Generalization in Deep Networks. II.

Peter Bartlett

UC Berkeley

March 20, 2019

Outline

- Uniform laws of large numbers
- Rademacher complexity and uniform laws
  (Concentration. Symmetrization. Restrictions.)
- Controlling Rademacher complexity:
  - Growth function
  - VC-dimension
  - Structural results for Rademacher complexity
- Neural networks
  - VC-dimension
  - Large margin classifiers
  - Rademacher averages for sigmoid networks
  - Rademacher averages for ReLU networks
- Interpolating prediction rules

VC-Dimension of Neural Networks

Theorem (Vapnik and Chervonenkis)

Suppose $\mathcal{F} \subseteq \{-1,1\}^{\mathcal{X}}$.
For every prob distribution P on $\mathcal{X} \times \{-1,1\}$,
with probability $1-\delta$ over n iid examples $(x_1,y_1),\ldots,(x_n,y_n)$,
every f in $\mathcal{F}$ satisfies

$$P(f(x) \neq y) \leq \frac{1}{n}\left|\{i : f(x_i) \neq y_i\}\right| + \left(\frac{c}{n}\left(\mathrm{VCdim}(\mathcal{F}) + \log(1/\delta)\right)\right)^{1/2}.$$

- For uniform bounds (that is, for all distributions and all $f \in \mathcal{F}$, proportions are close to probabilities), this inequality is tight within a constant factor.
- For neural networks, VC-dimension:
  - increases with number of parameters
  - depends on nonlinearity and depth

VC-Dimension of Neural Networks

Theorem

Consider the class $\mathcal{F}$ of $\{-1, 1\}$-valued functions computed by a network with L layers, p parameters, and k computation units with the following nonlinearities:

1. Piecewise constant (linear threshold units): $\mathrm{VCdim}(\mathcal{F}) = \tilde{O}(p)$.

   (Baum and Haussler, 1989)

2. Piecewise linear (ReLUs): $\mathrm{VCdim}(\mathcal{F}) = \tilde{O}(pL)$.

   (B., Harvey, Liaw, Mehrabian, 2017)

3. Piecewise polynomial: $\mathrm{VCdim}(\mathcal{F}) = \tilde{O}\left(pL^2\right)$.

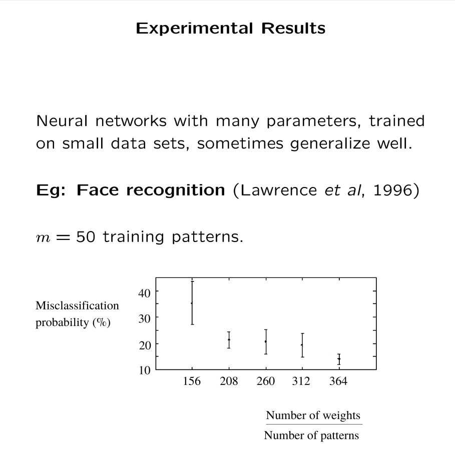
   (B., Maiorov, Meir, 1998)

4. Sigmoid: $\mathrm{VCdim}(\mathcal{F}) = \tilde{O}\left(p^2k^2\right)$.

   (Karpinsky and MacIntyre, 1994)

Generalization in Neural Networks: Number of Parameters

NeurIPS 1996



Experimental Results

Neural networks with many parameters, trained on small data sets, sometimes generalize well.

Eg: Face recognition (Lawrence *et al*, 1996)

$m = 50$ training patterns.

