RVM и EM-алгоритм

Сергей Николенко

Академия MADE — Mail.Ru

6 мая 2020 г.

Random facts:

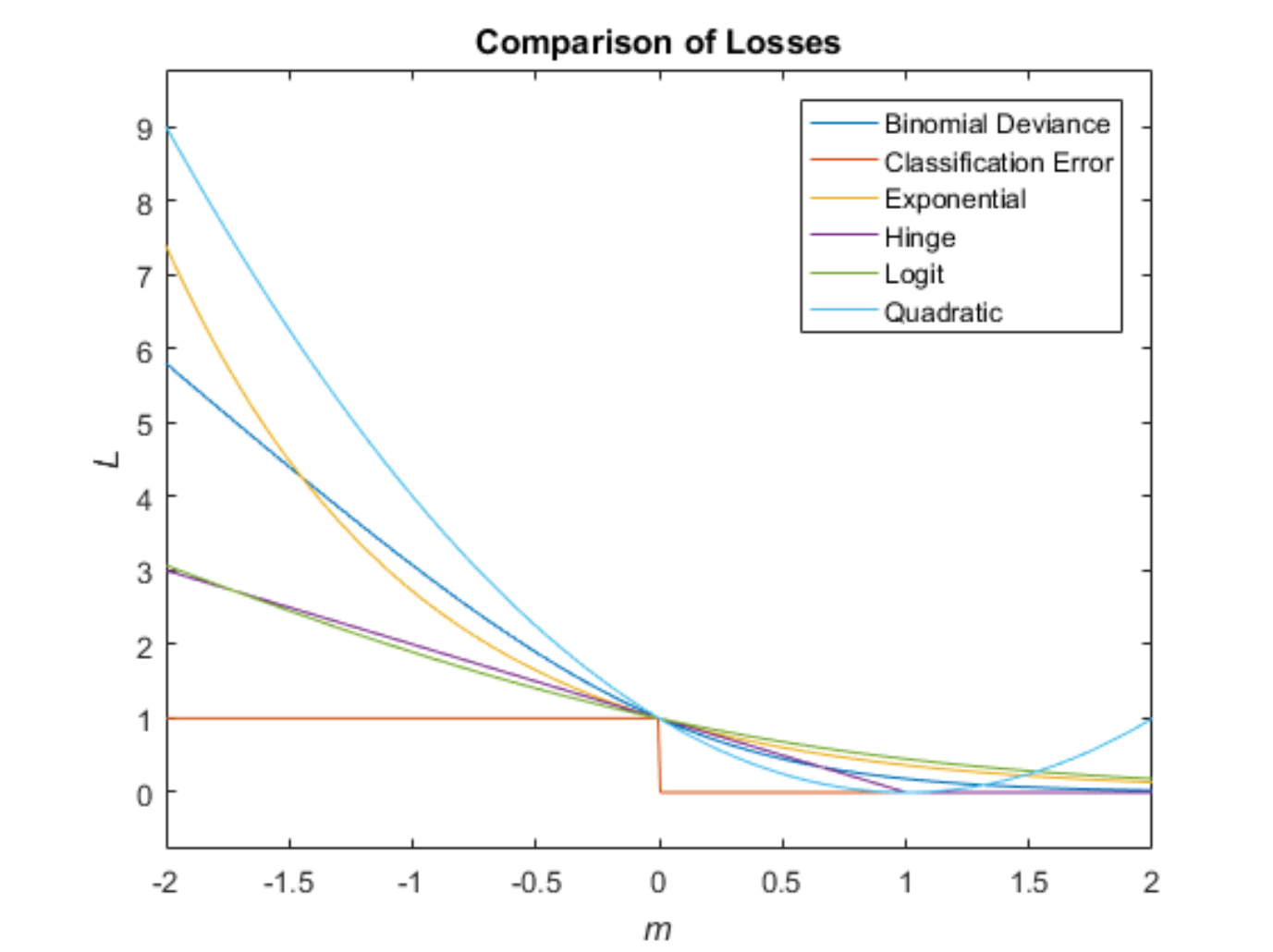
- 6 мая 1626 г. голландский колонист Петер Минёйт (Peter Minuit) за горсть бижутерии общей стоимостью 25 долларов приобрёл у индейцев остров Манхэттен
- 6 мая 1840 г. был выпущен в обращение «Чёрный пенни»
- 6 мая 1890 г. Церковь Иисуса Христа Святых последних дней официально отказалась от многожёнства, и в том же году Юта получила статус штата
- 6 мая 1949 г. в Кембридже свою первую программу запустил EDSAC (Electronic delay storage automatic calculator), первый настоящий практический компьютер на архитектуре фон Неймана; он вычислил списки квадратов и простых чисел
- 6 мая 2012 г. в Москве прошёл «Марш миллионов» для выражения протеста против инаугурации Владимира Путина

RVM

Функции потерь в классификации

- Сначала – ещё один важный 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, ещё кое-что осталось.

Функции потерь в классификации



Постановка задачи

- SVM – отличный метод. Но и у него есть недостатки.
  1. Выходы SVM – решения, а апостериорные вероятности непонятно как получить.
  2. SVM – для двух классов, обобщить на несколько проблематично.
  3. Есть параметр C (или ν, или ещё вдобавок ϵ), который надо подбирать.
  4. Предсказания – линейные комбинации ядер, которым необходимо быть положительно определёнными и которые центрированы на точках из датасета.
- Сейчас мы рассмотрим байесовский аналог SVM – *relevance vector machines* (RVM).

RVM для регрессии

- RVM удобнее сразу формулировать для регрессии.
- Вспомним обычную нашу линейную модель:

$$p(t \mid \mathbf{x}, \mathbf{w}, \beta) = \mathcal{N}(t \mid y(\mathbf{x}), \beta^{-1}), \text{ где}$$

$$y(\mathbf{x}) = \sum_{i=1}^{M} w_i \phi_i(\mathbf{x}) = \mathbf{w}^\top \boldsymbol{\phi}(\mathbf{x}).$$

RVM для регрессии

- RVM – это вариант такой модели, который старается работать как SVM.
- Рассмотрим

$$y(\mathbf{x}) = \sum_{n=1}^{N} w_n k(\mathbf{x}, \mathbf{x}_n) + b.$$

- Т.е. мы сразу ищем решение в форме линейной комбинации значений ядра (вспомним «эквивалентное ядро» для линейной регрессии), но, в отличие от SVM, теперь ядро никак не ограничивается.

