The Tradeoffs of Large Scale Learning

Стрельцов Антон
07.03.2017

О чём?

До этого момента все наши предположения об оптимизации машинного обучения исходили из стандартного компромисса между сложностью алгоритма и точностью предсказания.

Появление класса задач, в котором количество данных растёт быстрее, чем вычислительная мощность компьютеров, задействованных в обучении, определило необходимость в качественно другой постановке задачи оптимизации – теперь мы ищем компромисс между тремя параметрами: сложностью алгоритма, точностью предсказания и временем обучения.

Почему это важно?

- Непрерывный рост количества данных
  - Разрыв между вычислительной мощностью, генерирующей данные, и вычислительной мощностью, анализирующей эти данные, постоянно растёт. (не в пользу последней)
- «**Быстрее**, выше, сильнее»
  - Если в результате исследования мы сможем лучше делать свою работу, нам (возможно) будут платить больше, наши конкуренты будут умываться слезами, мир станет лучше.
- Типичные алгоритмы обучения не «успевают» за данными
  - Если количество данных растёт, нам становится всё сложнее обращать, перемножать и осуществлять операции с матрицами. Грубо говоря, при увеличении количества данных в два раза, количество требуемых операций возрастает в восемь раз.

Стандартный подход:

1. Предположение статистики:

   “Имеет смысл минимизировать такую функцию потерь, которая быстро вычисляется даже при росте количества наблюдений.”

2. Предположение оптимизации:

   “Чтобы эффективно обрабатывать большое количество данных, нужно выбирать оптимизационный алгоритм с хорошей сходимостью, например супер-линейный.”

3. Вывод:

   “Выбери хорошую функцию потерь, выбери хороший алгоритм и надейся, что всё будет хорошо.”

Стандартный подход:

1. Предположение статистики:

   “Имеет смысл минимизировать такую функцию потерь, которая быстро вычисляется даже при росте количества наблюдений.”

2. Предположение оптимизации:

   “Чтобы эффективно обрабатывать большое количество данных, нужно выбирать оптимизационный алгоритм с хорошей сходимостью, например супер-линейный.”

3. Вывод:

   “Выбери хорошую функцию потерь, выбери хороший алгоритм и надейся, что всё будет хорошо.”

Слишком просто

Цели:

- Сформулировать теорию, в которой будет учитываться новое ограничение на время обучения алгоритма.
- Узнать, что изменилось:
  - Как улучшатся наши предсказания с ростом количества наблюдений.
  - Как ухудшится время обучения с ростом количества наблюдений.

Немножко определений и предположений:

- Предположение:
  Наши данные независимо генерируются из неизвестного нам распределения $P(x, y)$.
- Ожидаемый риск:
  $E(f) = S\, L(f(x), y)\, dP(x, y)$.
- Эмпирический риск:
  $E_n(f) = 1/n * \Sigma\, L(f(x_i), y_i)$.
- Лучший алгоритм:
  $f^* = argmin(E(f))$

- Семейство наших алгоритмов $\mathcal{F}$:
  В общем случае, $f^* \notin \mathcal{F}$
- Лучший доступный нам алгоритм:
  $f_{\mathcal{F}}^* = argmin(E(f_{\mathcal{F}}))$

По предположению, $P(x, y)$ нам неизвестно (!)

- Будем искать $f_n \in \mathcal{F}$ минимизирующий $E_n(f)$.

Теория Вапника-Червоненкиса подскажет, сработает ли это.

Идея:

1. Точное вычисление f_n чаще всего дорого.

2. Так как мы уже сделали достаточно много допущений, зачем нам знать точную f_n ?

3. Будем считать, что наша оптимизация возвращает некоторую f'_n , такую что $E_n(f'_n) < E_n(f_n) + \rho$

Например, мы можем останавливать наш итеративный алгоритм задолго до того, как он сойдётся.

Разложение ошибки

$$E(f'_n) - E(f^*) =$$

$$= E(f_{\mathcal{F}}^*) - E(f^*)$$ – ошибка аппроксимации

$$+ E(f_n) - E(f_{\mathcal{F}}^*)$$ – ошибка оценки

$$+ E(f'_n) - E(f_n)$$ – ошибка оптимизации

Задача:

Выбрать $\mathcal{F}$, n и ρ такие, чтобы ошибка была минимальной, при ограничениях на максимальное n и максимальное время исполнения T.

