Machine learning lecture slides

COMS 4771 Fall 2020

Optimization I: Convex optimization

Outline

- Convex sets and convex functions
- Local minimizers and global minimizers
- Gradient descent
- Analysis for smooth objective functions
- Stochastic gradient method
- Gradient descent for least squares linear regression

Convex sets

- *Convex set*: a set that contains every line segment between pairs of points in the set.
- Examples:
  - All of $\mathbb{R}^d$
  - Empty set
  - Half-spaces
  - Intersections of convex sets
  - Convex hulls

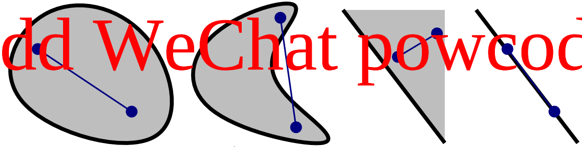

Figure 1: Which of these sets are convex?

Convex functions (1)

- *Convex function*: a function satisfying the two-point version of *Jensen's inequality*:

$$f((1-\alpha)w+\alpha w') \leq (1-\alpha)f(w)+\alpha f(w'), \quad w, w' \in \mathbb{R}^d, \alpha \in [0,1].$$

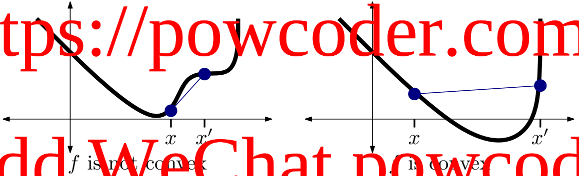

Figure 2: Which of these functions are convex?

Convex functions (2)

- Examples:
  - $f(w) = c$ for $c \in \mathbb{R}$
  - $f(w) = \exp(w)$ (on $\mathbb{R}$)
  - $f(w) = |w|^c$ for $c \geq 1$ (on $\mathbb{R}$)
  - $f(w) = b^\top w$ for $b \in \mathbb{R}^d$
  - $f(w) = \|w\|$ for any norm $\|\cdot\|$
  - $f(w) = w^\top A w$ for any symmetric positive semidefinite matrix A
  - $w \mapsto af(w) + g(w)$ for convex functions f, g and $a \geq 0$
  - $w \mapsto \max\{f(w), g(w)\}$ for convex functions f, g
  - $f(w) = \text{logsumexp}(w) = \ln\left(\sum_{i=1}^d \exp(w_i)\right)$
  - $w \mapsto f(g(w))$ for convex function f and affine function g

Verifying convexity of Euclidean norm

- Verify $f(w) = \|w\|$ is convex

Convexity of differentiable functions (1)

- Differentiable function f is convex iff

$$f(w) \geq f(w_0) + \nabla f(w_0)^\mathsf{T}(w - w_0) \quad \text{for all } w, w_0 \in \mathbb{R}^d.$$

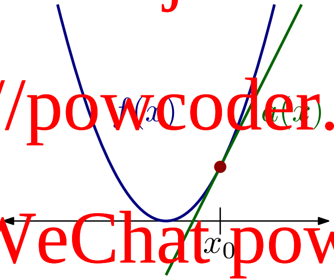

Figure 3: Affine approximation

- Twice-differentiable function f is convex iff $\nabla^2 f(w)$ is positive semidefinite for all $w \in \mathbb{R}^d$.

- Example: Verify $f(w) = w^4$ is convex
  - Use second-order condition

- Example: Verify $f(w) = e^{b^\mathsf{T} w}$ for $b \in \mathbb{R}^d$ is convex
  - Use first-order condition

Verifying convexity of least squares linear regression

- Verify $f(w) = \|Aw - b\|_2^2$ is convex

Verifying convexity of logistic regression MLE problem

- Verify $f(w) = \frac{1}{n}\sum_{i=1}^{n} \ln(1 + e^{-y_i x_i^\mathsf{T} w})$ is convex

- Say $w^\star \in \mathbb{R}^d$ is a *local minimizer* of $f\colon \mathbb{R}^d \to \mathbb{R}$ if there is an "open ball" $U = \{w \in \mathbb{R}^d : \|w - w^\star\|_2 < r\}$ of positive radius $r > 0$ such that $f(w^\star) \leq f(w)$ for all $w \in U$.
- I.e., nothing looks better in the immediate vicinity of $w^\star$.