RVM для регрессии

- Для N наблюдений вектора $\mathbf{x}$ (обозначим через $\mathbf{X}$) со значениями $\mathbf{t}$ получим правдоподобие

$$p(\mathbf{t} \mid \mathbf{X}, \mathbf{w}, \beta) = \prod_{n=1}^{N} p(t_n \mid \mathbf{x}_n, \mathbf{w}, \beta^{-1}).$$

- Априорное распределение тоже будет нормальное, но вместо единого гиперпараметра для всех весов мы введём отдельный гиперпараметр для каждого:

$$p(\mathbf{w} \mid \boldsymbol{\alpha}) = \prod_{i=1}^{M} \mathcal{N}(w_i \mid 0, \alpha_i^{-1}).$$

RVM для регрессии

- Отдельные гиперпараметры:

$$p(\mathbf{w} \mid \boldsymbol{\alpha}) = \prod_{i=1}^{M} \mathcal{N}(w_i \mid 0, \alpha_i^{-1}).$$

- Идея здесь в том, что при максимизации апостериорной вероятности большая часть α_i просто уйдёт на бесконечность, и соответствующие веса будут нулевыми.
- Сейчас увидим, как это получается.

RVM для регрессии

- Апостериорное распределение нам знакомо:

$$p(\mathsf{w} \mid \mathsf{t}, \mathsf{X}, \boldsymbol{\alpha}, \beta) = \mathcal{N}(\mathsf{w} \mid \mathsf{m}, \boldsymbol{\Sigma}), \text{ где}$$

$$\mathsf{m} = \beta \boldsymbol{\Sigma} \boldsymbol{\Phi}^\top \mathsf{t},$$
$$\boldsymbol{\Sigma} = \left(\mathsf{A} + \beta \boldsymbol{\Phi}^\top \boldsymbol{\Phi}\right)^{-1},$$

где $\mathsf{A} = \mathrm{diag}(\alpha_1, \ldots, \alpha_M)$, а $\boldsymbol{\Phi}$ в нашем случае – это K, симметрическая матрица с элементами $k(\mathsf{x}_n, \mathsf{x}_m)$.

RVM для регрессии

- Как найти $\boldsymbol{\alpha}$ и β? Нужно максимизировать маргинальное правдоподобие датасета

$$p(\mathbf{t} \mid \mathbf{X}, \boldsymbol{\alpha}, \beta) = \int p(\mathbf{t} \mid \mathbf{X}, \mathbf{w}, \beta) p(\mathbf{w} \mid \boldsymbol{\alpha}) d\mathbf{w}.$$

- Это свёртка двух гауссианов, тоже гауссиан:

$$\ln p(\mathbf{t} \mid \mathbf{X}, \boldsymbol{\alpha}, \beta) = \ln \mathcal{N}(\mathbf{t} \mid \mathbf{0}, \mathbf{C}) =$$
$$= -\frac{1}{2}\left[N \ln(2\pi) + \ln|\mathbf{C}| + \mathbf{t}^\top \mathbf{C}^{-1}\mathbf{t}\right], \text{ где } \mathbf{C} = \beta^{-1}\mathbf{I} + \boldsymbol{\Phi}\mathbf{A}^{-1}\boldsymbol{\Phi}^\top.$$

- Как это оптимизировать?

RVM для регрессии

- Можно подсчитать производные и получить

$$\alpha_i = \frac{\gamma_i}{m_i^2},$$
$$\beta^{-1} = \frac{\|\mathbf{t} - \mathbf{\Phi m}\|^2}{N - \sum_i \gamma_i},$$

  где $\gamma_i = 1 - \alpha_i \Sigma_{ii}$.

- Теперь можно просто итеративно пересчитывать $\boldsymbol{\alpha}, \beta$ из $\mathbf{m}, \mathbf{\Sigma}$, потом наоборот, потом опять наоборот, и до сходимости.

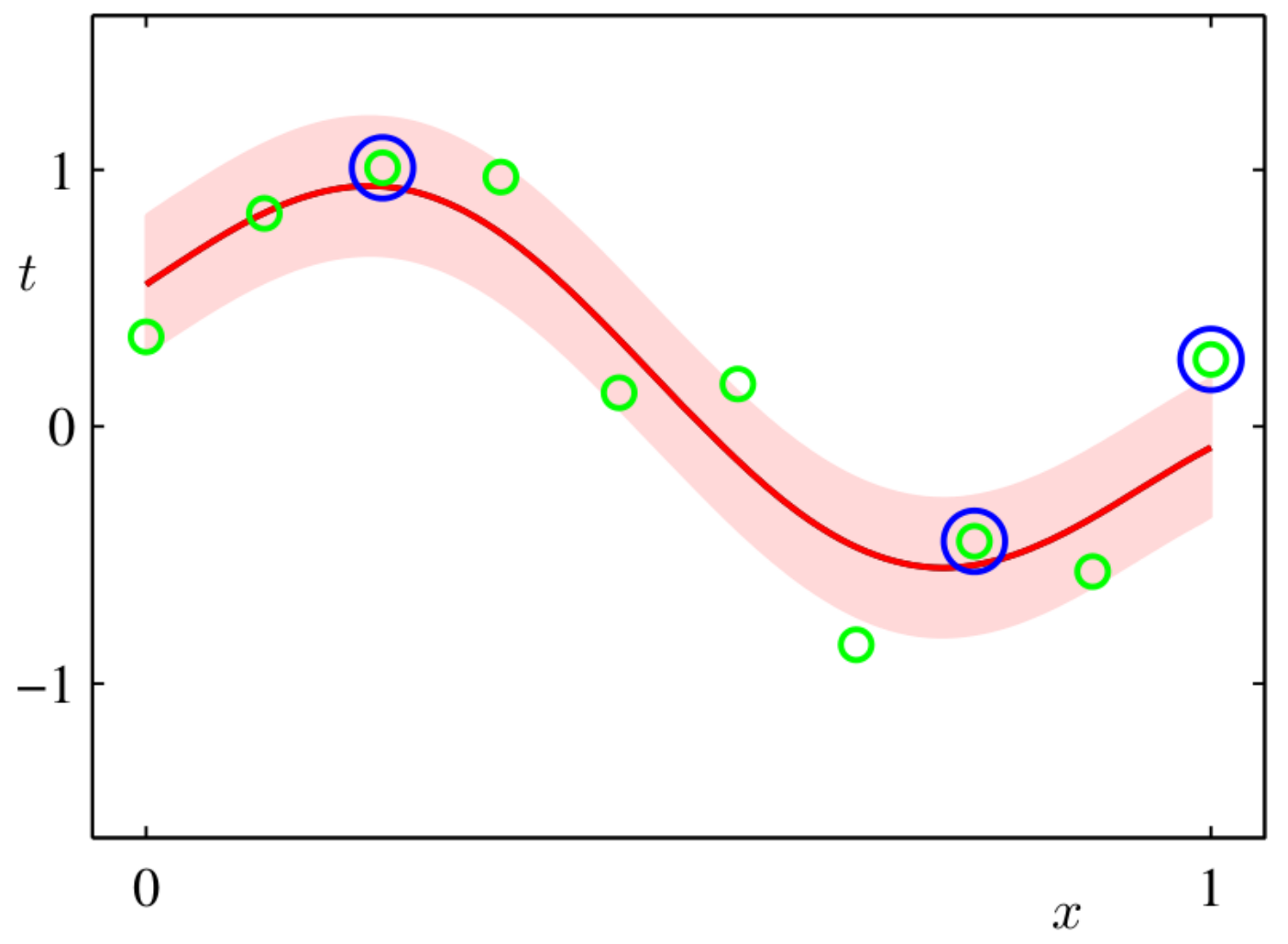
RVM для регрессии

- В результате получается обычно, что большинство α_i неограниченно растут, и соответствующие веса можно считать нулевыми.
- Оставшиеся называются *relevance vectors*, их обычно мало.
- Если теперь мы найдём $\boldsymbol{\alpha}^*, \beta^*$, то предсказывать в новых точках можно как

$$p(t \mid \mathbf{x}, \mathbf{X}, \mathbf{t}, \boldsymbol{\alpha}^*, \beta^*) = \int p(t \mid \mathbf{x}, \mathbf{w}, \beta^*) p(\mathbf{w} \mid \mathbf{X}, \mathbf{t}, \boldsymbol{\alpha}^*, \beta^*) d\mathbf{w} = \\ = \mathcal{N}(t \mid \mathbf{m}^\top \boldsymbol{\phi}(\mathbf{x}), \sigma^2(\mathbf{x})),$$

