CS 446/ECE 449: Machine Learning

A. G. Schwing

University of Illinois at Urbana-Champaign, 2020

Scribe & Exercises

L4: Optimization Primal

Goals of this lecture

- Understanding the basics of optimization

Reading Material

- S. Boyd and L. Vandenberghe; Convex Optimization; Chapters 2-4

Optimization problems that we have seen so far:

- Linear Regression

$$\min_{\boldsymbol{w}} \frac{1}{2} \sum_{(x^{(i)},y^{(i)})\in\mathcal{D}} \left(y^{(i)} - \phi(x^{(i)})^\top \boldsymbol{w}\right)^2$$

- Logistic Regression

$$\min_{\boldsymbol{w}} \sum_{(x^{(i)},y^{(i)})\in\mathcal{D}} \log\left(1 + \exp(-y^{(i)}\boldsymbol{w}^T\phi(x^{(i)}))\right)$$

Finding optimum:

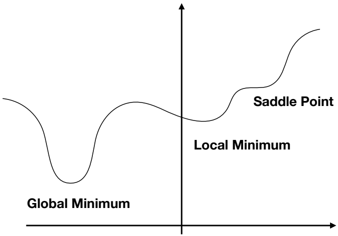
Analytically computable optimum vs. gradient descent

The Problem more generally:

$$\begin{array}{ll} \min_{\boldsymbol{w}} & f_0(\boldsymbol{w}) \\ \text{s.t.} & f_i(\boldsymbol{w}) \leq 0 \quad \forall i \in \{1, \ldots, C\} \end{array}$$

Solution:

Solution $\boldsymbol{w}^*$ has smallest value $f_0(\boldsymbol{w}^*)$ among all values that satisfy constraints



Questions:

- When can we find the optimum?
- Algorithms to search for the optimum?
- How long does it take to find the optimum?

When can we find the optimum?

When can we find the optimum?

- Least squares, linear and convex programs can be solved efficiently and reliably
- General optimization problems are very difficult to solve
- Often compromise between accuracy and computation time

What's the form of 'least squares,' 'linear,' and 'convex' programs?

- Least squares program

$$\min_{\boldsymbol{w}} \frac{1}{2} \sum_{(x^{(i)},y^{(i)})\in\mathcal{D}} \left(y^{(i)} - \phi(x^{(i)})^\top \boldsymbol{w}\right)^2$$

- Linear program

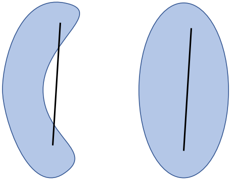
$$\min_{\boldsymbol{w}} \boldsymbol{c}^\top \boldsymbol{w} \quad \text{s.t.} \quad \boldsymbol{A}\boldsymbol{w} \leq \boldsymbol{b}$$

- Convex program when all f_i convex (generalizes the above)

$$\min_{\boldsymbol{w}} f_0(\boldsymbol{w}) \quad \text{s.t.} \quad f_i(\boldsymbol{w}) \leq 0 \quad \forall i \in \{1, \ldots, C\}$$

Convex set:

A set is convex if for any two points $\boldsymbol{w}_1$, $\boldsymbol{w}_2$ in the set, the line segment $\lambda\boldsymbol{w}_1 + (1-\lambda)\boldsymbol{w}_2$ for $\lambda \in [0,1]$ also lies in the set.



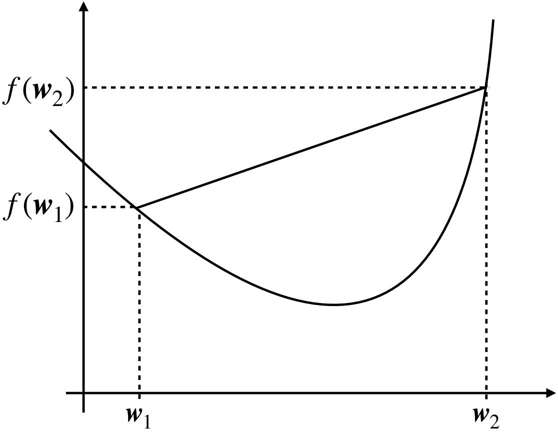
Example: Polyhedron

$$\{\boldsymbol{w} | \boldsymbol{A}\boldsymbol{w} \leq \boldsymbol{b}, \boldsymbol{C}\boldsymbol{w} = \boldsymbol{d}\}$$

