Классификация

Сергей Николенко

Академия больших данных MADE — VK

21 февраля 2022 г.

Random facts:

- 21 февраля в ЮНЕСКО — Международный день родного языка; около половины из 6 тысяч языков мира находятся под угрозой исчезновения; в частности, в 2009 году ЮНЕСКО признала таковыми 136 языков на территории России
- 21 февраля 1804 г. Ричард Тревитик впервые публично продемонстрировал самодвижущуюся паровую повозку на рельсах — прототип паровоза
- 21 февраля 1848 г. Карл Маркс и Фридрих Энгельс запустили в Европу призрак коммунизма, опубликовав «Манифест Коммунистической партии»
- 21 февраля 1953 г. на основании рентгеноструктурных данных, полученных Морисом Уилкинсом и Розалинд Франклин, Джеймс Уотсон и Френсис Крик предложили структурную модель ДНК — двойную спираль
- 21 февраля 2006 г. телескоп «Хаббл» зарегистрировал объект SCP 06F6, природу которого астрономы не могут объяснить до сих пор: он увеличивал яркость в течение 100 суток (дольше, чем сверхновые), а рентгеновское излучение было вдвое мощнее, чем у сверхновых

Введение в классификацию

Задача классификации

- Теперь классификация: определить вектор $\mathbf{x}$ в один из K классов $\mathcal{C}_k$.
- В итоге у нас так или иначе всё пространство разобьётся на эти классы.
- Т.е. на самом деле мы ищем *разделяющую поверхность* (decision surface, decision boundary).

Задача классификации

- Как кодировать? Бинарная задача – очень естественно, переменная t, $t = 0$ соответствует $\mathcal{C}_1$, $t = 1$ соответствует $\mathcal{C}_2$.
- Оценку t можно интерпретировать как вероятность (по крайней мере, мы постараемся, чтобы было можно).
- Если несколько классов – удобно 1-of-K:

$$\mathbf{t} = (0, \ldots, 0, 1, 0, \ldots)^\top.$$

- Тоже можно интерпретировать как вероятности – или пропорционально им.

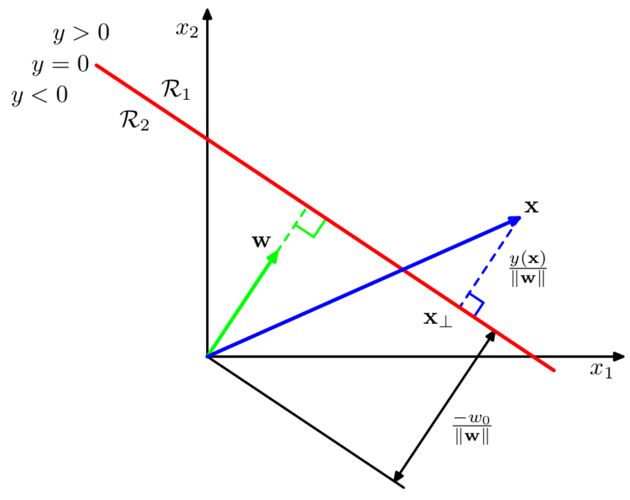
Разделяющая гиперплоскость

- Начнём с геометрии: рассмотрим линейную дискриминантную функцию

$$y(\mathbf{x}) = \mathbf{w}^\top \mathbf{x} + w_0.$$

- Это гиперплоскость, и $\mathbf{w}$ – нормаль к ней.
- Расстояние от начала координат до гиперплоскости равно $\frac{-w_0}{\|\mathbf{w}\|}$.
- $y(\mathbf{x})$ связано с расстоянием до гиперплоскости: $d = \frac{y(\mathbf{x})}{\|\mathbf{w}\|}$.

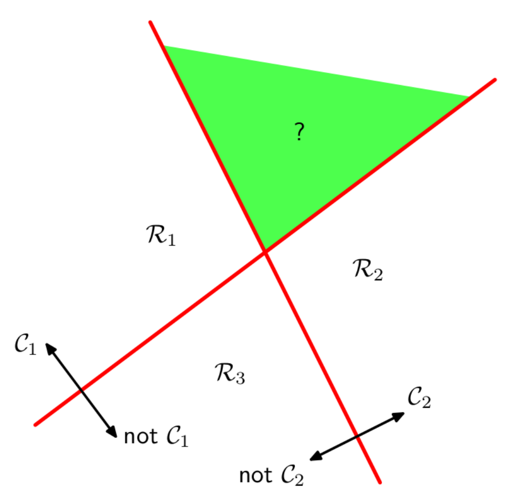
Разделяющая гиперплоскость



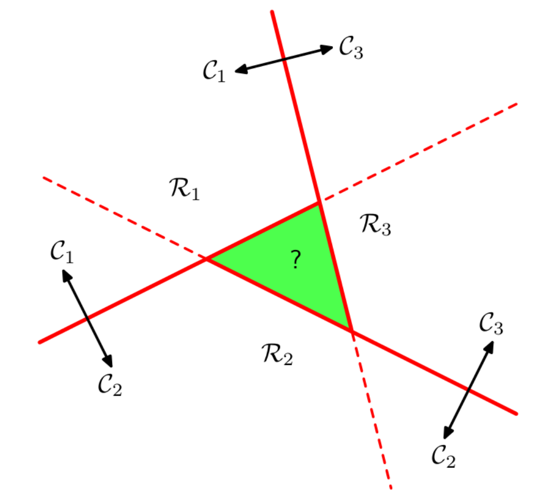
Несколько классов

- С несколькими классами выходит незадача.
- Можно рассмотреть K поверхностей вида «один против всех».
- Можно – $\binom{K}{2}$ поверхностей вида «каждый против каждого».
- Но всё это как-то нехорошо.

Несколько классов



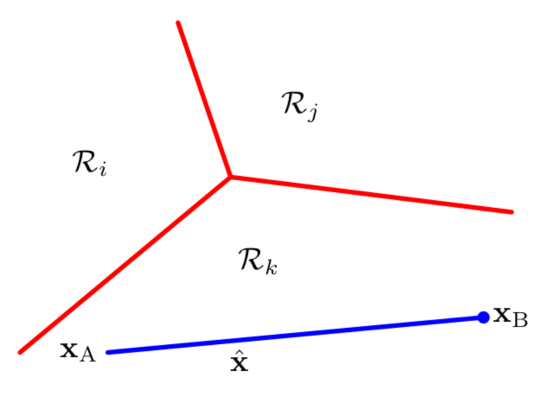
Несколько классов



Несколько классов

- Лучше рассмотреть единый дискриминант из K линейных функций:
$$y_k(\mathbf{x}) = \mathbf{w}_k^\top \mathbf{x} + w_{k0}.$$
- Классифицировать в $\mathcal{C}_k$, если $y_k(\mathbf{x})$ – максимален.
- Тогда разделяющая поверхность между $\mathcal{C}_k$ и $\mathcal{C}_j$ будет гиперплоскостью вида $y_k(\mathbf{x}) = y_j(\mathbf{x})$:
$$\left(\mathbf{w}_k - \mathbf{w}_j\right)^\top \mathbf{x} + \left(w_{k0} - w_{j0}\right).$$

Несколько классов



Упражнение. Докажите, что области, соответствующие классам, при таком подходе всегда односвязные и выпуклые.

