

ОБЪЕДИНЕНИЕ МОДЕЛЕЙ: УСРЕДНЕНИЕ И БУСТИНГ

Сергей Николенко

НИУ ВШЭ – Санкт-Петербург
28 апреля 2017 г.

Random facts:

- 28 апреля 1599 г. английский парламент утвердил общую молельную книгу для англиканской церкви
- а 28 апреля 1686 г. Ньютон представил Королевскому обществу первый том *Philosophiæ Naturalis Principia Mathematica*

ОБЪЕДИНЕНИЕ МОДЕЛЕЙ

КАК ОБЪЕДИНЯТЬ МОДЕЛИ

- До сих пор мы разрабатывали (и потом ещё будем разрабатывать) модели, которые делают предсказания (в основном для задач регрессии и классификации).
- Таким образом, мы можем попробовать обучить сразу много разных моделей!
- Model selection – это о том, как выбрать из них лучшую.
- Но, может быть, можно не выбирать, а использовать все сразу?

ОСНОВНЫЕ ПОДХОДЫ

- Комитет: обучаем L разных моделей, а потом так или иначе усредняем-комбинируем их результаты.
- Альтернатива: обучаем L разных моделей, а потом обучаем отдельную модель о том, какую из них использовать для предсказания (например, дерево принятия решений).

КОМБИНАЦИЯ МОДЕЛЕЙ И БАЙЕСОВСКОЕ УСРЕДНЕНИЕ

- Начнём с самого простого – байесовского усреднения.
- Мы уже знаем, что такое комбинация моделей – например, линейная смесь гауссианов:

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

- Если её обучать, мы обучим коэффициенты смеси, и результат будет порождён как бы двухуровневым процессом.

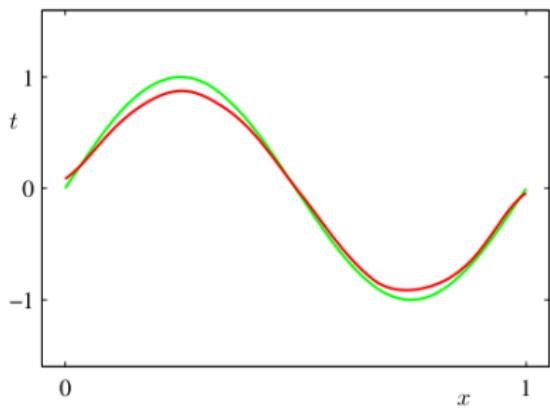
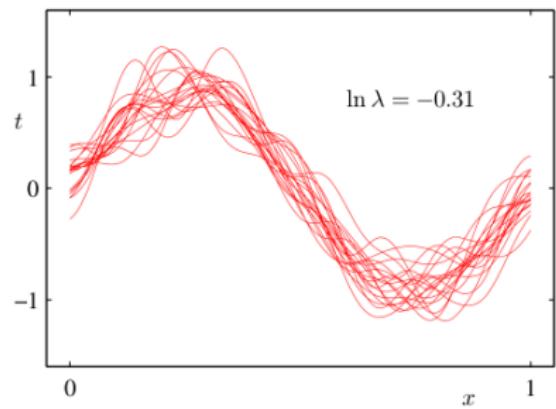
КОМБИНАЦИЯ МОДЕЛЕЙ И БАЙЕСОВСКОЕ УСРЕДНЕНИЕ

