

Math of Data Science: Lecture 5

Vlad Kobzar

APAM, Columbia

September 20, 2022

Course progress

- ▶ Last lecture:
 - ▶ Basics of probability: Probabilistic inequalities
 - ▶ Jointly distributed random variables: independence and covariance
- ▶ This lecture:
 - ▶ Finish probability review: LLNs, Gaussian RVs and CLT
 - ▶ Presentation based on reference [1] and [3]
 - ▶ Optimization review
 - ▶ Presentation based on references [2], [4] and [5]

Motivation for expectation/LLN

- ▶ As with the sample mean, we can think of $E[X]$ as indicating where the values taken by X 'typically' lie (even though $E[X]$ may not actually equal any of the possible values of X)
- ▶ There are plenty of other quantities that can be used this way (such as 'median' and 'mode' in statistics).
- ▶ But the expectation has a better theory and more computational tools available, making it more useful to solve problems.
- ▶ For example, if the loss function depends on random inputs, its expectation is a natural choice of the thing to minimize in machine learning problems
- ▶ The LLN connects expectations with long-run averages when we perform an experiment many independent times

- ▶ One of our basic intuitions about probability is this: If we perform an experiment independently many times, and E is an event that can happen for each performance of the experiment, then in the long-run average

frequency of occurrence of $E \approx P(E)$.

- ▶ For instance, if 37% (not a real statistic) of US citizens have visible dandruff, and we randomly select a few thousand citizens (a large number, but much less than US population), then we expect about 37% of those sampled to have visible dandruff.
- ▶ So this is saying that, under these long-run average conditions, this 'frequency random variable' settles down, in some approximate sense, to the fixed value $P(E)$.

LLN

- ▶ Instead of an event E , assume our basic experiment has a random variable X , e.g., $\mathbb{1}_E$ indicator function of the event E
- ▶ Independent repeats of the experiment give independent copies of this random variable, say X_1, X_2, \dots
- ▶ In general, a sequence of RVs X_1, X_2, \dots are independent and identically distributed ('i.i.d.') if (i) they are independent, and (ii) they all have the same distribution. Let

$$\mu_n = \frac{1}{n} \sum_i X_i$$

- ▶ (Weak Law of Large Numbers, 'WLLN'). For any $\epsilon > 0$, we have

$$P(|\mu_n - E[X]| \geq \epsilon) \rightarrow 0 \text{ as } n \rightarrow \infty$$

where μ_n is the sample mean defined previously

LLN

- ▶ *(Weak) Law of Large Numbers:*
- ▶ For any $\epsilon > 0$, we have

$$P(|\mu_n - E[X]| \geq \epsilon) \rightarrow 0 \text{ as } n \rightarrow \infty$$

- ▶ We will justify this result subject to the extra assumption that every X has a well-defined and finite variance (some RVs don't). This must be the same for every X_i - call it σ^2 .
- ▶ (LLN is actually true without this assumption.)

LLN justification

- ▶ By linearity of expectation, we have $E[\mu_n] = E[X]$
- ▶ We've previously shown that

$$\text{Var}(aX) = \text{Cov}(aX, aX) = a^2 \text{Cov}(X, X) = a^2 \text{Var}(X)$$

- ▶ Also since X_i s are independent,

$$\text{Var}\left(\sum_{i=1}^n X_i\right) = \sum_{i=1}^n \text{Var}(X_i) = \sum_{i=1}^n \sigma_i^2$$

- ▶ Using these results and that X_i 's have the same variance,

$$\text{Var}(\mu_n) = \text{Var}\left(\frac{1}{n} \sum_{i=1}^n X_i\right) = \frac{\sigma^2}{n}$$

Then, since $E|\mu_n - E[X]|^2 = \text{Var}(\mu_n)$, by Chebychev

$$P(|\mu_n - E[X]| \geq \epsilon) = P(|\mu_n - E[X]|^2 \geq \epsilon^2) \leq \frac{\sigma^2}{n\epsilon^2}$$

LLN

- ▶ Let X_i 's be i.i.d. with expectation $E[X]$, the sample mean is

$$\mu_n = \frac{1}{n} \sum_i X_i$$

- ▶ (*Weak*) *Law of Large Numbers*: Then for any $\epsilon > 0$,

$$P(|\mu_n - E[X]| \geq \epsilon) \rightarrow 0 \text{ as } n \rightarrow \infty$$

- ▶ How large n has to be depends on how good an approximation you want
- ▶ Our proof gave an explicit estimate for how long we have to wait, given ϵ .
- ▶ Another way of visualizing the n Bernoulli-trials case: once n is large, then $\frac{1}{n}\text{binom}(p, n)$, PMF puts almost all of its mass into a narrow window around the mean p .

