Introduction to Machine Learning (67577)
Lecture 7

Shai Shalev-Shwartz

School of CS and Engineering,
The Hebrew University of Jerusalem

Solving Convex Problems using SGD and RLM

Outline

1. Reminder: Convex learning problems
2. Learning Using Stochastic Gradient Descent
3. Learning Using Regularized Loss Minimization
4. Dimension vs. Norm bounds
   - Example application: Text categorization

Convex-Lipschitz-bounded learning problem

Definition (Convex-Lipschitz-Bounded Learning Problem)

A learning problem, $(\mathcal{H}, Z, \ell)$, is called Convex-Lipschitz-Bounded, with parameters ρ, B if the following holds:

- The hypothesis class $\mathcal{H}$ is a convex set and for all $\mathbf{w} \in \mathcal{H}$ we have $\|\mathbf{w}\| \le B$.
- For all $z \in Z$, the loss function, $\ell(\cdot, z)$, is a convex and ρ-Lipschitz function.

Convex-Lipschitz-bounded learning problem

Definition (Convex-Lipschitz-Bounded Learning Problem)

A learning problem, $(\mathcal{H}, Z, \ell)$, is called Convex-Lipschitz-Bounded, with parameters ρ, B if the following holds:

- The hypothesis class $\mathcal{H}$ is a convex set and for all $\mathbf{w} \in \mathcal{H}$ we have $\|\mathbf{w}\| \leq B$.
- For all $z \in Z$, the loss function, $\ell(\cdot, z)$, is a convex and ρ-Lipschitz function.

Example:

- $\mathcal{H} = \{\mathbf{w} \in \mathbb{R}^d : \|\mathbf{w}\| \leq B\}$
- $\mathcal{X} = \{\mathbf{x} \in \mathbb{R}^d : \|\mathbf{x}\| \leq \rho\}$, $\mathcal{Y} = \mathbb{R}$,
- $\ell(\mathbf{w}, (\mathbf{x}, y)) = |\langle \mathbf{w}, \mathbf{x} \rangle - y|$

Convex-Smooth-bounded learning problem

Definition (Convex-Smooth-Bounded Learning Problem)

A learning problem, $(\mathcal{H}, Z, \ell)$, is called Convex-Smooth-Bounded, with parameters β, B if the following holds:

- The hypothesis class $\mathcal{H}$ is a convex set and for all $\mathbf{w} \in \mathcal{H}$ we have $\|\mathbf{w}\| \leq B$.
- For all $z \in Z$, the loss function, $\ell(\cdot, z)$, is a convex, non-negative, and β-smooth function.

Convex-Smooth-bounded learning problem

Definition (Convex-Smooth-Bounded Learning Problem)

A learning problem, $(\mathcal{H}, Z, \ell)$, is called Convex-Smooth-Bounded, with parameters β, B if the following holds:

- The hypothesis class $\mathcal{H}$ is a convex set and for all $\mathbf{w} \in \mathcal{H}$ we have $\|\mathbf{w}\| \leq B$.
- For all $z \in Z$, the loss function, $\ell(\cdot, z)$, is a convex, non-negative, and β-smooth function.

Example:

- $\mathcal{H} = \{\mathbf{w} \in \mathbb{R}^d : \|\mathbf{w}\| \leq B\}$
- $\mathcal{X} = \{\mathbf{x} \in \mathbb{R}^d : \|\mathbf{x}\| \leq \beta/2\}$, $\mathcal{Y} = \mathbb{R}$,
- $\ell(\mathbf{w}, (\mathbf{x}, y)) = (\langle \mathbf{w}, \mathbf{x} \rangle - y)^2$

Outline

Learning Using Stochastic Gradient Descent

- Consider a learning problem.

Learning Using Stochastic Gradient Descent

- Consider a learning problem.
- Recall: our goal is to (probably approximately) solve:

$$\min_{\mathbf{w} \in \mathcal{H}} L_{\mathcal{D}}(\mathbf{w}) \quad \text{where} \quad L_{\mathcal{D}}(\mathbf{w}) = \mathop{\mathbb{E}}_{z \sim \mathcal{D}} [\ell(\mathbf{w}, z)]$$

Learning Using Stochastic Gradient Descent

- Consider a learning problem.
- Recall: our goal is to (probably approximately) solve:

$$\min_{\mathbf{w} \in \mathcal{H}} L_{\mathcal{D}}(\mathbf{w}) \quad \text{where} \quad L_{\mathcal{D}}(\mathbf{w}) = \mathop{\mathbb{E}}_{z \sim \mathcal{D}}[\ell(\mathbf{w}, z)]$$

- So far, learning was based on the empirical risk, $L_S(\mathbf{w})$

Learning Using Stochastic Gradient Descent

- Consider a learning problem.
- Recall: our goal is to (probably approximately) solve:

$$\min_{\mathbf{w}\in\mathcal{H}} L_{\mathcal{D}}(\mathbf{w}) \quad \text{where} \quad L_{\mathcal{D}}(\mathbf{w}) = \mathop{\mathbb{E}}_{z\sim\mathcal{D}}[\ell(\mathbf{w}, z)]$$

- So far, learning was based on the empirical risk, $L_S(\mathbf{w})$
- We now consider directly minimizing $L_{\mathcal{D}}(\mathbf{w})$

Stochastic Gradient Descent

$$\min_{\mathbf{w} \in \mathcal{H}} L_{\mathcal{D}}(\mathbf{w}) \quad \text{where} \quad L_{\mathcal{D}}(\mathbf{w}) = \mathop{\mathbb{E}}_{z \sim \mathcal{D}} [\ell(\mathbf{w}, z)]$$