Разложение ошибки

- Ошибка аппроксимации:
  - ➔ Уменьшается с ростом сложности $\mathcal{F}$

(теория аппроксимации)

- Ошибка оценки:
  - ➔ Уменьшается с ростом n
  - ➔ Увеличивается с ростом сложности $\mathcal{F}$

(теория Вапника-Червоненкиса)

- Ошибка оптимизации:
  - ➔ Увеличивается с ростом ρ

(теория Вапника-Червоненкиса и магия)

- Время исполнения T:
  - ➔ Уменьшается с ростом ρ
  - ➔ Увеличивается с ростом сложности $\mathcal{F}$
  - ➔ Увеличивается с ростом n

(строго говоря, зависит от алгоритма оптимизации)

Small-scale vs. Large-scale Learning

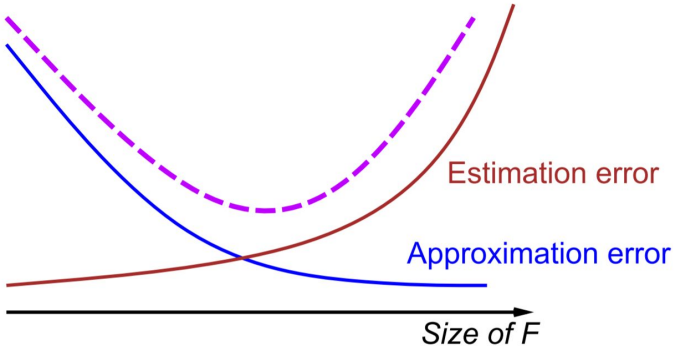
Теперь мы можем строго определить, что мы будем называть ..

- Проблемой малого масштаба ту, в которой активным ограничением будет максимальное количество наблюдений n;
- Проблемой большого масштаба ту, в которой активным ограничением будет максимальное время исполнения T.

Small-scale Learning

Активным ограничением является максимальное количество наблюдений.

- Чтобы уменьшить ошибку оценки, возьмём столько примеров для обучения, сколько у нас есть.
- Чтобы занулить ошибку оптимизации, возьмём $\rho = 0$.
- Далее нам нужно подобрать лучшее семейство алгоритмов.



На эту тему по слухам можно почитать Вапника, но я не читал.