Misclassification probability (%)
40
30
20
10
156
208
260
312
364
Number of weights / Number of patterns

Outline

- Uniform laws of large numbers
- Rademacher complexity and uniform laws
  (Concentration. Symmetrization. Restrictions.)
- Controlling Rademacher complexity:
  - Growth function
  - VC-dimension
  - Structural results for Rademacher complexity
- Neural networks
  - VC-dimension
  - Large margin classifiers
  - Rademacher averages for sigmoid networks
  - Rademacher averages for ReLU networks
- Interpolating prediction rules

Large-Margin Classifiers

- Consider a real-valued function $f : \mathcal{X} \to \mathbb{R}$ used for classification.
- The prediction on $x \in \mathcal{X}$ is $\mathrm{sign}(f(x)) \in \{-1, 1\}$.
- For a pattern-label pair $(x, y) \in \mathcal{X} \times \{-1, 1\}$, if $yf(x) > 0$ then f classifies x correctly.
- We call $yf(x)$ the *margin* of f on x.
- We can view a larger margin as a more confident correct classification.
- Minimizing a continuous loss, such as

  $$\sum_{i=1}^{n} \left(f(X_i) - Y_i\right)^2,$$

  encourages large margins.
- For large-margin classifiers, we should expect the fine-grained details of f to be less important.

Generalization: Margins and Size of Parameters

Theorem (B., 1996)

1. With high probability over n training examples $(X_1, Y_1), \ldots, (X_n, Y_n) \in \mathcal{X} \times \{\pm 1\}$, every $f \in \mathcal{F} \subset \mathbb{R}^{\mathcal{X}}$ has

$$\Pr(\mathrm{sign}(f(X)) \neq Y) \leq \frac{1}{n} \sum_{i=1}^{n} 1[Y_i f(X_i) \leq \gamma] + \tilde{O}\left(\sqrt{\frac{\mathrm{fat}_{\mathcal{F}}(\gamma)}{n}}\right).$$

2. If functions in $\mathcal{F}$ are computed by L-layer sigmoid networks with each unit's weights bounded in 1-norm, that is, $\|w\|_1 \leq B$, then

$$\mathrm{fat}_{\mathcal{F}}(\gamma) = \tilde{O}((B/\gamma)^L).$$

- The bound depends on the margin loss plus a complexity term.
- Minimizing quadratic loss or cross-entropy loss leads to large margins.
- $\mathrm{fat}_{\mathcal{F}}(\gamma)$ is a scale-sensitive version of VC-dimension. Unlike the VC-dimension, it need not grow with the number of parameters.

Outline

- Uniform laws of large numbers
- Rademacher complexity and uniform laws
  (Concentration. Symmetrization. Restrictions.)
- Controlling Rademacher complexity:
  - Growth function
  - VC-dimension
  - Structural results for Rademacher complexity
- Neural networks
  - VC-dimension
  - Large margin classifiers
  - Rademacher averages for sigmoid networks
  - Rademacher averages for ReLU networks
- Interpolating prediction rules

Recall: Rademacher Complexity Structural Results

Theorem

1. $F \subseteq G$ implies $\|R_n\|_F \le \|R_n\|_G$.
2. $\|R_n\|_{cF} = |c|\|R_n\|_F$.
3. For $|g(X)| \le 1$, $|\mathbb{E}\|R_n\|_{F+g} - \mathbb{E}\|R_n\|_F| \le \sqrt{2\log 2/n}$.
4. $\|R_n\|_{\mathrm{co}F} = \|R_n\|_F$, where $\mathrm{co}F$ is the convex hull of F.
5. If $\phi : \mathbb{R} \times \mathcal{Z}$ has $\alpha \mapsto \phi(\alpha, z)$ 1-Lipschitz for all z and $\phi(0, z) = 0$, then for $\phi(F) = \{z \mapsto \phi(f(z), z)\}$, $\mathbb{E}\|R_n\|_{\phi(F)} \le 2\mathbb{E}\|R_n\|_F$.

