http://goo.gl/$XIlNMN$

Course website

Kyoto University

Statistical Machine Learning Theory

Sparsity

Hisashi Kashima
kashima@i.Kyoto-u.ac.jp

Department of Intelligence Science and Technology

Topics:
Learning with sparsity

- L_1 regularization & Lasso
- Reduced rank regression

Lasso

Regression: Prediction of a continuous target variable

- Training dataset $\{(\mathbf{x}^{(1)}, y^{(1)}), \dots, (\mathbf{x}^{(N)}, y^{(N)})\}$
  - $\mathbf{x}^{(i)} \in \mathbb{R}^D$: feature vector
  - $y^{(i)} \in \mathbb{R}$: real-valued target value
- Linear regression model: $y = \mathbf{w}^\top \mathbf{x}$
- Least square solution:

$$\mathbf{w}^* = \mathrm{argmin}_{\mathbf{w}} \sum_{i=1}^{N} \left(y^{(i)} - \mathbf{w}^\top \mathbf{x}^{(i)}\right)^2$$
$$= \mathrm{argmin}_{\mathbf{w}} \|\mathbf{y} - \boldsymbol{X}\mathbf{w}\|_2^2$$
$$= (\boldsymbol{X}^\top \boldsymbol{X})^{-1} \boldsymbol{X}^\top \mathbf{y}$$

$$\boldsymbol{X} = \left(\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(N)}\right)^\top$$
$$\mathbf{y} = \left(y^{(1)}, y^{(2)}, \dots, y^{(N)}\right)^\top$$

Ridge regression:
L_2-Regularization for avoiding overfitting

- Overfitting to the training data
  - Especially when the training data is small compared with the input space dimensionality
- Regularized least square solution:

$$\begin{aligned}\mathbf{w}^* &= \mathrm{argmin}_{\mathbf{w}} \|\mathbf{y} - \boldsymbol{X}\mathbf{w}\|_2^2 + \gamma\|\mathbf{w}\|_2^2 \\ &= (\boldsymbol{X}^\top \boldsymbol{X} + \gamma \boldsymbol{I})^{-1}\boldsymbol{X}^\top \mathbf{y}\end{aligned}$$

  - $\|\mathbf{w}\|_2^2 = w_1^2 + w_2^2 + \cdots + w_D^2$: L_2-regularization term

L_1-regularization: A sparsity-inducing regularization

- Over-fitting sometimes occurs even with L_2-regularization
  - when the dimensionality is extremely large
  - when the true model uses only a small number of features
- L_1-regularization
  - $\|\mathbf{w}\|_1 = |w_1| + |w_2| + \cdots + |w_D|$: L_1-regularization term leads to sparse solutions
    - Sparse: Many w_d becomes 0 in the solutions
    - High interpretability and easy-to-implementability
  - L_1-regularized least square linear regression (LASSO):

$$\mathbf{w}^* = \mathrm{argmin}_{\mathbf{w}} \|\mathbf{y} - \boldsymbol{X}\mathbf{w}\|_2^2 + \gamma \|\mathbf{w}\|_1$$

Why does L_1-regularization induce sparse solutions?: Some intuitive explanations

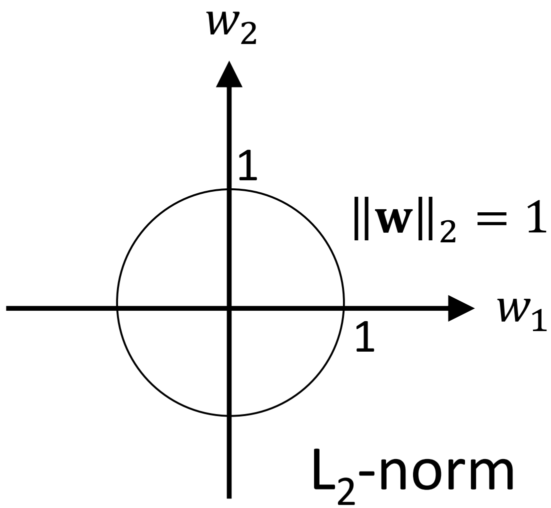
- L_1-regularization is equivalent to L_1-norm constraint:

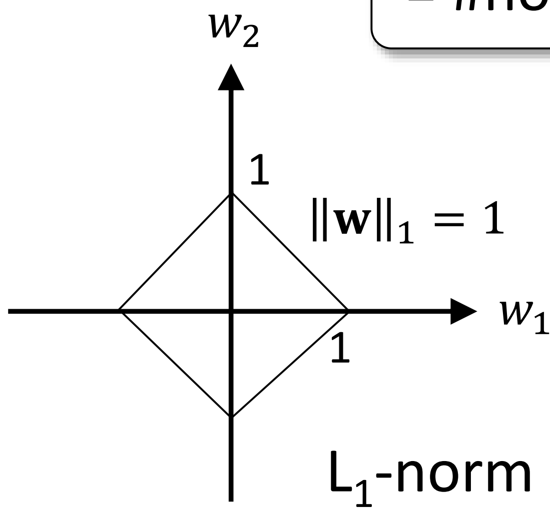
$$\mathrm{argmin}_{\mathbf{w}}\, f(\mathbf{w}) + \gamma\|\mathbf{w}\|_1 \Leftrightarrow \mathrm{argmin}_{\mathbf{w}}\, f(\mathbf{w}) \text{ s.t. } \|\mathbf{w}\|_1 \leq \lambda$$

- Some intuitive explanations for sparsity:

1. L_1-norm is a convex alternative to L_0-norm
2. Level curves of norms

= #nonzero elements





L_1-regularized least square linear regression: No closed-form solutions

- L_1-regularized least square linear regression (LASSO):

$$\mathbf{w}^* = \operatorname{argmin}_{\mathbf{w}} \|\mathbf{y} - \boldsymbol{X}\mathbf{w}\|_2^2 + \gamma \|\mathbf{w}\|_1$$

  – L_1-regularization with a convex loss function is a convex optimization problem

- LASSO has no closed form solution…
  ⇒ needs iterative solutions, e.g.:

  1. Optimization with respect to only one dimension
  2. Reduction to L_2-regularization

we will discuss this

An algorithm for lasso: Repeat optimization w.r.t only one dimension

- L_1-regularization term is cumbersome since:
  - it is not differentiable at $w_d = 0$
  - $w_d = 0$ tends to be a solution
- Observation: The objective function is easy to optimize if we focus only on a single dimension (e.g. w_d)
- Iterative algorithm:
  1. Choose an arbitrary d
  2. Optimize w_d (has a closed form solution)
  3. Repeat steps 1&2 until convergence

One dimensional optimization problem for LASSO: Sum of a quadratic function & an absolute value function

- L_1-regularized least square linear regression (LASSO):

$$\mathbf{w}^* = \mathrm{argmin}_{\mathbf{w}} \|\mathbf{y} - \boldsymbol{X}\mathbf{w}\|_2^2 + \gamma\|\mathbf{w}\|_1$$

- Consider optimization w.r.t. only w_d:
  - $w_d{}^* = \mathrm{argmin}_{w_d}\, q(w_d) + \gamma|w_d|$
    - $q(w_d) = a(w_d - \widetilde{w}_d)^2 + b \;\; (a > 0)$: quadratic function
      - $\widetilde{w}_d$ is the minimizer of $q(w_d)$ i.e. the solution of the one-variable optimization when $\gamma|w_d|$ is neglected