LLN

- ▶ The WLLN does not say that X_n is guaranteed to be close to p , only that this is very likely.
- ▶ Of course, if we're very unlucky, we might toss a fair coin but still get the outcome

HHHHHHHHHHHHHH or maybe

HHTHTTHTHTHHH

- ▶ For these very unlikely outcomes, the sample mean takes the values 1 and $2/3$ respectively, far away from the true mean, which is $1/2$.

LLN

- ▶ If we consider our running sequence of sample means μ_n , then WLLN says that, for each individual large value of n , μ_n is unlikely be far away from $E[X]$.
- ▶ But that's an infinite sequence of unlikely events.
- ▶ Even though their individual probabilities are small, we can still imagine that one of them occurs very occasionally.
- ▶ That is, it could be that μ_n mostly stays close to $E[X]$, but as n increases μ_n very occasionally makes a large deviation away from μ .
- ▶ Strong LLN says this doesn't happen.

$$P\left(\lim_{n \rightarrow \infty} \mu_n = E[X]\right) = 1$$

- ▶ Proof is more difficult than WLLN (e.g., can use estimates of the 4th moments $E[\mu_n^4]/n^4$)

- ▶ Back to our dandruff example:
 - ▶ If 37% of US citizens have visible dandruff, and we randomly select a thousand citizens (a large number, but much less than the US population), then we expect about 37% of those sampled to have visible dandruff.
 - ▶ Taking $X_i = \mathbb{1}_{i\text{-th person has dandruff}}$
 - ▶ Now SLLN ensures that

$$P\left(\lim_{n \rightarrow \infty} \mu_n = E[X] = P(\text{person has dandruff})\right) = 1$$

- ▶ But how confident can we be of this approximation? Is a sample of a thousand large enough for the effect to be reliable?

CLT

- ▶ In the LLN context our question was: Pick two error tolerances, $\epsilon > 0$ and $\alpha > 0$. How large does n have to be so that

$$P(|\mu_n - E[X]| \geq \epsilon) < \alpha$$

- ▶ (There were really two kinds of error tolerance involved all along: ϵ is how close you want μ_n to be to $E[X] = \rho$, and α is the small probability of error that you allow.
- ▶ Our proof of the WLLN gave

$$P(|\mu_n - E[X]| \geq \epsilon) \leq \frac{\sigma^2}{n\epsilon^2}$$

- ▶ I.e., bound on the probability of the tails

CLT

- ▶ Now our question is: effectively how wide are spikes around the mean as $n \rightarrow \infty$.
- ▶ E.g., for a Bernoulli X_1, \dots, X_n with $p = 1/2$, let

$$S_n = \sum_{i=1}^n X_i$$

- ▶ We want to approximate the 'shape' of the PMF

$$P(S_n = k)$$

when k/n is close to $1/2$, i.e. in a range

$$n/2 - n\epsilon < k < n/2 + n\epsilon$$

Gaussian

- ▶ The *normal distribution* with mean m and variance σ^2 , or $N(m, \sigma^2)$ is given by the Gaussian density

$$p(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-m)^2/(2\sigma^2)}$$

- ▶ The the CDF of standard normal $N(0, 1)$ is

$$\Phi(a) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^a e^{-x^2/2}$$

CLT

- ▶ Let X_1, \dots, X_n be i.i.d with mean $E[X]$ and $\text{Var}(X) = \sigma^2$ and

$$\mu_n = \frac{1}{n}(X_1 + \dots + X_n)$$

- ▶ *The Central Limit Theorem:* The limiting distribution of $\sqrt{n}(\mu_n - E[X])$ is $N(0, 1)$, in the following sense.

$$P(a < \frac{\sqrt{n}}{\sigma}(\mu_n - E[X]) < b) \rightarrow \Phi(b) - \Phi(a)$$

as $n \rightarrow \infty$.