Rademacher Complexity for Lipschitz Loss

Example

To analyze ERM over $F : \mathcal{X} \to \mathcal{Y}$ with loss ℓ, we want $\|P - P_n\|_{\ell_F}$ small, where

$$\ell_F := \{(x, y) \mapsto \ell(f(x), y) : f \in F\},$$

If $\ell(\cdot, y)$ is 1-Lipschitz, then we can define $\phi(\alpha, (x, y)) = \ell(\alpha, y) - \ell(0, y)$ and

$$\begin{aligned}\phi(F) &= \{(x, y) \mapsto \ell(f(x), y) - \ell(0, y) : f \in F\} \\ &= \ell_F - \ell_0.\end{aligned}$$

Then (5) implies $\mathbb{E}\|R_n\|_{\phi(F)} \leq 2\mathbb{E}\|R_n\|_F$.
And if $|\ell| \leq 1$, (3) implies $\mathbb{E}\|R_n\|_{\ell_F} \leq 2\mathbb{E}\|R_n\|_F + \sqrt{2 \log 2/n}$.

Rademacher Complexity for Lipschitz Loss

- Classification loss is not Lipschitz!
- Consider the $1/\gamma$-Lipschitz loss

$$\phi(\alpha) = \begin{cases} 1 & \text{if } \alpha \le 0, \\ 1 - \alpha/\gamma & \text{if } 0 < \alpha < \gamma, \\ 0 & \text{if } \alpha \ge 1. \end{cases}$$

- Large margin loss is an upper bound and classification loss is a lower bound:

$$1[Yf(X) \le 0] \le \phi(Yf(X)) \le 1[Yf(X) \le \gamma].$$

- So if we can relate the Lipschitz risk $P\phi(Yf(X))$ to the Lipschitz empirical risk $P_n\phi(Yf(X))$, we have a large margin bound:

$$P1[Yf(X) \le 0] \le P\phi(Yf(X)) \text{ c.f. } P_n\phi(Yf(X)) \le P_n 1[Yf(X) \le \gamma].$$

Rademacher Complexity for Lipschitz Loss

$$\begin{aligned}
P1[Yf(X) \leq 0] &\leq P\phi(Yf(X)) \\
&\leq P_n\phi(Yf(X)) + \frac{c}{\gamma}\mathbb{E}\|R_n\|_F + O(1/\sqrt{n}) \\
&\leq P_n1[Yf(X) \leq \gamma] + \frac{c}{\gamma}\mathbb{E}\|R_n\|_F + O(1/\sqrt{n})
\end{aligned}$$

with high probability.
Notice that we've turned a classification problem into a regression problem.
The VC-dimension (which captures arbitrarily fine-grained properties of the function class) is no longer important.
This is only an upper bound, but there are comparison theorems that relate the *excess* risk to the excess ϕ-risk.

Rademacher Averages for Sigmoid Networks

Theorem

Consider the following class $\mathcal{F}_B$ of two-layer neural networks:

$$\mathcal{F}_B = \left\{ x \mapsto \sum_{i=1}^{k} w_i \sigma\left(v_i^T x \right) : w_i \geq 0, \|w\|_1 \leq B, \|v_i\|_1 \leq B, k \geq 1 \right\},$$

where $B > 0$ and the nonlinear function $\sigma : \mathbb{R} \to \mathbb{R}$ satisfies the Lipschitz condition, $|\sigma(a) - \sigma(b)| \leq |a - b|$, and $\sigma(0) = 0$. Suppose that the distribution is such that $\|X\|_\infty \leq 1$ a.s. Then

$$\mathbb{E}\|R_n\|_{\mathcal{F}_B} \leq B^2 \sqrt{\frac{2 \log 2d}{n}},$$

where d is the dimension of the input space, $\mathcal{X} = \mathbb{R}^d$.

Rademacher Averages for Sigmoid Networks: Proof

Recall the notation

$$\mathrm{co}(F) = \left\{ \sum_{i=1}^{k} \alpha_i f_i \,:\, k \geq 1, \alpha_i \geq 0, \|\alpha\|_1 = 1, f_i \in F \right\}.$$

Define

$$\mathcal{G} := \{(x_1, \ldots, x_d) \mapsto x_j : 1 \leq j \leq d\},$$

$$\begin{aligned} \mathcal{V}_B &:= \left\{ x \mapsto v'x \,:\, \|v\|_1 = \sum_{i=1}^{d} |v_i| \leq B \right\} \\ &= B\,\mathrm{co}\left(\{0\} \cup \mathcal{G} \cup -\mathcal{G}\right) \\ &= B\,\mathrm{co}\left(\mathcal{G} \cup -\mathcal{G}\right) \end{aligned}$$