- Recall the gradient descent method in which we initialize $\mathbf{w}^{(1)} = \mathbf{0}$ and update $\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} - \eta \nabla L_{\mathcal{D}}(\mathbf{w})$

Stochastic Gradient Descent

$$\min_{\mathbf{w} \in \mathcal{H}} L_{\mathcal{D}}(\mathbf{w}) \quad \text{where} \quad L_{\mathcal{D}}(\mathbf{w}) = \mathop{\mathbb{E}}_{z \sim \mathcal{D}}[\ell(\mathbf{w}, z)]$$

- Recall the gradient descent method in which we initialize $\mathbf{w}^{(1)} = \mathbf{0}$ and update $\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} - \eta \nabla L_{\mathcal{D}}(\mathbf{w})$
- Observe: $\nabla L_{\mathcal{D}}(\mathbf{w}) = \mathbb{E}_{z \sim \mathcal{D}}[\nabla \ell(\mathbf{w}, z)]$

Stochastic Gradient Descent

$$\min_{\mathbf{w}\in\mathcal{H}} L_{\mathcal{D}}(\mathbf{w}) \quad \text{where} \quad L_{\mathcal{D}}(\mathbf{w}) = \mathop{\mathbb{E}}_{z\sim\mathcal{D}}[\ell(\mathbf{w}, z)]$$

- Recall the gradient descent method in which we initialize $\mathbf{w}^{(1)} = \mathbf{0}$ and update $\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} - \eta\nabla L_{\mathcal{D}}(\mathbf{w})$
- Observe: $\nabla L_{\mathcal{D}}(\mathbf{w}) = \mathbb{E}_{z\sim\mathcal{D}}[\nabla\ell(\mathbf{w}, z)]$
- We can't calculate $\nabla L_{\mathcal{D}}(\mathbf{w})$ because we don't know $\mathcal{D}$

Stochastic Gradient Descent

$$\min_{\mathbf{w}\in\mathcal{H}} L_{\mathcal{D}}(\mathbf{w}) \quad \text{where} \quad L_{\mathcal{D}}(\mathbf{w}) = \mathop{\mathbb{E}}_{z\sim\mathcal{D}}[\ell(\mathbf{w}, z)]$$

- Recall the gradient descent method in which we initialize $\mathbf{w}^{(1)} = \mathbf{0}$ and update $\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} - \eta\nabla L_{\mathcal{D}}(\mathbf{w})$
- Observe: $\nabla L_{\mathcal{D}}(\mathbf{w}) = \mathbb{E}_{z\sim\mathcal{D}}[\nabla\ell(\mathbf{w}, z)]$
- We can't calculate $\nabla L_{\mathcal{D}}(\mathbf{w})$ because we don't know $\mathcal{D}$
- But we can estimate it by $\nabla\ell(\mathbf{w}, z)$ for $z \sim \mathcal{D}$

Stochastic Gradient Descent

$$\min_{\mathbf{w}\in\mathcal{H}} L_{\mathcal{D}}(\mathbf{w}) \quad \text{where} \quad L_{\mathcal{D}}(\mathbf{w}) = \mathop{\mathbb{E}}_{z\sim\mathcal{D}}[\ell(\mathbf{w}, z)]$$

- Recall the gradient descent method in which we initialize $\mathbf{w}^{(1)} = \mathbf{0}$ and update $\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} - \eta\nabla L_{\mathcal{D}}(\mathbf{w})$
- Observe: $\nabla L_{\mathcal{D}}(\mathbf{w}) = \mathbb{E}_{z\sim\mathcal{D}}[\nabla\ell(\mathbf{w}, z)]$
- We can't calculate $\nabla L_{\mathcal{D}}(\mathbf{w})$ because we don't know $\mathcal{D}$
- But we can estimate it by $\nabla\ell(\mathbf{w}, z)$ for $z \sim \mathcal{D}$
- If we take a step based on the direction $\mathbf{v} = \nabla\ell(\mathbf{w}, z)$ then in expectation we're moving in the right direction

Stochastic Gradient Descent

$$\min_{\mathbf{w} \in \mathcal{H}} L_{\mathcal{D}}(\mathbf{w}) \quad \text{where} \quad L_{\mathcal{D}}(\mathbf{w}) = \mathop{\mathbb{E}}_{z \sim \mathcal{D}}[\ell(\mathbf{w}, z)]$$

- Recall the gradient descent method in which we initialize $\mathbf{w}^{(1)} = \mathbf{0}$ and update $\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} - \eta \nabla L_{\mathcal{D}}(\mathbf{w})$
- Observe: $\nabla L_{\mathcal{D}}(\mathbf{w}) = \mathbb{E}_{z \sim \mathcal{D}}[\nabla \ell(\mathbf{w}, z)]$
- We can't calculate $\nabla L_{\mathcal{D}}(\mathbf{w})$ because we don't know $\mathcal{D}$
- But we can estimate it by $\nabla \ell(\mathbf{w}, z)$ for $z \sim \mathcal{D}$
- If we take a step based on the direction $\mathbf{v} = \nabla \ell(\mathbf{w}, z)$ then in expectation we're moving in the right direction
- In other words, $\mathbf{v}$ is an unbiased estimate of the gradient

Stochastic Gradient Descent

$$\min_{\mathbf{w}\in\mathcal{H}} L_{\mathcal{D}}(\mathbf{w}) \quad \text{where} \quad L_{\mathcal{D}}(\mathbf{w}) = \mathop{\mathbb{E}}_{z\sim\mathcal{D}}[\ell(\mathbf{w}, z)]$$

