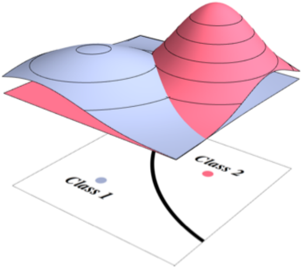
Генеративные модели

Виктор Китов

victorkitov.github.io

Курс поддержан фондом 'Интеллект'



Победитель конкурса VK среди курсов по IT

Содержание

Штрафы за неправильные классификации

- Предсказываем $y \in \{1, 2, ...C\}$
- λ_{yf} - штраф за прогноз класса y классом f.
- Примеры задач, где штрафы важны:
    - медицина: классификация болезни, методов лечения
    - финансы: детекция мошеннических сделок
    - почта: фильтрация спама
    - сети: обнаружение вторжений (intrusion detection)

Матрица штрафов

- Матрица штрафов

	$f=1$	$f=2$	$\cdots$	$f=C$
$y=1$	λ_{11}	λ_{12}	$\cdots$	λ_{1C}
$y=2$	λ_{21}	λ_{22}	$\cdots$	λ_{2C}
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
$y=C$	λ_{C1}	λ_{C2}	$\cdots$	λ_{CC}

- Ожидаемая цена прогноза $\widehat{y}(x)=f$:

$$\mathcal{L}(f)=\sum_{y}p(y|x)\lambda_{yf}$$

- Байесовское правило минимального риска
  - англ. Bayes minimum risk decision rule

$$\widehat{y}(x)=\arg\min_{f}\mathcal{L}(f)$$

Упрощение решающего правила

УПРОЩЕНИЕ 1: за *любые* ошибки на классе y платим λ_y.

$$\lambda_{yf} \equiv \lambda_y \mathbb{I}[y \neq f]$$

Упрощение решающего правила

УПРОЩЕНИЕ 1: за *любые* ошибки на классе y платим λ_y.

$$\lambda_{yf} \equiv \lambda_y \mathbb{I}[y \neq f]$$

Матрица штрафов:

	$f=1$	$f=2$	$\cdots$	$f=C$
$y=1$	0	λ_1	$\cdots$	λ_1
$y=2$	λ_2	0	$\cdots$	λ_2
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
$y=C$	λ_C	λ_C	$\cdots$	0

Упрощение решающего правила

- Ожидаемый штраф за прогноз f:

$$\mathcal{L}(f) = \sum_{y} p(y|x)\lambda_y \mathbb{I}[f \neq y] = \sum_{y} p(y|x)\lambda_y - p(f|x)\lambda_f$$

Упрощение решающего правила

- Ожидаемый штраф за прогноз f:

$$\mathcal{L}(f)=\sum_y p(y|x)\lambda_y \mathbb{I}[f\neq y]=\sum_y p(y|x)\lambda_y - p(f|x)\lambda_f$$

- Байесовское правило **минимального риска** становится:

$$\widehat{y}(x)=\arg\min_f \mathcal{L}(f)=\arg\min_f\left(\overbrace{\sum_y p(y|x)\lambda_y}^{const(f)} - p(f|x)\lambda_f\right)=$$
$$=\arg\min_f\left(-p(f|x)\lambda_f\right)=\arg\max_f \lambda_f p(f|x) \tag{1}$$

Упрощение решающего правила

- Ожидаемый штраф за прогноз f:

$$\mathcal{L}(f)=\sum_{y}p(y|x)\lambda_{y}\mathbb{I}[f\neq y]=\sum_{y}p(y|x)\lambda_{y}-p(f|x)\lambda_{f}$$

- Байесовское правило **минимального риска** становится:

$$\widehat{y}(x)=\arg\min_{f}\mathcal{L}(f)=\arg\min_{f}\left(\overbrace{\sum_{y}p(y|x)\lambda_{y}}^{const(f)}-p(f|x)\lambda_{f}\right)=$$
$$=\arg\min_{f}\left(-p(f|x)\lambda_{f}\right)=\arg\max_{f}\lambda_{f}p(f|x) \quad (1)$$

- Важна не только вероятность класса, но и штраф при пропуске класса!

