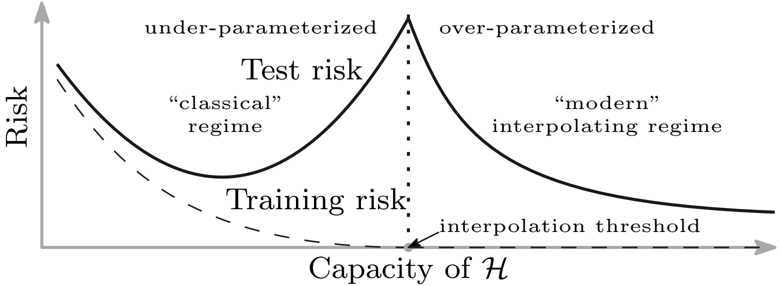
UPPSALA UNIVERSITET

No double descent in PCA: Training and pre-training in high dimensions

Daniel Gedon
Uppsala University

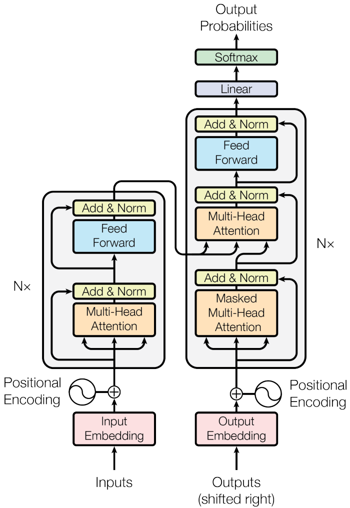
Belkin Lab weekly meeting
San Diego, March 06, 2023

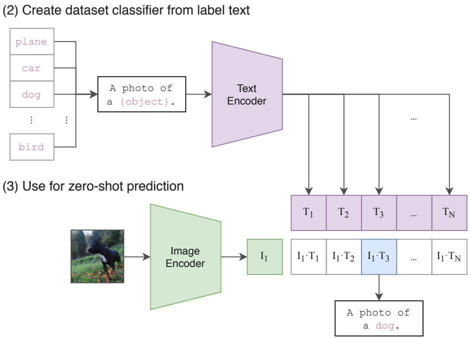


Found/analysed in: linear/logistic regression, random forests, adversarial training, neural networks, ...

Belkin et al., "Reconciling modern machine-learning practice and the classical bias–variance trade-off".

Introduction – Encoder-decoder models





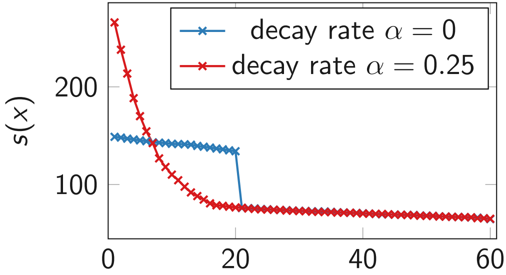
Vaswani et al., "Attention is all you need".
Radford et al., "Learning transferable visual models from natural language supervision".

Data generator

$$\begin{tikzcd}[column sep=small] & \boldsymbol{z}_i \in \mathbb{R}^d \arrow[dl] \arrow[dr] & \\ \boldsymbol{x}_i \in \mathbb{R}^m & & y_i \in \mathbb{R} \end{tikzcd}$$

$$\boldsymbol{x}_i = \boldsymbol{D}\boldsymbol{z}_i + \boldsymbol{e}_i$$

$$y_i = \boldsymbol{\theta}\boldsymbol{z}_i + \varepsilon_i$$



Model

$$\begin{tikzcd}[column sep=small] \boldsymbol{x}_i \arrow[dr, "\text{PCA}"'] & & \hat{y}_i \\ & \hat{\boldsymbol{z}}_i \in \mathbb{R}^{\hat{d}} \arrow[ur, "\text{lin. reg.}"'] & \end{tikzcd}$$

SVD: $\boldsymbol{X} \approx \hat{\boldsymbol{U}}\hat{\boldsymbol{\Sigma}}\hat{\boldsymbol{V}}^\top,$

PCA: $\hat{\boldsymbol{z}}_i = \hat{\boldsymbol{V}}^\top \boldsymbol{x}_i,$

lin. reg. $\hat{y}_i = \hat{\boldsymbol{\theta}}^\top \hat{\boldsymbol{z}}_i.$

Interested in risk: $R(\hat{\boldsymbol{\theta}}) = \mathbb{E}_{y_0}\left[(y_0 - \hat{y}_0)^2\right]$.
Write data generator directly from features to outputs as $\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}$ with $\boldsymbol{\beta} \in \mathbb{R}^m$.

