}_n-\boldsymbol{\mu}_1)=0 \Leftrightarrow$$

$$\sum_{n=1}^{N} t_n\,(\mathbf{x}_n-\boldsymbol{\mu}_1)=0 \overset{\sum_{n=1}^{N} t_n=N_1}{\Leftrightarrow} \sum_{n=1}^{N} t_n\,\mathbf{x}_n=N_1\boldsymbol{\mu}_1 \Leftrightarrow$$

$$\boldsymbol{\mu}_1=\frac{1}{N_1}\sum_{n=1}^{N} t_n\,\mathbf{x}_n$$

Аналогично, соответствующий результат для $\boldsymbol{\mu}_2$:

$$\boldsymbol{\mu}_2=\frac{1}{N_2}\sum_{n=1}^{N}(t_n-1)\mathbf{x}_n$$

Наконец, решение для общей ковариационной матрицы Σ похоже на решение, полученное для многомерного гауссовского распределения, где матрица Σ определена.

Подгонка гауссовых распределений к классам не является устойчивой к выбросам, поскольку оценка максимального правдоподобия гауссовой функции сама по себе не является устойчивой.

Дискретные признаки

Рассмотрим случай дискретных бинарных значений признаков $x_i \in \{0,1\}$. При наличии D входных данных общее распределение будет соответствовать $2^D - 1$ независимым переменным. При использовании *наивного байесовского* подхода мы имеем следующие функции плотности распределения, зависящие от класса:

$$p(\mathbf{x}|\mathcal{C}_k) = \prod_{i=1}^{D} \mu_{ki}^{x_i} (1 - \mu_{ki})^{1-x_i}$$

Для K классов, подставляя в $\alpha_k = \ln(p(\mathbf{x}|\mathcal{C}_k)p(\mathcal{C}_k))$, получаем,

$$\alpha_k(\mathbf{x}) = \sum_{i=1}^{D} (x_i \ln \mu_{ki} + (1 - x_i)\ln(1 - \mu_{ki})) + \ln p(\mathcal{C}_k)$$

В более общем случае, когда дискретные переменные могут принимать $M > 2$ состояния, функции плотности распределения, зависящие от класса, определяются следующим образом:

$$p(\mathbf{x}|\mathcal{C}_k) = \prod_{i=1}^{D} \prod_{m=1}^{M} \mu_{kim}^{\phi(x_i)_m}$$

где $\phi(x_i)$ создаётся схема двоичного 1 в M кодирования, где только одно из значений равно $\phi(x_i)_1, \ldots, \phi(x_i)_M, = 1,$ а все остальные равны 0. Таким образом, подставляя выражение выше в (4.63), получаем:

$$\alpha_k(\mathbf{x}) = \sum_{i=1}^{D} \sum_{m=1}^{M} (\phi(x_i)_m \ln \mu_{kim}) + \ln p(\mathcal{C}_k)$$