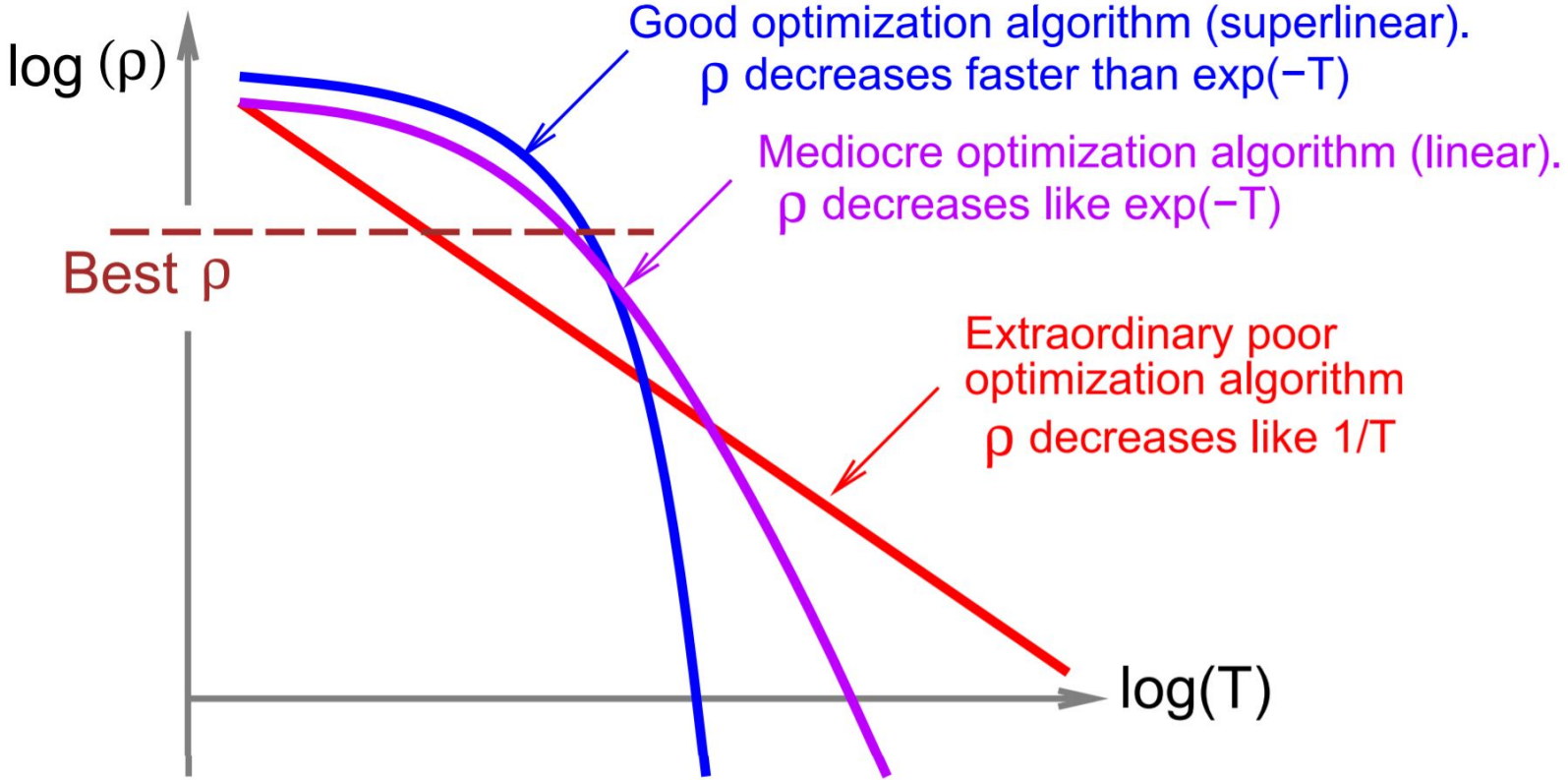
Large-scale Learning

Активным ограничением является время исполнения T.

- Поиск компромисса в данном случае будет сложнее, т. к. T зависит от всех переменных сразу.
- Точный компромисс всегда будет зависеть от конкретного оптимизационного алгоритма.

Зато мы можем строго сравнивать оптимизационные алгоритмы между собой!

Промежуточные итоги



Здесь могли быть выкладки из теории Вапника-Червоненкиса..

..но лучше не надо.

$$\text{Estimation error} \leq \mathcal{O}\left(\left[\frac{d}{n}\log\frac{n}{d}\right]^{\alpha}\right) \qquad \frac{1}{2} \leq \alpha \leq 1$$

$$\text{Estimation error} + \text{Optimization error} \leq \mathcal{O}\left(\left[\frac{d}{n}\log\frac{n}{d}\right]^{\alpha} + \rho\right)$$

Практическая часть

Исследование будет проводиться на фиксированном семействе алгоритмов с функциями, линейно зависящими от $w \in R^d$.

Будут рассмотрены четыре итеративных алгоритма оптимизации $E_n(f)$:

1. Градиентный спуск
2. Метод Ньютона (градиентный спуск второго порядка)
3. Стохастический градиентный спуск
4. Стохастический градиентный спуск второго порядка

Некоторые определения

- Эмпирический гессиан в эмпирическом оптимуме w_n :

$$H = \frac{\partial^2 E_n}{\partial w^2}(f_{w_n}) = \frac{1}{n}\sum_{i=1}^{n}\frac{\partial^2 \ell(f_n(x_i), y_i)}{\partial w^2}$$