Lemma

Sample covariance $\hat{\boldsymbol{C}} = \frac{1}{n}\boldsymbol{X}^\top\boldsymbol{X}$ and the true covariance $\boldsymbol{C}$. Orthogonal projectors $\boldsymbol{\Pi} = \boldsymbol{I}_m - \hat{\boldsymbol{V}}\hat{\boldsymbol{V}}^\top$. Then,

$$\mathbb{E}_{\boldsymbol{\epsilon}}\left[R(\hat{\boldsymbol{\theta}})\right] = \boldsymbol{\beta}^\top\boldsymbol{\Pi}\boldsymbol{C}\boldsymbol{\Pi}\boldsymbol{\beta} + \frac{\sigma_\epsilon^2}{n}\operatorname{Tr}(\hat{\boldsymbol{V}}^\top\boldsymbol{C}\hat{\boldsymbol{V}}\hat{\boldsymbol{V}}^\top\hat{\boldsymbol{C}}^+\hat{\boldsymbol{V}}) + \sigma_\epsilon^2.$$

Compare with Hastie et al. for direct linear regression:

$$\mathbb{E}_{\boldsymbol{\epsilon}}\left[R(\hat{\boldsymbol{\theta}})\right] = \boldsymbol{\beta}^\top\boldsymbol{\Pi}\boldsymbol{C}\boldsymbol{\Pi}\boldsymbol{\beta} + \frac{\sigma_\epsilon^2}{n}\operatorname{Tr}(\boldsymbol{C}\hat{\boldsymbol{C}}^+) + \sigma_\epsilon^2.$$

Hastie et al., "Surprises in high-dimensional ridgeless least squares interpolation".

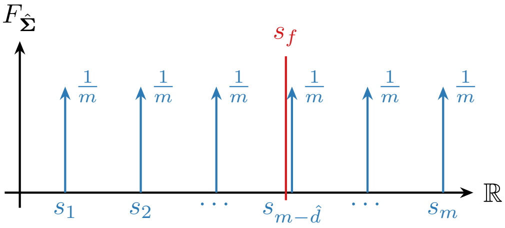
UPPSALA UNIVERSITET

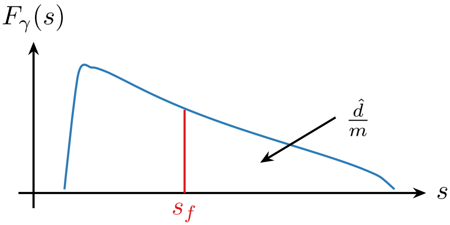
Theorem

Assume isotropic features $\boldsymbol{C} = \boldsymbol{I}_m$, which implies $d = m$ and choose constant $\hat{d}$. Then, as $m, n \to \infty$, such that $\frac{m}{n} \to \gamma$, the expected risk satisfies almost surely

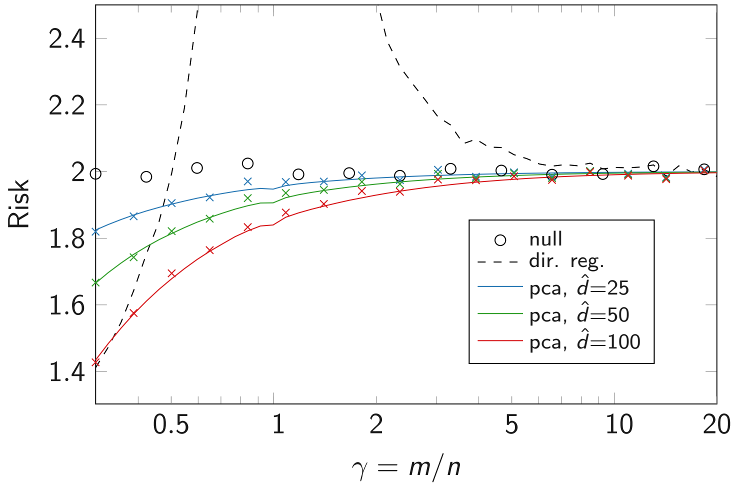
$$\mathbb{E}_\epsilon\left[R(\hat{\boldsymbol{\theta}})\right] \to \sigma_\epsilon^2 \frac{m}{n} \int_{s_f}^{\infty} \frac{1}{s} dF_\gamma(s) + \sigma_\epsilon^2 + \begin{cases} \beta^\top \beta \left(1 - \min(\hat{d}, m)/m\right) & \text{for} \quad \gamma < 1 \\ \beta^\top \beta \left(1 - \min(\hat{d}, n)/m\right) & \text{for} \quad \gamma > 1 \end{cases}$$