Метод наименьших квадратов

- Мы снова можем воспользоваться методом наименьших квадратов: запишем $y_k(\mathbf{x}) = \mathbf{w}_k^\top \mathbf{x} + w_{k0}$ вместе (спрятав свободный член) как
$$\mathbf{y}(\mathbf{x}) = \mathbf{W}^\top \mathbf{x}.$$
- Можно найти $\mathbf{W}$, оптимизируя сумму квадратов; функция ошибки:
$$E_D(\mathbf{W}) = \frac{1}{2}\mathrm{Tr}\left[(\mathbf{X}\mathbf{W} - \mathbf{T})^\top (\mathbf{X}\mathbf{W} - \mathbf{T})\right].$$
- Берём производную, решаем...

Метод наименьших квадратов

- …получается привычное

$$\mathbf{w} = \left(\mathbf{X}^\top \mathbf{X}\right)^{-1} \mathbf{X}^\top \mathbf{T} = \mathbf{X}^\dagger \mathbf{T},$$

  где $\mathbf{X}^\dagger$ – псевдообратная Мура-Пенроуза.

- Теперь можно найти и дискриминантную функцию:

$$y(\mathbf{x}) = \mathbf{W}^\top \mathbf{x} = \mathbf{T}^\top \left(\mathbf{X}^\dagger\right)^\top \mathbf{x}.$$

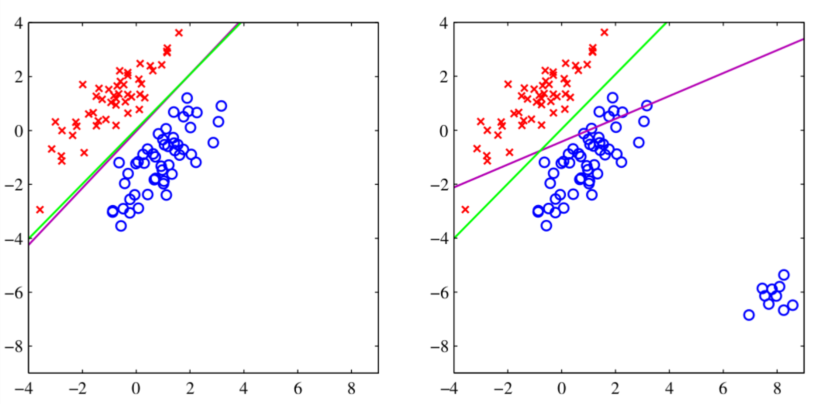
Метод наименьших квадратов

- Это решение сохраняет линейность.

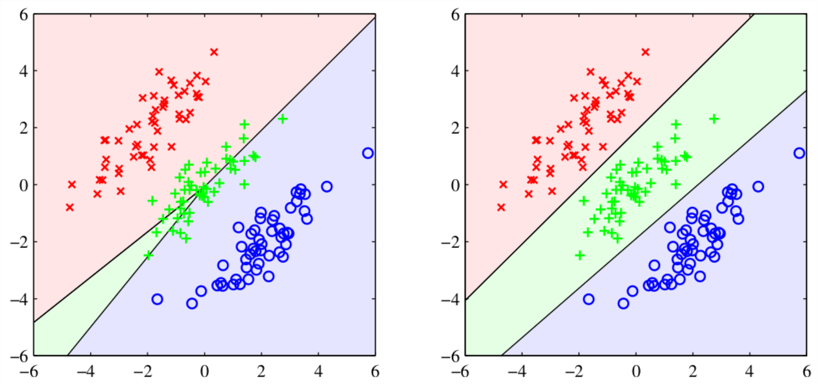
Упражнение. Докажите, что в схеме кодирования 1-of-K предсказания $y_k(\mathbf{x})$ для разных классов при любом $\mathbf{x}$ будут давать в сумме 1. Почему они всё-таки не будут разумными оценками вероятностей?

- Проблемы наименьших квадратов:
  - outliers плохо обрабатываются;
  - «слишком правильные» предсказания добавляют штраф.

Проблемы наименьших квадратов



Проблемы наименьших квадратов



Проблемы наименьших квадратов

- Почему так? Почему наименьшие квадраты так плохо работают?

Проблемы наименьших квадратов

- Почему так? Почему наименьшие квадраты так плохо работают?
- Они предполагают гауссовское распределение ошибки.
- Но, конечно, распределение у бинарных векторов далеко не гауссово.

Линейный дискриминант Фишера

- Другой взгляд на классификацию: в линейном случае мы хотим спроецировать точки в размерность 1 (на нормаль разделяющей гиперплоскости) так, чтобы в этой размерности 1 они хорошо разделялись.
- Т.е. классификация – это такой метод радикального сокращения размерности.
- Давайте посмотрим на классификацию с этих позиций и попробуем добиться оптимальности в каком-то смысле.

Линейный дискриминант Фишера

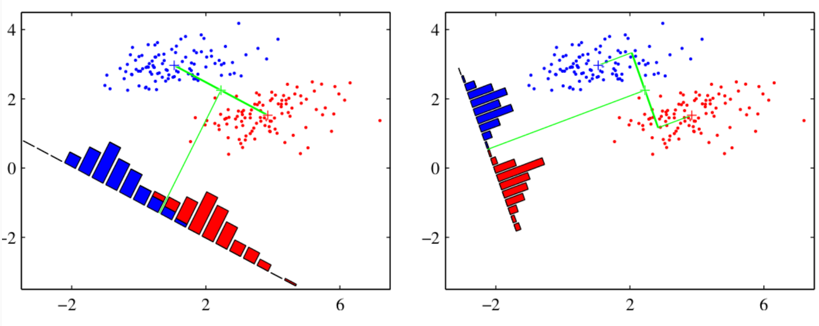
- Рассмотрим два класса $\mathcal{C}_1$ и $\mathcal{C}_2$ с N_1 и N_2 точками.
- Первая идея – надо найти серединный перпендикуляр между центрами кластеров

$$\mathbf{m}_1 = \frac{1}{N_1}\sum_{\mathcal{C}_1}\mathbf{x}, \text{ и } \mathbf{m}_2 = \frac{1}{N_2}\sum_{\mathcal{C}_2}\mathbf{x},$$

$$\text{т.е. максимизировать } \mathbf{w}^\top(\mathbf{m}_2 - \mathbf{m}_1).$$

- Надо ещё добавить ограничение $\|\mathbf{w}\| = 1$, но всё равно не ахти как работает.

Линейный дискриминант Фишера



Чем левая картинка хуже правой?