Упрощение решающего правила

- **УПРОЩЕНИЕ 2:** одинаковый штраф при любых ошибках $\lambda_y \equiv \lambda \, \forall y$.
- Байесовское правило минимального риска становится

$$\widehat{y}(x) = \arg\max_f p(f|x) \tag{2}$$

- Это Байесовское правило **минимальной ошибки**.
    - т.к. прогноз максимально вероятным классом минимизирует ожидаемое число ошибок.

Упрощение решающего правила

- **УПРОЩЕНИЕ 2:** одинаковый штраф при любых ошибках $\lambda_y \equiv \lambda \, \forall y$.
- Байесовское правило минимального риска становится

$$\widehat{y}(x) = \arg\max_f p(f|x) \tag{2}$$

- Это Байесовское правило **минимальной ошибки**.
  - т.к. прогноз максимально вероятным классом минимизирует ожидаемое число ошибок.
- **УПРОЩЕНИЕ 3:** Если x и y независимы, то $p(f|x) = p(f)$ и (2) становится

$$\widehat{y}(x) = \arg\max_f p(f|x) = \arg\max_f p(f)$$

Генеративные и дискриминативные модели

$$\widehat{y}(\mathbf{x}) = \arg\max_y p(y|\mathbf{x}) = \arg\max_y \frac{p(\mathbf{x}, y)}{p(\mathbf{x})} = \arg\max_y p(y)p(\mathbf{x}|y)$$

Генеративные и дискриминативные модели

$$\widehat{y}(\mathbf{x}) = \arg\max_y p(y|\mathbf{x}) = \arg\max_y \frac{p(\mathbf{x}, y)}{p(\mathbf{x})} = \arg\max_y p(y)p(\mathbf{x}|y)$$

Можно строить прогноз по

- $p(y|\mathbf{x})$: дискриминативная модель
  - моделируем только то, что нужно; простота оценивания
- $p(y)p(\mathbf{x}|y) = p(\mathbf{x}, y)$: генеративная модель
  - $p(y)$ легко оценить, $p(\mathbf{x}|y)$ - сложно

Генеративные и дискриминативные модели

$$\widehat{y}(\mathbf{x}) \quad = \quad \arg\max_{y} p(y|\mathbf{x}) = \arg\max_{y} \frac{p(\mathbf{x}, y)}{p(\mathbf{x})} = \arg\max_{y} p(y)p(\mathbf{x}|y)$$

Можно строить прогноз по

- $p(y|\mathbf{x})$: дискриминативная модель
  - моделируем только то, что нужно; простота оценивания
- $p(y)p(\mathbf{x}|y) = p(\mathbf{x}, y)$: генеративная модель
  - $p(y)$ легко оценить, $p(\mathbf{x}|y)$ - сложно
  - возможное упрощение: **предположение наивного Байеса**

$$p(\mathbf{x}|y) = p(x^1|y)p(x^2|y)...p(x^D|y)$$

  - можно подстраивать модель под изменяемые $p(y)$
  - если x^i пропущено, то оценивается

$$p\left(y\right)p\left(\mathbf{x}\backslash\left\{x^i\right\}|y\right) = p(y)\int_{x^i} p(\mathbf{x}|y)dx^i$$

  - легко фильтровать выбросы - малое $p(x)$

Содержание

Гауссов классификатор

- Гауссов классификатор - генеративная модель с $x|y \sim \mathcal{N}(\mu_y, \sigma_y^2)$:

$$p(x|y) = \frac{1}{(2\pi)^{D/2}|\Sigma_y|^{1/2}} \exp\left\{-\frac{1}{2}(x-\mu_y)^T \Sigma_y^{-1}(x-\mu_y)\right\}$$

Гауссов классификатор

- Гауссов классификатор - генеративная модель с $x|y \sim \mathcal{N}(\mu_y, \sigma_y^2)$:

$$p(x|y) = \frac{1}{(2\pi)^{D/2}|\Sigma_y|^{1/2}} \exp\left\{-\frac{1}{2}(x-\mu_y)^T\Sigma_y^{-1}(x-\mu_y)\right\}$$