with F_γ the Marčenko-Pastur law and s_f the value in $\mathbb{R}$ that satisfies $\frac{\hat{d}}{m} = \int_{s_f}^{\infty} dF_\gamma$.

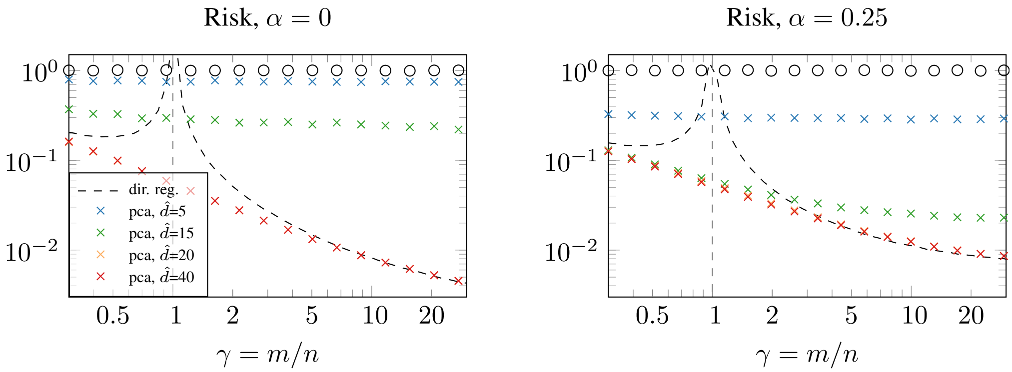




Isotropic data: Theorem from last slide.

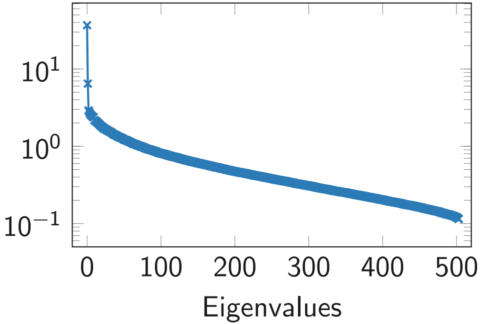


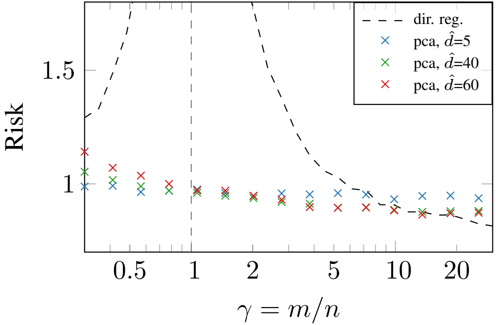
Latent variable data: No theorem but empirical results.



Real world example: Diverse MAGIC wheat genetics data set.

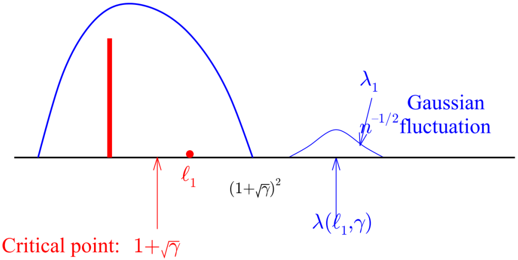
Input: genome sequence (=1.1M nucleotides) of 504 wheat lines.
Outcome: real-valued phenotypes.





Aim: Asymptotic risk for latent variable data generator.