Линейный дискриминант Фишера

- Слева больше дисперсия каждого кластера.
- Идея: минимизировать перекрытие классов, оптимизируя и проекцию расстояния, и дисперсию.
- Выборочные дисперсии в проекции: для $y_n = \mathbf{w}^\top \mathbf{x}_n$

$$s_1 = \sum_{n \in \mathcal{C}_1} (y_n - m_1)^2 \text{ и } s_1 = \sum_{n \in \mathcal{C}_2} (y_n - m_2)^2 .$$

Линейный дискриминант Фишера

- Критерий Фишера:

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

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

$$\mathbf{S}_W = \sum_{n \in \mathcal{C}_1} (\mathbf{x}_n - \mathbf{m}_1)(\mathbf{x}_n - \mathbf{m}_1)^\top + \sum_{n \in \mathcal{C}_2} (\mathbf{x}_n - \mathbf{m}_2)(\mathbf{x}_n - \mathbf{m}_2)^\top.$$

  (between-class covariance и within-class covariance).
- Дифференцируя по $\mathbf{w}$...

Линейный дискриминант Фишера

- ...получим, что $J(\mathbf{w})$ максимален при

$$\left(\mathbf{w}^\top \mathbf{S}_B \mathbf{w}\right) \mathbf{S}_W \mathbf{w} = \left(\mathbf{w}^\top \mathbf{S}_W \mathbf{w}\right) \mathbf{S}_B \mathbf{w}.$$

- Т.к. $\mathbf{S}_B = (\mathbf{m}_2 - \mathbf{m}_1)(\mathbf{m}_2 - \mathbf{m}_1)^\top$, $\mathbf{S}_B \mathbf{w}$ всё равно будет в направлении $\mathbf{m}_2 - \mathbf{m}_1$, а длина $\mathbf{w}$ нас не интересует.
- Поэтому получается

$$\mathbf{w} \propto \mathbf{S}_W^{-1} (\mathbf{m}_2 - \mathbf{m}_1).$$

- В итоге мы выбрали направление проекции, и осталось только разделить данные на этой проекции.

Линейный дискриминант Фишера

- Любопытно, что дискриминант Фишера тоже можно получить из наименьших квадратов.
- Давайте для класса $\mathcal{C}_1$ выберем целевое значение $\frac{N_1+N_2}{N_1}$, а для класса $\mathcal{C}_2$ возьмём $-\frac{N_1+N_2}{N_2}$.

Упражнение. Докажите, что при таких целевых значениях наименьшие квадраты – это дискриминант Фишера.

Линейный дискриминант Фишера

- А что будет с несколькими классами? Рассмотрим $\mathbf{y} = \mathbf{W}^\top \mathbf{x}$, обобщим внутреннюю дисперсию как

$$\mathbf{S}_W = \sum_{k=1}^{K} \mathbf{S}_k = \sum_{k=1}^{K} \sum_{n \in \mathcal{C}_k} (\mathbf{x}_n - \mathbf{m}_k)(\mathbf{x}_n - \mathbf{m}_k)^\top .$$

- Чтобы обобщить внешнюю (межклассовую) дисперсию, просто возьмём остаток полной дисперсии

$$\mathbf{S}_T = \sum_{n} (\mathbf{x}_n - \mathbf{m})(\mathbf{x}_n - \mathbf{m})^\top ,$$

$$\mathbf{S}_B = \mathbf{S}_T - \mathbf{S}_W.$$

Линейный дискриминант Фишера

- Обобщить критерий можно разными способами, например:

$$J(\mathrm{W}) = \mathrm{Tr}\left[\mathrm{s}_W^{-1}\mathrm{s}_B\right],$$

где s – ковариации в пространстве проекций на y:

$$\mathrm{s}_W = \sum_{k=1}^{K}\sum_{n\in\mathcal{C}_k}\left(\mathrm{y}_n - \boldsymbol{\mu}_k\right)\left(\mathrm{y}_n - \boldsymbol{\mu}_k\right)^\top,$$

$$\mathrm{s}_B = \sum_{k=1}^{K} N_k\left(\boldsymbol{\mu}_k - \boldsymbol{\mu}\right)\left(\boldsymbol{\mu}_k - \boldsymbol{\mu}\right)^\top,$$

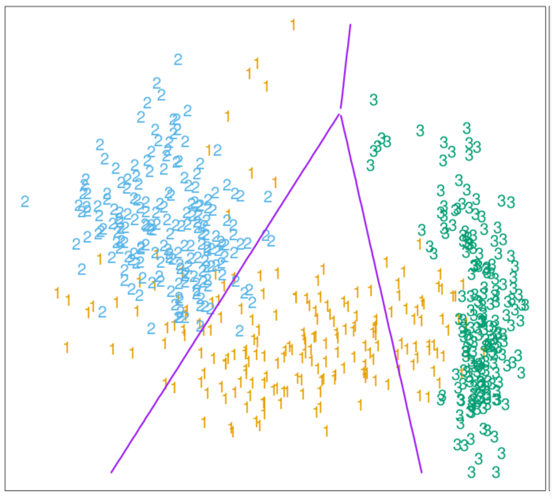
где $\boldsymbol{\mu}_k = \frac{1}{N_k}\sum_{n\in\mathcal{C}_k}\mathrm{y}_n$.

LDA и QDA

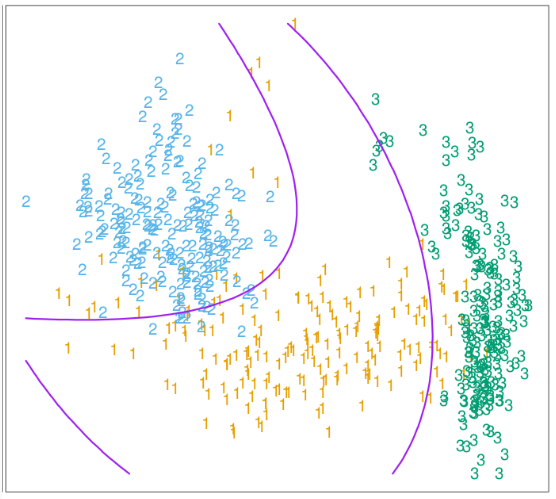
Нелинейные поверхности

- Мы учились проводить разделяющие гиперплоскости.
- Но как же нелинейные поверхности?
- Можно делать нелинейные из линейных, увеличивая размерность.

Нелинейные поверхности



Нелинейные поверхности



Генеративные модели

- Теперь классификация через генеративные модели: давайте каждому классу сопоставим плотность $p(\mathrm{x} \mid \mathcal{C}_k)$, найдём априорные распределения $p(\mathcal{C}_k)$, будем искать $p(\mathcal{C}_k \mid \mathrm{x})$ по теореме Байеса.
- Для двух классов:

$$p(\mathcal{C}_1 \mid \mathrm{x}) = \frac{p(\mathrm{x} \mid \mathcal{C}_1)p(\mathcal{C}_1)}{p(\mathrm{x} \mid \mathcal{C}_1)p(\mathcal{C}_1) + p(\mathrm{x} \mid \mathcal{C}_2)p(\mathcal{C}_2)}.$$

Генеративные модели

- Перепишем:

$$p(\mathcal{C}_1 \mid \mathbf{x}) = \frac{p(\mathbf{x} \mid \mathcal{C}_1)p(\mathcal{C}_1)}{p(\mathbf{x} \mid \mathcal{C}_1)p(\mathcal{C}_1) + p(\mathbf{x} \mid \mathcal{C}_2)p(\mathcal{C}_2)} = \frac{1}{1 + e^{-a}} = \sigma(a),$$

где

$$a = \ln \frac{p(\mathbf{x} \mid \mathcal{C}_1)p(\mathcal{C}_1)}{p(\mathbf{x} \mid \mathcal{C}_2)p(\mathcal{C}_2)}, \qquad \sigma(a) = \frac{1}{1 + e^{-a}}.$$

- $\sigma(a)$ – *логистический сигмоид*:

$$\sigma(a) = \frac{1}{1 + e^{-a}}$$

- $\sigma(-a) = 1 - \sigma(a)$.
- $a = \ln\left(\frac{\sigma}{1-\sigma}\right)$ – *логит-функция*.

