CSCI699: Theory of Machine Learning Fall 2021

Lecture 15: Convex Surrogates & Convex Optimization Algorithms

Instructor: Vatsal Sharan *Scribe: Bhavya Vasudeva*

Recap:

We started with how much data is needed for learning, then went on to the computational aspects, saw algorithms for learning problems efficiently. We also saw that learning can be computationally hard, sometimes it helps to do improper learning. We then studied the RCN framework and SQ model to understand what we can learn efficiently there. We've continued to look at such algorithms, like boosting, which allows us to go from a weak learning guarantee to a strong learning guarantee. The study of convex optimization, which we started the last time continues along this trend, where we look at problems for which efficient algorithms exist, what problems such algorithms can solve, and what algorithmic guarantees can be obtained.

1 Convex Optimization:

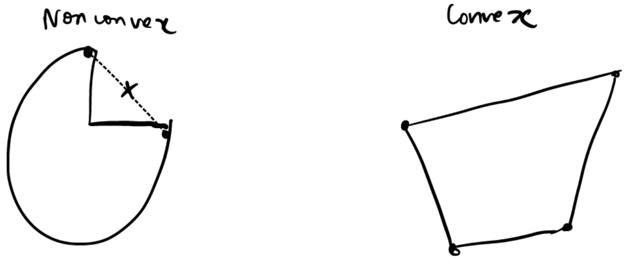


Figure 1: Non-convex and convex sets.

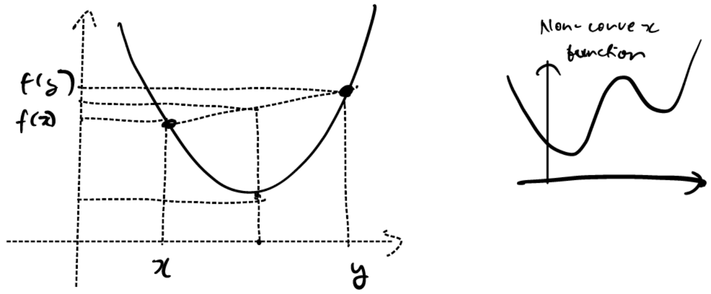


Figure 2: Convex and non-convex functions.

Lemma 1. *If f is differentiable, then f is convex if and only if domain(f) is convex and:*

$$f(y) - f(x) \geq \langle y - x, \nabla f(x) \rangle$$

$$f(y) - f(x) \leq \langle y - x, \nabla f(y) \rangle$$

- Convex optimization problems can be solved efficiently:

They are formulated as:

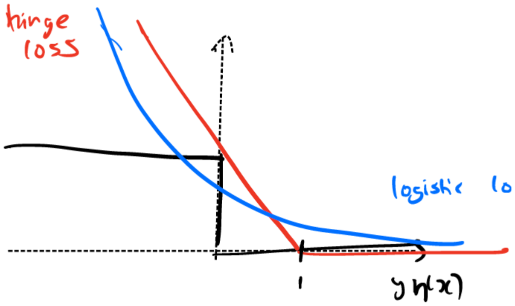
$$\min_{x\in A} f(x),$$

where 1) f is convex, 2) A is convex.
The optimization problem we want to solve is finding the ERM. Let $\mathcal{H} \subseteq \mathbb{R}^d$ parametrized by $w \in \mathcal{H}$.

$$\mathrm{ERM}_{\mathcal{H}}(\mathcal{S}) = \arg\min_{w\in\mathcal{H}} \frac{1}{n}\sum_{i=1}^{n} \ell(w, z_i).$$

Lemma 2. *If ℓ is a convex loss (in terms of w), and $\mathcal{H}$ is convex, then $ERM_{\mathcal{H}}(\mathcal{S})$ is a convex optimization problem.*

There are many losses which are non-convex. A common technique to handle a non-convex loss is to instead consider a convex surrogate. The figure below shows convex surrogates for the 0/1 loss.



$$\text{Hinge loss:}\quad \ell(h,(x,y)) = \max(1 - yh(x), 0)$$
$$\text{Logistic loss:}\quad \ell(h,(x,y)) = \log(1 + e^{-yh(x)})$$

Today:

- More on convex surrogates
- Algorithms for convex optimization

2 Convex Surrogates

Let $\mathcal{R}(h) = \mathbb{E}_{(x,y)\sim\mathcal{D}}\mathbb{1}(yh(x) \le 0)$ be the 0/1 risk. Let $\phi : \mathbb{R} \to \mathbb{R}$ be some other function (used as a surrogate) and define $\mathcal{R}_\phi(h) = \mathbb{E}_{(x,y)\sim\mathcal{D}}\phi(yh(x))$ to be the surrogate risk (plugging in ϕ in place of $\mathbb{1}$ in $\mathcal{R}$).