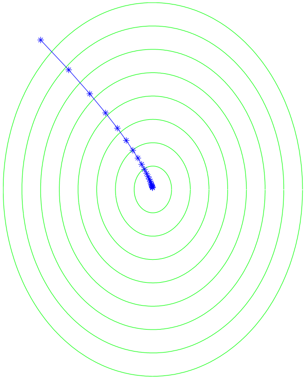
- Recall the gradient descent method in which we initialize $\mathbf{w}^{(1)} = \mathbf{0}$ and update $\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} - \eta\nabla L_{\mathcal{D}}(\mathbf{w})$
- Observe: $\nabla L_{\mathcal{D}}(\mathbf{w}) = \mathbb{E}_{z\sim\mathcal{D}}[\nabla\ell(\mathbf{w}, z)]$
- We can't calculate $\nabla L_{\mathcal{D}}(\mathbf{w})$ because we don't know $\mathcal{D}$
- But we can estimate it by $\nabla\ell(\mathbf{w}, z)$ for $z \sim \mathcal{D}$
- If we take a step based on the direction $\mathbf{v} = \nabla\ell(\mathbf{w}, z)$ then in expectation we're moving in the right direction
- In other words, $\mathbf{v}$ is an unbiased estimate of the gradient
- We'll show that this is good enough

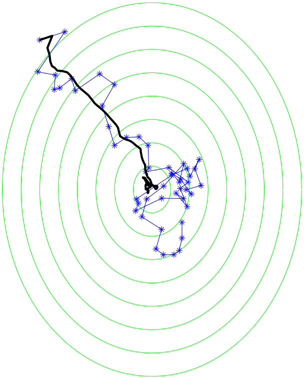
Stochastic Gradient Descent

- **initialize:** $\mathbf{w}^{(1)} = \mathbf{0}$
- **for** $t = 1, 2, \ldots, T$
  - choose $z_t \sim \mathcal{D}$
  - let $\mathbf{v}_t \in \partial \ell(\mathbf{w}^{(t)}, z_t)$ update $\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} - \eta \mathbf{v}_t$
- **output** $\bar{\mathbf{w}} = \frac{1}{T} \sum_{t=1}^{T} \mathbf{w}^{(t)}$

Stochastic Gradient Descent

- **initialize:** $\mathbf{w}^{(1)} = \mathbf{0}$
- **for** $t = 1, 2, \ldots, T$
  - choose $z_t \sim \mathcal{D}$
  - let $\mathbf{v}_t \in \partial \ell(\mathbf{w}^{(t)}, z_t)$ update $\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} - \eta \mathbf{v}_t$
- **output** $\bar{\mathbf{w}} = \frac{1}{T} \sum_{t=1}^{T} \mathbf{w}^{(t)}$





Analyzing SGD for convex-Lipschitz-bounded

By algebraic manipulations, for any sequence of $\mathbf{v}_1, \ldots, \mathbf{v}_T$, and any $\mathbf{w}^\star$,

$$\sum_{t=1}^{T} \langle \mathbf{w}^{(t)} - \mathbf{w}^\star, \mathbf{v}_t \rangle = \frac{\|\mathbf{w}^{(1)} - \mathbf{w}^\star\|^2 - \|\mathbf{w}^{(T+1)} - \mathbf{w}^\star\|^2}{2\eta} + \frac{\eta}{2} \sum_{t=1}^{T} \|\mathbf{v}_t\|^2$$

Analyzing SGD for convex-Lipschitz-bounded

By algebraic manipulations, for any sequence of $\mathbf{v}_1, \ldots, \mathbf{v}_T$, and any $\mathbf{w}^\star$,

$$\sum_{t=1}^{T} \langle \mathbf{w}^{(t)} - \mathbf{w}^\star, \mathbf{v}_t \rangle = \frac{\|\mathbf{w}^{(1)} - \mathbf{w}^\star\|^2 - \|\mathbf{w}^{(T+1)} - \mathbf{w}^\star\|^2}{2\eta} + \frac{\eta}{2} \sum_{t=1}^{T} \|\mathbf{v}_t\|^2$$

Assume that $\|\mathbf{v}_t\| \leq \rho$ for all t and that $\|\mathbf{w}^\star\| \leq B$ we obtain

$$\sum_{t=1}^{T} \langle \mathbf{w}^{(t)} - \mathbf{w}^\star, \mathbf{v}_t \rangle \leq \frac{B^2}{2\eta} + \frac{\eta\,\rho^2\,T}{2}$$

Analyzing SGD for convex-Lipschitz-bounded

By algebraic manipulations, for any sequence of $\mathbf{v}_1, \ldots, \mathbf{v}_T$, and any $\mathbf{w}^\star$,

$$\sum_{t=1}^{T} \langle \mathbf{w}^{(t)} - \mathbf{w}^\star, \mathbf{v}_t \rangle = \frac{\|\mathbf{w}^{(1)} - \mathbf{w}^\star\|^2 - \|\mathbf{w}^{(T+1)} - \mathbf{w}^\star\|^2}{2\eta} + \frac{\eta}{2} \sum_{t=1}^{T} \|\mathbf{v}_t\|^2$$

Assume that $\|\mathbf{v}_t\| \leq \rho$ for all t and that $\|\mathbf{w}^\star\| \leq B$ we obtain

$$\sum_{t=1}^{T} \langle \mathbf{w}^{(t)} - \mathbf{w}^\star, \mathbf{v}_t \rangle \leq \frac{B^2}{2\eta} + \frac{\eta\, \rho^2\, T}{2}$$

In particular, for $\eta = \sqrt{\frac{B^2}{\rho^2\, T}}$ we get

$$\sum_{t=1}^{T} \langle \mathbf{w}^{(t)} - \mathbf{w}^\star, \mathbf{v}_t \rangle \leq B\, \rho \sqrt{T} \; .$$