Упражнение. Докажите эти свойства.

Несколько классов

- В случае нескольких классов получится

$$p(\mathcal{C}_k \mid \mathrm{x}) = \frac{p(\mathrm{x} \mid \mathcal{C}_k)p(\mathcal{C}_k)}{\sum_j p(\mathrm{x} \mid \mathcal{C}_j)p(\mathcal{C}_j)} = \frac{e^{a_k}}{\sum_j e^{a_j}}.$$

- Здесь $a_k = \ln p(\mathrm{x} \mid \mathcal{C}_k)p(\mathcal{C}_k)$.
- $\frac{e^{a_k}}{\sum_j e^{a_j}}$ – нормализованная экспонента, или softmax-функция (сглаженный максимум).

Пример

- Давайте рассмотрим гауссовы распределения для классов:

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

- Сначала пусть Σ у всех одинаковые, а классов всего два.
- Посчитаем логистический сигмоид...

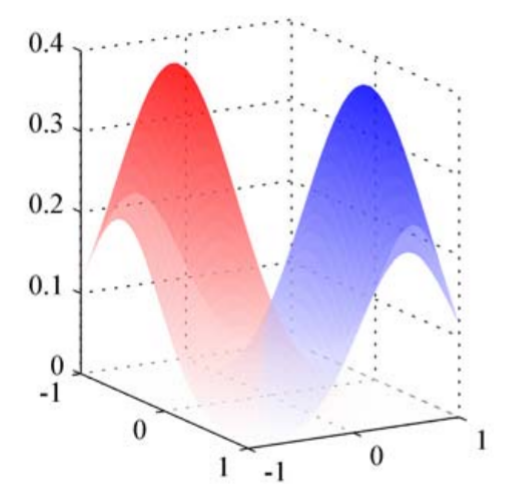
Пример

- …получится

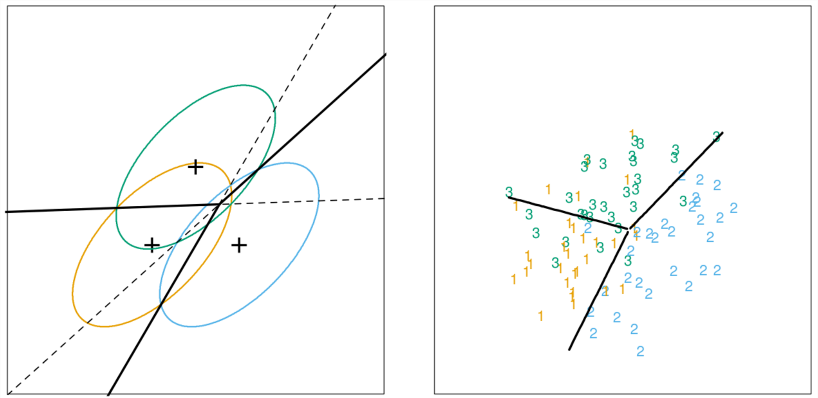
$$\begin{aligned} p(\mathcal{C}_1 \mid \mathbf{x}) &= \sigma(\mathbf{w}^\top \mathbf{x} + w_0), \text{ где} \\ \mathbf{w} &= \Sigma^{-1}\left(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2\right), \\ w_0 &= -\frac{1}{2}\boldsymbol{\mu}_1^\top \Sigma^{-1}\boldsymbol{\mu}_1 + \frac{1}{2}\boldsymbol{\mu}_2^\top \Sigma^{-1}\boldsymbol{\mu}_2 + \ln\frac{p(\mathcal{C}_1)}{p(\mathcal{C}_2)}. \end{aligned}$$

- Т.е. в аргументе сигмоида получается линейная функция от $\mathbf{x}$. Поверхности уровня – это когда $p(\mathcal{C}_1 \mid \mathbf{x})$ постоянно, т.е. гиперплоскости в пространстве $\mathbf{x}$. Априорные вероятности $p(\mathcal{C}_k)$ просто сдвигают эти гиперплоскости.

Разделяющая гиперплоскость

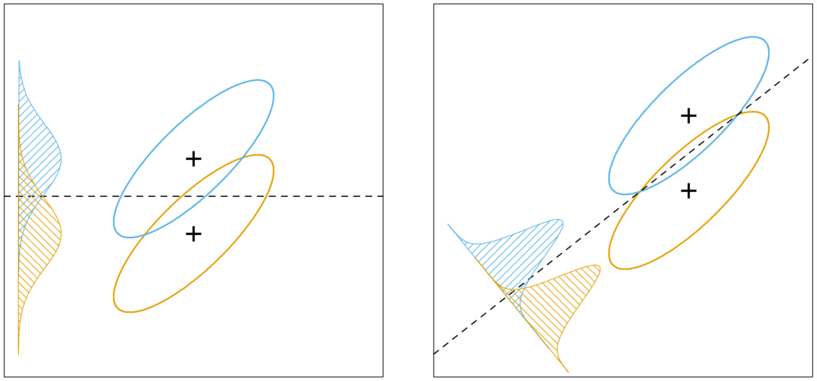


Разделяющая гиперплоскость



Дискриминант Фишера

Кстати, с дискриминантом Фишера эта разделяющая поверхность отлично сходится.



Несколько классов

- С несколькими классами получится тоже примерно так же:

$$\delta_k(\mathrm{x}) = \mathrm{x}^\top \Sigma^{-1} \boldsymbol{\mu}_k - \frac{1}{2} \boldsymbol{\mu}_k^\top \Sigma^{-1} \boldsymbol{\mu}_k + \ln \pi_k,$$

  где $\pi_k = p(\mathcal{C}_k)$.
- Получились линейные $\delta_k(\mathrm{x})$, и опять разделяющие поверхности линейные (тут разделяющие поверхности – когда две максимальных вероятности равны).
- Этот метод называется LDA – linear discriminant analysis.

Метод максимального правдоподобия

- Как оценить распределения $p(\mathbf{x} \mid \mathcal{C}_k)$, если даны только данные?
- Можно по методу максимального правдоподобия.
- Опять рассмотрим тот же пример: два класса, гауссианы с одинаковой матрицей ковариаций, и есть $D = \{\mathbf{x}_n, t_n\}_{n=1}^{N}$, где $t_n = 1$ значит $\mathcal{C}_1$, $t_n = 0$ значит $\mathcal{C}_2$.
- Обозначим $p(\mathcal{C}_1) = \pi$, $p(\mathcal{C}_2) = 1 - \pi$.