Вероятностные дискриминационные модели

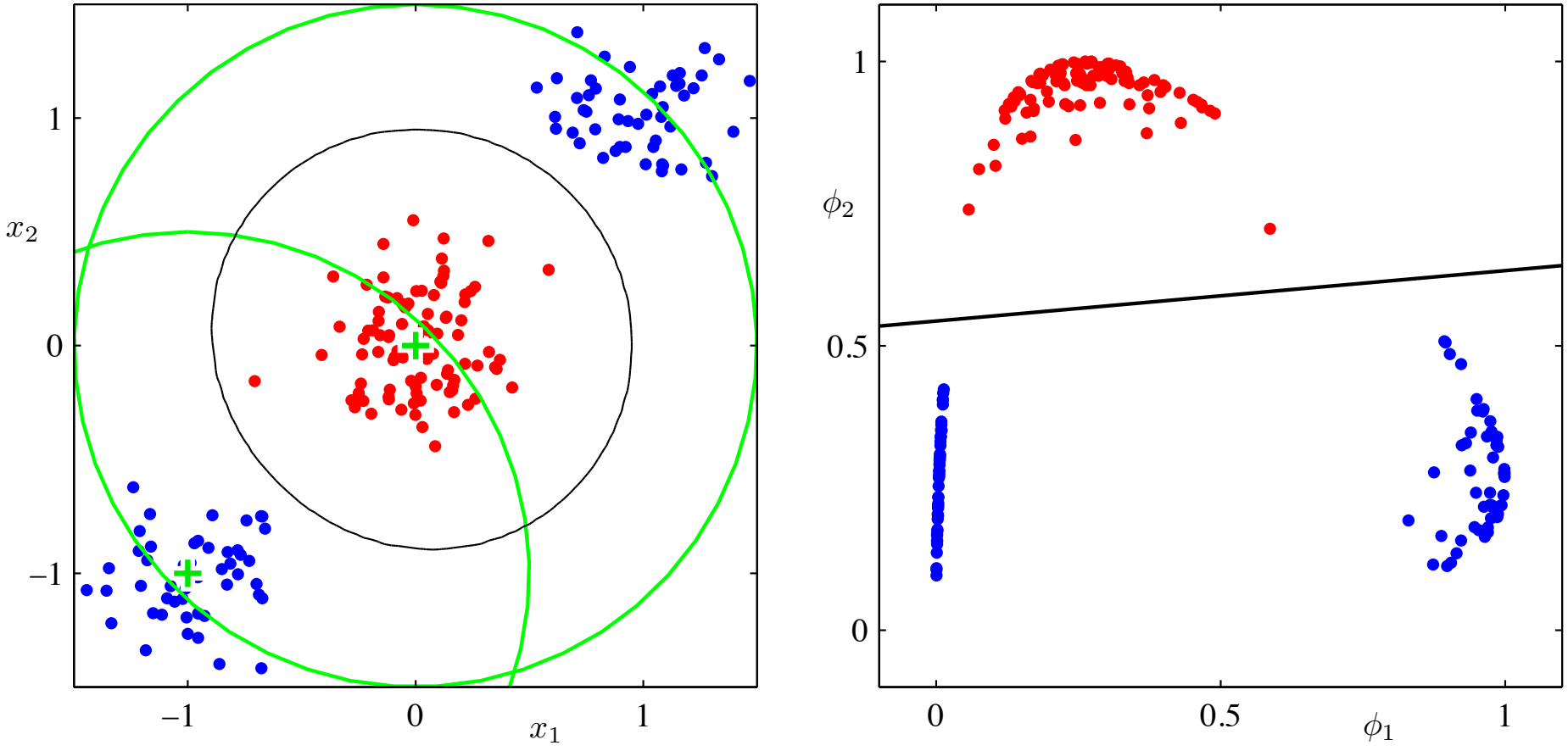
Альтернативный подход, называемый *дискриминационным обучением* , заключается в непосредственной максимизации функции правдоподобия, определенной через условное распределение $p(\mathcal{C}_k|\mathbf{x})$.

Рассмотрим задачу бинарной классификации. При анализе генеративных подходов мы увидели, что при достаточно общих предположениях апостериорная вероятность класса $\mathcal{C}_1$может быть выражена как логистический сигмоид, действующий на линейную функцию входных векторов $\mathbf{x}$ или вектора признаков $\boldsymbol{\phi}$, так что

$$p(\mathcal{C}_1|\boldsymbol{\phi}) = y(\boldsymbol{\phi}) = \sigma(\mathbf{w}^T\boldsymbol{\phi})$$

В терминологии статистики эта модель известна как *логистическая регрессия*, хотя это модель классификации.

Фиксированные базисные функции



Логистическая регрессия

Одним из преимуществ дискриминативного подхода является то, что обычно требуется определять меньше адаптивных параметров. Для M-мерного пространства признаков эта модель имеет Mнастраиваемые параметры. В отличие от этого, генеративная модель, использующая условные плотности гауссовых классов, использовала бы $2M$ параметров для средних значений и $M(M+1)/2$ параметров для (общей) ковариационной матрицы.