Rademacher Averages for Sigmoid Networks: Proof

$$
\begin{aligned}
\mathcal{F}_B &= \left\{ x \mapsto \sum_{i=1}^{k} w_i \sigma(v_i(x)) \mid k \geq 1, w_i \geq 0, \sum_{i=1}^{k} w_i \leq B, v_i \in \mathcal{V}_B \right\} \\
&= B \operatorname{co}\left(\{0\} \cup \sigma \circ \mathcal{V}_B\right) = B \operatorname{co}\left(\sigma \circ \mathcal{V}_B\right)
\end{aligned}
$$

$$
\begin{aligned}
R_n(\mathcal{F}_B) &= R_n\left(B \operatorname{co}\left(\sigma \circ \mathcal{V}_B\right)\right) \\
&= B R_n\left(\operatorname{co}\left(\sigma \circ \mathcal{V}_B\right)\right) \\
&= B R_n\left(\sigma \circ \mathcal{V}_B\right) \\
&\leq B R_n(\mathcal{V}_B) \\
&= B R_n\left(B \operatorname{co}\left(\mathcal{G} \cup -\mathcal{G}\right)\right) \\
&= B^2 R_n\left(\mathcal{G} \cup -\mathcal{G}\right) \\
&\leq B^2 \sqrt{\frac{2 \log(2d)}{n}}.
\end{aligned}
$$

Outline

- Uniform laws of large numbers
- Rademacher complexity and uniform laws
  (Concentration. Symmetrization. Restrictions.)
- Controlling Rademacher complexity:
  - Growth function
  - VC-dimension
  - Structural results for Rademacher complexity
- Neural networks
  - VC-dimension
  - Large margin classifiers
  - Rademacher averages for sigmoid networks
  - Rademacher averages for ReLU networks
- Interpolating prediction rules

ReLU Networks

- The sigmoid nonlinearity is convenient, because it ensures boundedness (in ℓ_∞) of the inputs to each layer.
- What about nonlinearities like the ReLU's, which is Lipschitz, but unbounded?
- We also need to keep control of the scale of the vectors that are computed throughout the network.

Networks with Lipschitz Nonlinearities

Theorem (B., Foster, Telgarsky, 2017)

With high probability over n training examples $(X_1, Y_1), \ldots, (X_n, Y_n) \in \mathcal{X} \times \{\pm 1\}$, every f_W with $R_W \leq r$ has

$$\Pr(\mathrm{sign}(f(X)) \neq Y) \leq \frac{1}{n} \sum_{i=1}^{n} 1[Y_i f(X_i) \leq \gamma] + \tilde{O}\left(\frac{rL}{\gamma\sqrt{n}}\right).$$

Here, f_W is computed in a network with L layers and parameters $W_1, \ldots, W_L$:

$$f_W(x) := \sigma_L(W_L \sigma_{L-1}(W_{L-1} \cdots \sigma_1(W_1 x) \cdots)),$$

where the σ_i are 1-Lipschitz, and we measure the scale of f_W using a product of norms of the matrices W_i,
for example, $r := \prod_{i=1}^{L} \|W_i\|_* \left(\sum_{i=1}^{L} \frac{\|W_i\|_{2,1}^{2/3}}{\|W_i\|_*^{2/3}}\right)^{3/2}$.

The proof uses a covering numbers argument.

ReLU Networks

Using the positive homogeneity property of the ReLU nonlinearity (that is, for all $\alpha \geq 0$ and $x \in \mathbb{R}$, $\sigma(\alpha x) = \alpha\sigma(x)$) gives an elegant argument (due to Gollowich, Rakhlin and Shamir) to bound the Rademacher complexity.