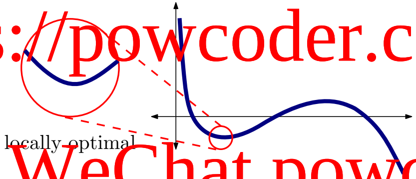

Figure 4: Local minimizer

Local minimizers of convex problems

- If f is convex, and $w^\star$ is a local minimizer, then it is also a global minimizer.
- "Local to global" phenomenon
- Local search is well-motivated for convex optimization problems

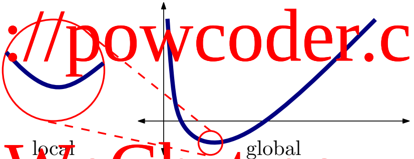

Figure 5: Local-to-global phenomenon

Gradient descent

- Consider (unconstrained) convex optimization problem

$$\min_{w \in \mathbb{R}^d} f(w).$$

- *Gradient descent*: iterative algorithm for (approximately) minimizing f
- Given *initial iterate* $w^{(0)} \in \mathbb{R}^d$ and *step size* $\eta > 0$,
  - For $t = 1, 2, \ldots$:

$$w^{(t)} := w^{(t-1)} - \eta \nabla f(w^{(t-1)}).$$

- (Lots of things unspecified here ...)

- Why move in direction of (negative) gradient?
- Affine approximation of $f(w + \delta)$ around w:

$$f(w + \delta) \approx f(w) + \nabla f(w)^{\mathsf{T}} \delta.$$

- Therefore, want δ such that $\nabla f(w)^{\mathsf{T}} \delta < 0$
- Use $\delta := -\eta \nabla f(w)$ for some $\eta > 0$:

$$\nabla f(w)^{\mathsf{T}} (-\eta \nabla f(w)) = -\eta \|\nabla f(w)\|_2^2 < 0$$

  as long as $\nabla f(w) \neq 0$.

- Need η to be small enough so still have improvement given error of affine approximation.

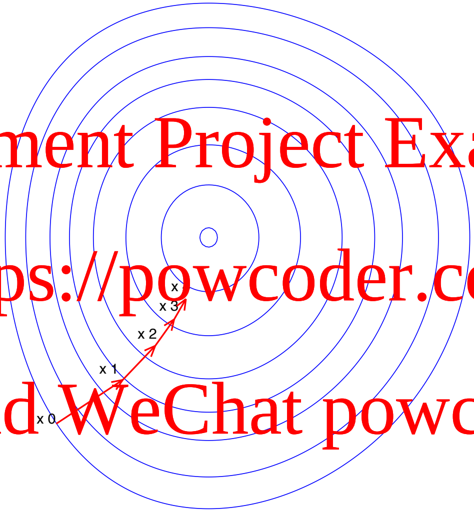

Figure 6: Trajectory of gradient descent

- Negative gradient of logistic loss on i-th training example: using chain rule,

$$
\begin{aligned}
-\nabla\{\ell_{\text{logistic}}(y_i x_i^{\mathsf{T}} w)\} &= -\ell'_{\text{logistic}}(y_i x_i^{\mathsf{T}} w) y_i x_i \\
&= \left(1 - \frac{1}{1 + \exp(-y_i x_i^{\mathsf{T}} w)}\right) y_i x_i \\
&= (1 - \sigma(y_i x_i^{\mathsf{T}} w)) y_i x_i
\end{aligned}
$$

where σ is the sigmoid function.
- Recall, $\Pr_w(Y = y \mid X = x) = \sigma(y x^{\mathsf{T}} w)$ for (X, Y) following the logistic regression model.

Example: Gradient descent for logistic regression

- Objective function:

$$f(w) = \frac{1}{n} \sum_{i=1}^{n} \ell_{\text{logistic}}(y_i x_i^\mathsf{T} w)$$

- Gradient descent: given *initial iterate* $w^{(0)} \in \mathbb{R}^d$ and *step size* $\eta > 0$,

- For $t = 1, 2, \ldots$:

$$\begin{aligned} w^{(t)} &:= w^{(t-1)} - \eta \nabla f(w^{(t-1)}) \\ &= w^{(t-1)} + \eta \frac{1}{n} \sum_{i=1}^{n} (1 - \sigma(y_i x_i^\mathsf{T} w^{(t-1)})) y_i x_i \end{aligned}$$

- Interpretation of update:
  - How much of $y_i x_i$ to add to $w^{(t-1)}$ is scaled by how far $\sigma(y_i x_i^\mathsf{T} w^{(t-1)})$ currently is from 1.

Convergence of gradient descent on smooth objectives

- **Theorem**: Assume f is twice-differentiable and convex, and $\lambda_{\max}(\nabla^2 f(w)) \leq \beta$ for all $w \in \mathbb{R}^d$ ("f is *β-smooth*"). Then gradient descent with step size $\eta = 1/\beta$ satisfies

$$f(w^{(t)}) \leq f(w^\star) + \frac{\beta \|w^{(0)} - w^\star\|_2^2}{2t}.$$

  - Same holds even if f only once-differentiable, as long as gradient $\nabla f(w)$ does not change too fast with w:

$$\|\nabla f(w) - \nabla f(w')\|_2 \leq \beta \|w - w'\|_2.$$

  - Note: it is possible to have convergence even with $\eta > 1/\beta$ in some cases; should really treat η as a hyperparameter.