- ▶ This implies

$$P(|\mu_n - E[X]| < \frac{c}{\sqrt{n}}) \rightarrow \Phi\left(\frac{c}{\sigma}\right) - \Phi\left(-\frac{c}{\sigma}\right)$$

- ▶ and taking $\epsilon = \frac{c}{\sqrt{n}}$

$$P(|\mu_n - E[X]| < \epsilon) \rightarrow \Phi\left(\frac{\epsilon\sqrt{n}}{\sigma}\right) - \Phi\left(-\frac{\epsilon\sqrt{n}}{\sigma}\right)$$

CLT

- ▶ Note that CLT gives

$$P(|\mu_n - E[X]| < \epsilon) \rightarrow \Phi\left(\frac{\epsilon\sqrt{n}}{\sigma}\right) - \Phi\left(-\frac{\epsilon\sqrt{n}}{\sigma}\right)$$

- ▶ On the other hand, if we wanted to bound the probability mass around the mean rather than the tails, the WLLN is not informative as $n \rightarrow \infty$

$$P(|\mu_n - E[X]| < \epsilon) \geq 1 - \frac{\sigma^2}{n\epsilon^2}$$

- ▶ This follows from

$$1 - P(|\mu_n - E[X]| < \epsilon) = P(|\mu_n - E[X]| \geq \epsilon) \leq \frac{\sigma^2}{n\epsilon^2}$$

Optimization review - convexity

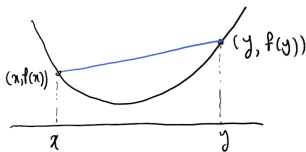
- ▶ K is a *convex set*: if $x, y \in K$, then so is the line segment from x to y , i.e.

$$px + (1 - p)y \in K$$

for $p \in [0, 1]$.

- ▶ f is a convex fcn if the set of points on and above the graph of F is convex, i.e.

$$f(px + (1 - p)y) \leq pf(x) + (1 - p)f(y)$$



for $p \in [0, 1]$

Convex optimization

- ▶ Convex optimization

$$\min_{x \in K} f(x)$$

where K is a convex set and $f(x)$ is a convex function

- ▶ Thm: Any local minimum is global, and there will be no isolated local min
- ▶ Justification: if that wasn't true, then the line between two local min would intersect with the graph of f

Gradient descent

- ▶ Let's start with an unconstrained problem

$$\min_x f(x)$$

- ▶ Gradient descent algorithm is an iterative method given by

$$x_{k+1} = x_k - s_k \nabla f(x_k)$$

- ▶ x_0 is an initial (often random) guess
- ▶ We'll discuss the step size (also called learning rate $s_k > 0$) shortly.

GD and learning

- ▶ Let's look at our linear regression objective (typically scale the loss by the number of data points in ML, but that shouldn't affect the minimization)

$$R(x) = \frac{1}{n} \|Ax - b\|^2 = \frac{1}{n} \sum_{i=1}^n (x^T a_i - b_i)^2$$

- ▶ To learn x , we can follow the GD descent algorithm

$$x_{k+1} = x_k - s_k \nabla R(x)$$

where

$$\nabla R(x) = \frac{1}{n} (2x^T A^T A - 2b^T A) = \frac{1}{n} \sum_{i=1}^n 2(x^T a_i - b_i) a_i$$

GD and learning

- ▶ Our linear regression objective uses a square loss to measure the difference between the prediction of a linear model $x^T a_i$ and the actual data b_i

$$\ell_i(x) = (x^T a_i - b_i)^2$$

- ▶ You can generalize this to other loss functions and learning algorithms

$$\ell_i(x) = \ell(F(x, a_i) - b_i)^2$$

- ▶ E.g., F can represent a neural network, which outputs a label $F(x, a)$ given features a and parameters x .
- ▶ The training, i.e., learning x given the data, can be done by minimizing

$$L(x) = \frac{1}{n} \sum_{i=1}^n \ell_i(x)$$

- ▶ Sometimes the above term is called empirical risk, and the training process is called empirical risk minimization

Learning rate

- ▶ Several ways to determine s_k depending on the algorithm
- ▶ For GD, a convergence guarantee is available for a fixed step size $s \leq 1/M$ where M is the Lipschitz constant of the gradient

$$\|\nabla f(x) - \nabla f(y)\| \leq M\|x - y\|$$

uniformly over the domain

- ▶ If f is C^2 , then M would be the bound of the eigenvalues λ_i of the Hessian

$$|\lambda_i| \leq M$$

for all i uniformly in x .