Theorem

With high probability over n training examples $(X_1, Y_1), \ldots, (X_n, Y_n) \in \mathcal{X} \times \{\pm 1\}$ with $\|X_i\| \leq 1$ a.s., every $f \in \mathcal{F}^F_{L,B}$ has

$$R_n(\mathcal{F}_{F,B}) \leq \frac{(2B)^L}{\sqrt{n}},$$

where $f \in \mathcal{F}_{F,B}$ is an L-layer network of the form

$$\mathcal{F}_{F,B} := W_L\sigma(W_{L-1} \cdots \sigma(W_1 x) \cdots),$$

σ is 1-Lipschitz, positive homogeneous (that is, for all $\alpha \geq 0$ and $x \in \mathbb{R}$, $\sigma(\alpha x) = \alpha\sigma(x)$), and applied componentwise, and $\|W_i\|_F \leq B$.
(W_L is a row vector.)

ReLU Networks: Proof

(Write $\mathbb{E}_\epsilon$ as the conditional expectation given the data.)

Lemma

$$\mathbb{E}_\epsilon \sup_{f \in F, \|W\|_F \le B} \frac{1}{n} \left\| \sum_{i=1}^n \epsilon_i \sigma(Wf(X_i)) \right\|_2 \le 2B \mathbb{E}_\epsilon \sup_{f \in F} \frac{1}{n} \left\| \sum_{i=1}^n \epsilon_i f(X_i) \right\|_2 .$$

Iterating this and using Jensen's inequality proves the theorem:

$$\begin{aligned} \mathbb{E}\left[\frac{1}{n} \left\| \sum_{i=1}^n \epsilon_i X_i \right\|_2 \middle| X_1, \ldots, X_n \right] &\le \frac{1}{n} \sqrt{\mathbb{E}\left[\left\| \sum_{i=1}^n \epsilon_i X_i \right\|_2^2 \middle| X_1, \ldots, X_n \right]} \\ &= \frac{1}{n} \sqrt{\sum_{i=1}^n \|X_i\|_2^2} \le \frac{1}{\sqrt{n}}. \end{aligned}$$

ReLU Networks: Proof

For $W^\top = (w_1 \cdots w_k)$, we use positive homogeneity:

$$\left\| \sum_{i=1}^{n} \epsilon_i \sigma(Wf(x_i)) \right\|^2 = \sum_{j=1}^{k} \left(\sum_{i=1}^{n} \epsilon_i \sigma(w_j^\top f(x_i)) \right)^2$$
$$= \sum_{j=1}^{k} \|w_j\|^2 \left(\sum_{i=1}^{n} \epsilon_i \sigma \left(\frac{w_j^\top}{\|w_j\|} f(x_i) \right) \right)^2,$$

and

$$\sup_{\|W\|_F \le B} \sum_{j=1}^{k} \|w_j\|^2 \left(\sum_{i=1}^{n} \epsilon_i \sigma \left(\frac{w_j^\top}{\|w_j\|} f(x_i) \right) \right)^2$$
$$= \sup_{\|w_j\|=1; \|\alpha\|_1 \le B^2} \sum_{j=1}^{k} \alpha_j \left(\sum_{i=1}^{n} \epsilon_i \sigma \left(w_j^\top f(x_i) \right) \right)^2 = B^2 \sup_{\|w\|=1} \left(\sum_{i=1}^{n} \epsilon_i \sigma \left(w^\top f(x_i) \right) \right)^2,$$

then apply the Ledoux-Talagrand contraction and Cauchy-Schwartz inequalities.

Outline

- Uniform laws of large numbers
- Rademacher complexity and uniform laws
  (Concentration. Symmetrization. Restrictions.)
- Controlling Rademacher complexity:
  - Growth function
  - VC-dimension
  - Structural results for Rademacher complexity
- Neural networks
  - VC-dimension
  - Large margin classifiers
  - Rademacher averages for sigmoid networks
  - Rademacher averages for ReLU networks
- Interpolating prediction rules

