Machine learning lecture slides

Optimization I: Convex optimization

Outline

- Convex sets and convex functions
- Local minimizers and global minimizers
- Gradient descent
- Analysis for smooth objective functions
-
-

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

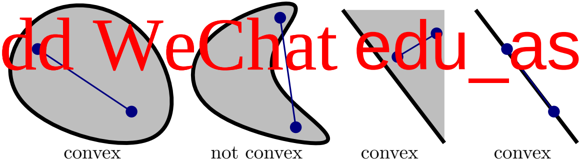


Figure 1: Which of these sets are convex?

Convex functions (1)

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

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

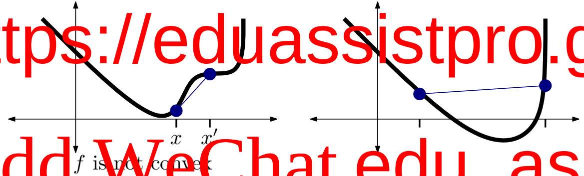


Figure 2: Which of these functions are c

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 $b \quad \mathbb{R}^d$

    A

    0

    $\mapsto \quad \{ \quad \}$
  - $f(w) = \operatorname{logsumexp}(w) = \ln \Big($
  - $w \mapsto f(g(w))$ for convex function

Verifying convexity of Euclidean norm

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

Convexity of differentiable functions (1)

- Differentiable function f is convex iff

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

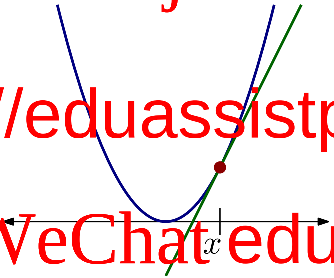


Figure 3: Affine approximation

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

Convexity of differentiable functions (2)

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

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

- 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 $w^\star$.

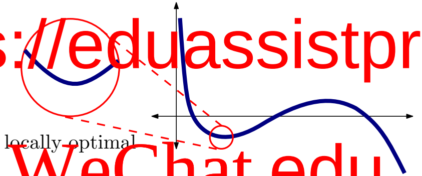


Figure 4: Local minimiz

- 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

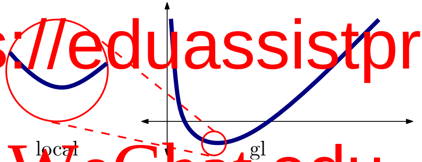


Figure 5: Local-to-global phe

Gradient descent

- Consider (unconstrained) convex optimization problem

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

-

-

$$w^{(t)} := w^{(t-1)}$$

- (Lots of things unspecified here . . .)

Motivation for gradient descent

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

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

-
-

$$\nabla f \quad w \qquad -\eta \nabla f \quad w \qquad \eta \qquad f \quad w \qquad <$$

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

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

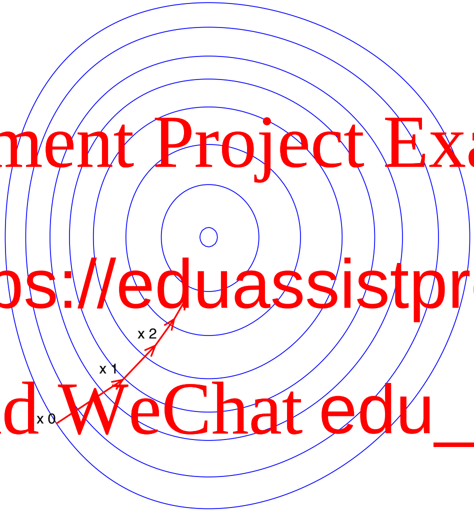


Figure 6: Trajectory of gradient descent

Example: Gradient of logistic loss

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

$$-\nabla\{\ell_{\text{logistic}}(y_i x_i^{\top} w)\} = -\ell'_{\text{logistic}}(y_i x_i^{\top} w) y_i x_i$$

$$y_i x_i$$

where σ is the sigmoid function.

- Recall, $\Pr_w(Y = y \mid X = x) = \sigma($ ing the logistic regression model.

- Objective function:

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

- *tep size*

-

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

$$= w^{(t-1)} + \eta \frac{1}{n} \sum_{i=1}^{n} (1$$

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

- **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 $\eta_t = 1/\beta$ satisfies

$$\frac{\beta \ w^{(0)} \quad w^{\star} \ ^2}{}$$

$\nabla f \ w$ w

$\|\nabla f(w) - \nabla f(w')$

- Note: it is possible to have convergence ev in some cases; should really treat η as a hyperparameter.