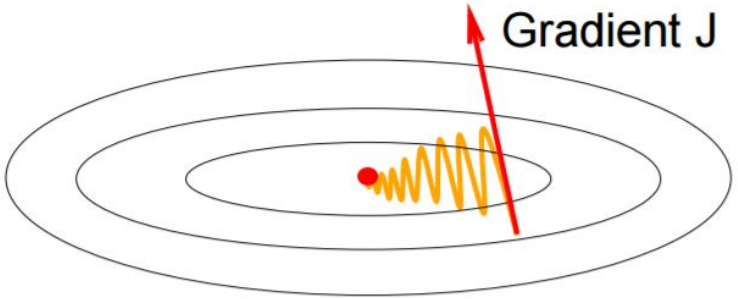
- Эмпирическая информационная матрица Фишера в w_n :

$$G = \frac{1}{n}\sum_{i=1}^{n}\left[\left(\frac{\partial \ell(f_n(x_i), y_i)}{\partial w}\right)\left(\frac{\partial \ell(f_n(x_i), y_i)}{\partial w}\right)'\right]$$

- $\mathrm{trace}(GH^{-1}) \approx \nu$; $\mathrm{spectrum}(H) \subset [\lambda_{min}, \lambda_{max}]$; $\kappa = \lambda_{max} / \lambda_{min}$;

Градиентный спуск

$$w_{t+1} \leftarrow w_t - \eta \frac{\partial E_n(f_{w_t})}{\partial w}$$



$$\eta = \frac{1}{\lambda_{\max}}$$

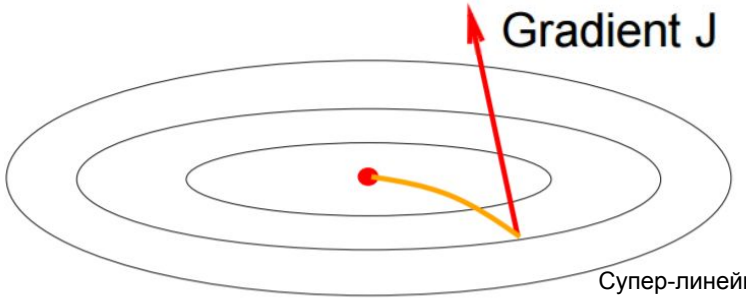
(e.g., Dennis & Schnabel, 1983)

одна итерация	кол-во итераций чтобы дойти до ρ	время чтобы дойти до ρ	время чтобы дойти до $E(f'_n) - E(f_{\mathcal{F}}^*) < \varepsilon$
$\mathcal{O}(nd)$	$\mathcal{O}\left(\kappa \log \frac{1}{\rho}\right)$	$\mathcal{O}\left(nd\kappa \log \frac{1}{\rho}\right)$	$\mathcal{O}\left(\frac{d^2 \kappa}{\varepsilon^{1/\alpha}} \log^2 \frac{1}{\varepsilon}\right)$

- ➔ В последней колонке ρ и n уже выбраны наилучшим образом
- ➔ Решаем относительно ε, чтобы найти лучшую ошибку в доступное время
- ➔ O() нотация не строгая

Метод Ньютона*

$$w_{t+1} \leftarrow w_t - H^{-1} \frac{\partial E_n(f_{w_t})}{\partial w}$$



* подразумевается, что гессиан известен и вычисляется бесплатно

Супер-линейная сходимость

(e.g., Dennis & Schnabel, 1983)

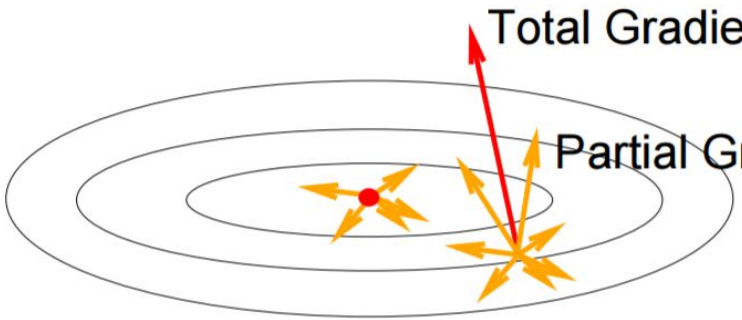
одна итерация	кол-во итераций чтобы дойти до ρ	время чтобы дойти до ρ	время чтобы дойти до $E(f'_n) - E(f_{\mathcal{F}}^*) < \varepsilon$
$\mathcal{O}\big(d(d+n)\big)$	$\mathcal{O}\left(\log\log\frac{1}{\rho}\right)$	$\mathcal{O}\left(d(d+n)\log\log\frac{1}{\rho}\right)$	$\mathcal{O}\left(\frac{d^2}{\varepsilon^{1/\alpha}}\log\frac{1}{\varepsilon}\log\log\frac{1}{\varepsilon}\right)$

➔ Необходимое время оптимизации значительно уменьшилось.