Analyzing SGD for convex-Lipschitz-bounded

Taking expectation of both sides w.r.t. the randomness of choosing $z_1, \ldots, z_T$ we obtain:

$$\mathop{\mathbb{E}}_{z_1,\ldots,z_T}\left[\sum_{t=1}^{T}\langle \mathbf{w}^{(t)} - \mathbf{w}^\star, \mathbf{v}_t\rangle\right] \;\leq\; B\,\rho\sqrt{T}\ .$$

Analyzing SGD for convex-Lipschitz-bounded

Taking expectation of both sides w.r.t. the randomness of choosing $z_1, \ldots, z_T$ we obtain:

$$\mathop{\mathbb{E}}_{z_1,\ldots,z_T}\left[\sum_{t=1}^{T}\langle \mathbf{w}^{(t)} - \mathbf{w}^\star, \mathbf{v}_t\rangle\right] \;\leq\; B\,\rho\,\sqrt{T}\ .$$

The law of total expectation: for every two random variables α, β, and a function g, $\mathbb{E}_\alpha[g(\alpha)] = \mathbb{E}_\beta\,\mathbb{E}_\alpha[g(\alpha)|\beta]$.

Analyzing SGD for convex-Lipschitz-bounded

Taking expectation of both sides w.r.t. the randomness of choosing $z_1, \ldots, z_T$ we obtain:

$$\mathop{\mathbb{E}}_{z_1,\ldots,z_T}\left[\sum_{t=1}^{T}\langle \mathbf{w}^{(t)} - \mathbf{w}^\star, \mathbf{v}_t\rangle\right] \;\leq\; B\,\rho\sqrt{T}\ .$$

The law of total expectation: for every two random variables α, β, and a function g, $\mathbb{E}_\alpha[g(\alpha)] = \mathbb{E}_\beta\,\mathbb{E}_\alpha[g(\alpha)|\beta]$. Therefore

$$\mathop{\mathbb{E}}_{z_1,\ldots,z_T}[\langle \mathbf{w}^{(t)} - \mathbf{w}^\star, \mathbf{v}_t\rangle] = \mathop{\mathbb{E}}_{z_1,\ldots,z_{t-1}}\ \mathop{\mathbb{E}}_{z_1,\ldots,z_T}[\langle \mathbf{w}^{(t)} - \mathbf{w}^\star, \mathbf{v}_t\rangle \,|\, z_1, \ldots, z_{t-1}]\ .$$

Analyzing SGD for convex-Lipschitz-bounded

Taking expectation of both sides w.r.t. the randomness of choosing $z_1, \ldots, z_T$ we obtain:

$$\mathop{\mathbb{E}}_{z_1,\ldots,z_T}\left[\sum_{t=1}^{T}\langle \mathbf{w}^{(t)} - \mathbf{w}^\star, \mathbf{v}_t\rangle\right] \;\leq\; B\,\rho\sqrt{T} \;.$$

The law of total expectation: for every two random variables α, β, and a function g, $\mathbb{E}_\alpha[g(\alpha)] = \mathbb{E}_\beta\,\mathbb{E}_\alpha[g(\alpha)|\beta]$. Therefore

$$\mathop{\mathbb{E}}_{z_1,\ldots,z_T}[\langle \mathbf{w}^{(t)} - \mathbf{w}^\star, \mathbf{v}_t\rangle] = \mathop{\mathbb{E}}_{z_1,\ldots,z_{t-1}}\;\mathop{\mathbb{E}}_{z_1,\ldots,z_T}[\langle \mathbf{w}^{(t)} - \mathbf{w}^\star, \mathbf{v}_t\rangle \,|\, z_1,\ldots,z_{t-1}] \;.$$

Once we know $z_1, \ldots, z_{t-1}$ the value of $\mathbf{w}^{(t)}$ is not random, hence,

$$\begin{aligned}\mathop{\mathbb{E}}_{z_1,\ldots,z_T}[\langle \mathbf{w}^{(t)} - \mathbf{w}^\star, \mathbf{v}_t\rangle \,|\, z_1,\ldots,z_{t-1}] &= \langle \mathbf{w}^{(t)} - \mathbf{w}^\star \;,\; \mathop{\mathbb{E}}_{z_t}[\nabla \ell(\mathbf{w}^{(t)}, z_t)]\rangle \\ &= \langle \mathbf{w}^{(t)} - \mathbf{w}^\star \;,\; \nabla L_{\mathcal{D}}(\mathbf{w}^{(t)})\rangle\end{aligned}$$

Analyzing SGD for convex-Lipschitz-bounded

We got:

$$\mathop{\mathbb{E}}_{z_1,\ldots,z_T}\left[\sum_{t=1}^{T}\langle \mathbf{w}^{(t)} - \mathbf{w}^\star \ , \ \nabla L_{\mathcal{D}}(\mathbf{w}^{(t)})\rangle\right] \ \leq \ B\,\rho\sqrt{T}$$

Analyzing SGD for convex-Lipschitz-bounded

We got:

$$\mathop{\mathbb{E}}_{z_1,\ldots,z_T}\left[\sum_{t=1}^{T}\langle \mathbf{w}^{(t)} - \mathbf{w}^\star \ , \ \nabla L_{\mathcal{D}}(\mathbf{w}^{(t)})\rangle\right] \ \le \ B\,\rho\,\sqrt{T}$$

By convexity, this means

$$\mathop{\mathbb{E}}_{z_1,\ldots,z_T}\left[\sum_{t=1}^{T}(L_{\mathcal{D}}(\mathbf{w}^{(t)}) - L_{\mathcal{D}}(\mathbf{w}^\star))\right] \ \le \ B\,\rho\,\sqrt{T}$$