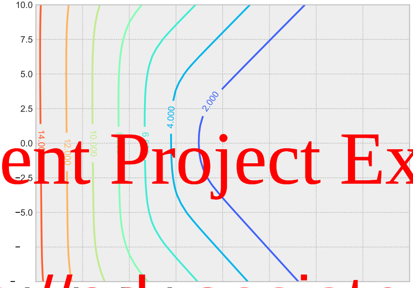
- Empirical risk with squared loss

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

So objective function is β-smooth with $\beta = \lambda \quad (A^\top A)$.

- Empirical risk with logistic loss

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


10.0
7.5
5.0
2.5
0.0
−2.5
−5.0
14.0
12.000
10.000
4.000
2.000

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

$$f(w+\delta) \leq f(w) + \nabla f(w)^\top \delta + \frac{\beta}{2}\|\delta\|_2^2.$$

-

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

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

- Plug-in this value of δ into above ine

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

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

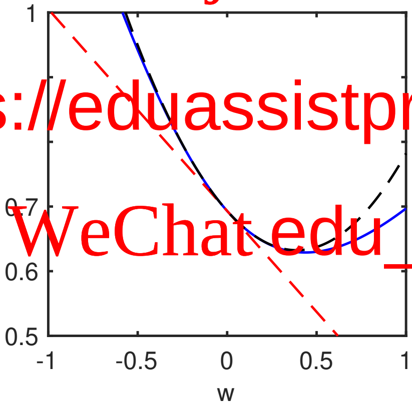


Figure 8: Linear and quadratic approximations to a convex function

Example: Text classification (1)

- Data: articles posted to various internet message boards
- Label: -1 for articles from “religion”, $+1$ for articles from “politics”
- Features:
- tions

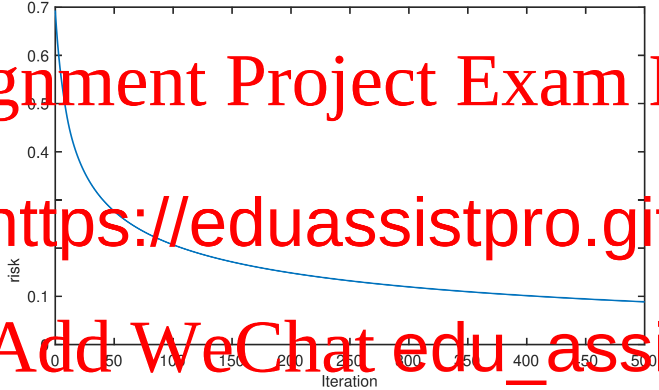


Figure 9: Objective value as a function of number of gradient descent iterations

Example: Text classification (3)

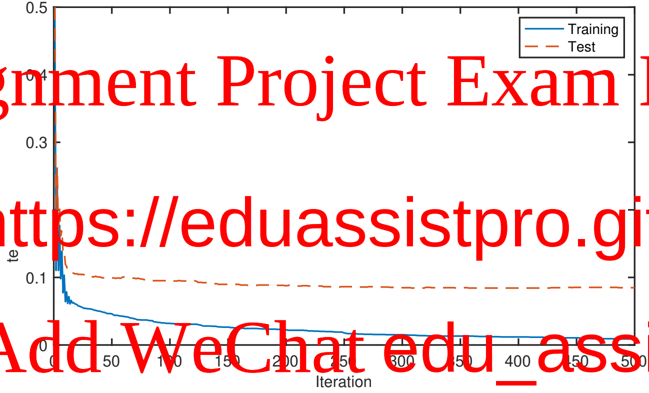


Figure 10: Error rate as a function of number of gradient descent iterations

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)*

$$\frac{1}{n}\sum_{j=1} \nabla \ell(\quad {}^{\top} \quad {}^{(t)}$$

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

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

  - What is expected value of this random vector?

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

$$\frac{1}{B} \sum^{B}$$

η.

- Alternative: instead of picking example un
shuffle order of training examples, and take
this order.

  - Verify that expected value is same!
  - Seems to reduce variance as well, but not fully understood.

Example: SGD for logistic regression

- 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 $p = 1, 2, \ldots$:

- 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
- (0) d
  - Time to multiply matrix by vector is linear i
  - So each iteration takes time $\mathcal{O}($
- Can describe behavior of gradient descent f (empirical risk) objective very precisely.

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

-
  - If we choose η such that $2\eta\lambda_i$

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

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!

- There are many optimization algorithms for convex optimization
  - Gradient descent, Newton's method, BFGS, coordinate descent, mirror descent, etc.
  - Stochastic variants thereof
-
-
  - E.g., want coordinates of w to lie i
- The algorithmic inductive bias not always
  it is there!