Используем метод максимального правдоподобия для определения параметров модели логистической регрессии. Для набора данных $\{\boldsymbol{\phi}_n, t_n\}$, где $t_n \in \{0,1\}$, функция правдоподобия определяется как:

$$p(\mathbf{t}|\boldsymbol{\Phi}, \mathbf{w}) = \prod_{n=1}^{N} p\,(\mathcal{C}_1|\boldsymbol{\phi}_n)^{t_n}(1 - p(\mathcal{C}_1|\boldsymbol{\phi}_n))^{1-t_n} = \prod_{n=1}^{N} y_n^{t_n}\,\{1 - y_n\}^{1-t_n}$$

Максимальное правдоподобие эквивалентно минимуму отрицательного логарифма правдоподобия, что дает *функцию ошибки кросс-энтропии* ,

$$E(\mathbf{w}) = -\ln p(\mathbf{t}|\boldsymbol{\Phi}, \mathbf{w}) = -\sum_{n=1}^{N} t_n \ln y_n + (1 - t_n)\ln(1 - y_n)$$

Почему функция ошибок называется кросс-энтропией?

Перекрёстная энтропия для дискретных распределений вероятностей p и q определяется как

$$H(p,q) = \sum_x p\ (x)\log q(x)$$

Поскольку мы предполагаем, что целевые переменные t_n — это вероятности,

принимающие только экстремальные значения , 0 или 1, а y_n — распределение вероятностей, то $E(\mathbf{w})$можно интерпретировать как перекрёстную энтропию целевых переменных и апостериорного распределения вероятностей.

Поиск весов в лог. регресии

Взяв градиент функции ошибки по $\mathbf{w}$, получим,

$$\nabla E(\mathbf{w}) = -\nabla \ln p(\mathbf{t}|\mathbf{\Phi}, \mathbf{w}) = -\nabla \sum_{n=1}^{N} t_n \ln y_n + (1 - t_n)\ln(1 - y_n)$$

$$= -\sum_{n=1}^{N} \frac{d}{dy_n} t_n \ln y_n + \frac{d}{dy_n}(1 - t_n)\ln(1 - y_n) \overset{\frac{d}{dx}\ln f(x) = \frac{f'(x)}{f(x)}}{=} -\sum_{n=1}^{N} \frac{d}{dy_n} t_n \ln y_n + \frac{d}{dy_n}(1 - t_n)\ln(1 - y_n)$$

$$= -\sum_{n=1}^{N} \frac{t_n}{y_n}\frac{d}{da_n} y_n \frac{d}{d\mathbf{w}} a_n - \frac{1 - t_n}{1 - y_n}\frac{d}{da_n} y_n \frac{d}{d\mathbf{w}} a_n = -\sum_{n=1}^{N} \left(\frac{t_n}{y_n} - \frac{1 - t_n}{1 - y_n}\right)\frac{d}{da_n} y_n \frac{d}{d\mathbf{w}} a_n$$

$$= -\sum_{n=1}^{N} \left(\frac{t_n}{y_n} - \frac{1 - t_n}{1 - y_n}\right) y_n(1 - y_n)\mathbf{\phi}_n = -\sum_{n=1}^{N} \frac{t_n - y_n}{y_n(1 - t_n)} y_n(1 - y_n)\mathbf{\phi}_n$$

$$= \sum_{n=1}^{N} (y_n - t_n)\mathbf{\phi}_n$$

Обратите внимание, что градиент имеет ту же форму, что и градиент функции ошибок суммы квадратов, однако y_n содержит нелинейную функцию. В этом случае мы можем использовать последовательный алгоритм (градиентного спуска) для оптимизации параметров.

Итеративный метод наименьших квадратов с пересчетом весов

Для логистической регрессии аналитического решения больше не существует из-за нелинейности логистической сигмоидальной функции. Однако функция ошибки по-прежнему выпукла и может быть минимизирована эффективным итерационным методом, основанным на итеративной схеме оптимизации *Ньютона*. Этот алгоритм использует локальное квадратичное приближение логарифмического правдоподобия и принимает вид

$$\mathbf{w}^{new} = \mathbf{w}^{old} - \mathbf{H}^{-1}\nabla E(\mathbf{w})$$

где $\mathbf{H}$ — матрица Гессе, элементы которой содержат вторые производные $E(\mathbf{w})$по $\mathbf{w}$.

Обратите внимание, что если мы применим алгоритм к модели линейной регрессии, мы получим стандартное решение методом наименьших квадратов.