Analyzing SGD for convex-Lipschitz-bounded

We got:

$$\mathop{\mathbb{E}}_{z_1,\ldots,z_T}\left[\sum_{t=1}^{T}\langle \mathbf{w}^{(t)} - \mathbf{w}^\star \; , \; \nabla L_{\mathcal{D}}(\mathbf{w}^{(t)})\rangle\right] \;\le\; B\,\rho\,\sqrt{T}$$

By convexity, this means

$$\mathop{\mathbb{E}}_{z_1,\ldots,z_T}\left[\sum_{t=1}^{T}(L_{\mathcal{D}}(\mathbf{w}^{(t)}) - L_{\mathcal{D}}(\mathbf{w}^\star))\right] \;\le\; B\,\rho\,\sqrt{T}$$

Dividing by T and using convexity again,

$$\mathop{\mathbb{E}}_{z_1,\ldots,z_T}\left[L_{\mathcal{D}}\left(\frac{1}{T}\sum_{t=1}^{T}\mathbf{w}^{(t)}\right)\right] \;\le\; L_{\mathcal{D}}(\mathbf{w}^\star) + \frac{B\,\rho}{\sqrt{T}}$$

Learning convex-Lipschitz-bounded problems using SGD

Corollary

Consider a convex-Lipschitz-bounded learning problem with parameters ρ, B. *Then, for every* $\epsilon > 0$, *if we run the SGD method for minimizing* $L_{\mathcal{D}}(\mathbf{w})$ *with a number of iterations (i.e., number of examples)*

$$T \geq \frac{B^2 \rho^2}{\epsilon^2}$$

and with $\eta = \sqrt{\frac{B^2}{\rho^2 T}}$, *then the output of SGD satisfies:*

$$\mathbb{E}\left[L_{\mathcal{D}}(\bar{\mathbf{w}})\right] \leq \min_{\mathbf{w} \in \mathcal{H}} L_{\mathcal{D}}(\mathbf{w}) + \epsilon \ .$$

Learning convex-Lipschitz-bounded problems using SGD

Corollary

Consider a convex-Lipschitz-bounded learning problem with parameters ρ, B. Then, for every $\epsilon > 0$, if we run the SGD method for minimizing $L_{\mathcal{D}}(\mathbf{w})$ with a number of iterations (i.e., number of examples)

$$T \geq \frac{B^2 \rho^2}{\epsilon^2}$$

and with $\eta = \sqrt{\frac{B^2}{\rho^2 T}}$, then the output of SGD satisfies:

$$\mathbb{E}\left[L_{\mathcal{D}}(\bar{\mathbf{w}})\right] \leq \min_{\mathbf{w} \in \mathcal{H}} L_{\mathcal{D}}(\mathbf{w}) + \epsilon \ .$$

- Remark: Can obtain high probability bound using "boosting the confidence" (Lecture 4)

Convex-smooth-bounded problems

Similar result holds for smooth problems:

Corollary

Consider a convex-smooth-bounded learning problem with parameters β, B*. Assume in addition that* $\ell(\mathbf{0}, z) \leq 1$ *for all* $z \in Z$*. For every* $\epsilon > 0$*, set* $\eta = \frac{1}{\beta(1+3/\epsilon)}$*. Then, running SGD with* $T \geq 12B^2\beta/\epsilon^2$ *yields*

$$\mathbb{E}[L_{\mathcal{D}}(\bar{\mathbf{w}})] \leq \min_{\mathbf{w} \in \mathcal{H}} L_{\mathcal{D}}(\mathbf{w}) + \epsilon \ .$$

Outline

1. Reminder: Convex learning problems
2. Learning Using Stochastic Gradient Descent
3. Learning Using Regularized Loss Minimization
4. Dimension vs. Norm bounds
   - Example application: Text categorization

Regularized Loss Minimization (RLM)

Given a regularization function $R : \mathbb{R}^d \to \mathbb{R}$, the RLM rule is:

$$A(S) = \operatorname*{argmin}_{\mathbf{w}} \left(L_S(\mathbf{w}) + R(\mathbf{w})\right) \ .$$

Regularized Loss Minimization (RLM)

Given a regularization function $R : \mathbb{R}^d \to \mathbb{R}$, the RLM rule is:

$$A(S) = \operatorname*{argmin}_{\mathbf{w}} \left(L_S(\mathbf{w}) + R(\mathbf{w})\right) \ .$$

We will focus on Tikhonov regularization

$$A(S) = \operatorname*{argmin}_{\mathbf{w}} \left(L_S(\mathbf{w}) + \lambda \|\mathbf{w}\|^2\right) \ .$$

Why to regularize ?

- Similar to MDL: specify "prior belief" in hypotheses. We bias ourselves toward "short" vectors.
- Stabilizer: we'll show that Tikhonov regularization makes the learner stable w.r.t. small perturbation of the training set, which in turn leads to generalization

Stability

- Informally: an algorithm A is stable if a small change of its input S will lead to a small change of its output hypothesis

Stability

- Informally: an algorithm A is stable if a small change of its input S will lead to a small change of its output hypothesis
- Need to specify what is "small change of input" and what is "small change of output"

Stability

- Replace one sample: given $S = (z_1, \ldots, z_m)$ and an additional example z', let $S^{(i)} = (z_1, \ldots, z_{i-1}, z', z_{i+1}, \ldots, z_m)$

Stability

- Replace one sample: given $S = (z_1, \ldots, z_m)$ and an additional example z', let $S^{(i)} = (z_1, \ldots, z_{i-1}, z', z_{i+1}, \ldots, z_m)$