- Finally what we want is

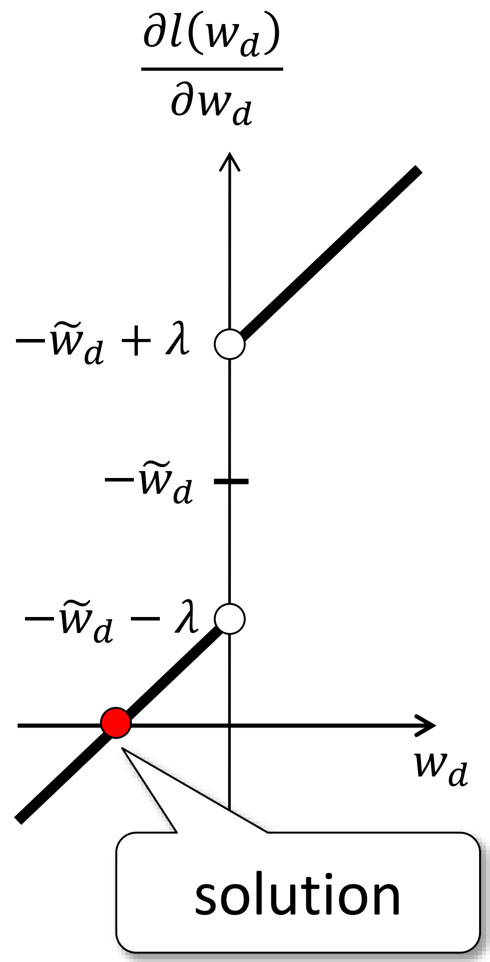
$$w_d{}^* = \mathrm{argmin}_{w_d}\, \frac{1}{2}(w_d - \widetilde{w}_d)^2 + \lambda|w_d| \quad \left(\lambda = \frac{1}{2a}\gamma\right)$$

Solution of the one-dimensional optimization: Find the stationary point

- Find the minimizer of $l(w_d) = \frac{1}{2}(w_d - \widetilde{w}_d)^2 + \lambda |w_d|$

- Taking the derivative of $l(w_d)$,

$$\frac{\partial l(w_d)}{\partial w_d} = \begin{cases} w_d - \widetilde{w}_d + \lambda & (\text{if } w_d > 0) \\ w_d - \widetilde{w}_d - \lambda & (\text{if } w_d < 0) \\ \text{undefined} & (\text{otherwise}) \end{cases}$$



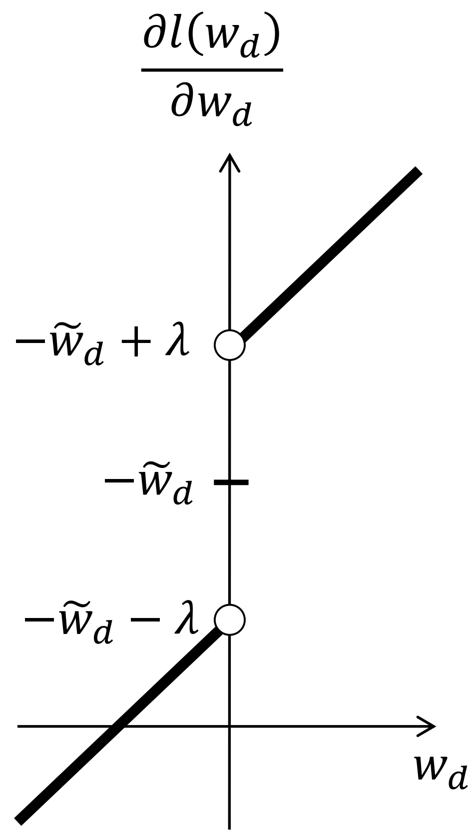
- Solution: $w_d = w_d^*$ s.t. $\left.\frac{\partial l(w_d)}{\partial w_d}\right|_{w_d = w_d^*} = 0$
  - lies at $\frac{\partial l(w_d)}{\partial w_d}$ hits the x-axis

Sparsity of lasso solutions:
Solutions close to zero are rounded to zero

- We have 3 cases:

1. $-\widetilde{w}_d + \lambda < 0$ (i.e. $\widetilde{w}_d > \lambda$),
   - Solution: $w_d^* = \widetilde{w}_d - \lambda$
2. $-\widetilde{w}_d - \lambda > 0$ (i.e. $\widetilde{w}_d < -\lambda$),
   - Solution: $w_d^* = \widetilde{w}_d + \lambda$
3. $-\lambda \le \widetilde{w}_d \le \lambda$
   - Solution: $w_d^* = 0$
     - if $w_d^* > 0$, we have a contradiction $\left.\frac{\partial l(w_d)}{\partial w_d}\right|_{w_d = w_d^*} = w_d^* - \widetilde{w}_d + \lambda = 0 \Rightarrow w_d^* = \widetilde{w}_d - \lambda \le 0$
     - Similarly, assuming $w_d{}^* < 0$ yields a contradiction $w_d^* \ge 0$

sparse solution



Dimension Reduction

Multivariate regression:
Prediction of multiple continuous variables

- Multivariate regression is a regression problem to predict multiple output variables
  - $f: \mathbb{R}^D \Rightarrow \mathbb{R}^{D'}$
- Training dataset $\left\{ \left(\mathbf{x}^{(1)}, \mathbf{y}^{(1)}\right), \dots, \left(\mathbf{x}^{(N)}, \mathbf{y}^{(N)}\right) \right\}$
  - $\mathbf{x}^{(i)} \in \mathbb{R}^D$: feature vector
  - $\mathbf{y}^{(i)} \in \mathbb{R}^{D'}$: real-valued target values
- Multivariate linear regression model: $\mathbf{y} = \boldsymbol{W}^\top \mathbf{x}$
  - $\boldsymbol{W} \in \mathbb{R}^{D' \times D}$: Matrix parameter

Solution of multivariate regression: Closed form least square solution

- Least square solution:

$$\boldsymbol{W}^* = \mathrm{argmin}_{\boldsymbol{W}\in\mathbb{R}^{D'\times D}} \sum_{i=1}^{N} \left\|\mathbf{y}^{(i)} - \boldsymbol{W}^\top \mathbf{x}^{(i)}\right\|_2^2$$

$$= \mathrm{argmin}_{\boldsymbol{W}} \|\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{W}\|_\mathrm{F}^2$$

$$= (\boldsymbol{X}^\top\boldsymbol{X})^{-1}\boldsymbol{X}^\top\boldsymbol{Y}$$

$$\boldsymbol{X} = \left(\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(N)}\right)^\top$$

$$\boldsymbol{Y} = \left(\mathbf{y}^{(1)}, \mathbf{y}^{(2)}, \dots, \mathbf{y}^{(N)}\right)^\top$$

$$\frac{\partial \mathrm{tr}(\boldsymbol{A}\boldsymbol{B})}{\partial \boldsymbol{A}} = \boldsymbol{B}^\top$$

- Regularized version
  - $\|\boldsymbol{W}\|_\mathrm{F}^2 = \sum_{(i,j)} w_{ij}^2$: L_2-regularization term
  - $\boldsymbol{W}^* = (\boldsymbol{X}^\top\boldsymbol{X} + \gamma\boldsymbol{I})^{-1}\boldsymbol{X}^\top\boldsymbol{Y}$