Метод максимального правдоподобия

- Для одной точки в классе $\mathcal{C}_1$:

$$p(\mathbf{x}_n, \mathcal{C}_1) = p(\mathcal{C}_1)p(\mathbf{x}_n \mid \mathcal{C}_1) = \pi\mathcal{N}(\mathbf{x}_n \mid \mu_1, \Sigma).$$

- В классе $\mathcal{C}_2$:

$$p(\mathbf{x}_n, \mathcal{C}_2) = p(\mathcal{C}_2)p(\mathbf{x}_n \mid \mathcal{C}_2) = (1-\pi)\mathcal{N}(\mathbf{x}_n \mid \mu_2, \Sigma).$$

- Функция правдоподобия:

$$p(\mathbf{t} \mid \pi, \mu_1, \mu_2, \Sigma) = \\ = \prod_{n=1}^{N} \left[\pi\mathcal{N}(\mathbf{x}_n \mid \mu_1, \Sigma)\right]^{t_n} \left[(1-\pi)\mathcal{N}(\mathbf{x}_n \mid \mu_2, \Sigma)\right]^{1-t_n}.$$

Метод максимального правдоподобия

- Максимизируем логарифм правдоподобия. Сначала по π, там останется только

$$\sum_{n=1}^{N} \left[t_n \ln \pi + (1 - t_n) \ln(1 - \pi) \right],$$

и, взяв производную, получим, совершенно неожиданно,

$$\hat{\pi} = \frac{N_1}{N_1 + N_2}.$$

Метод максимального правдоподобия

- Теперь по μ_1; всё, что зависит от μ_1:

$$\sum_n t_n \ln \mathcal{N}(\mathrm{x}_n \mid \mu_1, \Sigma) = -\frac{1}{2} \sum_n t_n \left(\mathrm{x}_n - \boldsymbol{\mu}_1\right)^\top \Sigma^{-1} \left(\mathrm{x}_n - \boldsymbol{\mu}_1\right) + C.$$

- Берём производную, и получается, опять внезапно,

$$\hat{\mu}_1 = \frac{1}{N_1} \sum_{n=1}^{N} t_n \mathrm{x}_n.$$

- Аналогично,

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

Метод максимального правдоподобия

- Для матрицы ковариаций придётся постараться; в результате получится

$$\hat{\Sigma} = \frac{N_1}{N_1 + N_2}\mathrm{S}_1 + \frac{N_2}{N_1 + N_2}\mathrm{S}_2, \text{ где}$$

$$\mathrm{S}_1 = \frac{1}{N_1}\sum_{n\in\mathcal{C}_1}\left(\mathrm{x}_n - \boldsymbol{\mu}_1\right)\left(\mathrm{x}_n - \boldsymbol{\mu}_1\right)^\top,$$

$$\mathrm{S}_2 = \frac{1}{N_2}\sum_{n\in\mathcal{C}_2}\left(\mathrm{x}_n - \boldsymbol{\mu}_2\right)\left(\mathrm{x}_n - \boldsymbol{\mu}_2\right)^\top.$$

- Тоже совершенно неожиданно: взвешенное среднее оценок для двух матриц ковариаций.

Метод максимального правдоподобия

- Это самым прямым образом обобщается на случай нескольких классов.

Упражнение. Сделайте это.

Разные матрицы ковариаций

- А вот с разными матрицами ковариаций уже будет по-другому.
- Квадратичные члены не сократятся.
- Разделяющие поверхности станут квадратичными; QDA – quadratic discriminant analysis.

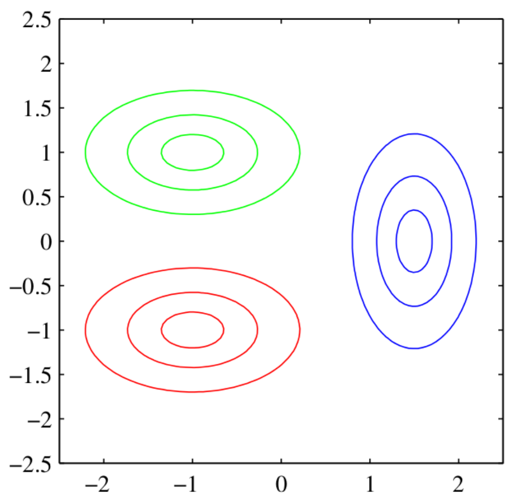
Разные матрицы ковариаций

- В QDA получится

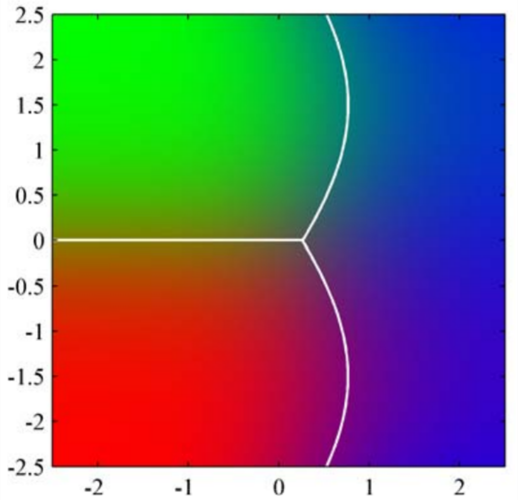
$$\delta_k(\mathrm{x}) = -\frac{1}{2}\log|\Sigma_k| - \frac{1}{2}(\mathrm{x} - \boldsymbol{\mu}_k)^{-1}\Sigma_k^{-1}(\mathrm{x} - \boldsymbol{\mu}_k) + \log\pi_k.$$

- Разделяющая поверхность между $\mathcal{C}_i$ и $\mathcal{C}_j$ – это $\{\mathrm{x} \mid \delta_i(\mathrm{x}) = \delta_j(\mathrm{x})\}$.
- Оценки максимального правдоподобия такие же, только надо отдельно матрицы ковариаций оценивать.

Разные матрицы ковариаций

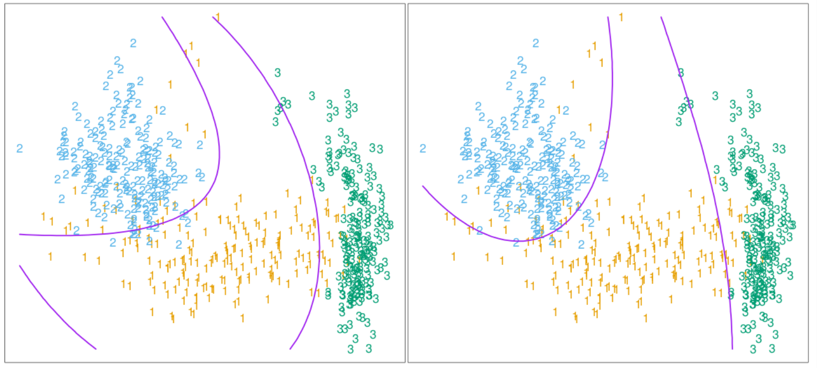


Разные матрицы ковариаций



LDA vs. QDA

Разница между LDA с квадратичными членами и QDA обычно невелика.