➔ Время обучения практически не изменилось, не считая пропажи к.

SGD

$$w_{t+1} \leftarrow w_t - \frac{\eta}{t} \frac{\partial \ell(f_{w_t}(x_t), y_t)}{\partial w}$$



$$\eta = \frac{1}{\lambda_{\min}}$$

(see Murata, 1998; Bottou & LeCun, 2004)

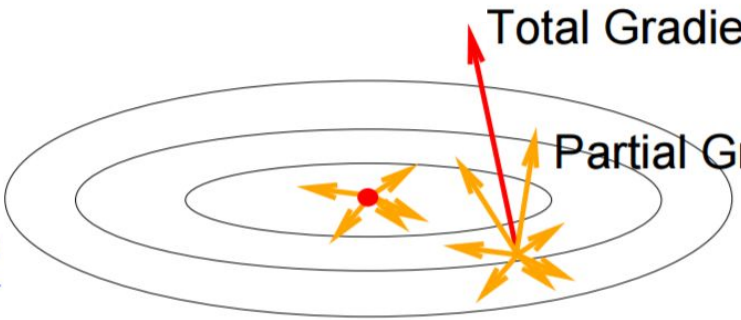
одна итерация	кол-во итераций чтобы дойти до ρ	время чтобы дойти до ρ	время чтобы дойти до $E(f'_n) - E(f_{\mathcal{F}}^*) < \varepsilon$
$\mathcal{O}(d)$	$\frac{\nu k}{\rho} + o\left(\frac{1}{\rho}\right)$	$\mathcal{O}\left(\frac{d \nu k}{\rho}\right)$	$\mathcal{O}\left(\frac{d \nu k}{\varepsilon}\right)$

$1 \leq k \leq \kappa^2$

- ➔ Время оптимизации *исключительно* плохо.
- ➔ Последняя колонка больше не зависит от α, но **к** вернулось.
- ➔ Теперь время обучения очень хорошо масштабируется.

2SGD

$$w_{t+1} \leftarrow w_t - \frac{1}{t} H^{-1} \frac{\partial \ell(f_{w_t}(x_t), y_t)}{\partial w}$$



$$\frac{\eta}{t} \rightarrow \frac{1}{t} H^{-1}$$

(see Murata, 1998; Bottou & LeCun, 2004)

одна итерация	кол-во итераций чтобы дойти до ρ	время чтобы дойти до ρ	время чтобы дойти до $E(f'_n) - E(f_{\mathcal{F}}^*) < \varepsilon$
$\mathcal{O}(d^2)$	$\frac{\nu}{\rho} + o\left(\frac{1}{\rho}\right)$	$\mathcal{O}\left(\frac{d^2 \nu}{\rho}\right)$	$\mathcal{O}\left(\frac{d^2 \nu}{\varepsilon}\right)$

- ➔ Каждая итерация в d раз дороже.
- ➔ Количество итераций уменьшилось в k раз (κ^2 или меньше)
- ➔ Второй порядок качественно асимптотики не изменил.

Выводы

- SGD и 2SGD не зависят от α.

- Алгоритмы второго порядка в общем случае несильно, но улучшают асимптотики (т.к. основной вклад всё равно у $1/\varepsilon$). В конкретных задачах эти изменения могут серьёзно повлиять на выбор алгоритма.

- Стохастические алгоритмы, несмотря на наихудшие показатели по стандартным критериям, показывают наилучшую производительность в нашем случае.

Спасибо за внимание!

При подготовке использовались:

Bousquet, Olivier, and Léon Bottou. "The tradeoffs of large scale learning." *Advances in neural information processing systems*. 2008.

https://papers.nips.cc/paper/3323-the-tradeoffs-of-large-scale-learning.pdf

Léon Bottou. “Learning with Large Datasets”

http://leon.bottou.org/slides/largescale/lstut.pdf