где $\sigma^2(\mathbf{x}) = (\beta^*)^{-1} + \boldsymbol{\phi}(\mathbf{x})^\top \boldsymbol{\Sigma} \boldsymbol{\phi}(\mathbf{x})$.

RVM для регрессии



RVM для классификации

- Можно сделать то же самое и для классификации. Рассмотрим классификацию с двумя классами, $t \in \{0, 1\}$:

$$y(\mathbf{x}, \mathbf{w}) = \sigma(\mathbf{w}^\top \boldsymbol{\phi}(\mathbf{x})).$$

- И добавим сюда, опять же, априорное распределение с разными α_i для каждого веса:

$$p(\mathbf{w} \mid \boldsymbol{\alpha}) = \prod_{i=1}^{M} \mathcal{N}(w_i \mid 0, \alpha_i^{-1}).$$

- Идея: инициализируем $\boldsymbol{\alpha}$, считаем лапласовское приближение к апостериорному распределению, максимизируем, получаем новое $\boldsymbol{\alpha}$, и т.д.

RVM для классификации

- Апостериорное распределение:

$$\ln p(\mathbf{w} \mid \mathbf{t}, \boldsymbol{\alpha}) = \ln \left(p(\mathbf{t} \mid \mathbf{w}) p(\mathbf{w} \mid \boldsymbol{\alpha})\right) - \ln p(\mathbf{t} \mid \boldsymbol{\alpha}) =$$
$$= \sum_{n=1}^{N} \left[t_n \ln y_n + (1 - t_n) \ln(1 - y_n)\right] - \frac{1}{2} \mathbf{w}^\top \mathbf{A} \mathbf{w} + \text{const}.$$

- Мы уже обсуждали, как его максимизировать – IRLS; для этого подсчитаем

$$\nabla \ln p(\mathbf{w} \mid \mathbf{t}, \boldsymbol{\alpha}) = \boldsymbol{\Phi}^\top (\mathbf{t} - \mathbf{y}) - \mathbf{A} \mathbf{w},$$
$$\nabla \nabla \ln p(\mathbf{w} \mid \mathbf{t}, \boldsymbol{\alpha}) = -\left(\boldsymbol{\Phi}^\top \mathbf{B} \boldsymbol{\Phi} + \mathbf{A}\right),$$

где $\mathbf{B}$ – диагональная матрица с элементами $b_n = y_n(1 - y_n)$.

RVM для классификации

- Лапласовское приближение получится из $\nabla \ln p(\mathbf{w} \mid \mathbf{t}, \boldsymbol{\alpha})$, и получится

$$\mathbf{w}^* = \mathbf{A}^{-1}\boldsymbol{\Phi}^\top (\mathbf{t} - \mathbf{y}),$$
$$\boldsymbol{\Sigma} = \left(\boldsymbol{\Phi}^\top \mathbf{B}\boldsymbol{\Phi} + \mathbf{A}\right)^{-1},$$

и распределение для предсказания получится

$$p(\mathbf{t} \mid \boldsymbol{\alpha}) = \int p(\mathbf{t} \mid \mathbf{w}) p(\mathbf{w} \mid \boldsymbol{\alpha}) d\mathbf{w} \approx$$
$$\approx p(\mathbf{t} \mid \mathbf{w}^*) p(\mathbf{w}^* \mid \boldsymbol{\alpha}) (2\pi)^{M/2} |\boldsymbol{\Sigma}|^{1/2}.$$

RVM для классификации

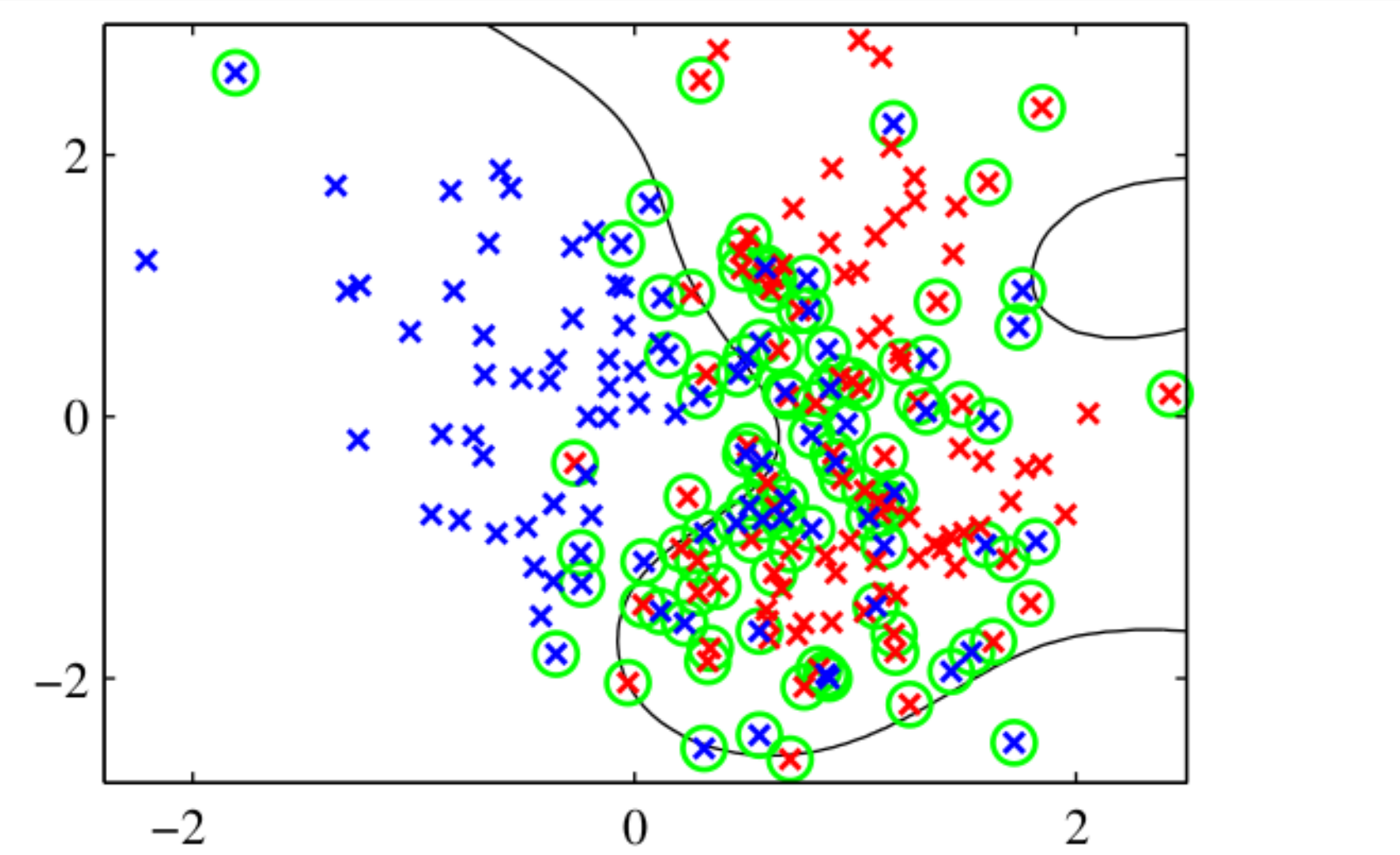
- $p(\mathbf{t} \mid \boldsymbol{\alpha}) = \int p(\mathbf{t} \mid \mathbf{w}) p(\mathbf{w} \mid \boldsymbol{\alpha}) d\mathbf{w} \approx$
  $p(\mathbf{t} \mid \mathbf{w}^*) p(\mathbf{w}^* \mid \boldsymbol{\alpha}) (2\pi)^{M/2} |\boldsymbol{\Sigma}|^{1/2}$.
- Теперь мы оптимизируем это по $\boldsymbol{\alpha}$: берём производную, получаем