LDA vs. QDA

- LDA и QDA неплохо работают на практике. Часто это первая идея в классификации.
- Число параметров:
  - у LDA $(K-1)(d+1)$ параметр: по $d+1$ на каждую разницу вида $\delta_k(\mathrm{x}) - \delta_K(\mathrm{x})$;
  - у QDA $(K-1)(d(d+3)/2+1)$ параметр, но он выглядит гораздо лучше своих лет.

LDA vs. QDA

- Почему хорошо работают?
- Скорее всего, потому, что линейные и квадратичные оценки достаточно стабильны: даже если bias относительно большой (как будет, если данные всё-таки не гауссианами порождены), variance будет маленькой.

RDA

- Компромисс между LDA и QDA – регуляризованный дискриминантный анализ, RDA.
- Стянем ковариации каждого класса к общей матрице ковариаций:
$$\hat{\Sigma}_k(\alpha) = \alpha\hat{\Sigma}_k + (1-\alpha)\hat{\Sigma},$$
где $\hat{\Sigma}_k$ – оценка из QDA, $\hat{\Sigma}$ – оценка из LDA.
- Или стянем к единичной матрице:
$$\hat{\Sigma}_k(\gamma) = \gamma\hat{\Sigma}_k + (1-\gamma)\hat{\sigma}^2 \boldsymbol{I}.$$

Снижение ранга в LDA

- Предположим, что размерность d больше, чем число классов K.
- Тогда центроиды классов $\hat{\mu}_k$ лежат в подпространстве размерности $\leq K - 1$.
- И когда мы определяем ближайший центроид, нам достаточно считать расстояния только в этом подпространстве.
- Таким образом, можно сократить ранг задачи.

Снижение ранга в LDA

- Куда именно проецировать? Не обязательно само подпространство, порождённое центроидами, будет оптимальным.
- Это мы уже проходили: для размерности 1 это линейный дискриминант Фишера.
- Это он и есть: оптимальное подпространство будет там, где межклассовая дисперсия максимальна по отношению к внутриклассовой.

Логистическая регрессия

В прошлый раз

- Итак, мы рассмотрели логистический сигмоид:

$$p(\mathcal{C}_1 \mid \mathbf{x}) = \frac{p(\mathbf{x} \mid \mathcal{C}_1)p(\mathcal{C}_1)}{p(\mathbf{x} \mid \mathcal{C}_1)p(\mathcal{C}_1) + p(\mathbf{x} \mid \mathcal{C}_2)p(\mathcal{C}_2)} = \frac{1}{1 + e^{-a}} = \sigma(a),$$

$$\text{где } a = \ln \frac{p(\mathbf{x} \mid \mathcal{C}_1)p(\mathcal{C}_1)}{p(\mathbf{x} \mid \mathcal{C}_2)p(\mathcal{C}_2)}, \qquad \sigma(a) = \frac{1}{1 + e^{-a}}.$$

- Вывели из него LDA и QDA, обучили их методом максимального правдоподобия, а потом отвлеклись на naive Bayes.

Два класса

- Возвращаемся к задаче классификации.
- Два класса, и апостериорное распределение – логистический сигмоид на линейной функции:

$$p(\mathcal{C}_1 \mid \phi) = y(\phi) = \sigma(\mathbf{w}^\top \phi), \quad p(\mathcal{C}_2 \mid \phi) = 1 - p(\mathcal{C}_1 \mid \phi).$$

- *Логистическая регрессия* – это когда мы напрямую оптимизируем $\mathbf{w}$.

Два класса

- Для датасета $\{\phi_n, t_n\}$, $t_n \in \{0, 1\}$, $\phi_n = \phi(\mathbf{x}_n)$:

$$p(\mathbf{t} \mid \mathbf{w}) = \prod_{n=1}^{N} y_n^{t_n}(1 - y_n)^{1-t_n}, \quad y_n = p(\mathcal{C}_1 \mid \phi_n).$$

- Ищем параметры максимального правдоподобия, минимизируя $-\ln p(\mathbf{t} \mid \mathbf{w})$:

$$E(\mathbf{w}) = -\ln p(\mathbf{t} \mid \mathbf{w}) = -\sum_{n=1}^{N} \left[t_n \ln y_n + (1 - t_n) \ln(1 - y_n)\right].$$

Два класса

- Пользуясь тем, что $\sigma' = \sigma(1-\sigma)$, берём градиент (похоже на перцептрон):
$$\nabla E(\mathbf{w}) = \sum_{n=1}^{N} (y_n - t_n)\phi_n.$$
- Если теперь сделать градиентный спуск, получим как раз разделяющую поверхность.
- Заметим, правда, что если данные действительно разделимы, то может получиться жуткий оверфиттинг: $\|\mathbf{w}\| \to \infty$, и сигмоид превращается в функцию Хевисайда. Надо регуляризовать.

IRLS

- В логистической регрессии не получается замкнутого решения из-за сигмоида.
- Но функция $E(\mathbf{w})$ всё равно выпуклая, и можно воспользоваться методом Ньютона-Рапсона – на каждом шаге использовать локальную квадратичную аппроксимацию к функции ошибки:

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

где $\mathbf{H}$ (Hessian) – матрица вторых производных $E(\mathbf{w})$.

- Замечание: давайте применим Ньютона-Рапсона к обычной линейной регрессии с квадратической ошибкой:

$$\nabla E(\mathbf{w}) = \sum_{n=1}^{N} \left(\mathbf{w}^\top \phi_n - t_n\right) \phi_n = \Phi^\top \Phi \mathbf{w} - \Phi^\top \mathbf{t},$$

$$\nabla\nabla E(\mathbf{w}) = \sum_{n=1}^{N} \phi_n \phi_n^\top = \Phi^\top \Phi,$$

и шаг оптимизации будет

$$\mathbf{w}^{\text{new}} = \mathbf{w}^{\text{old}} - \left(\Phi^\top \Phi\right)^{-1} \left[\Phi^\top \Phi \mathbf{w}^{\text{old}} - \Phi^\top \mathbf{t}\right] = \\ = \left(\Phi^\top \Phi\right)^{-1} \Phi^\top \mathbf{t},$$

т.е. мы за один шаг придём к решению.

IRLS

- Для логистической регрессии:

$$\nabla E(\mathbf{w}) = \sum_{n=1}^{N} (y_n - t_n)\, \phi_n = \Phi^\top (\mathbf{y} - \mathbf{t})\,,$$

$$\mathrm{H} = \nabla\nabla E(\mathbf{w}) = \sum_{n=1}^{N} y_n(1 - y_n)\phi_n\phi_n^\top = \Phi^\top \boldsymbol{R}\Phi$$

для диагональной матрицы $\boldsymbol{R}$ с $R_{nn} = y_n(1 - y_n)$.