Example: smoothness of empirical risk with squared loss

- Empirical risk with squared loss

$$\nabla^2 \left\{ \|Aw - b\|_2^2 \right\} = A^\top A.$$

So objective function is β-smooth with $\beta = \lambda_{\max}(A^\top A)$.

- Empirical risk with logistic loss

$$\nabla^2 \left\{ \frac{1}{n} \sum_{i=1}^{n} \ln(1 + \exp(-y_i x_i^\mathsf{T} w)) \right\}$$

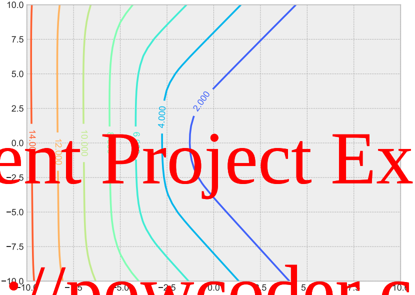

Figure 7: Gradient descent for logistic regression

- By Taylor's theorem, can upper-bound $f(w+\delta)$ by quadratic:

$$f(w+\delta) \le f(w) + \nabla f(w)^{\mathsf{T}}\delta + \frac{\beta}{2}\|\delta\|_2^2.$$

- Gradient descent is based on making local quadratic upper-bounds, and minimizing that quadratic:

$$\min_{\delta \in \mathbb{R}^d} f(w) + \nabla f(w)^{\mathsf{T}}\delta + \frac{\beta}{2}\|\delta\|_2^2.$$

Minimized by $\delta := -\frac{1}{\beta}\nabla f(w)$.

- Plug-in this value of δ into above inequality to get

$$f\left(w - \frac{1}{\beta}\nabla f(w)\right) - f(w) \le -\frac{1}{2\beta}\|\nabla f(w)\|_2^2.$$

- If f is convex (in addition to β-smooth), then repeatedly making such local changes is sufficient to approximately minimize f.

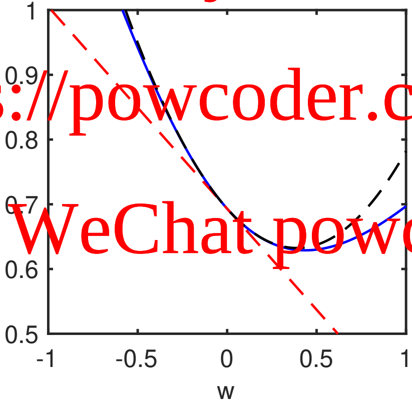

Figure 8: Linear and quadratic approximations to a convex function

Example: Text classification (1)

- Data: articles posted to various internet message boards
- Label: -1 for articles from "religion", $+1$ for articles from "politics"
- Features:
  - Vocabulary of $d = 61188$ words
  - Each document is a binary vector $x \in \{0,1\}^d$, where $x_i = 1_{\{\text{document contains } i\text{th vocabulary word}\}}$
- Executed gradient descent with $\eta = 0.25$ for 500 iterations

Example: Text classification (2)

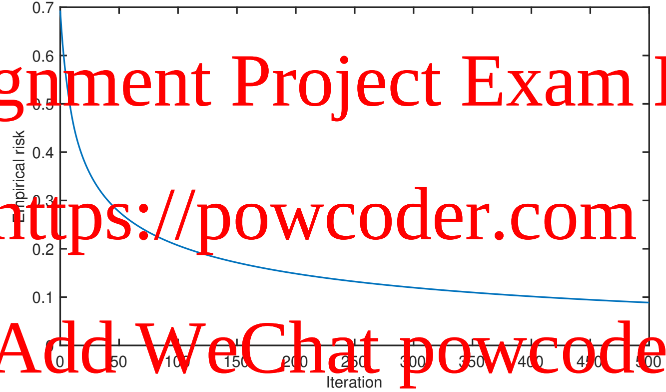

Figure 9: Objective value as a function of number of gradient descent iterations

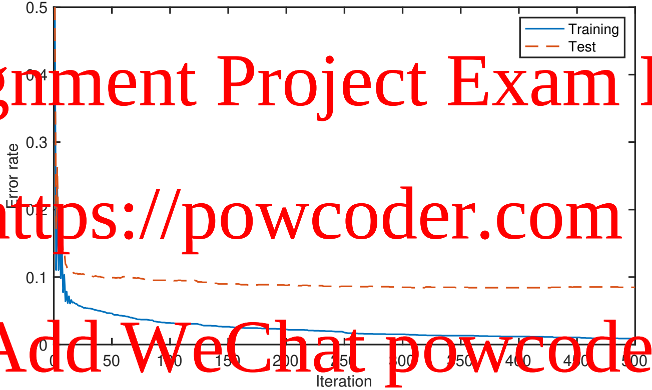

Figure 10: Error rate as a function of number of gradient descent iterations

Stochastic gradient method (1)