Functions which depend on $yh(x)$ can be used as ϕ. All the functions mentioned earlier (hinge loss, logistic loss) fit this description.

Recall that $\mathcal{R}^* = \inf_{f:\mathcal{X}\to\mathcal{Y}} \mathcal{R}(f)$ is the Bayes optimal risk.

Let $\mathcal{R}^*_\phi = \inf_{f:\mathcal{X}\to\mathcal{Y}} \mathcal{R}_\phi(f)$ denote the Bayes optimal ϕ-risk.

Definition 3. *We say surrogate loss ϕ is classification calibrated if for any sequence of functions f_i and every distribution $\mathcal{D}$ over (x, y),*

$$\mathcal{R}_\phi(f_i) \to \mathcal{R}_\phi^* \implies \mathcal{R}(f_i) \to \mathcal{R}^*$$

This says that if you can find a hypothesis which is Bayes optimal according to ϕ-risk, then it will also be optimal according to the 0/1 loss, *i.e.* the surrogate is a good surrogate.

Note. We want $\mathcal{R}_\phi(f_i) \to \mathcal{R}_\phi^*$ (Bayes optimal predictor for ϕ-risk). If we optimize our risk over some limited hypothesis class which doesn't include the Bayes optimal predictor for ϕ-risk, the above definition is not meaningful. It is applicable in the realizable setup, where our hypothesis class contains the target function.

Theorem 4 (Bartlett, Jordan, McAuliffe'06)**.** *Consider a surrogate loss $\phi(yh(x))$. If ϕ is convex, then it is classification calibrated if and only if $\phi'(0)$ exists and $\phi'(0) < 0$ (the derivatives are taken w.r.t. $yh(x)$).*

Corollary 5. *Hinge loss and logistic loss are classification calibrated.*

It is easy to verify this by taking their derivatives.

Note. Squared loss $l(h; (x, y)) = (h(x) - y)^2$ doesn't fit the form of $\phi(yh(x)$, but we can show that it is also classification calibrated.

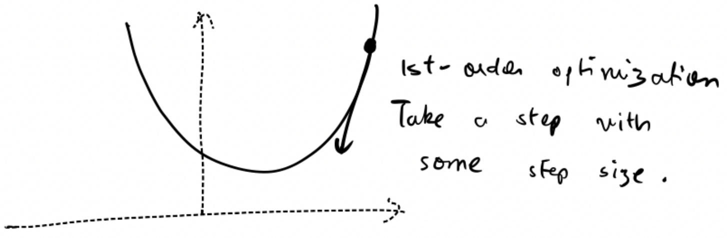
3 Gradient Descent (GD)

Consider an unconstrained optimization problem: $\min_{x \in \mathbb{R}^d} f(x)$, where $f(x)$ is differentiable.

Algorithm 1: GD

```
Initialize $w_1$
for t=$1,2,...,T$ do
    w$_{t+1} \leftarrow w_t - \eta \nabla f(w_t)$
```

The output has not been stated as it is not always the same. Commonly, w_T or $\frac{1}{T}\sum_{t=1}^{T} w_t$ is output.



1st-order optimization
Take a step with
some step size.

For any point, we are considering the first-order approximation of f and taking a step in that direction with step size η. The figure above shows this for a 1-D problem.

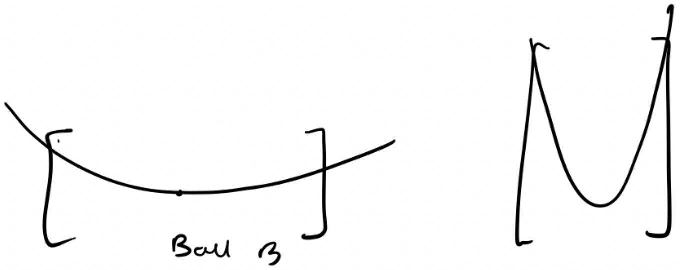
Theorem 6 (Convergence rate of GD)**.** *Let f be a convex ρ-Lipschitz function, i.e.*