IRLS

- Формула шага оптимизации:

$$\mathbf{w}^{\text{new}} = \mathbf{w}^{\text{old}} - \left(\Phi^\top R\Phi\right)^{-1}\Phi^\top(\mathbf{y}-\mathbf{t}) = \left(\Phi^\top R\Phi\right)^{-1}\Phi^\top R\mathbf{z},$$

  где $\mathbf{z} = \Phi\mathbf{w}^{\text{old}} - R^{-1}(\mathbf{y}-\mathbf{t})$.
- Получилось как бы решение взвешенной задачи минимизации квадратического отклонения с матрицей весов R.
- Отсюда название: iterative reweighted least squares (IRLS).

Несколько классов

- В случае нескольких классов

$$p(\mathcal{C}_k \mid \phi) = y_k(\phi) = \frac{e^{a_k}}{\sum_j e^{a_j}} \text{ для } a_k = \mathbf{w}_k^\top \phi.$$

- Опять выпишем максимальное правдоподобие; во-первых,

$$\frac{\partial y_k}{\partial a_j} = y_k \left([k = j] - y_j \right).$$

Несколько классов

- Теперь запишем правдоподобие – для схемы кодирования 1-of-K будет целевой вектор $\mathbf{t}_n$ и правдоподобие

$$p(\mathbf{T} \mid \mathbf{w}_1, \ldots, \mathbf{w}_K) = \prod_{n=1}^{N} \prod_{k=1}^{K} p(\mathcal{C}_k \mid \phi_n)^{t_{nk}} = \prod_{n=1}^{N} \prod_{k=1}^{K} y_{nk}^{t_{nk}}$$

для $y_{nk} = y_k(\phi_n)$; берём логарифм:

$$E(\mathbf{w}_1, \ldots, \mathbf{w}_K) = -\ln p(\mathbf{T} \mid \mathbf{w}_1, \ldots, \mathbf{w}_K) = -\sum_{n=1}^{N} \sum_{k=1}^{K} t_{nk} \ln y_{nk}, \text{ и}$$

$$\nabla_{\mathbf{w}_j} E(\mathbf{w}_1, \ldots, \mathbf{w}_K) = -\sum_{n=1}^{N} \left(y_{nj} - t_{nj}\right) \phi_n.$$

Несколько классов

- Оптимизировать опять можно по Ньютону-Рапсону; гессиан получится как

$$\nabla_{\mathbf{w}_k}\nabla_{\mathbf{w}_j}E(\mathbf{w}_1,\ldots,\mathbf{w}_K) = -\sum_{n=1}^{N} y_{nk}\left([k=j]-y_{nj}\right)\phi_n\phi_n^\top.$$

Пробит-регрессия

- А что если у нас другая форма сигмоида?
- Мы по-прежнему в той же постановке: два класса, $p(t = 1 \mid a) = f(a)$, $a = \mathbf{w}^\top \boldsymbol{\phi}$, f – функция активации.
- Давайте установим функцию активации с порогом θ: для каждого $\boldsymbol{\phi}_n$, вычисляем $a_n = \mathbf{w}^\top \boldsymbol{\phi}_n$, и

$$\begin{cases} t_n = 1, & \text{если } a_n G_{\mathrm{enh}} \theta, \\ t_n = 0, & \text{если } a_n < \theta. \end{cases}$$

Пробит-регрессия

- Если θ берётся по распределению $p(\theta)$, это соответствует

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

- Пусть, например, $p(\theta)$ – гауссиан с нулевым средним и единичной дисперсией. Тогда

$$f(a) = \Phi(a) = \int_{-\infty}^{a} \mathcal{N}\left(\theta \mid 0, 1\right) \mathrm{d}\theta.$$

Пробит-регрессия

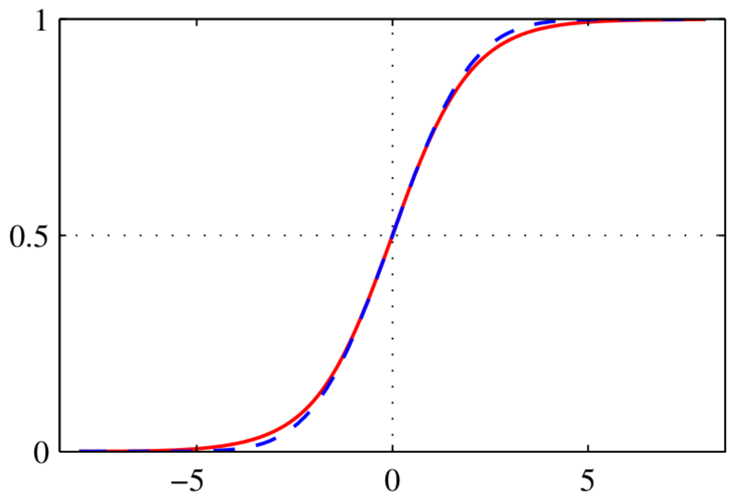
- Это называется *пробит-функцией* (probit); неэлементарная, но тесно связана с

$$\mathrm{erf}(a) = \frac{2}{\sqrt{\pi}} \int_0^a e^{-\frac{\theta^2}{2}} \mathrm{d}\theta :$$

$$\Phi(a) = \frac{1}{2} \left[1 + \frac{1}{\sqrt{2}} \mathrm{erf}(a) \right].$$

- Пробит-регрессия – это модель с пробит-функцией активации.

σ и Φ



Логистическая регрессия по-байесовски

Байесовская логистическая регрессия

- Теперь давайте обработаем логистическую регрессию по-байесовски.
- Логистическую регрессию так просто не выпишешь, как линейную – точного ответа из произведения логистических сигмоидов не получается.
- Будем приближать по Лапласу.

Байесовская логистическая регрессия

- Априорное распределение выберем гауссовским:

$$p(\mathbf{w}) = \mathcal{N}(\mathbf{w} \mid \boldsymbol{\mu}_0, \Sigma_0).$$

- Тогда апостериорное будет

$$\begin{aligned} p(\mathbf{w} \mid \mathbf{t}) \propto & p(\mathbf{w}) p(\mathbf{t} \mid \mathbf{w}), \text{ и} \\ \ln p(\mathbf{w} \mid \mathbf{t}) = & -\frac{1}{2} (\mathbf{w} - \boldsymbol{\mu}_0)^\top \Sigma_0^{-1} (\mathbf{w} - \boldsymbol{\mu}_0) \\ & + \sum_{n=1}^{N} [t_n \ln y_n + (1 - t_n) \ln(1 - y_n)] + \mathrm{const}, \\ \text{где } y_n = & \sigma(\mathbf{w}^\top \boldsymbol{\phi}_n). \end{aligned}$$

Байесовская логистическая регрессия

- Чтобы приблизить, сначала находим максимум $\mathbf{w}_{\mathrm{MAP}}$, а потом матрица коварииаций – это матрица вторых производных

$$\Sigma_N = -\nabla\nabla \ln p(\mathbf{w} \mid \mathbf{t}) = \Sigma_0^{-1} + \sum_{n=1}^{N} y_n(1 - y_n)\phi_n\phi_n^\top.$$

- Наше приближение – это

$$q(\mathbf{w}) = \mathcal{N}(\mathbf{w} \mid \mathbf{w}_{\mathrm{MAP}}, \Sigma_N).$$