Итеративный метод наименьших квадратов с пересчетом весов

Применяя обновление к функции ошибок кросс-энтропии для модели логистической регрессии, получаем:

$$\nabla E(\mathbf{w}) = \sum_{n=1}^{N} (y_n - t_n)\boldsymbol{\phi}_n = \boldsymbol{\Phi}^T(\mathrm{y} - \mathrm{t})$$

и

$$\mathbf{H} = \nabla\nabla E(\mathbf{w}) = \nabla \sum_{n=1}^{N} (y_n - t_n)\boldsymbol{\phi}_n$$

$$= \nabla \sum_{n=1}^{N} y_n \boldsymbol{\phi}_n = \sum_{n=1}^{N} y_n (1 - y_n)\boldsymbol{\phi}_n \frac{d}{d\mathbf{w}} \mathbf{w}^T \boldsymbol{\phi}_n$$

$$= \sum_{n=1}^{N} y_n (1 - y_n)\boldsymbol{\phi}_n \boldsymbol{\phi}_n^T = \boldsymbol{\Phi}^T \mathbf{R} \boldsymbol{\Phi}$$

Итеративный метод наименьших квадратов с пересчетом весов

$$\mathbf{H} = \mathbf{\Phi}^T \mathbf{R} \mathbf{\Phi}$$

где $\mathbf{R}$— диагональная матрица, элементы которой равны $R_{nn} = y_n(1 - y_n)$. Тогда формула обновления принимает вид:

$$\mathbf{w}^{new} = \mathbf{w}^{old} - (\mathbf{\Phi}^T \mathbf{R} \mathbf{\Phi})^{-1} \mathbf{\Phi}^T (\mathrm{y} - \mathrm{t})$$

Обратите внимание, что гессиан зависит от $\mathbf{w}$ матрицы весов $\mathbf{R}$, что соответствует тому факту, что функция ошибки больше не является квадратичной. Таким образом, формулу обновления необходимо применять итеративно, каждый раз используя новый вектор весов $\mathbf{w}$ для вычисления изменённой матрицы весов $\mathbf{R}$. В связи с этим алгоритм называется *итеративным методом наименьших квадратов с пересчетом весов* (*IRLS)*.

Элементы $\mathbf{R}$ можно интерпретировать как дисперсии, заданные формулой:

$$\mathbb{E}[t] = \sum_{t \in \{0,1\}} t \, p(t|\mathbf{x}) = \sigma(x)$$

$$\mathrm{var}[t] = \mathbb{E}[t^2] - \mathbb{E}[t]^2 \overset{t^2=t}{=} \mathbb{E}[t] - \mathbb{E}[t]^2 = \sigma(x) - \sigma(x)^2 = y(1 - y)$$

Пробит-регрессия

Для многих условных по классу плотностей вероятности, принадлежащих экспоненциальному семейству, полученные апостериорные вероятности классов определяются логистическим (или softmax) преобразованием, действующим на линейную функцию признаков. Однако не все виды условной по классу плотности вероятности приводят к такой простой форме для апостериорных вероятностей (например, если условные по классу плотности вероятности моделируются с использованием смесей нормальных распределений). Это говорит о том, что, возможно, стоит изучить другие типы дискриминантных вероятностных моделей.

$$p(t = 1|a) = f(a),$$

где $a = \boldsymbol{w}^t \boldsymbol{\phi}$, а f – функция активации. Целевое значение t задается согласно формуле:

$$\begin{cases} t = 1, \text{если } a \geq \theta \\ t = 0, \text{иначе} \end{cases}$$

Тогда

$$f(a) = \int_{-\infty}^{a} p(\theta) d\theta$$

Частный случай $f(a) = \int_{-\infty}^{a} \mathcal{N}(\theta|0,1) d\theta$ - обратная пробит-функция

Обобщенная линейная модель, основанная на пробит-функции активации, называется пробит-регрессией.

Учет ошибки измерений

$$p(t|x) = (1 - \varepsilon)\sigma(x) + \varepsilon(1 - \sigma(x))$$

ε – вероятность правильной разметки.

Байесовская логистическая регрессия

Точный байесовский вывод для логистической регрессии невозможен. В частности, вычисление апостериорного распределения требует нормировки произведения априорного распределения и функции правдоподобия, которая сама по себе со держит произведение логистических сигмоид, по одной для каждой точки наблюдения. Вычисление прогностического распределения также невозможно.

Возможно применение аппроксимации Лапласа к проблеме байесовской логистической регрессии.