$$-\frac{1}{2}(w_i^*)^2 + \frac{1}{2\alpha_i} - \frac{1}{2}\Sigma_{ii} = 0, \text{ т.е.}$$

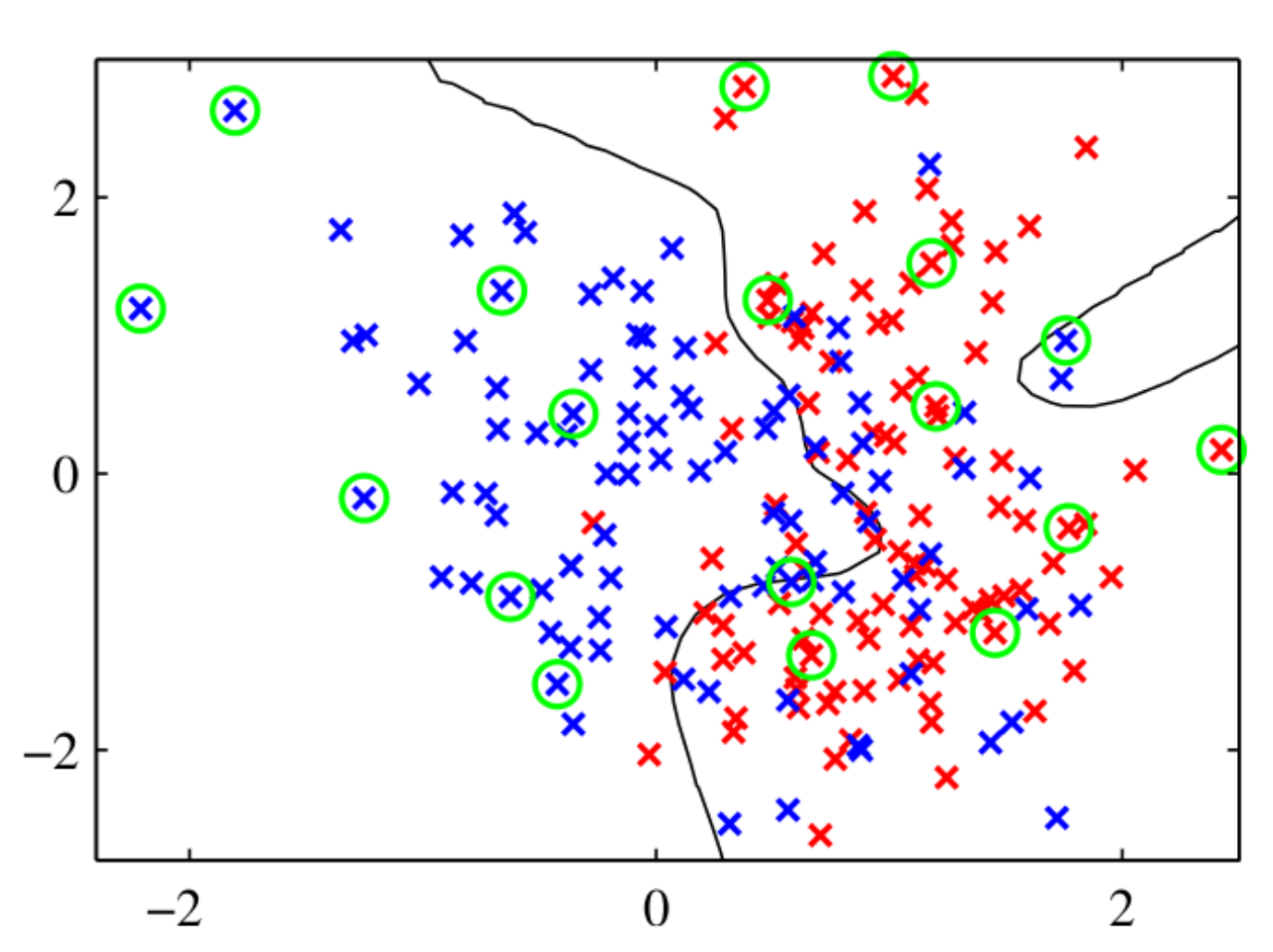
$$\alpha_i = \frac{\gamma_i}{(w_i^*)^2}, \ \gamma_i = 1 - \alpha_i \Sigma_{ii}.$$

- Т.е. формула получилась точно такая же, как в случае регрессии.