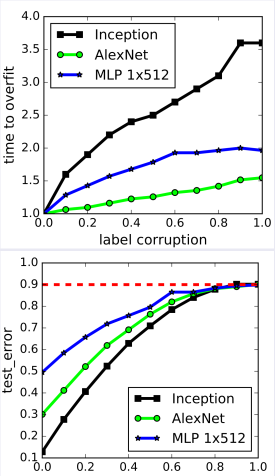
Generalization: Margins and Size of Parameters

- A *classification* problem becomes a *regression* problem if we use a loss function that doesn't vary too quickly.
- For regression, the complexity of a neural network is controlled by the *size* of the parameters, and can be independent of the number of parameters.
- We have a tradeoff between the fit to the training data (margins) and the complexity (size of parameters):

$$\Pr(\mathrm{sign}(f(X)) \neq Y) \leq \frac{1}{n}\sum_{i=1}^{n} \ell(Y_i, f(X_i)) + p_n(\mathcal{F})$$

- Even if the training set is classified correctly, it might be worthwhile to increase the complexity, to improve this loss function.

Interpolation in Deep Networks:
A New Challenge for Statistical Learning Theory



(Zhang, Bengio, Hardt, Recht, Vinyals, 2017)

- Deep networks can be trained to zero training error (for *regression* loss)
- ... with near state-of-the-art performance
- ... even for noisy problems.
- No tradeoff between fit to training data and complexity!

$$\Pr(\mathrm{sign}(f(X)) \neq Y) \leq \frac{1}{n}\sum_{i=1}^{n} \ell(Y_i, f(X_i)) + p_n(\mathcal{F})$$

(Belkin, Hsu, Ma, Mandal, 2018)

Interpolating Prediction Rules

Progress on interpolating prediction

- Interpolating nearest neighbor rules in high dimensions

  (Belkin, Hsu, Mitra, 2018)

- Kernel regression with polynomial kernels

  (Liang and Rakhlin, 2018)

- Kernel smoothing with singular kernels

  (Belkin, Rakhlin, Tsybakov, 2018)

Interpolation in Linear Regression

Phil Long

Gábor Lugosi

Alexander Tsigler

Linear regression

- Training data $(x_1, y_1), \ldots, (x_n, y_n) \in \mathbb{R}^p \times \mathbb{R}$.
- Linear functions: $f_\theta(x) = x^\top \theta$.
- Squared error: $\ell(y, f_\theta(x)) = (y - f_\theta(x))^2$.
- Least squares linear prediction: θ^* minimizes $\mathbb{E}\ell(y, f_\theta(x))$.
- Choose $\hat{\theta}$ to interpolate: $\displaystyle \frac{1}{n}\sum_{i=1}^{n} \ell(y_i, f_\theta(x_i)) = 0$.

  Hence, $y_1 = f_{\hat{\theta}}(x_1), \ldots, y_n = f_{\hat{\theta}}(x_n)$ (need $p \geq n$).
- Which interpolating f_θ? Choose $\hat{\theta}$ to minimize $\|\theta\|$.

Interpolation in Linear Regression

Think of this optimization as

$$\begin{aligned} \min_{\theta} \quad & \|\theta\| \\ \text{s.t.} \quad & \sum_{i=1}^{n} \ell(y_i, f_\theta(x_i)) \leq C, \end{aligned}$$

with $C = 0$. Compare this to

$$\min_{\theta} \quad \sum_{i=1}^{n} \ell(y_i, f_\theta(x_i)) + \lambda\|\theta\|,$$

$$\text{or} \quad \begin{aligned} \min_{\theta} \quad & \sum_{i=1}^{n} \ell(y_i, f_\theta(x_i)) \\ \text{s.t.} \quad & \|\theta\| \leq B. \end{aligned}$$