$$|f(x) - f(y)| \leq \rho||x - y|| \;\forall\; x, y \in \mathbb{R}^d.$$

Let $w^* = \underset{w \in \mathbb{R}^d}{\arg\min} f(w)$, $||w^* - w_1|| \leq B$, where w_1 is the initialization for GD. Suppose we run GD for T steps with $\eta = \sqrt{\frac{B^2}{\rho^2 T}}$ and let $\bar{w} = \frac{1}{T}\sum_{t=1}^{T} w_t$. Then $\bar{w}$ satisfies $f(\bar{w}) - f(w^*) \leq \frac{B\rho}{\sqrt{T}}$.

If $T \geq \frac{B^2\rho^2}{\epsilon^2}$, $f(\bar{w}) - f(w^*) \leq \epsilon$. This convergence rate is without any restrictions.

Aside: T depends on both ρ and B. For a function which is flat, ρ is small. When we initialize within a ball of radius B, the value on any point in this ball is not too different from the best value, so we don't need to take many steps to get small error, hence T is small when ρ is small. If the function is steep (ρ is large) and we initialize in the same ball, we will need more steps to get small error, so T would be large. These cases are shown in the figure below.



Proof.

$$\begin{aligned} f(\bar{w}) - f(w^*) &= f\left(\frac{1}{T}\sum_{t=1}^{T} w_t\right) - f(w^*) \\ &\leq \frac{1}{T}\sum_{t=1}^{T} f(w_t) - f(w^*) \\ &= \frac{1}{T}\sum_{t=1}^{T}(f(w_t) - f(w^*)). \end{aligned} \tag{1}$$

Because f is convex, we have:

$$f(w_t) - f(w^*) \leq \langle w_t - w^*, \nabla f(w_t)\rangle. \tag{2}$$

Combining (1) and (2),

$$f(\bar{w}) - f(w^*) \leq \frac{1}{T}\sum_{t=1}^{T}\langle w_t - w^*, \nabla f(w_t)\rangle. \tag{3}$$

(This is the only step where we use convexity.) Now, we will upper bound $\sum_{t=1}^{T} \langle w_t - w^*, \nabla f(w_t) \rangle$.

We might use Cauchy-Schwarz to do this, however, that will give $TB\rho$, $\implies f(\bar{w}) - f(w^*) \leq B\rho$, which is a constant, it doesn't go down with iterations.

Lemma 7 (Iterative Update)**.** *Let $v_1, ..., v_T$ be an arbitrary sequence of vectors. Consider any algorithm with an update rule $w_{t+1} = w_t - \eta v_t$. Then,*

$$\sum_{t=1}^{T} \langle w_t - w^*, v_t \rangle \leq \frac{||w^* - w_1||^2}{2\eta} + \frac{\eta}{2} \sum_{t=1}^{T} ||v_t||^2.$$

If $||w^* - w_1|| \leq B$, $||v_t|| \leq \rho$, and $\eta = \sqrt{\frac{B^2}{\rho^2 T}}$,

$$\sum_{t=1}^{T} \langle w_t - w^*, v_t \rangle \leq B\rho\sqrt{T}.$$

Proof.

$$\begin{aligned}
\langle w_t - w^*, v_t \rangle &= \frac{1}{\eta} \langle w_t - w^*, \eta v_t \rangle \\
&= \frac{1}{2\eta} (-||w_t - w^* - \eta v_t||^2 + ||w_t - w^*||^2 + \eta^2 ||v_t||^2) \\
&= \frac{1}{2\eta} (-||w_{t+1} - w^*||^2 + ||w_t - w^*||^2) + \frac{\eta}{2} ||v_t||^2 \\
\sum_{t=1}^{T} \langle w_t - w^*, v_t \rangle &= \frac{1}{2\eta} \sum_{t=1}^{T} (-||w_{t+1} - w^*||^2 + ||w_t - w^*||^2) + \sum_{t=1}^{T} \frac{\eta}{2} ||v_t||^2 \\
&= \frac{1}{2\eta} (-||w_{T+1} - w^*||^2 + ||w_1 - w^*||^2) + \frac{\eta}{2} \sum_{t=1}^{T} ||v_t||^2 \qquad \text{(other terms cancel out due to telescopic sum)} \\
&\leq \frac{1}{2\eta} ||w_1 - w^*||^2 + \frac{\eta}{2} \sum_{t=1}^{T} ||v_t||^2 \\
&\leq \frac{B^2}{2\eta} + \frac{\eta}{2} T\rho^2 \\
\implies \frac{1}{T} \sum_{t=1}^{T} \langle w_t - w^*, v_t \rangle &\leq \frac{B^2}{2\eta T} + \frac{\eta}{2} \rho^2.
\end{aligned}$$