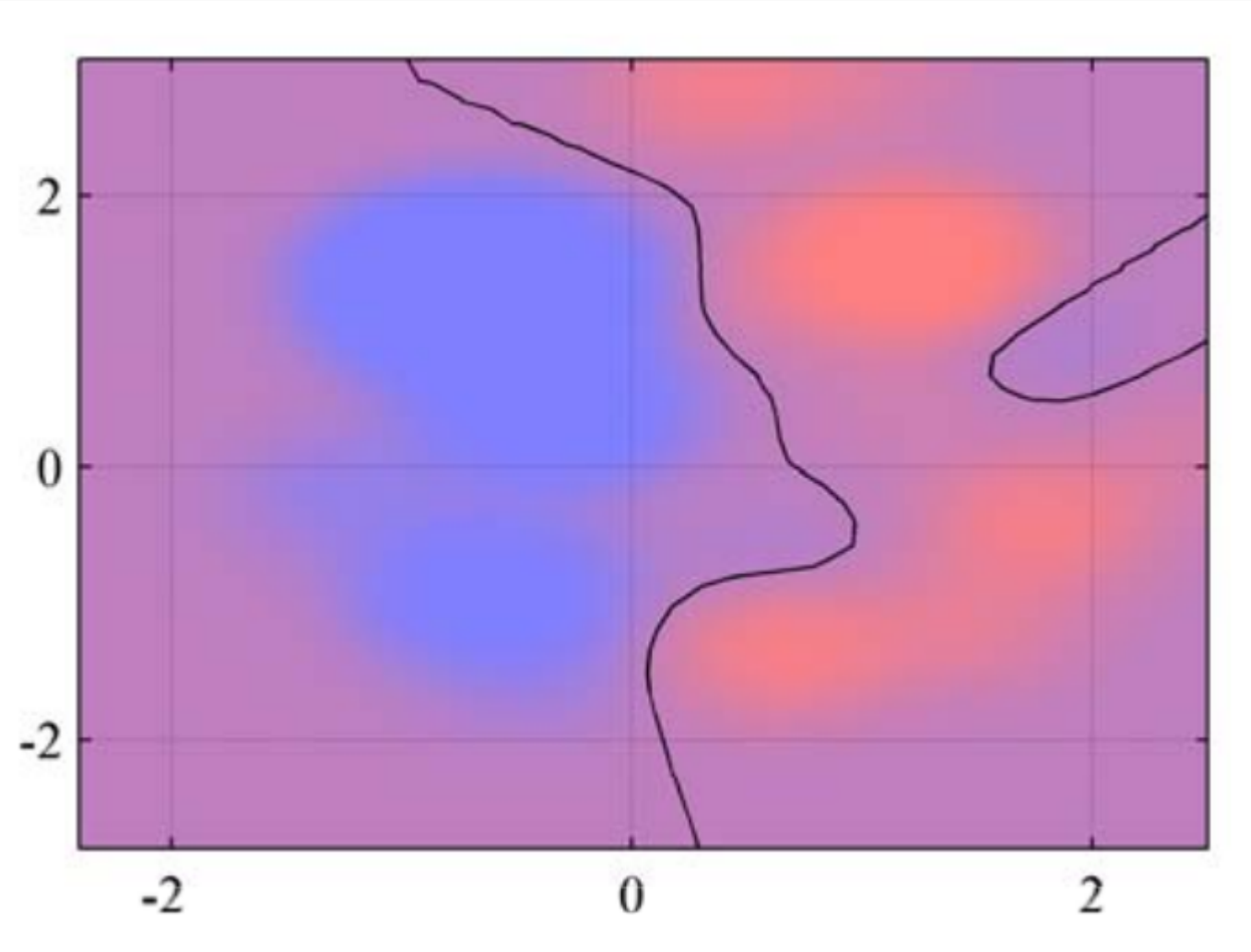
Было: SVM



Стало: RVM



Стало: RVM



RVM для нескольких классов

- На несколько классов теперь обобщается естественным образом:

$$a_k = \mathbf{w}_k^\top \mathbf{x}, \quad y_k(\mathbf{x}) = \frac{e^{a_k}}{\sum_j e^{a_j}}.$$

- И дальше всё то же самое.

Сравнение SVM и RVM

- RVM как-то получилось лучше со всех сторон.
- Главный минус – в RVM обучение гораздо дольше (хотя есть алгоритмы и побыстрее, чем мы рассматривали, но всё равно дольше).
- Но даже это не то чтобы минус, потому что в SVM нужна кросс-валидация для подбора параметров, т.е. на самом деле обучение SVM дольше, чем кажется.
- Есть и (даже более важный) плюс с точки зрения скорости – в RVM гораздо быстрее применение модели к новым точкам, потому что опорных векторов гораздо меньше.

Быстрый алгоритм обучения RVM

- В RVM для регрессии получается правдоподобие

$$\ln p(\mathbf{t} \mid \mathbf{X}, \boldsymbol{\alpha}, \beta) = \ln \mathcal{N}(\mathbf{t} \mid \mathbf{0}, \mathbf{C}) =$$
$$= -\frac{1}{2}\left[N \ln(2\pi) + \ln|\mathbf{C}| + \mathbf{t}^\top \mathbf{C}^{-1}\mathbf{t}\right], \text{ где } \mathbf{C} = \beta^{-1}\mathbf{I} + \boldsymbol{\Phi}\mathbf{A}^{-1}\boldsymbol{\Phi}^\top.$$

Быстрый алгоритм обучения RVM

- Выделим вклад в C одного компонента α_i:

$$\mathsf{C} = \beta^{-1}\boldsymbol{I} + \sum_{j \neq i} \alpha_j^{-1}\boldsymbol{\varphi}_j\boldsymbol{\varphi}_j^\top + \alpha_i^{-1}\boldsymbol{\varphi}_i\boldsymbol{\varphi}_i^\top = \\ = \mathsf{C}_{-i} + \alpha_i^{-1}\boldsymbol{\varphi}_i\boldsymbol{\varphi}_i^\top,$$

где $\boldsymbol{\varphi}_i$ – i-я строка $\boldsymbol{\Phi}$ ($\boldsymbol{\phi}_n$ был n-м столбцом).

Быстрый алгоритм обучения RVM

- $\mathsf{C} = \mathsf{C}_{-i} + \alpha_i^{-1}\boldsymbol{\varphi}_i\boldsymbol{\varphi}_i^\top$.
- Верны следующие тождества для определителя и обратной матрицы:

$$|\mathsf{C}| = |\mathsf{C}_{-i}||1 + \alpha_i^{-1}\boldsymbol{\varphi}_i^\top \mathsf{C}_{-i}^{-1}\boldsymbol{\varphi}_i|,$$

$$\mathsf{C}^{-1} = \mathsf{C}_{-i}^{-1} - \frac{\mathsf{C}_{-i}^{-1}\boldsymbol{\varphi}_i\boldsymbol{\varphi}_i^\top \mathsf{C}_{-i}^{-1}}{\alpha_i + \boldsymbol{\varphi}_i^\top \mathsf{C}_{-i}^{-1}\boldsymbol{\varphi}_i}.$$

Упражнение. Докажите это.