Приближение Лапласа

Ищем гауссовское представление для апостериорного распределения параметров, поэтому мы используем гауссовское (сопряженное) априорное распределение в общей форме,

$$p(\mathbf{w}) = \mathcal{N}(\mathbf{w}|\mathbf{m}_0, \mathbf{S}_0)$$

где $\mathbf{m}_0, \mathbf{S}_0$– фиксированные гиперпараметры. Тогда апостериорная функция $\mathbf{w}$ определяется как

$$p(\mathbf{w}|\mathrm{t}) \propto p(\mathrm{t}|\mathbf{w})p(\mathbf{w})$$

Взяв натуральный логарифм с обеих сторон и подставив вместо него априорную вероятность и вероятность, получаем

$$\ln p(\mathbf{w}|\mathrm{t}) = \ln\left(\prod_{n=1}^{N} y_n^{t_n}\{1-y_n\}^{1-t_n}\right)\ln\mathcal{N}(\mathbf{w}|\mathbf{m}_0, \mathbf{S}_0)$$

$$= \sum_{n=1}^{N}\{t_n \ln y_n + (1-t_n)\ln(1-y_n)\} + \frac{1}{(2\pi)^{D/2}|\mathbf{S}_0|^{1/2}} - \frac{1}{2}(\mathbf{w}-\mathbf{m}_0)^T\mathbf{S}_0^{-1}(\mathbf{w}-\mathbf{m}_0)$$

Приближение Лапласа

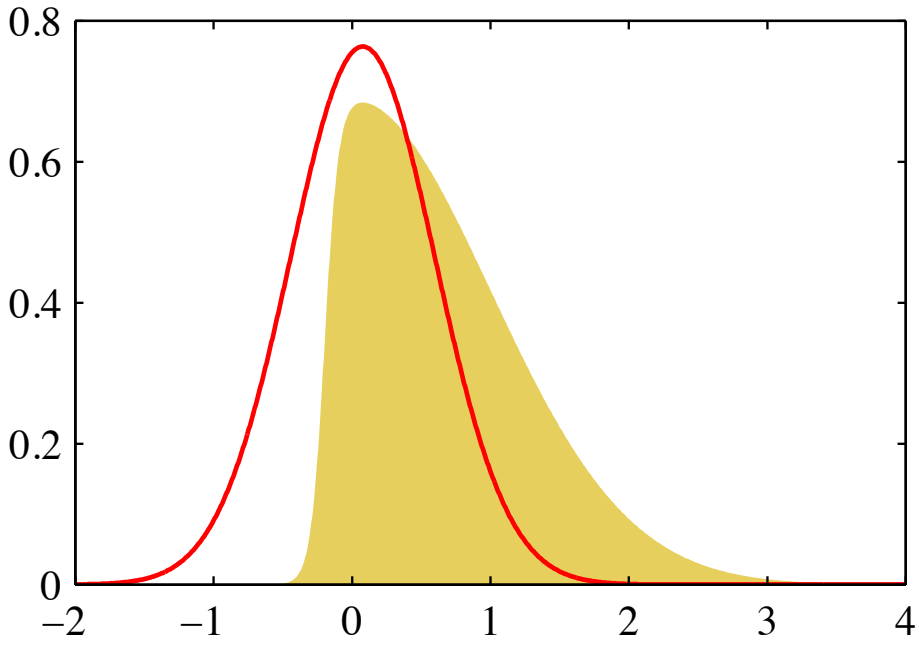
Чтобы получить гауссовскую аппроксимацию апостериорной вероятности, сначала максимизируем апостериорную вероятность, чтобы получить решение MAP $\mathbf{w}_{MAP}$, которое соответствует среднему значению аппроксимированной гауссовой вероятности. Ковариационная матрица затем задаётся матрицей Гессе отрицательного логарифмического правдоподобия:

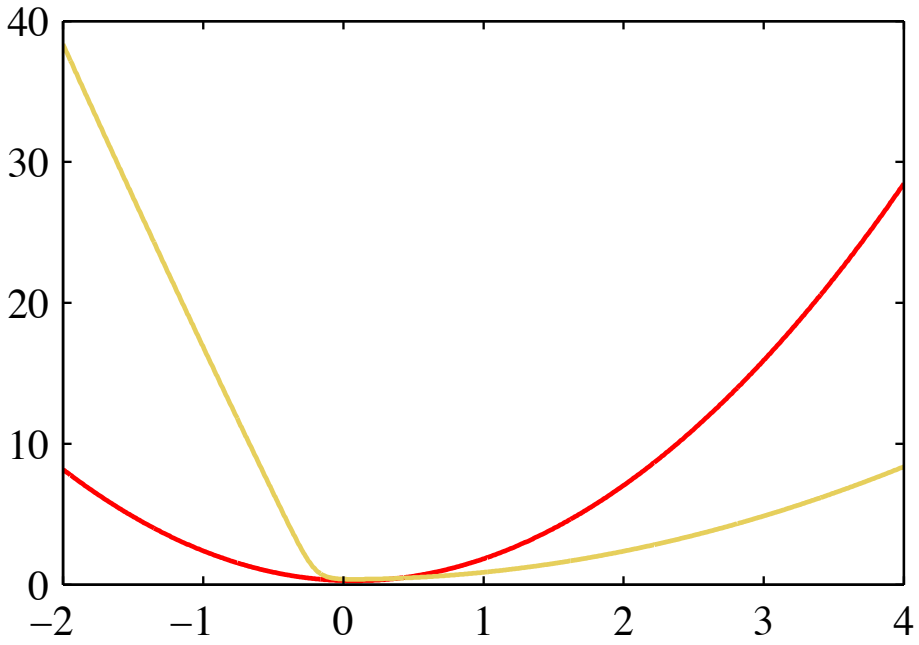
$$\mathbf{S}_N^{-1} = -\nabla\nabla \ln p(\mathbf{w}|\mathrm{t}) = \mathbf{S}_0^{-1} + \sum_{n=1}^{N} y_n\,(1 - y_n)\boldsymbol{\Phi}_n \boldsymbol{\Phi}_n^T$$

где мы используем (4.97). Таким образом, гауссовское приближение апостериорной функции принимает вид:

$$q(\mathbf{w}) = \mathcal{N}(\mathbf{w}|\mathbf{w}_{MAP}, \mathbf{S}_N)$$

Приближение Лапласа





Прогностическое распределение

Прогностическое распределение для класса $\mathcal{C}_1$, учитывая новый вектор признаков $\boldsymbol{\phi}_{unseen}$, получается путем маргинализации по апостериорному распределению $p(\mathbf{w}|\mathrm{t})$, которое аппроксимируется формулой $q(\mathbf{w})$, так что,

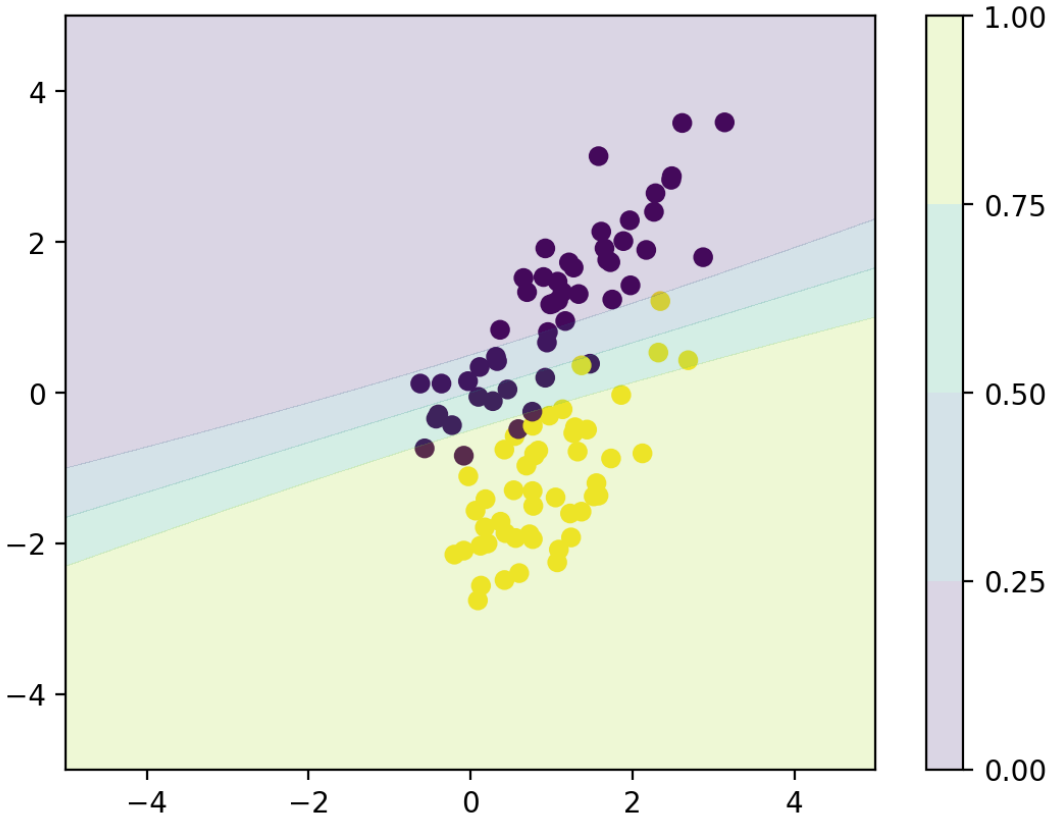
$$p(\mathcal{C}_1|\boldsymbol{\phi}_{unseen}, \mathrm{t}, \boldsymbol{\Phi}) = \int p(\mathcal{C}_1|\boldsymbol{\phi}_{unseen}, \mathbf{w}) p(\mathbf{w}|\mathrm{t}, \boldsymbol{\Phi}) \mathrm{d}\mathbf{w} \approx \int \sigma(\mathbf{w}^T \boldsymbol{\phi}) q(\mathbf{w}) \mathrm{d}\mathbf{w}$$