Reduced rank regression: Multivariate regression with rank constraint

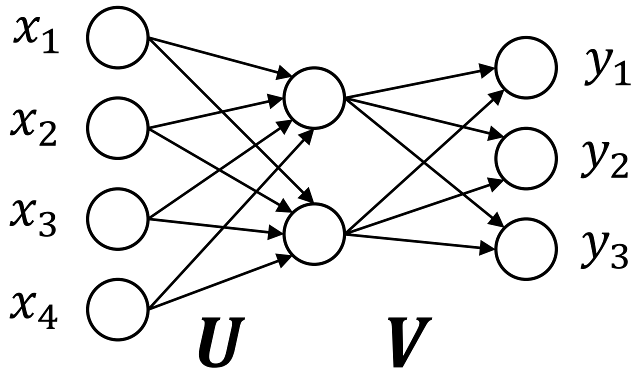
- Multivariate regression is equivalent to D'-independent univariate regressions
  - exploits no shared information
- Low-rank assumption $\boldsymbol{W} = \boldsymbol{U}\boldsymbol{V}^\top$
  - $\boldsymbol{U} \in \mathbb{R}^{D \times K}, \boldsymbol{V} \in \mathbb{R}^{D' \times K}$ i.e. rank of $\boldsymbol{W}$ is K
    - $K < \min(D, D')$
  - D' output variables share K–dimensional latent space
- Reduced rank regression:

$$\boldsymbol{W}^* = \mathrm{argmin}_{\boldsymbol{W}} \|\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{W}\|_{\mathrm{F}}^2 \text{ s.t. } \mathrm{rank}(\boldsymbol{W}) \le K$$

Sparsity in reduced rank regression: Sparse parameters in terms of matrix singular values

- Parameter $\boldsymbol{W}$ in the reduced rank regression $\mathbf{y} = \boldsymbol{W}^\top \mathbf{x}$ is dense in terms of matrix elements

- $\boldsymbol{W}$ is sparse in terms of singular values
  - $\boldsymbol{W} = \boldsymbol{U}\boldsymbol{V}^\top$ is low-rank
    - $\boldsymbol{U} \in \mathbb{R}^{D \times K}, \boldsymbol{V} \in \mathbb{R}^{D' \times K}, K < \min(D, D')$
  - Rank = L_0 norm of singular values: $\mathrm{rank}(\boldsymbol{W}) = \|\boldsymbol{\sigma}(\boldsymbol{W})\|_0$



Solution of reduced rank regression (1/2):
Best rank-K approximation of a matrix

- Objective function to be minimized:

$$\|\boldsymbol{Y}-\boldsymbol{X}\boldsymbol{W}\|_{\mathrm{F}}^2 = \mathrm{tr}\{(\boldsymbol{Y}-\boldsymbol{X}\boldsymbol{W})^\top(\boldsymbol{Y}-\boldsymbol{X}\boldsymbol{W})\}$$
$$= \mathrm{tr}\{\boldsymbol{Y}^\top\boldsymbol{Y}-2\boldsymbol{W}^\top\boldsymbol{X}^\top\boldsymbol{Y}+\boldsymbol{W}^\top\boldsymbol{X}^\top\boldsymbol{X}\boldsymbol{W}\}$$

(Let $\boldsymbol{X}^\top\boldsymbol{X}=\boldsymbol{P}^\top\boldsymbol{\Lambda}\boldsymbol{P}$ be the eigendecomposition)

$\boldsymbol{P}^\top\boldsymbol{P}=\boldsymbol{P}\boldsymbol{P}^\top=\boldsymbol{I}$
($\boldsymbol{P}$: orthogonal)

$$= \mathrm{tr}\left\{\boldsymbol{Y}^\top\boldsymbol{Y}-2\widetilde{\boldsymbol{W}}^\top\boldsymbol{\Lambda}^{-\frac{1}{2}}\boldsymbol{P}\boldsymbol{X}^\top\boldsymbol{Y}+\widetilde{\boldsymbol{W}}^\top\widetilde{\boldsymbol{W}}\right\}$$

where $\widetilde{\boldsymbol{W}}=\boldsymbol{\Lambda}^{\frac{1}{2}}\boldsymbol{P}\boldsymbol{W}$

$$= \left\|\widetilde{\boldsymbol{W}}-\boldsymbol{\Lambda}^{-\frac{1}{2}}\boldsymbol{P}\boldsymbol{X}^\top\boldsymbol{Y}\right\|_{\mathrm{F}}^2+\text{const.}$$

- Find the best rank-K approximation of $\boldsymbol{\Lambda}^{-\frac{1}{2}}\boldsymbol{P}\boldsymbol{X}^\top\boldsymbol{Y}$