Convex function

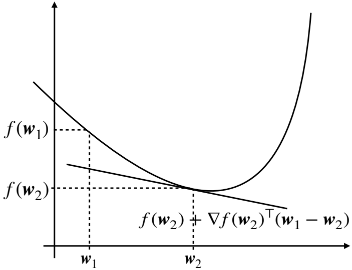
A function f is convex if its domain is a convex set and for any points $\boldsymbol{w}_1$, $\boldsymbol{w}_2$ in the domain and any $\lambda \in [0, 1]$

$$f((1-\lambda)\boldsymbol{w}_1 + \lambda \boldsymbol{w}_2) \leq (1-\lambda) f(\boldsymbol{w}_1) + \lambda f(\boldsymbol{w}_2)$$



Recognizing convex functions

- If f is differentiable, then f is convex if and only if its domain is convex and $f(\boldsymbol{w}_1) \geq f(\boldsymbol{w}_2) + \nabla f(\boldsymbol{w}_2)^\top(\boldsymbol{w}_1 - \boldsymbol{w}_2)\ \forall \boldsymbol{w}_1, \boldsymbol{w}_2$ in the domain



- If f is differentiable, then f is convex if and only if its domain is convex and $\forall \boldsymbol{w}_1, \boldsymbol{w}_2$ in the domain

$$(\nabla f(\boldsymbol{w}_1) - \nabla f(\boldsymbol{w}_2))^\top(\boldsymbol{w}_1 - \boldsymbol{w}_2) \geq 0 \qquad \text{monotone mapping}$$

- If f is twice differentiable, then f is convex if and only if its domain is convex and $\nabla^2 f(\boldsymbol{w}) \succeq 0\ \forall \boldsymbol{w}$ in the domain

Examples of convex functions

- Exponential: $\exp(ax)$ convex on $x \in \mathbb{R}\ \forall a \in \mathbb{R}$
- Negative Logarithm: $-\log(x)$ is convex on $x \in \mathbb{R}_{++}$
- Negative Entropy: $-H(x) = x\log(x)$ is convex on $x \in \mathbb{R}_{++}$
- Norms: $\|\boldsymbol{w}\|_p$ for $p \geq 1$
- Log-Sum-Exp: $\log(\exp(w_1) + \ldots + \exp(w_d))$

Operations which preserve convexity

- Non-negative weighted sums: $\alpha_i \geq 0$; if f_i convex $\forall i$, so is

$$g = \alpha_1 f_1 + \alpha_2 f_2 + \ldots$$

- Composition with an affine mapping: if f is convex, so is

$$g(\boldsymbol{w}) = f(\boldsymbol{A}\boldsymbol{w} + \boldsymbol{b})$$

- Pointwise maximum: if f_1, f_2 are convex, so is

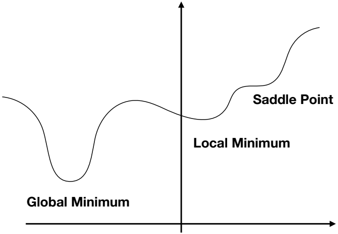
$$g(\boldsymbol{w}) = \max\{f_1(\boldsymbol{w}), f_2(\boldsymbol{w})\}$$

log(exp(x)+exp(x2))
convex
after mapping
Aw+b

Show that $\log(1 + \exp(x))$ is convex for $x \in \mathbb{R}$

Optimality of convex optimization

- A point $\boldsymbol{w}^*$ is locally optimal if $f(\boldsymbol{w}^*) \leq f(\boldsymbol{w})\ \forall \boldsymbol{w}$ in a neighborhood of $\boldsymbol{w}^*$; globally optimal if $f(\boldsymbol{w}^*) \leq f(\boldsymbol{w})\ \forall \boldsymbol{w}$



For convex problems global optimality follows directly from local optimality.