To minimize the bound, set both terms equal to get $\eta = \frac{B}{\rho\sqrt{T}}$, which gives:

$$\frac{1}{T} \sum_{t=1}^{T} \langle w_t - w^*, v_t \rangle \leq \frac{B\rho}{\sqrt{T}}.$$

□

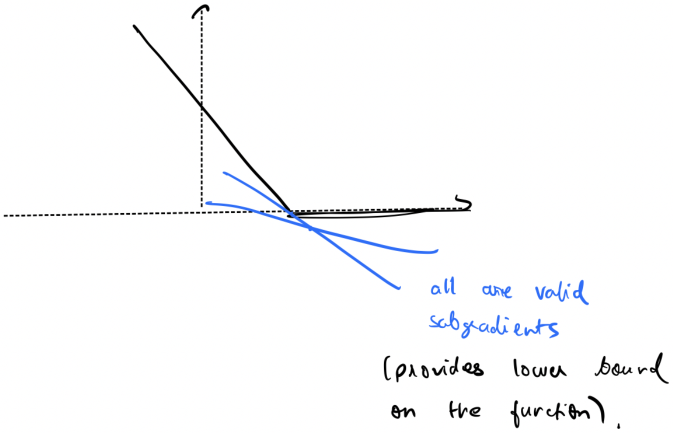
Claim 8. *If f is convex and ρ-Lipschitz, then $||\nabla f(x)|| \leq \rho \;\forall\; x$.*

Proof. By convexity, $f(x') - f(x) \geq \langle x' - x, \nabla f(x)\rangle$.
Take $x' = x + \epsilon \dfrac{\nabla f(x)}{||\nabla f(x)||}$, then $f(x') - f(x) \geq \epsilon ||\nabla f(x)||$.
By ρ-Lipschitzness, $|f(x') - f(x)| \leq \rho ||x' - x|| = \rho\epsilon$.
$\therefore \epsilon ||\nabla f(x)|| \leq \rho\epsilon, \implies ||\nabla f(x)|| \leq \rho$. □

Using (3), Lipschitz condition and Lemma 7, the proof is complete. □

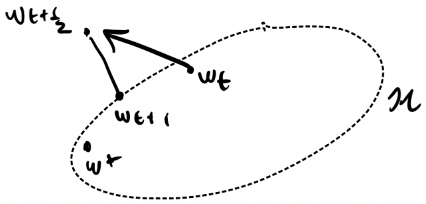
4 Variants and Other Properties

• If f is not differentiable, we can use subgradient descent. A subgradient $\partial f(x)$ is any vector which satisfies the first order definition of convexity, $f(y) - f(x) \leq \langle y - x, \partial f(y)\rangle$.
E.g. Hinge loss is not differentiable at 0 but there are many valid subgradients, as shown in the figure below.



• If we are doing constrained optimization, we can use projected gradient descent.
Suppose we have: $\min_{x \in \mathcal{H}} f(x)$, then the updates are given as:

$$w_{t+\frac{1}{2}} = w_t - \eta \nabla f(w_t)$$
$$w_{t+1} = \arg\min_{w \in \mathcal{H}} ||w - w_{t+\frac{1}{2}}||.$$



The second step finds the projection of $w_{t+\frac{1}{2}}$ in the convex set $\mathcal{H}$ which is closest to it.

Projected GD is visualized in the figure above.

Lemma 9. *For any $v \in \mathcal{H}$, $||w_{t+1} - v|| \leq ||w_{t+\frac{1}{2}} - v||$.*

Proof. $v \in \mathcal{H}$ and projection finds a point $w_{t+1} \in \mathcal{H}$, closest to $w_{t+\frac{1}{2}}$. As $w_{t+\frac{1}{2}} \notin \mathcal{H}$, w_{t+1} is closer to v. □

Lemma 10. *Projected GD has the same convergence rate as GD.*

Proof. Intuition: The projection step gets us closer to the optimal point, it never takes us farther away. Then, we can repeat the analysis for GD. As GD was working and projection step doesn't hurt, projected GD will also work. □

Note that the projection should be efficiently computable for this.