Definition (on-average-replace-one-stable)

Let $\epsilon : \mathbb{N} \to \mathbb{R}$ be a monotonically decreasing function. We say that a learning algorithm A is on-average-replace-one-stable with rate $\epsilon(m)$ if for every distribution $\mathcal{D}$

$$\mathop{\mathbb{E}}_{(S,z') \sim \mathcal{D}^{m+1}, i \sim U(m)} [\ell(A(S^{(i)}, z_i)) - \ell(A(S), z_i)] \leq \epsilon(m) \ .$$

Stable rules do not ovefit

Theorem

if A is on-average-replace-one-stable with rate $\epsilon(m)$ then

$$\mathop{\mathbb{E}}_{S \sim \mathcal{D}^m}[L_{\mathcal{D}}(A(S)) - L_S(A(S))] \le \epsilon(m) ~.$$

Stable rules do not ovefit

Theorem

if A is on-average-replace-one-stable with rate $\epsilon(m)$ then

$$\mathop{\mathbb{E}}_{S\sim\mathcal{D}^m}[L_{\mathcal{D}}(A(S)) - L_S(A(S))] \le \epsilon(m) \ .$$

Proof.

Since S and z' are both drawn i.i.d. from $\mathcal{D}$, we have that for every i,

$$\mathop{\mathbb{E}}_{S}[L_{\mathcal{D}}(A(S))] = \mathop{\mathbb{E}}_{S,z'}[\ell(A(S), z')] = \mathop{\mathbb{E}}_{S,z'}[\ell(A(S^{(i)}), z_i)] \ .$$

On the other hand, we can write

$$\mathop{\mathbb{E}}_{S}[L_S(A(S))] = \mathop{\mathbb{E}}_{S,i}[\ell(A(S), z_i)] \ .$$

The proof follows from the definition of stability. □

Tikhonov Regularization as Stabilizer

Theorem

Assume that the loss function is convex and ρ-Lipschitz. Then, the RLM rule with the regularizer $\lambda\|\mathbf{w}\|^2$ is on-average-replace-one-stable with rate $\frac{2\,\rho^2}{\lambda\,m}$. It follows that

$$\mathop{\mathbb{E}}_{S\sim\mathcal{D}^m}[L_{\mathcal{D}}(A(S)) - L_S(A(S))] \le \frac{2\,\rho^2}{\lambda\,m} ~.$$

Similarly, for convex, β-smooth, and non-negative, loss the rate is $\frac{48\beta C}{\lambda m}$, where C is an upper bound on $\max_z \ell(\mathbf{0}, z)$.

Tikhonov Regularization as Stabilizer

Theorem

Assume that the loss function is convex and ρ-Lipschitz. Then, the RLM rule with the regularizer $\lambda\|\mathbf{w}\|^2$ is on-average-replace-one-stable with rate $\frac{2\,\rho^2}{\lambda\,m}$. It follows that

$$\mathop{\mathbb{E}}_{S\sim\mathcal{D}^m}[L_{\mathcal{D}}(A(S)) - L_S(A(S))] \le \frac{2\,\rho^2}{\lambda\,m}~.$$

Similarly, for convex, β-smooth, and non-negative, loss the rate is $\frac{48\beta C}{\lambda m}$, where C is an upper bound on $\max_z \ell(\mathbf{0}, z)$.

The proof relies on the notion of strong convexity and can be found in the book.

The Fitting-Stability Tradeoff

Observe:

$$\mathop{\mathbb{E}}_{S}[L_{\mathcal{D}}(A(S))] = \mathop{\mathbb{E}}_{S}[L_S(A(S))] + \mathop{\mathbb{E}}_{S}[L_{\mathcal{D}}(A(S)) - L_S(A(S))] \; .$$

The Fitting-Stability Tradeoff

Observe:

$$\mathop{\mathbb{E}}_S[L_{\mathcal{D}}(A(S))] = \mathop{\mathbb{E}}_S[L_S(A(S))] + \mathop{\mathbb{E}}_S[L_{\mathcal{D}}(A(S)) - L_S(A(S))] \ .$$

- The first term is how good A fits the training set
- The 2nd term is the overfitting, and is bounded by the stability of A

The Fitting-Stability Tradeoff

Observe:

$$\mathop{\mathbb{E}}_{S}[L_{\mathcal{D}}(A(S))] = \mathop{\mathbb{E}}_{S}[L_S(A(S))] + \mathop{\mathbb{E}}_{S}[L_{\mathcal{D}}(A(S)) - L_S(A(S))] \ .$$

- The first term is how good A fits the training set
- The 2nd term is the overfitting, and is bounded by the stability of A
- λ controls the tradeoff between the two terms

The Fitting-Stability Tradeoff

- Let A be the RLM rule

The Fitting-Stability Tradeoff

- Let A be the RLM rule
- We saw (for convex-Lipschitz losses)

$$\mathop{\mathbb{E}}_{S}[L_{\mathcal{D}}(A(S)) - L_S(A(S))] \leq \frac{2\,\rho^2}{\lambda\, m}$$

The Fitting-Stability Tradeoff

- Let A be the RLM rule
- We saw (for convex-Lipschitz losses)

$$\mathop{\mathbb{E}}_{S}[L_{\mathcal{D}}(A(S)) - L_S(A(S))] \leq \frac{2\,\rho^2}{\lambda\, m}$$

- Fix some arbitrary vector $\mathbf{w}^*$, then:

$$L_S(A(S)) \leq L_S(A(S)) + \lambda\|A(S)\|^2 \leq L_S(\mathbf{w}^*) + \lambda\|\mathbf{w}^*\|^2 \ .$$