- А байесовское усреднение будет выглядеть как

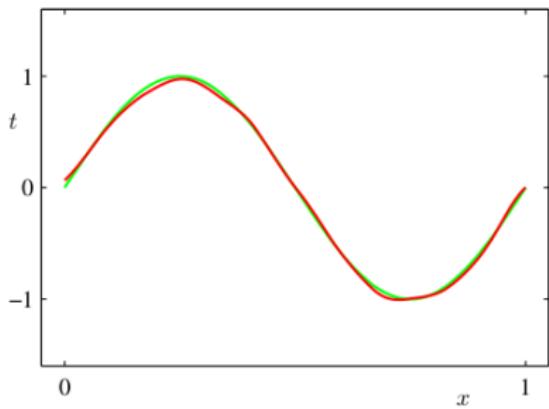
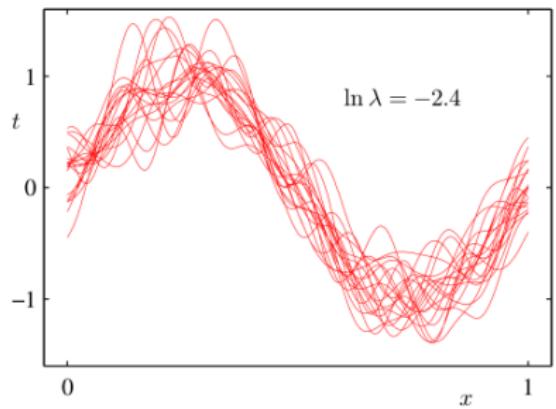
$$p(\mathbf{X}) = \sum_{h=1}^H p(\mathbf{X} \mid h)p(h).$$

- Смысл теперь в том, что генерирует \mathbf{X} только одна модель, но мы просто не знаем какая именно; когда \mathbf{X} , апостериорные распределения $p(h \mid \mathbf{X})$ сужаются, и мы выбираем то, что надо.
- Но метод очень простой: взять много моделей и усреднить.
- Где-то мы это уже видели...

ГДЕ-ТО МЫ ЭТО УЖЕ ВИДЕЛИ



ГДЕ-ТО МЫ ЭТО УЖЕ ВИДЕЛИ



- На этих картинках – модели с высоким *bias*, которые обучены по разным датасетам, сгенерированным одним и тем же распределением.
- И если их усреднить, получится как раз то, что надо.
- Но в жизни у нас нет возможности генерировать много датасетов: сколько данных есть, столько есть.
- Просто разбивать датасет на части – не поможет. Что делать?

- Пусть у нас есть датасет $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$.
- Давайте сгенерируем много датасетов так: будем выбирать из \mathbf{X} N точек с замещением, т.е. в новом датасете некоторые точки будут повторяться.
- Этот метод называется *bootstrapping*.

- Мы сделаем так M датасетов размера N (с повторяющимися точками), потом обучим M моделей, а потом образуем из них комитет и будем предсказывать как

$$y(\mathbf{x}) = \frac{1}{M} \sum_{m=1}^M y_m(\mathbf{x}).$$

- Это называется *bagging* (bootstrap aggregation).
- На первый взгляд кажется, что это какая-то ерунда: мы пытаемся получить что-то из ничего...

- Пусть настоящая функция, которую мы пытаемся предсказать
 - $h(\mathbf{x})$, т.е. модели наши выглядят как

$$y_m(\mathbf{x}) = h(\mathbf{x}) + \epsilon_m(\mathbf{x}).$$

- Тогда средняя ошибка модели – это

$$\mathbb{E}_{\mathbf{x}} [(y_m(\mathbf{x}) - h(\mathbf{x}))^2] = \mathbb{E}_{\mathbf{x}} [\epsilon_m(\mathbf{x})^2].$$

- И средняя ошибка тех моделей, которые мы обучаем, получается

$$E_{\text{avg}} = \frac{1}{M} \sum_{m=1}^M \mathbb{E}_{\mathbf{x}} [\epsilon_m(\mathbf{x})^2].$$

- И средняя ошибка тех моделей, которые мы обучаем, получается

$$E_{\text{avg}} = \frac{1}{M} \sum_{m=1}^M E_x [\epsilon_m(\mathbf{x})^2].$$

- А ошибка комитета – это

$$\begin{aligned} E_{\text{com}} &= E_x \left[\left(\frac{1}{M} \sum_{m=1}^M (y_m(\mathbf{x}) - h(\mathbf{x})) \right)^2 \right] = \\ &= E_x \left[\left(\frac{1}{M} \sum_{m=1}^M \epsilon_m(\mathbf{x}) \right)^2 \right]. \end{aligned}$$