Вычисление этого интеграла довольно сложно и требует значительного количества шагов. Подробнее см. соответствующий раздел книги.
Окончательное приближенное предсказательное распределение имеет вид:

$$p(\mathcal{C}_1|\boldsymbol{\phi}_{unseen}, \mathrm{t}) = \sigma(\kappa(\sigma_\alpha^2)\mu_\alpha)$$

где $\kappa(\sigma_\alpha^2) = (1 + \pi\sigma_\alpha^2/8)^{-1/2}$, $\mu_a = \mathbf{w}_{MAP}^T \boldsymbol{\phi}_{unseen}$, и $\sigma_\alpha = \boldsymbol{\phi}_{unseen}^T \mathbf{S}_N \boldsymbol{\phi}_{unseen}$.

Прогностическое распределение



Оценка качества классификации

Пусть есть два класса Y={0,1}. Пусть банк использует систему классификации заёмщиков на кредитоспособных и некредитоспособных. Обнаружение некредитоспособного заёмщика (y=1) можно рассматривать как "сигнал тревоги", сообщающий о возможных рисках.

Возможны следующие исходы классификации:

- Некредитоспособный заёмщик классифицирован как некредитоспособный, т.е. положительный класс распознан как положительный (True Positive — TP).
- Кредитоспособный заёмщик классифицирован как кредитоспособный, т.е. отрицательный класс распознан как отрицательный. (True Negative — TN).
- Кредитоспособный заёмщик классифицирован как некредитоспособный, т.е. имела место ошибка, в результате которой отрицательный класс был распознан как положительный (False Positive — FP) – это ошибка I рода (ложная тревога).
- Некредитоспособный заёмщик распознан как кредитоспособный, т.е. имела место ошибка, в результате которой положительный класс был распознан как отрицательный (False Negative — FN) – это ошибка II рода (пропуск цели).

Вопрос для самоконтроля

Где ошибка первого рода и где ошибка второго рода?

Метрики качества классификации

Аккуратность (англ. Accuracy) – доля правильных ответов. Бесполезна в задачах с неравными классами.

Точность (англ. Precision) - доля правильных ответов модели в пределах класса:

$$\text{Precision} = \frac{TP}{TP + FP}$$

Полнота (англ. Recall) - это доля истинно положительных классификаций:

$$\text{Recall} = \frac{TP}{TP + FN}$$

F-мера (англ. F-score) – гармоническое среднее между точностью и полнотой.

Сравнение метрик

Модель 1

F1=0.34, accuracy=0.48, precision=0.21, recall=0.95

[[51 77]
 [1 20]]

Модель 2

F1=0.54, accuracy=0.82, precision= 0.42, recall=0.76

[[106 22]
[5 16]]