- ▶ Often M is not known, but we can extend the convergence guarantees to *exact line search*

$$s_k = \arg \min_{s \geq 0} f(x_k - s\nabla f(x_k))$$

- ▶ And *backtracking line search*, which iteratively reduces s_k until

$$f(x_{k+1}) \leq f(x_k) - \frac{1}{2}s_k\|\nabla f(x_k)\|_2^2$$

Gradient descent convergence - fixed step size

- ▶ Let $G = \|\nabla f(x_k)\|_2^2$: the 2nd order Taylor expansion is

$$\begin{aligned} & f(x_{k+1}) \\ &= f(x_k) + \langle \nabla f(x_k), x_{k+1} - x_k \rangle + \frac{1}{2}(x_{k+1} - x_k)^T H(\xi)(x_{k+1} - x_k) \\ &= f(x_k) - sG + \frac{1}{2}(x_{k+1} - x_k)^T H(\xi)(x_{k+1} - x_k) \end{aligned}$$

for some ξ on the segment between x_{k+1} and x_k (mean value form of the remainder)

- ▶ If the eigenvalues λ_i of H are $|\lambda_i| \leq M$ for all i uniformly in x

$$f(x_{k+1}) \leq f(x_k) - sG + \frac{s^2 M}{2} G = f(x_k) - \frac{1}{2M} G$$

- ▶ The RHS is minimized taking the step size $s = \frac{1}{M}$.
- ▶ But any $s < \frac{2}{M}$ will reduce f

Gradient descent convergence - fixed step size

- ▶ From the previous slide

$$f(x_{k+1}) \leq f(x_k) - \frac{1}{2M} G$$

taking the step size $s = \frac{1}{M}$ where $G = \|\nabla f(x_k)\|_2^2$.

- ▶ Assume that f is bounded below by f^* (reasonable in ML since the loss function is typically nonnegative)
- ▶ Idea of a convergence argument
 - ▶ Start at some $f(x_0)$ and at each step decrease f by at least $\frac{1}{2M} G$
 - ▶ We can't decrease $f(x_k)$ below f^*
 - ▶ So G must be going to zero "fast enough"

Gradient descent convergence - fixed step size

- ▶ From the previous slide

$$f(x_{k+1}) \leq f(x_k) - \frac{1}{2M} G$$

taking the step size $s = \frac{1}{M}$ where $G = \|\nabla f(x_k)\|_2^2$.

- ▶ rearrange

$$G \leq 2M(f(x_k) - f(x_{k+1}))$$

- ▶ sum over time periods

$$\sum_{k=0}^t \|\nabla f(x_k)\|_2^2 \leq 2M \sum_{k=0}^t [f(x_k) - f(x_{k+1})]$$

Gradient descent convergence - fixed step size

- ▶ From the previous slide

$$\sum_{k=0}^{t-1} \|\nabla f(x_k)\|_2^2 \leq 2M \sum_{k=0}^{t-1} [f(x_k) - f(x_{k+1})]$$

- ▶ Replacing $\nabla f(x_k)$ with $\min_k \nabla f(x_k)$ on the LHS, using the telescoping sum on the RHS and then the fact that $f(x_t) \geq f^*$

$$t \min_{0 \leq k \leq t-1} \|\nabla f(x_k)\|_2^2 \leq 2M[f(x_0) - f(x_t)] \leq 2M[f(x_0) - f^*]$$

- ▶ Therefore

$$\min_{0 \leq k \leq t-1} \|\nabla f(x_k)\|_2^2 \leq \frac{2M[f(x_0) - f^*]}{t} = O(1/t)$$

Gradient descent convergence - fixed step size

- ▶ From the previous slide

$$\min_{0 \leq k \leq t-1} \|\nabla f(x_k)\|_2^2 \leq \frac{2M[f(x_0) - f^*]}{t} = O(1/t)$$

- ▶ So the norm is below ϵ if

$$\frac{2M[f(x_0) - f^*]}{t} \leq \epsilon$$

- ▶ This is guaranteed for

$$\frac{2M[f(x_0) - f^*]}{\epsilon} \leq t$$

- ▶ So GD requires $t = O(1/\epsilon)$ iterations to achieve $\|\nabla f(x_k)\|_2^2 \leq \epsilon$

Gradient descent convergence - fixed step size

- ▶ The previous argument didn't assume that f is convex, so the GD was converging to a local minimum (theoretically could also converge to a saddle point).
- ▶ Guaranteeing convergence to a global minimum of a nonconvex function requires multiple random initializations or grid search
- ▶ This requires $t = O(1/\epsilon^d)$ iterations to achieve $\|\nabla f(x_k)\|_2^2 \leq \epsilon$ for functions with Lipschitz continuous gradients and $x \in \mathbb{R}^d$
- ▶ In practice gradient-based methods work well for non-convex functions used in ML/NN even though there are not theoretical convergence guarantees
- ▶ If f is convex, then $\nabla f(x^*) = 0$ at a global minimizer x^*
- ▶ Therefore the above argument guarantees convergence to the global min at the above rate