- $E_{\text{avg}} = \frac{1}{M} \sum_{m=1}^M E_{\mathbf{x}} [\epsilon_m(\mathbf{x})^2], E_{\text{com}} = E_{\mathbf{x}} \left[\left(\frac{1}{M} \sum_{m=1}^M \epsilon_m(\mathbf{x}) \right)^2 \right].$
- Если предположить, что $E_{\mathbf{x}} [\epsilon_m(\mathbf{x})] = 0$, и ошибки некоррелированы: $E_{\mathbf{x}} [\epsilon_m(\mathbf{x})\epsilon_l(\mathbf{x})] = 0$, мы получим

$$E_{\text{com}} = \frac{1}{M} E_{\text{avg}}.$$

- $E_{\text{com}} = \frac{1}{M} E_{\text{avg}}$!
- Это кажется совершенно невероятным. На самом деле всё не так хорошо – конечно, ошибки на самом деле сильно коррелированы.
- И, конечно, на самом деле обычно уменьшение ошибки не такое большое.
- Но можно показать, что в любом случае $E_{\text{com}} \leq E_{\text{avg}}$, так что хуже от этого не будет, а лучше стать может.

БУСТИНГ: ADABOOST

- Следующая идея объединения моделей: предположим, что у нас есть возможность обучать какую-нибудь простую модель (weak learner) на подмножестве данных.
- Тогда можно делать так: обучили модель, посмотрели, где она хорошо работает, обучили следующую модель на том подмножестве, где она работает плохо, повторили.
- Этот метод называется бустинг (boosting).

- AdaBoost: самый простой вариант. Рассмотрим задачу бинарной классификации; данные – это $\mathbf{x}_1, \dots, \mathbf{x}_N$ с ответами $t_1, \dots, t_N, t_i \in \{-1, 1\}$.
- Снабдим каждый тестовый пример весом w_i ; изначально положим $w_i = \frac{1}{N}$.
- Предположим, что у нас есть процедура, которая обучает некоторый классификатор, выдающий $y(\mathbf{x}) \in \{-1, 1\}$, на взвешенных данных (минимизируя взвешенную ошибку).

- Тогда в алгоритме AdaBoost мы инициализируем $w_n^{(1)} := 1/N$, а потом для $m = 1..M$:
 1. обучаем классификатор $y_m(\mathbf{x})$, который минимизирует функцию ошибки

$$J_m = \sum_{n=1}^N w_n^{(m)} [y_m(\mathbf{x}_n) \neq t_n];$$

2. вычисляем

$$\epsilon_m = \frac{\sum_{n=1}^N w_n^{(m)} [y_m(\mathbf{x}_n) \neq t_n]}{\sum_{n=1}^N w_n^{(m)}}, \quad \alpha_m = \ln \left(\frac{1 - \epsilon_m}{\epsilon_m} \right);$$

3. пересчитываем новые веса

$$w_n^{(m+1)} = w_n^{(m)} e^{\alpha_m [y_m(\mathbf{x}_n) \neq t_n]}.$$

- После обучения предсказываем как $Y_M(\mathbf{x}) = \text{sign} \left(\sum_{m=1}^M \alpha_m y_m(\mathbf{x}) \right)$.

- Смысл именно такой, как мы говорили: сначала тренируем абстрактно лучший классификатор. Потом увеличиваем веса неправильно классифицированным примерам, обучаем новый классификатор, и т.д.