The Fitting-Stability Tradeoff

- Let A be the RLM rule
- We saw (for convex-Lipschitz losses)

$$\mathop{\mathbb{E}}_{S}[L_{\mathcal{D}}(A(S)) - L_S(A(S))] \leq \frac{2\,\rho^2}{\lambda\, m}$$

- Fix some arbitrary vector $\mathbf{w}^*$, then:

$$L_S(A(S)) \leq L_S(A(S)) + \lambda\|A(S)\|^2 \leq L_S(\mathbf{w}^*) + \lambda\|\mathbf{w}^*\|^2 \ .$$

- Taking expectation of both sides with respect to S and noting that $\mathbb{E}_S[L_S(\mathbf{w}^*)] = L_{\mathcal{D}}(\mathbf{w}^*)$, we obtain that

$$\mathop{\mathbb{E}}_{S}[L_S(A(S))] \leq L_{\mathcal{D}}(\mathbf{w}^*) + \lambda\|\mathbf{w}^*\|^2 \ .$$

The Fitting-Stability Tradeoff

- Let A be the RLM rule
- We saw (for convex-Lipschitz losses)

$$\mathop{\mathbb{E}}_{S}[L_{\mathcal{D}}(A(S)) - L_S(A(S))] \le \frac{2\,\rho^2}{\lambda\, m}$$

- Fix some arbitrary vector $\mathbf{w}^*$, then:

$$L_S(A(S)) \le L_S(A(S)) + \lambda \|A(S)\|^2 \le L_S(\mathbf{w}^*) + \lambda \|\mathbf{w}^*\|^2 \ .$$

- Taking expectation of both sides with respect to S and noting that $\mathbb{E}_S[L_S(\mathbf{w}^*)] = L_{\mathcal{D}}(\mathbf{w}^*)$, we obtain that

$$\mathop{\mathbb{E}}_{S}[L_S(A(S))] \le L_{\mathcal{D}}(\mathbf{w}^*) + \lambda \|\mathbf{w}^*\|^2 \ .$$

- Therefore:

$$\mathop{\mathbb{E}}_{S}[L_{\mathcal{D}}(A(S))] \le L_{\mathcal{D}}(\mathbf{w}^*) + \lambda \|\mathbf{w}^*\|^2 + \frac{2\,\rho^2}{\lambda\, m}$$

The Regularization Path

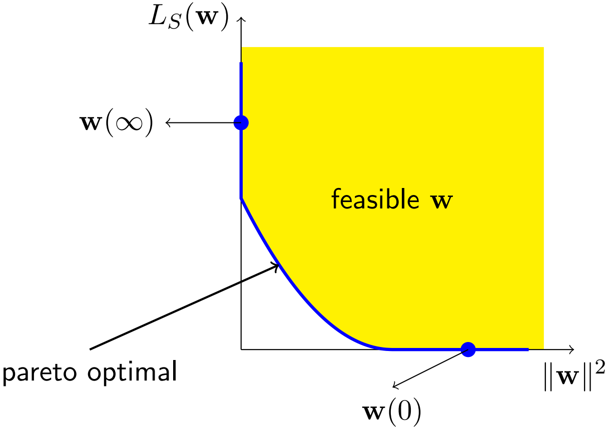
The RLM rule as a function of λ is $\mathbf{w}(\lambda) = \operatorname{argmin}_{\mathbf{w}} L_S(\mathbf{w}) + \lambda \|\mathbf{w}\|^2$

The Regularization Path

The RLM rule as a function of λ is $\mathbf{w}(\lambda) = \operatorname{argmin}_{\mathbf{w}} L_S(\mathbf{w}) + \lambda \|\mathbf{w}\|^2$
Can be seen as a pareto objective: minimize both $L_S(\mathbf{w})$ and $\|\mathbf{w}\|^2$

The Regularization Path

The RLM rule as a function of λ is $\mathbf{w}(\lambda) = \mathrm{argmin}_{\mathbf{w}} L_S(\mathbf{w}) + \lambda\|\mathbf{w}\|^2$
Can be seen as a pareto objective: minimize both $L_S(\mathbf{w})$ and $\|\mathbf{w}\|^2$



How to choose λ ?

- Bound minimization: choose λ according to the bound on $L_{\mathcal{D}}(\mathbf{w})$ usually far from optimal as the bound is worst case
- Validation: calculate several pareto optimal points on the regularization path (by varying λ) and use validation set to choose the best one

Outline

Dimension vs. Norm Bounds

- Previously in the course, when we learnt d parameters the sample complexity grew with d
- Here, we learn d parameters but the sample complexity depends on the norm of $\|\mathbf{w}^\star\|$ and on the Lipschitzness/smoothness, rather than on d
- Which approach is better depends on the properties of the distribution

Example: document categorization

Signs all encouraging for Phelps in comeback. He did not win any gold medals or set any world records but Michael Phelps ticked all the boxes he needed in his comeback to competitive swimming.

?

About sport ?

Bag-of-words representation

Signs all encouraging for Phelps in comeback. He did not win any gold medals or set any world records but Michael Phelps ticked all the boxes he needed in his comeback to competitive swimming.