- Some faster convergence guarantees for GD:

Definition 11 (Smoothness)**.** *We say a function f is β-smooth if $\forall\ x, y \in$ domain(f),*

$$||\nabla f(x) - \nabla f(y)|| \leq \beta ||x - y||.$$

The gradient itself is a Lipschitz function, the function can't be too steep. This also implies that

$$f(y) \leq f(x) + \langle y - x, \nabla f(x) \rangle + \frac{\beta}{2}\|y - x\|^2.$$

Theorem 12. *Let f be a convex function that is β-smooth. Then GD with step size $\eta = \frac{1}{\beta}$ satisfies*

$$f(w_T) - f(w^*) \leq \frac{B^2\beta}{2T}.$$

In this case, ϵ error can be achieved in $T \geq \frac{B^2\beta}{2\epsilon}$ iterations, which means we get faster convergence without averaging.

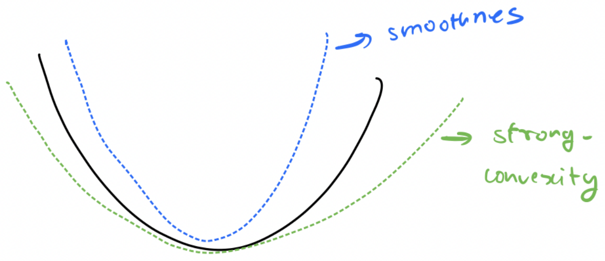
Recall:

Definition 13 (Strong Convexity)**.** *A function f is λ-strongly convex if its domain is convex and $\forall\ x, y$,*

$$f(y) \geq f(x) + \langle y - x, \nabla f(x) \rangle + \frac{\lambda}{2}\|y - x\|^2.$$

Theorem 14. *Let f be a convex function that is β-smooth and λ-strongly convex. Define condition number $\kappa = \frac{\beta}{\lambda} > 1$. Then GD with $\eta = \frac{2}{\beta + \lambda}$ satisfies*

$$f(w_T) - f(w^*) \leq e^{-\frac{T}{\kappa}} \frac{B^2\beta}{2T}.$$



Smoothness and strong convexity give upper and lower bounds on f, respectively, as shown in the figure above. This means that $\beta > \lambda$.

In this case, ϵ error can be achieved in $T \geq \kappa \log\left(\frac{B^2\beta}{2\epsilon}\right)$ or $\mathcal{O}\left(\kappa \log\left(\frac{1}{\epsilon}\right)\right)$ iterations. It is known as linear convergence in literature as when we plot it on a log scale, the error reduces linearly with iterations. This is the fastest convergence rate for GD.

To understand optimization better, think about linear regression: $f(x) = \min\limits_{x\in\mathbb{R}^d} x^T Ax$. It is convex if $A \succcurlyeq 0$ (positive semi-definite). In this case, β and λ are the largest and smallest eigenvalues of A, respectively.

When x is a vector, the problem will be well-conditioned if $f(x)$ has similar curvatures in all dimensions. Otherwise, the bounds could be very different in different dimensions, the problem would not be well-conditioned.

Bonus: There are accelerated methods which get faster (and optimal!) convergence rates ($\sqrt{\kappa}$ in place of κ in this case and T^2 in place of T in the previous case).

5 Stochastic Gradient Descent (SGD)

Algorithm 2: SGD

```
1 Initialize $w_1$
2 for t=1,2,...,$T$ do
3     Choose any $v_t$ s.t. $\mathbb{E}(v_t|w_t) = \nabla f(w_t)$
4     w$_{t+1} \leftarrow w_t - \eta v_t$
```

Step 3 requires that v_t is a valid gradient in expectation. If we are minimizing training loss over n points, we can just sample one point (x_i, y_i) uniformly at random and take gradient at (x_i, y_i), instead of computing gradient for the entire training set. Since

$$\nabla_w\left(\frac{1}{n}\sum_{i=1}^{n} l(w, z_i)\right) = \frac{1}{n}\sum_{i=1}^{n} \nabla_w l(w, z_i),$$

v_t gives an unbiased estimate of $\nabla f(w_t)$ because of linearity of expectation.