- For a local minimum of f, $\nabla f(\boldsymbol{w}^*) = 0$
- If f convex, then $\nabla f(\boldsymbol{w}^*) = 0$ sufficient for global optimality

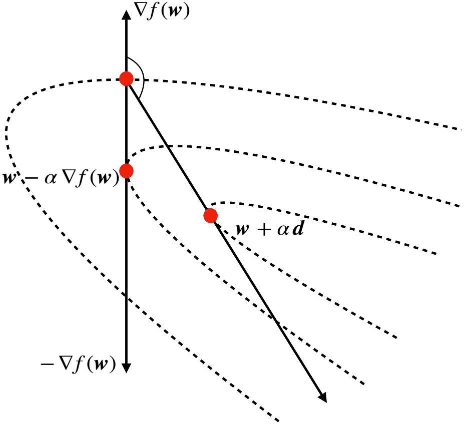
This makes convex optimization special!

Algorithms to search for the optimum?

Descent methods

$$\min_{\boldsymbol{w}} f(\boldsymbol{w})$$

Intuition (find a stationary point with $\nabla f(\boldsymbol{w}) = 0$)



Iterative algorithm

- Start with some guess $\boldsymbol{w}$
- Iterate k = 1, 2, 3, ...
  - Select direction $\boldsymbol{d}_k$ and stepsize α_k
  - $\boldsymbol{w} \leftarrow \boldsymbol{w} + \alpha_k \boldsymbol{d}_k$
  - Check whether we should stop (e.g., if $\nabla f(\boldsymbol{w}) \approx 0$)

Descent direction d_k satisfies $\nabla f(\boldsymbol{w})^\top \boldsymbol{d}_k < 0$

How to select direction:

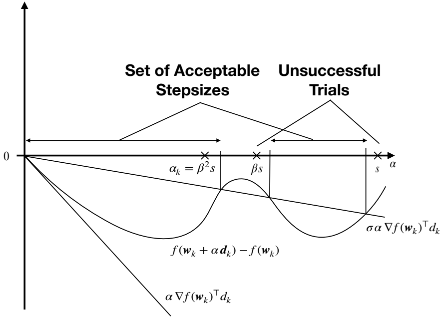
- Steepest descent: $\boldsymbol{d}_k = -\nabla f(\boldsymbol{w}_k)$
- Scaled gradient: $\boldsymbol{d}_k = -\boldsymbol{D}_k \nabla f(\boldsymbol{w}_k)$ for $\boldsymbol{D}_k \succ 0$
  - E.g., Newton's method: $\boldsymbol{D}_k = [\nabla^2 f(\boldsymbol{w}_k)]^{-1}$
- ...

How to select stepsize:

- Exact: $\alpha_k = \arg\min_{\alpha \geq 0} f(\boldsymbol{w}_k + \alpha \boldsymbol{d}_k)$
- Constant: $\alpha_k = 1/L$ (for suitable L)
- Diminishing: $\alpha_k \to 0$ but $\sum_k \alpha_k = \infty$ (e.g., $\alpha_k = 1/k$)
- Armijo Rule:

Start with $\alpha = s$ and continue with $\alpha = \beta s$, $\alpha = \beta^2 s, \ldots$, until $\alpha = \beta^m s$ falls within the set of α with

$$f(\boldsymbol{w}_k + \alpha \boldsymbol{d}_k) - f(\boldsymbol{w}_k) \leq \sigma \alpha \nabla f(\boldsymbol{w}_k)^\top \boldsymbol{d}_k$$



How long does it take to find the optimum?

Goal:
How many iterations k for

$$f(\boldsymbol{w}_k) - f(\boldsymbol{w}^*) \leq \epsilon$$

Two important properties:

- Lipschitz continuous gradient
- Strong convexity

Properties: Lipschitz continuous gradient

$$\|\nabla f(\boldsymbol{w}_1) - \nabla f(\boldsymbol{w}_2)\|_2 \leq L\|\boldsymbol{w}_1 - \boldsymbol{w}_2\|_2 \qquad \forall \boldsymbol{w}_1, \boldsymbol{w}_2$$

Intuition:

Lipschitz continuous gradient, then $g(\boldsymbol{w}) = \frac{L}{2}\|\boldsymbol{w}\|_2^2 - f(\boldsymbol{w})$ convex