Interpolation in Linear Regression

We have

$$\begin{aligned}\hat{\theta} &= (X^\top X)^\dagger X^\top y \\ &= (X^\top X)^\dagger X^\top (X\theta^* + \epsilon),\end{aligned}$$

so

$$\begin{aligned}&\mathbb{E}(x^\top \hat{\theta} - y)^2 - \mathbb{E}(x^\top \theta^* - y)^2 \\ &= \mathbb{E}{\theta^*}^\top \left(I - \hat{\Sigma}\hat{\Sigma}^\dagger\right)\left(\Sigma - \hat{\Sigma}\right)\left(I - \hat{\Sigma}^\dagger\hat{\Sigma}\right)\theta^* + \mathbb{E}\mathrm{Tr}\left(\Sigma\left(X^\top X\right)^\dagger\right).\end{aligned}$$

Interpolation in Linear Regression

Interpolation for linear prediction

- Excess expected loss, $\mathbb{E}\ell(y, f_{\hat{\theta}}(x)) - \mathbb{E}\ell(y, f_{\theta^*}(x))$ has two components: (corresponding to $f_{\theta*}(x)$ and $y - f_{\theta*}(x)$)
  1. $\hat{\theta}$ is a distorted version of θ^*, because the sample $x_1, \dots, x_n$ distorts our view of the covariance of x.

     Not a problem, even in high dimensions ($p > n$).
  2. $\hat{\theta}$ is corrupted by the noise in $y_1, \dots, y_n$.

     Problematic in high dimensions.
- When can we hide the label noise in $\hat{\theta}$ without hurting predictive accuracy?

Interpolation in Linear Regression

Accurate interpolating prediction as dimension p_n grows

- Split the covariance of x into two pieces:
    - a big piece of dimension k, and
    - a 'tail' (of dimension $p_n - k$))—that gets longer and flatter with n.
- Denote the variance in the first k directions as $\lambda_1 \geq \cdots \geq \lambda_k$,
- and the variance in the 'tail' directions as $\lambda_{k+1} \geq \cdots \geq \lambda_{p_n}$.
- Denote $r_k(\Sigma) = \dfrac{1}{\lambda_{k+1}} \displaystyle\sum_{i=k+1}^{p_n} \lambda_i$.

  (This is the scale of the variance tail, relative to its highest variance.)
- Also write $r_0(\Sigma) = \dfrac{1}{\lambda_1} \displaystyle\sum_{i=1}^{p_n} \lambda_i$.

Interpolation in Linear Regression

Theorem

If $k = o(n)$ and the 'tail' is long and flat:

- a small proportion of variance in any direction, $r_k(\Sigma) = \omega(n)$, that is, $\frac{\lambda_{k+1}}{\sum_{i>k} \lambda_i} = o(1/n)$,
- total variance not too large, $r_0(\Sigma) = o(n)$,

then for jointly gaussian (x, y),

$$\mathbb{E}\ell(y, f_{\hat{\theta}}(x)) - \mathbb{E}\ell(y, f_{\theta^*}(x)) = \tilde{O}\left(\sqrt{\frac{r_0(\Sigma)}{n}} + \frac{n}{r_k(\Sigma)} + \frac{k}{n}\right) \to 0,$$

where $r_k(\Sigma) = \frac{1}{\lambda_{k+1}} \sum_{i=k+1}^{\infty} \lambda_i$.

There is also a (weaker) lower bound in terms of $n/r_k(\Sigma)$.

Interpolating Prediction

- Interpolation: far from the regime of a tradeoff between fit to training data and complexity.
- In high-dimensional linear regression, if the covariance has a long, flat tail, the minimum norm interpolant can hide the noise in these many unimportant directions.
  - Relies on overparameterization
  - ... and lots of unimportant parameters
- Can we extend these results to interpolating deep networks?
  - What is the analog of the minimum norm linear prediction rule?
  - What role does the optimization method play?

Outline

- Uniform laws of large numbers
- Rademacher complexity and uniform laws
  (Concentration. Symmetrization. Restrictions.)
- Controlling Rademacher complexity:
    - Growth function
    - VC-dimension
    - Structural results for Rademacher complexity
- Neural networks
    - VC-dimension
    - Large margin classifiers
    - Rademacher averages for sigmoid networks
    - Rademacher averages for ReLU networks
- Interpolating prediction rules: A new challenge for statistical learning theory