ТЕОРЕТИЧЕСКИЕ СВОЙСТВА ADABOOST

- Изначально, когда AdaBoost придумали [Freund, Shapire, 1997], мотивация была такая: предположим, что ошибка каждого слабого классификатора h_t не превышает $\epsilon_t = \frac{1}{2} - \gamma_t$.
- Тогда можно показать, что окончательная ошибка не превосходит

$$\prod_t \left(2\sqrt{\epsilon_t(1-\epsilon_t)} \right) = \prod_t \sqrt{1 - 4\gamma_t^2} \leq e^{-2\sum_t \gamma_t^2}.$$

- Однако на самом деле гарантий на γ_t обычно нету, и практические результаты AdaBoost лучше, чем можно было бы ожидать из этой оценки.

ТЕОРЕТИЧЕСКИЕ СВОЙСТВА ADABoost

- Основная идея [Friedman et al., 2000]: давайте определим экспоненциальную ошибку

$$E = \sum_{n=1}^N e^{-t_n f_m(\mathbf{x}_n)},$$

где f_m – линейная комбинация базовых классификаторов:

$$f_m(\mathbf{x}) = \frac{1}{2} \sum_{l=1}^m \alpha_l y_l(\mathbf{x}).$$

- Мы хотим минимизировать E по α_l и параметрам $y_l(\mathbf{x})$.

ТЕОРЕТИЧЕСКИЕ СВОЙСТВА ADABoost

- Минимизируем $E = \sum_{n=1}^N e^{-t_n f_m(\mathbf{x}_n)}$.
- Вместо глобальной оптимизации будем действовать жадно:
пусть $y_1(\mathbf{x}), \dots, y_{m-1}(\mathbf{x})$ и $\alpha_1, \dots, \alpha_{m-1}$ уже зафиксированы.
Тогда ошибка получается

$$E = \sum_{n=1}^N e^{-t_n f_{m-1}(\mathbf{x}_n) - \frac{1}{2} t_n \alpha_m y_m(\mathbf{x})} = \sum_{n=1}^N w_n^{(m)} e^{-\frac{1}{2} t_n \alpha_m y_m(\mathbf{x})},$$

где $w_N^{(m)} = e^{-t_n f_{m-1}(\mathbf{x}_n)}$ – это как раз и есть наши веса, и их теперь можно считать константами.

ТЕОРЕТИЧЕСКИЕ СВОЙСТВА ADABoost

- На правильных классификациях произведение -1 , на неправильных $+1$:

$$\begin{aligned} E &= e^{-\frac{\alpha_m}{2}} \sum_{\text{correct}} w_n^{(m)} + e^{\frac{\alpha_m}{2}} \sum_{\text{wrong}} w_n^{(m)} = \\ &= \left(e^{\frac{\alpha_m}{2}} - e^{-\frac{\alpha_m}{2}} \right) \sum_{n=1}^N w_n^{(m)} [y_m(\mathbf{x}_n) \neq t_n] + e^{-\frac{\alpha_m}{2}} \sum_{n=1}^N w_n^{(m)}, \end{aligned}$$

и достаточно минимизировать $J_m = \sum_{n=1}^N w_n^{(m)} [y_m(\mathbf{x}_n) \neq t_n]$.

ТЕОРЕТИЧЕСКИЕ СВОЙСТВА ADABoost

- Ну а когда мы обучим $y_m(\mathbf{x})$, из $E = \sum_{n=1}^N w_n^{(m)} e^{-\frac{1}{2}t_n \alpha_m y_m(\mathbf{x}_n)}$ получится

$$w_n^{(m+1)} = w_n^{(m)} e^{-\frac{1}{2}t_n \alpha_m y_m(\mathbf{x}_n)} = w_n^{(m)} e^{-\frac{1}{2}\alpha_m} e^{\alpha_m [y_m(\mathbf{x}_n) \neq t_n]},$$

и на $e^{-\frac{1}{2}\alpha_m}$ можно все веса сократить.

- Таким образом, бустинг можно рассматривать как оптимизацию экспоненциальной ошибки.

СПАСИБО!

Спасибо за внимание!