Proof:

$$\begin{aligned}
&(\nabla f(\boldsymbol{w}_1) - \nabla f(\boldsymbol{w}_2))^\top(\boldsymbol{w}_1 - \boldsymbol{w}_2) \\
&\quad \leq \|\nabla f(\boldsymbol{w}_1) - \nabla f(\boldsymbol{w}_2)\|_2\|\boldsymbol{w}_1 - \boldsymbol{w}_2\|_2 \quad \text{Cauchy-Schwartz} \\
&\quad \leq L\|\boldsymbol{w}_1 - \boldsymbol{w}_2\|_2^2
\end{aligned}$$

$$\nabla g(\boldsymbol{w}_1) - \nabla g(\boldsymbol{w}_2) = L(\boldsymbol{w}_1 - \boldsymbol{w}_2) - (\nabla f(\boldsymbol{w}_1) - \nabla f(\boldsymbol{w}_2))$$

$$\begin{aligned}
&(\nabla g(\boldsymbol{w}_1) - \nabla g(\boldsymbol{w}_2))^\top(\boldsymbol{w}_1 - \boldsymbol{w}_2) \\
&\quad = L\|\boldsymbol{w}_1 - \boldsymbol{w}_2\|_2^2 - (\nabla f(\boldsymbol{w}_1) - \nabla f(\boldsymbol{w}_2))^\top(\boldsymbol{w}_1 - \boldsymbol{w}_2) \\
&\quad \geq 0 \qquad \text{monotone mapping}
\end{aligned}$$

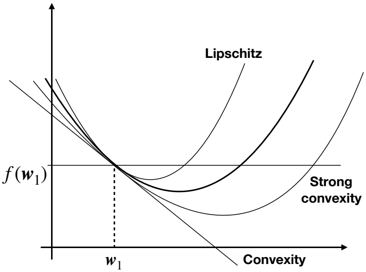
If $g(\boldsymbol{w}) = \frac{L}{2}\|\boldsymbol{w}\|_2^2 - f(\boldsymbol{w})$ convex, then

$$f(\boldsymbol{w}_2) \leq f(\boldsymbol{w}_1) + \nabla f(\boldsymbol{w}_1)^\top(\boldsymbol{w}_2 - \boldsymbol{w}_1) + \frac{L}{2}\|\boldsymbol{w}_2 - \boldsymbol{w}_1\|_2^2 \qquad \forall \boldsymbol{w}_1, \boldsymbol{w}_2$$

Proof: plug definition of g into $g(\boldsymbol{w}_2) \geq g(\boldsymbol{w}_1) + \nabla g(\boldsymbol{w}_1)^\top(\boldsymbol{w}_2 - \boldsymbol{w}_1)$ and re-arrange

Properties: Strong convexity

$$f(\boldsymbol{w}_2) \geq f(\boldsymbol{w}_1) + \nabla f(\boldsymbol{w}_1)^\top(\boldsymbol{w}_2 - \boldsymbol{w}_1) + \frac{\sigma}{2}\|\boldsymbol{w}_2 - \boldsymbol{w}_1\|_2^2 \qquad \forall \boldsymbol{w}_1, \boldsymbol{w}_2$$



if f twice differentiable

$$\sigma I \prec \nabla^2 f(\boldsymbol{w}) \prec LI \qquad \forall \boldsymbol{w}$$

How many iterations k such that

$$f(\boldsymbol{w}_k) - f(\boldsymbol{w}^*) \leq \epsilon \qquad \text{for} \quad \boldsymbol{w}_{k+1} = \boldsymbol{w}_k + \alpha \boldsymbol{d}_k$$

How to pick $\alpha \boldsymbol{d}_k$? Minimize w.r.t. $\boldsymbol{w}_{k+1}$ right-hand-side of upper bound

$$f(\boldsymbol{w}_{k+1}) \leq f(\boldsymbol{w}_k) + \nabla f(\boldsymbol{w}_k)^\top (\boldsymbol{w}_{k+1} - \boldsymbol{w}_k) + \frac{L}{2}\|\boldsymbol{w}_{k+1} - \boldsymbol{w}_k\|_2^2$$

Hence

$$\begin{array}{rcl} \alpha \boldsymbol{d}_k & = & -\frac{1}{L}\nabla f(\boldsymbol{w}_k) \\ f(\boldsymbol{w}_{k+1}) & \leq & f(\boldsymbol{w}_k) - \frac{1}{2L}\|\nabla f(\boldsymbol{w}_k)\|_2^2 \quad \text{Bound on guaranteed progress} \end{array}$$

Bound on sub-optimality from strong convexity:

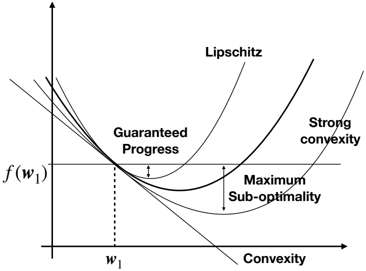
$$f(\boldsymbol{w}^*) \geq f(\boldsymbol{w}_k) - \frac{1}{2\sigma}\|\nabla f(\boldsymbol{w}_k)\|_2^2$$

Guaranteed progress:

$$f(\boldsymbol{w}_{k+1}) \leq f(\boldsymbol{w}_k) - \frac{1}{2L}\|\nabla f(\boldsymbol{w}_k)\|_2^2$$

Maximum sub-optimality:

$$f(\boldsymbol{w}^*) \geq f(\boldsymbol{w}_k) - \frac{1}{2\sigma}\|\nabla f(\boldsymbol{w}_k)\|_2^2$$



Distance to go:

$$f(\boldsymbol{w}_k) - f(\boldsymbol{w}^*) \leq \frac{1}{2\sigma}\|\nabla f(\boldsymbol{w}_k)\|_2^2$$

in 'guaranteed progress':

$$\begin{aligned} f(\boldsymbol{w}_k) - f(\boldsymbol{w}^*) &\leq f(\boldsymbol{w}_{k-1}) - f(\boldsymbol{w}^*) - \tfrac{\sigma}{L}(f(\boldsymbol{w}_{k-1}) - f(\boldsymbol{w}^*)) \\ &\leq \left(1 - \tfrac{\sigma}{L}\right)^k (f(\boldsymbol{w}_0) - f(\boldsymbol{w}^*)) \qquad (\sigma < L) \end{aligned}$$

Rate:

$$c\left(1 - \frac{\sigma}{L}\right)^k \leq \epsilon \quad \Longrightarrow \quad k \geq O(\log(1/\epsilon)) \quad (\text{sometimes } O(e^k))$$

No strong convexity assumption:
Lipschitz bound and $\boldsymbol{w}_2 = \boldsymbol{w}_1 - \alpha\nabla f(\boldsymbol{w}_1)$ yields

$$f(\boldsymbol{w}_2) \leq f(\boldsymbol{w}_1) - (1 - \frac{L\alpha}{2})\alpha\|\nabla f(\boldsymbol{w}_1)\|_2^2$$

Combined with convexity: $f(\boldsymbol{w}_1) + \nabla f(\boldsymbol{w}_1)^\top(\boldsymbol{w}^* - \boldsymbol{w}_1) \leq f(\boldsymbol{w}^*)$

$$f(\boldsymbol{w}_2) \leq f(\boldsymbol{w}^*) + \nabla f(\boldsymbol{w}_1)(\boldsymbol{w}_1 - \boldsymbol{w}^*) - \frac{\alpha}{2}\|\nabla f(\boldsymbol{w}_1)\|_2^2$$

Using $\boldsymbol{w}_2 - \boldsymbol{w}_1 = -\alpha\nabla f(\boldsymbol{w}_1)$ and rearranging terms gives

$$f(\boldsymbol{w}_2) \leq f(\boldsymbol{w}^*) + \frac{1}{2\alpha}\left(\|\boldsymbol{w}_1 - \boldsymbol{w}^*\|_2^2 - \|\boldsymbol{w}_2 - \boldsymbol{w}^*\|_2^2\right)$$

Summing over all iterations

$$\sum_{i=1}^{k}(f(\boldsymbol{w}_i) - f(\boldsymbol{w}^*)) \leq \frac{1}{2\alpha}\|\boldsymbol{w}_0 - \boldsymbol{w}^*\|_2^2$$

$f(\boldsymbol{w}_i)$ non-increasing:

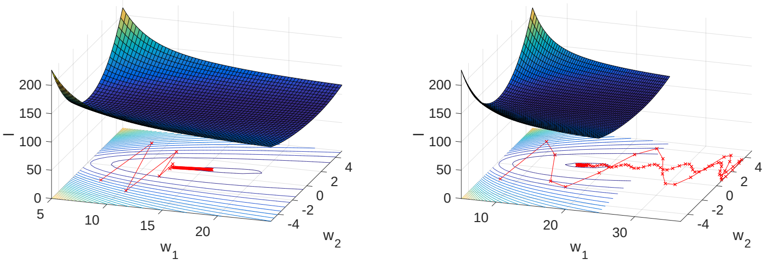
$$f(\boldsymbol{w}_k) - f(\boldsymbol{w}^*) \leq \frac{1}{k}\sum_{i=1}^{k}(f(\boldsymbol{w}_i) - f(\boldsymbol{w}^*)) \leq \frac{1}{2k\alpha}\|\boldsymbol{w}_0 - \boldsymbol{w}^*\|_2^2 \leq \epsilon$$

Consequently:

$$k \geq O(1/\epsilon)$$

Are these rates optimal?

Gradient with momentum
Intuition:



Video

- Polyak's method (aka heavy-ball)

$$\boldsymbol{w}_{k+1} = \boldsymbol{w}_k - \alpha_k \nabla f(\boldsymbol{w}_k) + \beta_k(\boldsymbol{w}_k - \boldsymbol{w}_{k-1})$$

- Momentum method in deep learning

$$\begin{aligned} \boldsymbol{v}_{k+1} &= \beta \boldsymbol{v}_k + \nabla f(\boldsymbol{w}_k) \\ \boldsymbol{w}_{k+1} &= \boldsymbol{w}_k - \alpha \boldsymbol{v}_{k+1} \end{aligned}$$

Recall the structure of our optimization problems:

$$\min_{w} \sum_{(x^{(i)},y^{(i)})\in\mathcal{D}} \ell(y_i, F(x^{(i)}, \boldsymbol{w}))$$

- So far we didn't consider the time for computing the gradient
- Iteration complexity is linear in the number of samples $|\mathcal{D}|$
- A large dataset makes gradient computation slow

How to deal with this?

Stochastic gradient descent

Consider a subset of samples and approximate the gradient based on this batch of data.

- Select a subset of samples $\mathcal{B}_k$
- Gradient update using approximation

$$\nabla f(\boldsymbol{w}) \approx \sum_{(x^{(i)},y^{(i)})\in\mathcal{B}_k} \nabla\ell(y^{(i)}, F(x^{(i)}, \boldsymbol{w}))$$

Convergence rates for stochastic gradient descent:

- Lipschitz continuous gradient and strongly convex: $k \geq O(1/\epsilon)$
- Lipschitz continuous gradient: $k \geq O(1/\epsilon^2)$

Stochastic vs. deterministic (strongly convex)

Batch gradient descent:

- Convergence rate: $O(\log 1/\epsilon)$
- Iteration complexity: linear in $|\mathcal{D}|$

Stochastic gradient descent:

- Convergence rate: $O(1/\epsilon)$
- Iteration complexity: independent of $|\mathcal{D}|$

Can we get the best of both worlds?

Many related algorithms:

- SAG (Le Roux, Schmidt, Bach 2012)
- SDCA (Shalev-Shwartz and Zhang 2013)
- SVRG (Johnnson and Zhang 2013)
- MISO (Mairal 2015)
- Finito (Defazio 2014)
- SAGA (Defazio, Bach, Lacoste-Julien 2014)
- ...

Idea: variance reduction

Example: SVRG

- Initialize $\hat{\boldsymbol{w}}$
- For epoch 1, 2, 3, ...
  - Compute $\nabla f(\hat{\boldsymbol{w}}) = \sum_{i \in \mathcal{D}} \nabla \ell_i(\hat{\boldsymbol{w}})$
  - Initialize $\boldsymbol{w}_0 = \hat{\boldsymbol{w}}$
  - For t in length of epochs

$$\boldsymbol{w}_t = \boldsymbol{w}_{t-1} - \alpha \left[\nabla f(\hat{\boldsymbol{w}}) + \nabla \ell_{i(t)}(\boldsymbol{w}_{t-1}) - \nabla \ell_{i(t)}(\hat{\boldsymbol{w}}) \right]$$

  - Update $\hat{\boldsymbol{w}} = \boldsymbol{w}_t$
- Output $\hat{\boldsymbol{w}}$

Quiz:

- Stepsize/Learning rate rules?
- Descent directions?
- Properties of convex functions?
- Convergence rates?
- Improvements?

Important topics of this lecture

- Convex optimization basics
- Algorithm choices
- Rates

Up next:

- How to deal with constraints