Data generator = spiked covariance model $\boldsymbol{C} = \sigma_0 \boldsymbol{I} + \sum_{i=0}^{K} s_i \boldsymbol{v}_i \boldsymbol{v}_i^\top$



Need eigenvector product: $\mathbb{E}_\epsilon \left[R(\hat{\boldsymbol{\theta}}) \right] = \boldsymbol{\beta}^\top \boldsymbol{\Pi} \boldsymbol{C} \boldsymbol{\Pi} \boldsymbol{\beta} + \frac{\sigma_\epsilon^2}{n} \operatorname{Tr}(\hat{\boldsymbol{V}}^\top \boldsymbol{C} \hat{\boldsymbol{V}} \hat{\boldsymbol{V}}^\top \hat{\boldsymbol{C}}^+ \hat{\boldsymbol{V}}) + \sigma_\epsilon^2.$

Results:

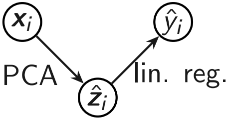
$$(\hat{\boldsymbol{v}}_i^\top \boldsymbol{v}_i)^2 \to \begin{cases} \frac{1-\gamma/(\ell_i-1)^2}{1-\gamma/(\ell_i-1)} & \text{for } \ell_i > 1+\sqrt{\gamma} \\ 0 & \text{for } \ell_i \in [1, 1+\sqrt{\gamma}] \end{cases}$$

Johnstone and Paul, "PCA in High Dimensions: An Orientation".
Paul, "Asymptotics of sample eigenstructure for a large dimensional spiked covariance model".

Pre-training the PCA – Setup

Use two data sets:

- Pre-training data set $\{\boldsymbol{x}_i\}_{i=1}^{n_p}$
- Training data set $\{\boldsymbol{x}_i, y_i\}_{i=1}^{n}$



Two step training procedure:

1. Unsupervised pre-training of PCA.
2. Train linear regression on the PCA features $\hat{\boldsymbol{z}}_i$.

For technical reasons: orthogonalize features and noise $\boldsymbol{x}_i = \boldsymbol{D}\boldsymbol{z}_i + \boldsymbol{D}_\perp \boldsymbol{e}_i$. Then:

$$\begin{aligned} \text{Model:} \qquad & \hat{\boldsymbol{z}}_i = \hat{\boldsymbol{V}}^\top \boldsymbol{x}_i, \\ \text{Data generator:} \qquad & \boldsymbol{z}_i = \boldsymbol{D}^+(\boldsymbol{x}_i - \boldsymbol{D}_\perp \boldsymbol{e}_i) = \boldsymbol{D}^+ \boldsymbol{x}_i. \end{aligned}$$

Define projection loss:

$$\mathcal{L}(\boldsymbol{D}) = \mathbb{E}\left[\|\boldsymbol{x}\|_2^2 - \|\boldsymbol{D}^+\boldsymbol{x}\|_2^2\right]; \qquad \mathcal{L}(\hat{\boldsymbol{V}}) = \mathbb{E}\left[\|\boldsymbol{x}\|_2^2 - \|\hat{\boldsymbol{V}}^\top\boldsymbol{x}\|_2^2\right]$$

Lemma

$$\mathcal{L}(\hat{\boldsymbol{V}}) - \mathcal{L}(\boldsymbol{D}) = \sum_{i=1}^{\min(d,\hat{d})} \sum_{j=1}^{m} (\hat{\boldsymbol{v}}_i^\top \boldsymbol{v}_j)^2 (s_i - s_j) + \underbrace{\sum_{i=\hat{d}}^{d} s_i}_{=0 \text{ for } \hat{d} \geq d} + \underbrace{\sum_{i=d}^{\hat{d}} \sum_{j=1}^{m} (\hat{\boldsymbol{v}}_i^\top \boldsymbol{v}_j)^2 s_j}_{=0 \text{ for } \hat{d} \leq d}.$$

$\rightarrow$ Correct estimation of eigenvectors $\hat{\boldsymbol{V}}$ crucial for small loss difference.