Solution of reduced rank regression (2/2):
Closed form solution using SVD

- The best rank-K approximation of $\boldsymbol{\Lambda}^{-\frac{1}{2}}\boldsymbol{P}\boldsymbol{X}^\top\boldsymbol{Y}$ is given as $\widetilde{\boldsymbol{W}}^* = \boldsymbol{U}^*\boldsymbol{\Sigma}^*\boldsymbol{V}^{*\top}$
  - $\boldsymbol{V}^*$ is top-K eigenvectors of
$$\boldsymbol{Y}^\top\boldsymbol{X}\boldsymbol{P}^\top\boldsymbol{\Lambda}^{-\frac{1}{2}}\boldsymbol{\Lambda}^{-\frac{1}{2}}\boldsymbol{P}\boldsymbol{X}^\top\boldsymbol{Y} = \boldsymbol{Y}^\top\boldsymbol{X}(\boldsymbol{X}^\top\boldsymbol{X})^{-1}\boldsymbol{X}^\top\boldsymbol{Y}$$
  - $\boldsymbol{\Sigma}^*$: a diagonal matrix with K largest singular values
  - $\boldsymbol{U}^* = \boldsymbol{\Lambda}^{-\frac{1}{2}}\boldsymbol{P}\boldsymbol{X}^\top\boldsymbol{Y}\,\boldsymbol{V}^*\boldsymbol{\Sigma}^{*-1}$
- The solution is $\boldsymbol{W}^* = \boldsymbol{P}^\top\boldsymbol{\Lambda}^{-\frac{1}{2}}\widetilde{\boldsymbol{W}}^* = \boldsymbol{P}^\top\boldsymbol{\Lambda}^{-\frac{1}{2}}\,\boldsymbol{U}^*\boldsymbol{\Sigma}^*\boldsymbol{V}^{*\top} = (\boldsymbol{X}^\top\boldsymbol{X})^{-1}\boldsymbol{X}^\top\boldsymbol{Y}\boldsymbol{V}^*\boldsymbol{V}^{*\top}$

[Supplement 1] Eigenvalue decomposition of symmetric matrix

- $\boldsymbol{A} = \boldsymbol{P}^\top \boldsymbol{\Lambda} \boldsymbol{P}$: eigen-decomposition of symmetric matrix $\boldsymbol{A}$
  - $\boldsymbol{\Lambda}$: diagonal matrix $\boldsymbol{\Lambda} = \mathrm{diag}(\lambda_1, \lambda_2, \dots, \lambda_D)$, where $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_D \geq 0$ (eigenvalues)
  - $\boldsymbol{P}$: orthogonal matrix $\boldsymbol{P}^\top \boldsymbol{P} = \boldsymbol{P}\boldsymbol{P}^\top = \boldsymbol{I}$

[Supplement 2] Singular value decomposition (SVD) and best rank-K approximation :

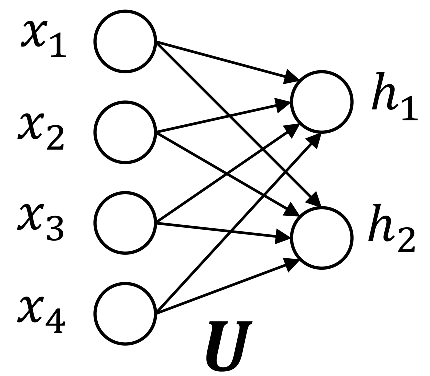
- $\boldsymbol{B} = \boldsymbol{U\Sigma V}^\top$: SVD of rank- R real matrix $\boldsymbol{B}$
  - $\boldsymbol{\Sigma}$: diagonal matrix $\boldsymbol{\Sigma} = \mathrm{diag}(\sigma_1, \sigma_2, \dots, \sigma_R, 0, \dots, 0)$, where $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_D \geq 0$ (singular values)
    - $\boldsymbol{\Sigma}$ is sqrt of eigenvalues of $\boldsymbol{B}\boldsymbol{B}^\top$ or $\boldsymbol{B}^\top\boldsymbol{B}$
  - $\boldsymbol{U}$, $\boldsymbol{V}$: orthogonal matrices
    - $\boldsymbol{U}$ is eig.vecs of $\boldsymbol{B}\boldsymbol{B}^\top$, $\boldsymbol{V}$ is eig.vecs of $\boldsymbol{B}^\top\boldsymbol{B}$, $\mathbf{u}_i = \frac{1}{\sigma_i}\boldsymbol{B}^\top\mathbf{v}_i$
- Best rank-K approximation problem of matrix $\boldsymbol{B}$:

$$\widehat{\boldsymbol{B}}^* = \mathrm{argmin}_{\widehat{\boldsymbol{B}}} \left\| \boldsymbol{B} - \widehat{\boldsymbol{B}} \right\|_{\mathrm{F}}^2 \text{ s.t. } \mathrm{rank}\left(\widehat{\boldsymbol{B}}\right) \leq K$$

  - Find K largest singular values $\boldsymbol{\Sigma}^* = \mathrm{diag}(\sigma_1, \dots, \sigma_K)$, and corresponding vectors $\boldsymbol{U}^* = (\mathbf{u}_1, \dots, \mathbf{u}_K)$, $\boldsymbol{V}^* = (\mathbf{v}_1, \dots, \mathbf{v}_K)$, and let $\widehat{\boldsymbol{B}}^* = \boldsymbol{U}^*\boldsymbol{\Sigma}^*\boldsymbol{V}^{*\top}$

Dimension reduction:
Find low-dimensional representations of high-dim. data

- Dimension reduction:
  - Find a low-dimensional mapping $f: \mathbb{R}^D \Rightarrow \mathbb{R}^K$ $(D > K)$
    - for interpretability, computational/space efficiency, generalization abilities, …
    - (Lossy) compression: keep the original information as much as possible
- Linear dimension reduction: $\mathbf{h} = \boldsymbol{U}^\top \mathbf{x}$
  - $\boldsymbol{U} : D \times K$ matrix



Basic idea behind dimension reduction: Find a coding & decoding function for lossy compression

- Coding and decoding process:

$$\mathbf{x} \xrightarrow[\text{coding}]{f} \mathbf{h} \xrightarrow[\text{decoding}]{g} \tilde{\mathbf{x}}$$

- If f and g are appropriately designed so that $\mathbf{x} \fallingdotseq \tilde{\mathbf{x}}$,
  $\mathbf{h}$ must be a good low-dimensional representation of $\mathbf{x}$

- Optimization problem
  - $(f, g) = \mathrm{argmin}_{f,g} \sum_{i=1}^{N} \mathrm{loss}\left(\mathbf{x}^{(i)}, g\left(f\left(\mathbf{x}^{(i)}\right)\right)\right)$

Principal component analysis: Dimension reduction using reduced rank regression

- Linear dimension reduction with coding & decoding functions
  - linear coding function $f : \mathbf{h} = \boldsymbol{U}^\top \mathbf{x}$ ($\boldsymbol{U} : D \times K$ matrix)
  - linear decoding function $g : \tilde{\mathbf{x}} = \boldsymbol{V}\mathbf{h}$ ($\boldsymbol{V} : K \times D$ matrix)
  - $\tilde{\mathbf{x}} = \boldsymbol{V}\boldsymbol{U}^\top \mathbf{x}$
- Reduced rank regression finds the solution by taking the training dataset as $\{ (\mathbf{x}^{(1)}, \mathbf{x}^{(1)}), \ldots, (\mathbf{x}^{(N)}, \mathbf{x}^{(N)}) \}$
  - Solution will be $\boldsymbol{V} = \boldsymbol{U}^\top$