Байесовская логистическая регрессия

- Теперь можно описать байесовское предсказание:

$$p(\mathcal{C}_1 \mid \boldsymbol{\phi}, \mathbf{t}) = \int p(\mathcal{C}_1 \mid \boldsymbol{\phi}, \mathbf{w}) p(\mathbf{w} \mid \mathbf{t}) d\mathbf{w} \approx \int \sigma(\mathbf{w}^\top \boldsymbol{\phi}) q(\mathbf{w}) \mathrm{d}\mathbf{w}.$$

- Заметим, что $\sigma(\mathbf{w}^\top \boldsymbol{\phi})$ зависит от $\mathbf{w}$ только через его проекцию на $\boldsymbol{\phi}$.
- Обозначим $a = \mathbf{w}^\top \boldsymbol{\phi}$:

$$\sigma(\mathbf{w}^\top \boldsymbol{\phi}) = \int \delta(a - \mathbf{w}^\top \boldsymbol{\phi}) \sigma(a) \mathrm{d}a.$$

Байесовская логистическая регрессия

- $\sigma(\mathbf{w}^\top \phi) = \int \delta(a - \mathbf{w}^\top \phi)\sigma(a)\mathrm{d}a$, а значит,

$$\int \sigma(\mathbf{w}^\top \phi) q(\mathbf{w}) d\mathbf{w} = \int \sigma(a) p(a) \mathrm{d}a,$$
$$\text{где } p(a) = \int \delta(a - \mathbf{w}^\top \phi) q(\mathbf{w}) \mathrm{d}\mathbf{w}.$$

- $p(a)$ – это маргинализация гауссиана $q(\mathbf{w})$, где мы интегрируем по всему, что ортогонально ϕ.

Байесовская логистическая регрессия

- $p(a)$ – это маргинализация гауссиана $q(\mathbf{w})$, где мы интегрируем по всему, что ортогонально $\boldsymbol{\phi}$.
- Значит, $p(a)$ – тоже гауссиан; найдём его моменты:

$$\mu_a = \mathrm{E}[a] = \int a p(a) \mathrm{d}a = \int q(\mathbf{w}) \mathbf{w}^\top \boldsymbol{\phi} \mathrm{d}\mathbf{w} = \mathbf{w}_{\mathrm{MAP}}^\top \boldsymbol{\phi},$$

$$\sigma_a^2 = \int \left(a^2 - \mathrm{E}[a]\right)^2 p(a) \mathrm{d}a =$$

$$= \int q(\mathbf{w}) \left[(\mathbf{w}^\top \boldsymbol{\phi})^2 - (\mu_N^\top \boldsymbol{\phi})^2\right]^2 \mathrm{d}\mathbf{w} = \boldsymbol{\phi}^\top \Sigma_N \boldsymbol{\phi}.$$

- Итого получили, что

$$p(\mathcal{C}_1 \mid \mathbf{t}) = \int \sigma(a) p(a) \mathrm{d}a = \int \sigma(a) \mathcal{N}(a \mid \mu_a, \sigma_a^2) \mathrm{d}a.$$

Байесовская логистическая регрессия

- $p(\mathcal{C}_1 \mid \mathbf{t}) = \int \sigma(a)\mathcal{N}(a \mid \mu_a, \sigma_a^2)\mathrm{d}a$.
- Этот интеграл так просто не взять, потому что сигмоид сложный, но можно приблизить, если приблизить $\sigma(a)$ через пробит: $\sigma(a) \approx \Phi(\lambda a)$ для $\lambda = \sqrt{\pi/8}$.

Упражнение. Докажите, что для $\lambda = \sqrt{\pi/8}$ у σ и Φ одинаковый наклон в нуле.

- А если мы перейдём к пробит-функции, то её свёртка с гауссианом будет просто другим пробитом:

$$\int \Phi(\lambda a)\mathcal{N}(a \mid \mu, \sigma^2)\mathrm{d}a = \Phi\left(\frac{\mu}{\sqrt{\frac{1}{\lambda^2} + \sigma^2}}\right).$$

Упражнение. Докажите это.

Байесовская логистическая регрессия

- В итоге получается аппроксимация

$$\int \sigma(a)\mathcal{N}(a \mid \mu, \sigma^2)\mathrm{d}a \approx \sigma\left(\kappa(\sigma^2)\mu\right),$$

$$\text{где } \kappa(\sigma^2) = \frac{1}{\sqrt{1+\frac{\pi}{8}\sigma^2}}.$$

- И теперь, собирая всё вместе, мы получили распределение предсказаний:

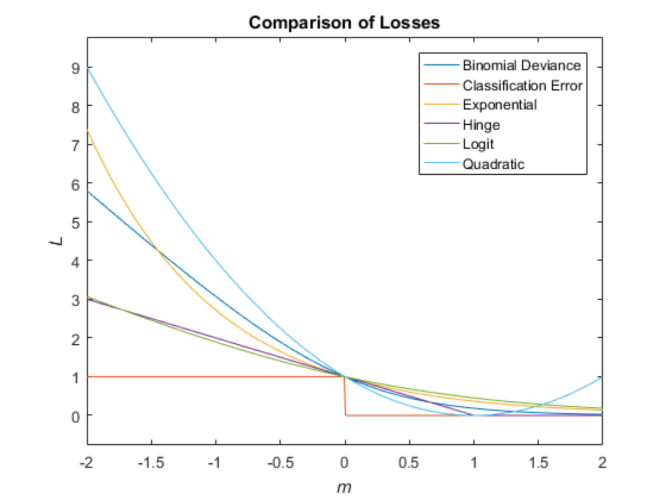
$$\begin{aligned} p(\mathcal{C}_1 \mid \boldsymbol{\phi}, \mathbf{t}) =& \sigma\left(\kappa(\sigma_a^2)\mu_a\right), \text{ где} \\ \mu_a =& \mathbf{w}_{\mathrm{MAP}}^\top \boldsymbol{\phi}, \\ \sigma_a^2 =& \boldsymbol{\phi}^\top \Sigma_N \boldsymbol{\phi}, \\ \kappa(\sigma^2) =& \frac{1}{\sqrt{1 + \frac{\pi}{8}\sigma^2}}. \end{aligned}$$

- Кстати, разделяющая поверхность $p(\mathcal{C}_1 \mid \boldsymbol{\phi}, \mathbf{t}) = \frac{1}{2}$ задаётся уравнением $\mu_a = 0$, и тут нет никакой разницы с просто использованием $\mathbf{w}_{\mathrm{MAP}}$. Разница будет только для более сложных критериев.

Функции потерь в классификации

- И напоследок немножко другой взгляд: разные методы классификации отличаются друг от друга тем, какую функцию ошибки они оптимизируют.
- У классификации проблема с «правильной» функцией ошибки, то есть ошибкой собственно классификации:
    - она и не везде дифференцируема,
    - и производная её никому не нужна.
- Давайте посмотрим на разные функции потерь (loss functions); мы уже несколько видели, но ещё немало осталось.

Функции потерь в классификации



Спасибо!

Спасибо за внимание!