Pre-training the PCA – Analysis

UPPSALA UNIVERSITET

Theorem

Take $t > 0$, $k_j^2 = s_j(s_j + \mathrm{Tr}(\boldsymbol{C}))$, then

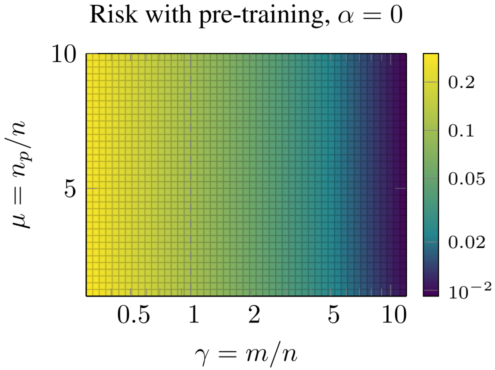
$$P\left(\mathcal{L}(\hat{\boldsymbol{V}}) - \mathcal{L}(\boldsymbol{D}) > t\right) \leq$$
$$\leq \frac{4}{t\, n_p}\left(\sum_{i=1}^{\min(d,\hat{d})}\sum_{j=i+1}^{m}\frac{k_j^2}{|s_i - s_j|} + \sum_{i=\hat{d}}^{d}\sum_{j=1}^{m}\frac{k_j^2 s_i}{(s_i - s_j)^2} + \sum_{i=d}^{\hat{d}}\sum_{j=1}^{m}\frac{k_j^2 s_j}{(s_i - s_j)^2}\right).$$

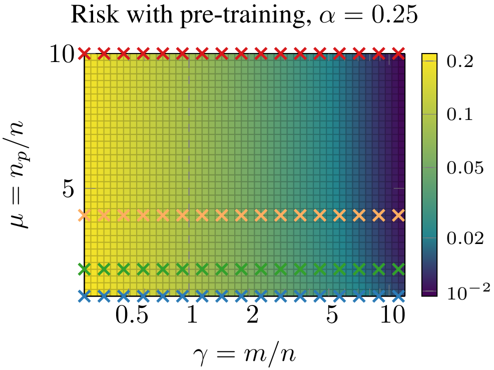
Tighter bound by:

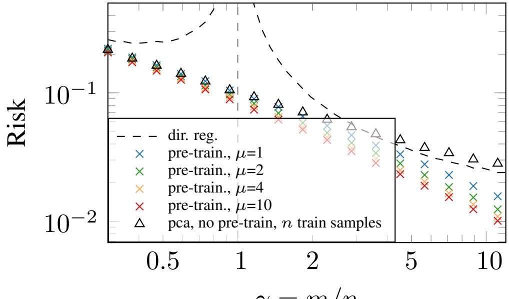
- rapidly decaying eigenvalues $\rightarrow$ large $|s_i - s_j| \geq 0$,
- more pre-training samples n_p.

Loukas, "How close are the eigenvectors of the sample and actual covariance matrices?"

Pre-training the PCA – Numerical results







Pre-training the PCA – Open problem

UPPSALA UNIVERSITET

Aim: Closed form for the risk with pre-training.

Move away from sample complexity towards asymptotic results.

$\rightarrow$ Use results for $(\hat{\boldsymbol{v}}_i^\top \boldsymbol{v}_i)^2 \rightarrow \begin{cases} \frac{1-\gamma/(\ell_i-1)^2}{1-\gamma/(\ell_i-1)} & \text{for } \ell_i > 1+\sqrt{\gamma} \\ 0 & \text{for } \ell_i \in [1, 1+\sqrt{\gamma}] \end{cases}$

to quantify asymptotic risk for spiked covariance model

Extensions:

- Use real-world nonlinear data
- Neural networks: multi-layer approaches
- Nonlinear setting: replace PCA with kernel PCA
- Distribution shift between pre-training and training / testing

Paul, "Asymptotics of sample eigenstructure for a large dimensional spiked covariance model".

Conclusion

Supervised case:

- Generalize results from Hastie et al. for direct regression
- Selecting sufficiently large latent dimension $\hat{d}$ is crucial for low risk

$\rightarrow$ formal guarantees for performance of PCA-regression on real-world data structures

Pre-training:

- more pre-training data only help to improve eigenvector estimates
- certain decay rate α is necessary such that more pre-training data are helpful

Thank you!

Daniel Gedon, Uppsala University

E-mail: `daniel.gedon@it.uu.se`
Web: `dgedon.github.io`
Twitter: `@danigedon`

Supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by Knut and Alice Wallenberg Foundation.