Быстрый алгоритм обучения RVM

- Значит, $L(\boldsymbol{\alpha})$ можно переписать в виде

$$L(\boldsymbol{\alpha}) = L(\boldsymbol{\alpha}_{-i}) + \lambda(\alpha_i), \text{ где}$$

$$\lambda(\alpha_i) = \frac{1}{2}\left[\ln \alpha_i - \ln(\alpha_i + s_i) + \frac{q_i^2}{\alpha_i + s_i}\right].$$

- Здесь

$$\begin{aligned} s_i &= \boldsymbol{\varphi}_i^\top \mathbf{C}_{-i}^{-1} \boldsymbol{\varphi}_i && \text{sparsity } \boldsymbol{\varphi}_i \\ q_i &= \boldsymbol{\varphi}_i^\top \mathbf{C}_{-i}^{-1} \mathbf{t} && \text{quality } \boldsymbol{\varphi}_i. \end{aligned}$$

Быстрый алгоритм обучения RVM

- $s_i = \boldsymbol{\varphi}_i^\top \mathbf{C}_{-i}^{-1} \boldsymbol{\varphi}_i$, $q_i = \boldsymbol{\varphi}_i^\top \mathbf{C}_{-i}^{-1} \mathbf{t}$.
- Sparsity – то, насколько $\boldsymbol{\varphi}_i$ перекрывается с остальными векторами модели.
- Quality – то, насколько $\boldsymbol{\varphi}_i$ сонаправлен с ошибкой между $\mathbf{t}$ и $\mathbf{y}_{-i}$ (ошибкой модели без $\boldsymbol{\varphi}_i$).
- Чем больше sparsity и чем меньше quality, тем более вероятно, что этот базисный вектор из модели исключат (т.е. $\alpha_i \to \infty$).

Быстрый алгоритм обучения RVM

- $\lambda(\alpha_i) = \frac{1}{2}\left[\ln \alpha_i - \ln(\alpha_i + s_i) + \frac{q_i^2}{\alpha_i + s_i}\right]$.
- Возьмём производную, приравняем нулю, получим (т.к. $\alpha_i \geq 0$)

$$\alpha_i = \begin{cases} \infty, & q_i^2 \leq s_i, \\ \frac{s_i^2}{q_i^2 - s_i}, & q_i^2 > s_i. \end{cases}$$

- Как мы и ожидали.

Быстрый алгоритм обучения RVM

- И алгоритм теперь получается такой:
  1. инициализировать β, $\boldsymbol{\varphi}_1$, $\alpha_1 = s_1^2/(q_1^2 - s_1)$, остальные $\alpha_j = \infty$;
  2. вычислить $\boldsymbol{\Sigma}$, $\mathbf{m}$, q_i и s_i для всех i;
  3. выбрать i, проапдейтить α_i, проапдейтить β;
  4. goto 2 и так пока не сойдётся.

Алгоритм EM

Постановка задачи

- Часто возникает ситуация, когда в имеющихся данных некоторые переменные присутствуют, а некоторые — отсутствуют.
- Даны результаты сэмплирования распределения вероятностей с несколькими параметрами, из которых известны не все.

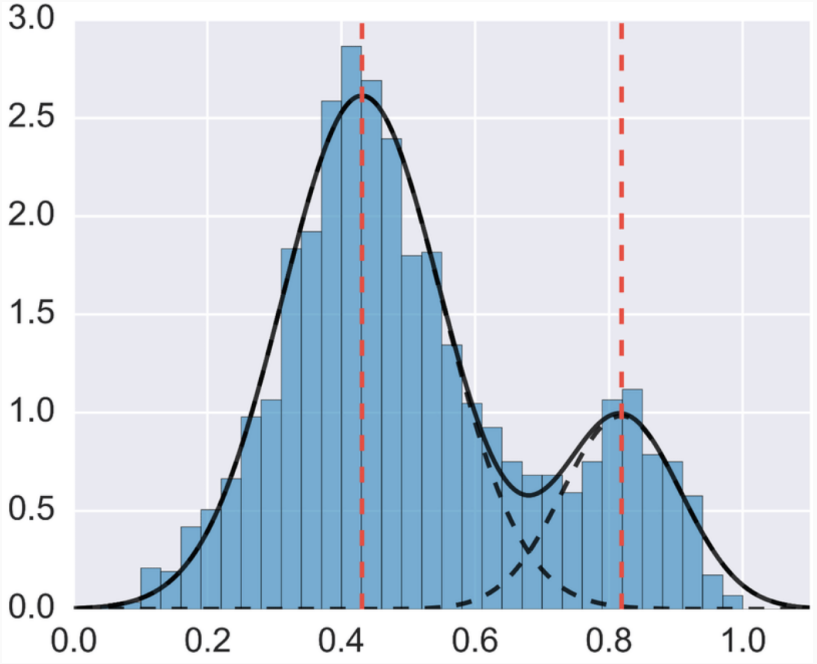
Постановка задачи

- Эти неизвестные параметры тоже расцениваются как случайные величины.
- Задача — найти наиболее вероятную гипотезу, то есть ту гипотезу h, которая максимизирует

$$E[\ln p(D|h)].$$

Частный случай

- Построим один из простейших примеров применения алгоритма EM. Пусть случайная переменная y сэмплируется из суммы двух нормальных распределений. Дисперсии даны (одинаковые), нужно найти только средние μ_1, μ_2.



- Какое тут правдоподобие? Как его оптимизировать?

Два распределения

- Нельзя понять, какие y_i были порождены каким распределением — классический пример *скрытых переменных*.
- Один тестовый пример полностью описывается как тройка $\langle y_i, z_{i1}, z_{i2} \rangle$, где $z_{ij} = 1$ iff y_i был сгенерирован j-м распределением.

Суть алгоритма EM

- Сгенерировать какую-нибудь гипотезу $h = (\mu_1, \mu_2)$.
- Пока не дойдем до локального максимума:
  - Вычислить ожидание $E(z_{ij})$ в предположении текущей гипотезы (E–шаг).
  - Вычислить новую гипотезу $h' = (\mu_1', \mu_2')$, предполагая, что z_{ij} принимают значения $E(z_{ij})$ (M–шаг).

В примере с гауссианами

- В примере с гауссианами:

$$E(z_{ij}) = \frac{p(y = y_i|\mu = \mu_j)}{p(y = y_i|\mu = \mu_1) + p(y = y_i|\mu = \mu_2)} = \\ = \frac{e^{-\frac{1}{2\sigma^2}(y_i - \mu_j)^2}}{e^{-\frac{1}{2\sigma^2}(y_i - \mu_1)^2} + e^{-\frac{1}{2\sigma^2}(y_i - \mu_2)^2}}.$$

- Мы подсчитываем эти ожидания, а потом подправляем гипотезу:

$$\mu_j \leftarrow \frac{1}{m}\sum_{i=1}^{m} E(z_{ij})y_i.$$

- Звучит логично, но с какой стати это всё работает?

Обоснование алгоритма EM

- Дадим формальное обоснование алгоритма EM.
- Мы решаем задачу максимизации правдоподобия по данным $\mathsf{y} = \{y_1, \ldots, y_N\}$.

$$L(\boldsymbol{\theta} \mid \mathcal{Y}) = p(\mathcal{Y} \mid \boldsymbol{\theta}) = \prod p(y_i \mid \boldsymbol{\theta})$$

  или, что то же самое, максимизации $\ell(\boldsymbol{\theta} \mid \mathcal{Y}) = \log L(\boldsymbol{\theta} \mid \mathcal{Y})$.
- EM может помочь, если этот максимум трудно найти аналитически.