Gradient descent convergence - fixed step size

- ▶ If f is *strongly* convex, i.e, the eigenvalues λ_j of the Hessian H are also $0 < m \leq \lambda_j$ uniformly in x , the convergence $O((1 - \frac{m}{M})^k)$ for $0 < c < 1$.
- ▶ This means that a bound of

$$f(x_k) - f(x^*) \leq \epsilon$$

can be achieved using only $O(\log(1/\epsilon))$ iterations.

- ▶ This rate is called “linear convergence” for historic reasons (the error lies below a line on a log-linear plot of the error vs iteration number)
- ▶ Many loss functions in ML are not strongly convex, e.g., Relu and softmax are convex but not strongly convex
- ▶ Adding an ℓ^2 regularization term will make them such and can improve convergence

Gradient descent convergence

- ▶ Let $G = \|\nabla f(x_k)\|_2^2$: the 2nd order Taylor expansion is

$$\begin{aligned} & f(x_{k+1}) \\ &= f(x_k) + \langle \nabla f(x_k), x_{k+1} - x_k \rangle + \frac{1}{2}(x_{k+1} - x_k)^T H(\xi)(x_{k+1} - x_k) \\ &= f(x_k) - sG + \frac{1}{2}(x_{k+1} - x_k)^T H(\xi)(x_{k+1} - x_k) \end{aligned}$$

for some ξ on the segment between x_{k+1} and x_k (mean value form of the remainder)

- ▶ If the eigenvalues λ_i of H are $m \leq \lambda_i \leq M$ for all i

$$f(x_{k+1}) \leq f(x_k) - sG + \frac{s^2 M}{2} G = f(x_k) - \frac{1}{2M} G$$

taking $s = \frac{1}{M}$ (exact line search)

- ▶ For the optimal x^* ,

$$f(x_{k+1}) - f(x^*) \leq f(x_k) - f(x^*) - \frac{1}{2M} G$$

Gradient descent convergence

- ▶ By a similar argument

$$\begin{aligned} f(x^*) &= f(x_k) + \langle \nabla f(x_k), x^* - x_k \rangle + \frac{1}{2}(x^* - x_k)^T H(\xi)(x^* - x_k) \\ &\geq f(x_k) + \langle \nabla f(x_k), \tilde{x} - x_k \rangle + \frac{m}{2} \|\tilde{x} - x_k\|^2 \\ &= f(x_k) - \frac{1}{2m} G \end{aligned}$$

where $\tilde{x} = x_k - \frac{1}{m} \nabla f(x_k)$ is determined by minimizing the above expression with respect to \tilde{x}

- ▶ we get

$$f(x_{k+1}) - f(x^*) \leq \left(1 - \frac{m}{M}\right)(f(x_k) - f(x^*))$$

Learning rate-GD line search

- ▶ The previous discussion suggested a fixed step size of $1/M$
- ▶ In practice, the Lipschitz constant of the gradient is not known
- ▶ so “try a big step-size, and decrease it if isn't satisfying a progress bound.”
- ▶ Another alternative is *exact line search*:

$$s_k = \arg \min_s f(x_k - s \nabla f(x_k))$$

- ▶ The previous bounds relied on

$$f(x_{k+1}) \leq f(x_k) - \frac{1}{2M} G$$

where x_{k+1} is determined using a fixed step size $s = \frac{1}{M}$.

- ▶ For x_{k+1}^* determined by an exact line search

$$f(x_{k+1}^*) \leq f(x_{k+1})$$

so all the convergence guarantees still hold

Learning rate-GD backtracking line search

- ▶ Exact line search can be also computationally expensive since we need to optimize f in 1D
- ▶ For $\alpha \in (0, 0.5)$, $\beta \in (0, 1)$, initial $t = 1$, backtracking line search entails iteratively reducing $s = \alpha t$ via

$$t = \beta t$$

until $f(x_{k+1}) \leq f(x_k) - \alpha t \|\nabla f(x_k)\|_2^2$

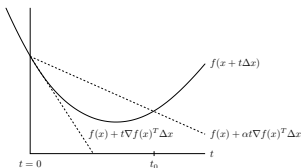


Figure: The backtracking condition is that f lies below the upper dashed line Fig 9.1 from [5]

- ▶ Since $-\nabla f(x_k)$ is a descent direction, the backtracking condition is satisfied for a sufficiently small $t \in (0, t_0]$.
- ▶ Therefore, the line search stops with $t = 1$ or $t \in (\beta t_0, t_0]$.