- Дискриминатная функция

$$\begin{aligned}
\log p(y|x) &= \log p(x|y) + \log p(y) - \log p(x) \\
&= -\frac{1}{2}(x-\mu_y^T)\Sigma_y^{-1}(x-\mu_y) - \frac{1}{2}\log|\Sigma_y| \\
&\quad -\frac{D}{2}\log(2\pi) + \log p(y) - \log p(x)
\end{aligned}$$

Гауссов классификатор

- Гауссов классификатор - генеративная модель с $x|y \sim \mathcal{N}(\mu_y, \sigma_y^2)$:

$$p(x|y) = \frac{1}{(2\pi)^{D/2}|\Sigma_y|^{1/2}} \exp\left\{-\frac{1}{2}(x-\mu_y)^T\Sigma_y^{-1}(x-\mu_y)\right\}$$

- Дискриминатная функция

$$\begin{aligned} \log p(y|x) &= \log p(x|y) + \log p(y) - \log p(x) \\ &= -\frac{1}{2}(x-\mu_y^T)\Sigma_y^{-1}(x-\mu_y) - \frac{1}{2}\log|\Sigma_y| \\ &\quad -\frac{D}{2}\log(2\pi) + \log p(y) - \log p(x) \end{aligned}$$

- Уберем общие для всех дискр. ф-ций константы:

$$g_y(x) = \log p(y) - \frac{1}{2}\log|\Sigma_y| - \frac{1}{2}(x-\mu_y)^T\Sigma_y^{-1}(x-\mu_y) \quad (3)$$

Практическое применение

- Заменяем $p(y), \mu_y, \Sigma_y$ их оценками макс. правдоподобия:

$$\widehat{p}(y) = \frac{N_y}{N}, \quad \widehat{\mu}_y = \frac{1}{N_y} \sum_{n: y_n = y} x_n$$

$$\widehat{\Sigma}_y = \frac{1}{N_y} \sum_{n: y_n = y} (x_n - \widehat{\mu}_y)(x_n - \widehat{\mu}_y)^T$$

- Модель опирается на предположение $x|y \sim \mathcal{N}(\mu_y, \sigma_y^2)$, в частности, унимодальность распределения. # параметров?

Практическое применение

- Заменяем $p(y), \mu_y, \Sigma_y$ их оценками макс. правдоподобия:

$$\widehat{p}(y) = \frac{N_y}{N}, \quad \widehat{\mu}_y = \frac{1}{N_y} \sum_{n: y_n = y} x_n$$

$$\widehat{\Sigma}_y = \frac{1}{N_y} \sum_{n: y_n = y} (x_n - \widehat{\mu}_y)(x_n - \widehat{\mu}_y)^T$$

- Модель опирается на предположение $x|y \sim \mathcal{N}(\mu_y, \sigma_y^2)$, в частности, унимодальность распределения. # параметров?
- $p(y)$: 1 параметр, μ_y : D параметров
- Σ_y : $\frac{D(D+1)}{2}$ параметров
- Всего параметров для $y = 1, 2, \ldots C$:

$$C\left(1 + D + \frac{D(D+1)}{2}\right)$$

Упрощение модели

- Гауссов классификатор квадратично зависит от #признаков.
- Упрощающие предположения:
    - $\Sigma_1, \Sigma_2, ...\Sigma_C$ - диагональные (naive Bayes)
    - уменьшить #признаков (отбор признаков / снижение размерности)
    - $\Sigma_1 = \Sigma_2 = ... = \Sigma_C = \Sigma$
    - $\Sigma_1 = \alpha_1\Sigma,\ \Sigma_2 = \alpha_2\Sigma,\ ...\ \Sigma_C = \alpha_C\Sigma.$

Регуляризация модели

- Если число наблюдений класса y мало, а D велико, то Σ_y может получиться вырожденной.
- Регуляризация для обратимости и плавного контроля сложности:

$$\Sigma_y' = \Sigma_y + \lambda I$$

$$\Sigma_y' = \Sigma_y + \lambda \operatorname{diag}\{\Sigma_y\}$$

$$\lambda > 0$$

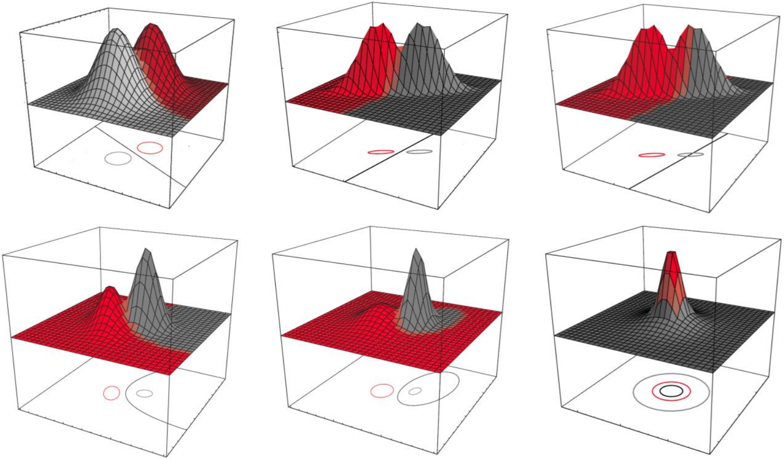
QDA vs. LDA

Метод Гауссова классификатора называется:

- **квадратичным дискриминантным анализом**, когда $\Sigma_1, \Sigma_2, ...\Sigma_C$ - произвольные.
  - границы между классами квадратичные[1]
- **линейным дискриминантным анализом**, когда $\Sigma_1 = \Sigma_2 = ... = \Sigma_C$ общие
  - др. название - линейный дискриминант Фишера
  - границы между классами линейные[2]

[1] Докажите.
[2] Докажите.

Линейный и квадратичный дискриминант



LDA (вверху) и QDA (внизу): $p(x|y)$, границы.

Содержание

Токены в текстах

Требуется представить текст вектором $\in \mathbb{R}^D$

Будем учитывать встречаемость D токенов $w_1, w_2, ...w_D$

- в простейшем случае: все уникальные слова языка
    - можно в разных формах или нормализованной
        - единственное число, именительный падеж, начальная форма глагола.
- убрать слишком частые слова и слишком редкие
- убрать неинформативные "стоп-слова" из словаря
    - а, но, если, конечно, зато, или, ...

Токены в текстах

- можно ограничить словами предметной области
- можно добавить биграммы/триграммы:
    - мне фильм не понравился -> 'мне фильм', 'фильм не', 'не понравился'.
    - мне фильм не понравился -> 'мне фильм не', 'фильм не понравился'.
- либо можно добавить только коллокации (неслучайно часто совстречающиеся слова)
    - линейная регрессия показала точность... -> 'линейная регрессия'

$$\frac{p(w_1 w_2)}{p(w_1)p(w_2)} > threshold$$

Генеративные модели классификации текстов

- Генеративные модели классификации текстов:
    - Модель Бернулли
        - $x^i = \mathbb{I}[w_i$ встретилось в документе$]$
    - Мультиномиальная
        - $x^i =$[сколько раз w_i встретилось в документе]
- Могут применяться и в др. областях:
    - изображения (кодирование bag-of-visual-words)
    - ДНК - цепочка нуклеотидов
    - покупки в магазине
    - использованные услуги в тарифе

Модель Бернулли[3]

- $w_1, w_2, ...w_D$ - токены
- $x \in \mathbb{R}^D$, $x^d = \mathbb{I}[w_d$ встретилось в документе$]$, $d = \overline{1, D}$
- $N = \#[$документов$]$, $N^y := \#[$документов класса $y]$
- $N_d^y = \#[$документов класса y, содержащих d-й токен$]$
- $\theta_d^y = p(x^d = 1|y)$
- Частотные оценки (макс. правдоподобия):

$$p(y) \approx \frac{N^y}{N}, \qquad \theta_d^y \approx \frac{N_d^y}{N^y}$$

[3] Является ли она линейным классификатором?

Модель Бернулли[6]

- Решающее правило (минимальной ошибки):

$$\widehat{y}(x) = \arg\max_{y} p(y)p(x|y)$$

- Генерация документа класса y: для каждого токена w_d генерируется его присутствие в документе $\sim Bernoulli(\theta_y^d)$.
    - не зависит от встречаемости др. токенов (naive Bayes)
- $p(x|y) = \prod_{d=1}^{D} \left(\theta_d^y\right)^{x^d} \left(1 - \theta_d^y\right)^{1-x^d}$
- Сглаживание Лапласа[4,5]: $\theta_y^d = \frac{N_d^y + \alpha}{N^y + 2\alpha}$

[4] Проинтерпретируйте добавлением новых наблюдений в выборку.
[5] Как сглаживать, чтобы приближать к априорному распределению слов?
[6] Оцените сложность обучения модели Бернулли.