Обоснование алгоритма EM

- Давайте предположим, что в данных есть *скрытые компоненты*, такие, что если бы мы их знали, задача была бы проще.
- Замечание: совершенно не обязательно эти компоненты должны иметь какой-то физический смысл. :) Может быть, так просто удобнее.
- В любом случае, получается набор данных $\mathcal{X} = (\mathcal{Y}, \mathcal{Z})$ с совместной плотностью

$$p(\mathsf{x} \mid \boldsymbol{\theta}) = p(\mathsf{y}, \mathsf{z} \mid \boldsymbol{\theta}) = p(\mathsf{z} \mid \mathsf{y}, \boldsymbol{\theta}) p(\mathsf{y} \mid \boldsymbol{\theta}).$$

- Получается полное правдоподобие $L(\boldsymbol{\theta} \mid \mathcal{X}) = p(\mathcal{Y}, \mathcal{Z} \mid \boldsymbol{\theta})$. Это случайная величина (т.к. $\mathcal{Z}$ неизвестно).

Обоснование алгоритма EM

- Заметим, что настоящее правдоподобие $L(\boldsymbol{\theta}) = E_Z\left[p(\mathcal{Y}, \mathcal{Z} \mid \boldsymbol{\theta}) \mid \mathcal{Y}, \boldsymbol{\theta}\right]$.
- E-шаг алгоритма EM вычисляет условное ожидание (логарифма) полного правдоподобия при условии $\mathcal{Y}$ и текущих оценок параметров $\boldsymbol{\theta}_n$:

$$Q(\boldsymbol{\theta}, \boldsymbol{\theta}_n) = E\left[\log p(\mathcal{Y}, \mathcal{Z} \mid \boldsymbol{\theta}) \mid \mathcal{Y}, \boldsymbol{\theta}_n\right].$$

- Здесь $\boldsymbol{\theta}_n$ — текущие оценки, а $\boldsymbol{\theta}$ — неизвестные значения (которые мы хотим получить в конечном счёте); т.е. $Q(\boldsymbol{\theta}, \boldsymbol{\theta}_n)$ — это функция от $\boldsymbol{\theta}$.

Обоснование алгоритма EM

- E-шаг алгоритма EM вычисляет условное ожидание (логарифма) полного правдоподобия при условии $\mathcal{Y}$ и текущих оценок параметров $\boldsymbol{\theta}$:

$$Q(\boldsymbol{\theta}, \boldsymbol{\theta}_n) = E\left[\log p(\mathcal{Y}, \mathcal{Z} \mid \boldsymbol{\theta}) \mid \mathcal{Y}, \boldsymbol{\theta}_n\right].$$

- Условное ожидание — это

$$E\left[\log p(\mathcal{Y}, \mathcal{Z} \mid \boldsymbol{\theta}) \mid \mathcal{Y}, \boldsymbol{\theta}_n\right] = \int_{\mathsf{z}} \log p(\mathcal{Y}, \mathsf{z} \mid \boldsymbol{\theta}) p(\mathsf{z} \mid \mathcal{Y}, \boldsymbol{\theta}_n) \mathrm{d}\mathsf{z},$$

  где $p(\mathsf{z} \mid \mathcal{Y}, \boldsymbol{\theta}_n)$ — маргинальное распределение скрытых компонентов данных.
- EM лучше всего применять, когда это выражение легко подсчитать, может быть, даже аналитически.
- Вместо $p(\mathsf{z} \mid \mathcal{Y}, \boldsymbol{\theta}_n)$ можно подставить $p(\mathsf{z}, \mathcal{Y} \mid \boldsymbol{\theta}_n) = p(\mathsf{z} \mid \mathcal{Y}, \boldsymbol{\theta}_n) p(\mathcal{Y} \mid \boldsymbol{\theta}_n)$, от этого ничего не изменится.

Обоснование алгоритма EM

- В итоге после E-шага алгоритма EM мы получаем функцию $Q(\boldsymbol{\theta}, \boldsymbol{\theta}_n)$.
- На M-шаге мы максимизируем

$$\boldsymbol{\theta}_{n+1} = \arg\max_{\boldsymbol{\theta}} Q(\boldsymbol{\theta}, \boldsymbol{\theta}_n).$$

- Затем повторяем процедуру до сходимости.
- В принципе, достаточно просто находить $\boldsymbol{\theta}_{n+1}$, для которого $Q(\boldsymbol{\theta}_{n+1}, \boldsymbol{\theta}_n) > Q(\boldsymbol{\theta}_n, \boldsymbol{\theta}_n)$ — Generalized EM.
- Осталось понять, что значит $Q(\boldsymbol{\theta}, \boldsymbol{\theta}_n)$ и почему всё это работает.

Обоснование алгоритма EM

- Мы хотели перейти от $\boldsymbol{\theta}_n$ к $\boldsymbol{\theta}$, для которого $\ell(\boldsymbol{\theta}) > \ell(\boldsymbol{\theta}_n)$.

$$
\begin{aligned}
&\ell(\boldsymbol{\theta}) - \ell(\boldsymbol{\theta}_n) = \\
&\qquad = \log\left(\int_{\mathsf{z}} p(\mathcal{Y} \mid \mathsf{z}, \boldsymbol{\theta}) p(\mathsf{z} \mid \boldsymbol{\theta}) \mathrm{d}\mathsf{z}\right) - \log p(\mathcal{Y} \mid \boldsymbol{\theta}_n) = \\
&= \log\left(\int_{\mathsf{z}} p(\mathsf{z} \mid \mathcal{Y}, \boldsymbol{\theta}_n) \frac{p(\mathcal{Y} \mid \mathsf{z}, \boldsymbol{\theta}) p(\mathsf{z} \mid \boldsymbol{\theta})}{p(\mathsf{z} \mid \mathcal{Y}, \boldsymbol{\theta}_n)} \mathrm{d}\mathsf{z}\right) - \log p(\mathcal{Y} \mid \boldsymbol{\theta}_n) \geq \\
&\geq \int_{\mathsf{z}} p(\mathsf{z} \mid \mathcal{Y}, \boldsymbol{\theta}_n) \log\left(\frac{p(\mathcal{Y} \mid \mathsf{z}, \boldsymbol{\theta}) p(\mathsf{z} \mid \boldsymbol{\theta})}{p(\mathsf{z} \mid \mathcal{Y}, \boldsymbol{\theta}_n)}\right) \mathrm{d}\mathsf{z} - \log p(\mathcal{Y} \mid \boldsymbol{\theta}_n) = \\
&\qquad = \int_{\mathsf{z}} p(\mathsf{z} \mid \mathcal{Y}, \boldsymbol{\theta}_n) \log\left(\frac{p(\mathcal{Y} \mid \mathsf{z}, \boldsymbol{\theta}) p(\mathsf{z} \mid \boldsymbol{\theta})}{p(\mathcal{Y} \mid \boldsymbol{\theta}_n) p(\mathsf{z} \mid \mathcal{Y}, \boldsymbol{\theta}_n)}\right) \mathrm{d}\mathsf{z}.
\end{aligned}
$$