- Every iteration of gradient descent takes $\Theta(nd)$ time.
  - Pass through all training examples to make a single update.
  - If n is enormous, too expensive to make many passes.
- Alternative: *Stochastic gradient descent (SGD)*
  - Another example of plug-in principle!
  - Use one or a few training examples to estimate the gradient.
  - Gradient at $w^{(t)}$:

$$\frac{1}{n}\sum_{j=1}^{n} \nabla \ell(y_j x_j^{\mathsf{T}} w^{(t)}).$$

    (A.k.a. *full batch gradient*.)
  - Pick term J uniformly at random:

$$\nabla \ell(y_J x_J^{\mathsf{T}} w^{(t)}).$$

  - What is expected value of this random vector?

Stochastic gradient method (2)

- Minibatch
  - To reduce variance of estimate, use several random examples $J_1, \dots, J_B$ and average—called *minibatch gradient*.

$$\frac{1}{B}\sum_{b=1}^{B} \nabla \ell(y_{J_b} x_{J_b}^{\mathsf{T}} w^{(t)}).$$

  - Rule of thumb: larger batch size $B \to$ larger step size η.
- Alternative: instead of picking example uniformly at random, shuffle order of training examples, and take next example in this order.
  - Verify that expected value is same!
  - Seems to reduce variance as well, but not fully understood.

- Logistic regression MLE for data $(x_1, y_1), \ldots, (x_n, y_n) \in \mathbb{R}^d \times \{-1, +1\}$.
- Start with $w^{(0)} \in \mathbb{R}^d$, $\eta > 0$, $t = 1$
- For epoch $p = 1, 2, \ldots$:
  - For each training example (x, y) in a random order:

$$w^{(t)} := w^{(t-1)} + \eta(1 - \sigma(y x^\top w^{(t-1)})) y x$$

$$t := t + 1.$$

Optimization for linear regression

- Back to considering ordinary least squares.
- Gaussian elimination to solve normal equations can be slow when d is large (time is $\mathcal{O}(nd^2)$).
- Alternative: find approximate solution using gradient descent
- Algorithm: start with some $w^{(0)} \in \mathbb{R}^d$ and $\eta > 0$.
  - For $t = 1, 2, \ldots$:

$$w^{(t)} := w^{(t-1)} - 2\eta A^{\mathsf{T}}(Aw^{(t-1)} - b)$$

  - Time to multiply matrix by vector is linear in matrix size.
  - So each iteration takes time $\mathcal{O}(nd)$.
- Can describe behavior of gradient descent for least squares (empirical risk) objective very precisely.

- **Theorem**: Let $\hat{w}$ be the minimum Euclidean norm solution to normal equations. Assume $w^{(0)} = 0$. Write eigendecomposition $A^{\mathsf{T}}A = \sum_{i=1}^{r} \lambda_i v_i v_i^{\mathsf{T}}$ with $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_r > 0$. Then $w^{(t)} \in \operatorname{range}(A^{\mathsf{T}})$ and

$$v_i^{\mathsf{T}} w^{(t)} = \left(2\eta\lambda_i \sum_{k=0}^{t-1} (1 - 2\eta\lambda_i)^k\right) v_i^{\mathsf{T}} \hat{w}, \quad i = 1, \ldots, r.$$

- Implications:
  - If we choose η such that $2\eta\lambda_i < 1$, then

$$2\eta\lambda_i \sum_{k=0}^{t-1} (1 - 2\eta\lambda_i)^k = 1 - (1 - 2\eta\lambda_i)^t,$$

    which converges to 1 as $t \to \infty$.
  - So, when $2\eta\lambda_1 < 1$, we have $w^{(t)} \to \hat{w}$ as $t \to \infty$.
  - *Rate of convergence* is geometric, i.e., "exponentially fast convergence".
  - Algorithmic inductive bias!

Postscript

- There are many optimization algorithms for convex optimization
  - Gradient descent, Newton's method, BFGS, coordinate descent, mirror descent, etc.
  - Stochastic variants thereof
- Many also usable even when objective function is non-convex
  - Typically just converge to a local minimizer or stationary point
- Can also handle constraints on the optimization variable
  - E.g., want coordinates of w to lie in a specific range
- The algorithmic inductive bias not always well-understood, but it is there!