Мультиномиальная модель

- $w_1, w_2, ...w_D$ - токены
- Решающее правило (минимальной ошибки):

$$\widehat{y}(x) = \arg\max_y p(y)p(x|y)$$

- $x \in \mathbb{R}^D$, $x^i =$[сколько раз w_i встретилось в документе].
- Генерация документа класса y: для каждой словопозиции $i = 1, 2, ...n_{document}$ сгенерировать токен $z_i \sim Categorical\left(\theta_1^y, \theta_2^y, ...\theta_D^y\right)$.
- $\theta_i^y =$[вероятность w_i на словопозиции]
    - не зависит от встречаемости др. токенов (naive Bayes)

Мультиномиальная модель[7]

$$p(x|y)=\frac{\left(\sum_i x^i\right)!}{\prod_i (x^i)!}\prod_{i=1}^{D}(\theta_i^y)^{x^i}$$ мультиномиальное распределение

Интерпретация мультиномиального коэффициента:

- $\left(\sum_i x^i\right)!=n!$ - число перестановок различимых токенов
- перестановки токена 1 неразличимы => делим на $\left(x^1\right)!$
- перестановки токена 2 неразличимы => делим на $\left(x^2\right)!$
- ... в итоге: $\frac{\left(\sum_i x^i\right)!}{\prod_i (x^i)!}$ - #способов расставить $w_1,...w_D$ в количествах $x^1,...x^D$ по $n=\sum_i x^i$ ячейкам.

[7] Является ли она линейным классификатором?

Вывод мультиномиального коэффициента

#число способов, как можно расставить D слов по n позициям в количествах $x_1, ...x_D$:

$$
\begin{aligned}
&C_{x_1}^{n} C_{x_2}^{n-x_1} ... C_{x_{D-1}}^{n-x_1-...-x_{D-2}} C_{x_D}^{n-x_1-...x_{D-1}} \\
&= \frac{n!}{x_1!(n-x_1)!} \times \frac{(n-x_1)!}{x_2!(n-x_1-x_2)!} \times \\
... \times &\frac{(n-x_1-x_{D-2})!}{x_{D-1}!(n-x_1-...x_{D-1})!} \frac{(n-x_1-...-x_{D-1})!}{x_D!(n-x_1-...-x_D)!} \\
&= \frac{\left(\sum_i x^i\right)!}{\prod_i (x^i)!}
\end{aligned}
$$

Оценки параметров[10]

- Частотные оценки (макс. правдоподобия):
  - $\widehat{p}(y) = \frac{N^y}{N}$, где
    - N=#[документов]
    - $N^y = $ #[документов $\in y$],
  - $\theta_i^y \approx n_i^y / n^y$, где
    - n_i^y=#[токен w_i встречался в документах$\in y$],
    - n^y=#[токенов в документе$\in y$],
- Сглаживание Лапласа[8,9]:

$$\theta_y^d = \frac{n_{yd} + \alpha}{n_y + \alpha D}$$

[8] Проинтерпретируйте добавлением новых наблюдений в выборку.
[9] Как сглаживать, чтобы приближать к априорному распределению слов?
[10] Оцените сложность обучения мультиномиальной модели.

Заключение

- Байесовское правило
  - минимального риска: минимизирует ожидаемые потери
  - минимальной ошибки: минимизирует #ошибок
    - оптимально в случае одинакового штрафа для любых ошибок
- Генеративные модели моделируют $p(y)p(x|y) = p(x, y)$.
- Предположение наивного Байеса:

$$p(x|y) = p(x^1|y)p(x^2|y)...p(x^D|y)$$

- Дискриминативные модели моделируют только $p(y|x)$.
  - предпочтительны в большинстве случаев
- Модели Бернулли и мультиномиальная имеют линейную относительно длин текстов сложность оценивания.
  - естественный бейзлайн для классификации текстов.