Learning rate-GD backtracking line search

- ▶ From previous the line search stops with $t = 1$ or $t \in (\beta t_0, t_0]$.
 $t \geq \min(t, \beta t_0)$
- ▶ By convexity of $g(t) = -t + Mt^2/2$

$$g(x/M + (1 - x) \cdot 0) \leq xg(1/M) - (1 - x)g(0)$$

- ▶ This implies

$$-t + Mt^2/2 \leq -t/2$$

Learning rate-GD backtracking line search

- ▶ By the previous slide

$$-t + Mt^2/2 \leq -t/2$$

- ▶ By an earlier computation, for $G = \|\nabla f(x_k)\|_2^2$

$$\begin{aligned} f(x_{k+1}) &\leq f(x_k) - tG + \frac{t^2 M}{2} G \\ &\leq f(x_k) - \frac{t}{2} G \\ &\leq f(x_k) - \alpha t G \end{aligned}$$

- ▶ Since the first inequality holds with equality for $t = 1/M$, the backtracking line search terminates with $t = 1$ or $t \geq \beta/M$:

$$f(x_{k+1}) \leq f(x_k) - \min\{\alpha, \beta\alpha/M\} G$$

Learning rate-GD backtracking line search

- ▶ By the previous slide

$$f(x_{k+1}) \leq f(x_k) - \min\{\alpha, \beta\alpha/M\}G$$

- ▶ We can repeat the earlier argument with a different constant prefactor

$$\min_{0 \leq k \leq t-1} \|\nabla f(x_k)\|_2^2 \leq \frac{[f(x_0) - f^*]}{\min\{\alpha, \beta\alpha/M\}t} = O(1/t)$$

Nesterov accelerated descent

- ▶ If f is convex (but not necessarily strongly convex) is the $t = O(1/\epsilon)$ “sublinear convergence” optimal?
- ▶ So called Nesterov accelerated descent

$$x_{k+1} = y_k - s \nabla f(y_k)$$

$$y_{k+1} = x_{k+1} + \beta_k (x_{k+1} - x_k)$$

achieves error of $O(1/t^2)$ after t iterations.

- ▶ Can use $s = 1/M$ and $\beta_k = (k - 1)/(k + 2)$
- ▶ So only needs $t = O(1/\sqrt{\epsilon})$ to get within ϵ of the solution
- ▶ It not straightforward to understand why this method works better
- ▶ One observation is that this is not a descent method, i.e., the steps may overshoot the minimum and oscillate around it, rather than converging from one direction.
- ▶ Used in practice to optimize convex and nonconvex function in ML

Second order methods: Newton's method

- ▶ A second order approximation of a convex C^2 function is

$$g(y) = f(x) + \nabla f(y - x) + \frac{1}{2}(y - x)^T \nabla^2 f(x)(y - x)$$

- ▶ If the Hessian is positive definite,

$$\arg \min_y g(y) = x - (\nabla^2 f(x))^{-1} \nabla f(x)$$

- ▶ This idea leads to Newton's method

$$x_{k+1} = x_k - (\nabla^2 f(x_k))^{-1} \nabla f(x_k)$$

which has quadratic convergence under certain assumptions

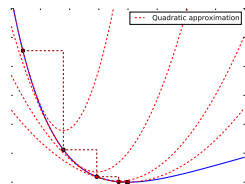







Figure: Quadratic approximation of 1D convex function (Fig14 from [4])

Next steps

- ▶ PCA (Sec I.9)
- ▶ Conjugate gradients (Sec. II.1) and least squares (Sec. II.2)

References I

-  [1] Tim Austin, *Theory of Probability unpublished lecture notes*, 2016
-  [2] Strang, *Linear Algebra and Learning from Data*, Wellesley Cambridge Press, 2019 and
-  [3] Ross, *A First Course in Probability* (9th ed., 2014)
-  [4] Carlos Fernandez-Granda, *DS-GA 1013 / MATH-GA 2821 Optimization-based Data Analysis, Lecture Notes*, 2017
https://math.nyu.edu/~cfgranda/pages/OBDA_fall17/index.html
-  [5] Boyd, Vandenberghe, *Convex Optimization*