Обоснование алгоритма EM

- Получили

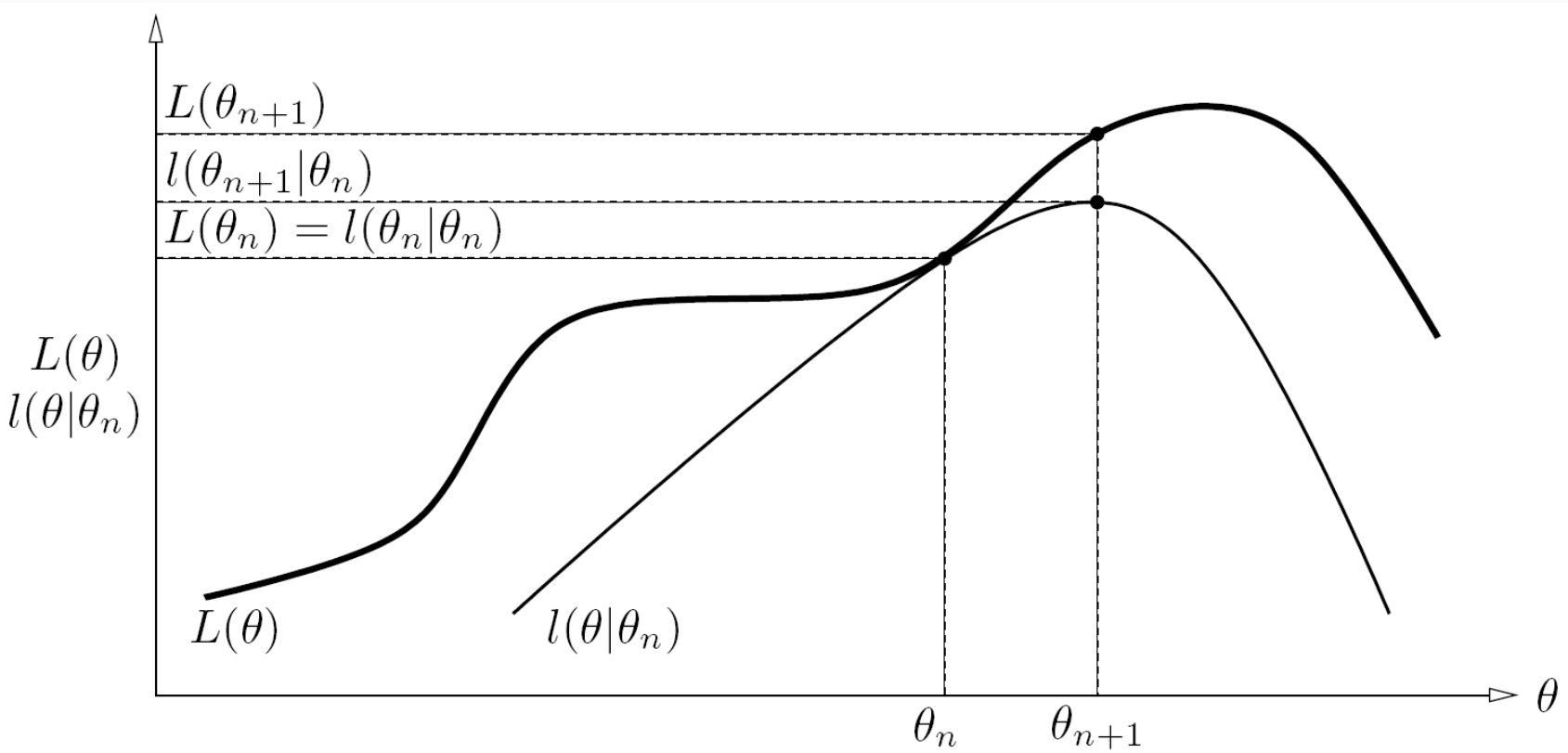
$$\ell(\boldsymbol{\theta}) \geq \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\theta}_n) =$$
$$= \ell(\boldsymbol{\theta}_n) + \int_{\mathsf{z}} p(\mathsf{z} \mid \mathcal{Y}, \boldsymbol{\theta}_n) \log \left(\frac{p(\mathcal{Y} \mid \mathsf{z}, \boldsymbol{\theta}) p(\mathsf{z} \mid \boldsymbol{\theta})}{p(\mathcal{Y} \mid \boldsymbol{\theta}_n) p(\mathsf{z} \mid \mathcal{Y}, \boldsymbol{\theta}_n)} \right) \mathrm{d}\mathsf{z}.$$

Упражнение. Докажите, что $\mathcal{L}(\boldsymbol{\theta}_n, \boldsymbol{\theta}_n) = \ell(\boldsymbol{\theta}_n)$.

Обоснование алгоритма EM

- Иначе говоря, мы нашли нижнюю оценку на $\ell(\boldsymbol{\theta})$ везде, касание происходит в точке $\boldsymbol{\theta}_n$.
- Т.е. мы нашли нижнюю оценку для правдоподобия и смещаемся в точку, где она максимальна (или хотя бы больше текущей).
- Такая общая схема называется *MM-алгоритм* (minorization-maximization). Мы к ним ещё вернёмся.

Обоснование алгоритма EM



Обоснование алгоритма EM

- Осталось только понять, что максимизировать можно Q.

$$\boldsymbol{\theta}_{n+1} = \arg\max_{\boldsymbol{\theta}} l(\boldsymbol{\theta}, \boldsymbol{\theta}_n) = \arg\max_{\boldsymbol{\theta}} \{\ell(\boldsymbol{\theta}_n) + \\ + \int_{\mathsf{z}} f(y \mid \mathcal{X}, \boldsymbol{\theta}_n) \log \left(\frac{p(\mathcal{X} \mid y, \boldsymbol{\theta}) f(y \mid \boldsymbol{\theta})}{p(\mathcal{X} \mid \boldsymbol{\theta}_n) f(y \mid \mathcal{X}, \boldsymbol{\theta}_n)} \right) \mathrm{d}\mathsf{z} \Big\} = \\ = \arg\max_{\boldsymbol{\theta}} \left\{ \int_{\mathsf{z}} p(\mathsf{z} \mid \mathcal{X}, \boldsymbol{\theta}_n) \log \left(p(\mathcal{X} \mid y, \boldsymbol{\theta}) p(\mathsf{z} \mid \boldsymbol{\theta}) \right) \mathrm{d}\mathsf{z} \right\} = \\ = \arg\max_{\boldsymbol{\theta}} \left\{ \int_{\mathsf{z}} p(\mathsf{z} \mid \mathcal{X}, \boldsymbol{\theta}_n) \log p(\mathcal{X}, y \mid \boldsymbol{\theta}) \mathrm{d}\mathsf{z} \right\} = \\ = \arg\max_{\boldsymbol{\theta}} \{Q(\boldsymbol{\theta}, \boldsymbol{\theta}_n)\},$$

а остальное от $\boldsymbol{\theta}$ не зависит. Вот и получился EM.

Спасибо!

Спасибо за внимание!