0	0	1	0	0	1	0	0	0	0
		swimming			*world*		*elephant*		

Document categorization

- Let $\mathcal{X} = \{\mathbf{x} \in \{0,1\}^d : \|\mathbf{x}\|^2 \leq R^2, x_d = 1\}$

Document categorization

- Let $\mathcal{X} = \{\mathbf{x} \in \{0,1\}^d : \|\mathbf{x}\|^2 \le R^2, x_d = 1\}$
- Think on $\mathbf{x} \in \mathcal{X}$ as a text document represented as a bag of words:

Document categorization

- Let $\mathcal{X} = \{\mathbf{x} \in \{0,1\}^d : \|\mathbf{x}\|^2 \leq R^2, x_d = 1\}$
- Think on $\mathbf{x} \in \mathcal{X}$ as a text document represented as a bag of words:
  - At most $R^2 - 1$ words in each document

Document categorization

- Let $\mathcal{X} = \{\mathbf{x} \in \{0,1\}^d : \|\mathbf{x}\|^2 \le R^2, x_d = 1\}$
- Think on $\mathbf{x} \in \mathcal{X}$ as a text document represented as a bag of words:
    - At most $R^2 - 1$ words in each document
    - $d - 1$ is the size of the dictionary

Document categorization

- Let $\mathcal{X} = \{\mathbf{x} \in \{0,1\}^d : \|\mathbf{x}\|^2 \le R^2, x_d = 1\}$
- Think on $\mathbf{x} \in \mathcal{X}$ as a text document represented as a bag of words:
  - At most $R^2 - 1$ words in each document
  - $d - 1$ is the size of the dictionary
  - Last coordinate is the bias

Document categorization

- Let $\mathcal{X} = \{\mathbf{x} \in \{0,1\}^d : \|\mathbf{x}\|^2 \le R^2, x_d = 1\}$
- Think on $\mathbf{x} \in \mathcal{X}$ as a text document represented as a bag of words:
  - At most $R^2 - 1$ words in each document
  - $d - 1$ is the size of the dictionary
  - Last coordinate is the bias
- Let $\mathcal{Y} = \{\pm 1\}$ (e.g., the document is about sport or not)

Document categorization

- Let $\mathcal{X} = \{\mathbf{x} \in \{0,1\}^d : \|\mathbf{x}\|^2 \le R^2, x_d = 1\}$
- Think on $\mathbf{x} \in \mathcal{X}$ as a text document represented as a bag of words:
  - At most $R^2 - 1$ words in each document
  - $d - 1$ is the size of the dictionary
  - Last coordinate is the bias
- Let $\mathcal{Y} = \{\pm 1\}$ (e.g., the document is about sport or not)
- Linear classifiers $\mathbf{x} \mapsto \mathrm{sign}(\langle \mathbf{w}, \mathbf{x} \rangle)$

Document categorization

- Let $\mathcal{X} = \{\mathbf{x} \in \{0,1\}^d : \|\mathbf{x}\|^2 \le R^2, x_d = 1\}$
- Think on $\mathbf{x} \in \mathcal{X}$ as a text document represented as a bag of words:
  - At most $R^2 - 1$ words in each document
  - $d - 1$ is the size of the dictionary
  - Last coordinate is the bias
- Let $\mathcal{Y} = \{\pm 1\}$ (e.g., the document is about sport or not)
- Linear classifiers $\mathbf{x} \mapsto \mathrm{sign}(\langle \mathbf{w}, \mathbf{x} \rangle)$
- Intuitively: w_i is large (positive) for words indicative to sport while w_i is small (negative) for words indicative to non-sport

Document categorization

- Let $\mathcal{X} = \{\mathbf{x} \in \{0,1\}^d : \|\mathbf{x}\|^2 \le R^2, x_d = 1\}$
- Think on $\mathbf{x} \in \mathcal{X}$ as a text document represented as a bag of words:
  - At most $R^2 - 1$ words in each document
  - $d - 1$ is the size of the dictionary
  - Last coordinate is the bias
- Let $\mathcal{Y} = \{\pm 1\}$ (e.g., the document is about sport or not)
- Linear classifiers $\mathbf{x} \mapsto \mathrm{sign}(\langle \mathbf{w}, \mathbf{x} \rangle)$
- Intuitively: w_i is large (positive) for words indicative to sport while w_i is small (negative) for words indicative to non-sport
- Hinge-loss: $\ell(\mathbf{w}, (\mathbf{x}, y)) = [1 - y\langle \mathbf{w}, \mathbf{x} \rangle]_+$

Dimension vs. Norm

- VC dimension is d, but d can be extremely large (number of words in English)

Dimension vs. Norm

- VC dimension is d, but d can be extremely large (number of words in English)
- Loss function is convex and R Lipschitz

Dimension vs. Norm

- VC dimension is d, but d can be extremely large (number of words in English)
- Loss function is convex and R Lipschitz
- Assume that the number of relevant words is small, and their weights is not too large, then there is a $\mathbf{w}^\star$ with small norm and small $L_{\mathcal{D}}(\mathbf{w}^\star)$

Dimension vs. Norm

- VC dimension is d, but d can be extremely large (number of words in English)
- Loss function is convex and R Lipschitz
- Assume that the number of relevant words is small, and their weights is not too large, then there is a $\mathbf{w}^\star$ with small norm and small $L_{\mathcal{D}}(\mathbf{w}^\star)$
- Then, can learn it with sample complexity that depends on $R^2\|\mathbf{w}^\star\|^2$, and does not depend on d at all !

Dimension vs. Norm

- VC dimension is d, but d can be extremely large (number of words in English)
- Loss function is convex and R Lipschitz
- Assume that the number of relevant words is small, and their weights is not too large, then there is a $\mathbf{w}^\star$ with small norm and small $L_{\mathcal{D}}(\mathbf{w}^\star)$
- Then, can learn it with sample complexity that depends on $R^2\|\mathbf{w}^\star\|^2$, and does not depend on d at all !
- But, there are of course opposite cases, in which d is much smaller than $R^2\|\mathbf{w}^\star\|^2$

Summary

- Learning convex learning problems using SGD
- Learning convex learning problems using RLM
- The regularization path
- Dimension vs. Norm