Regression

Zoltán Szabó – CMAP, École Polytechnique

Data Science @ HEC Paris
May 7, 2019

Contact information

- Email:

  zoltan (dot) szabo (at) polytechnique (dot) edu

- Web:

  http://www.cmap.polytechnique.fr/~zoltan.szabo/

Outline

- Linear regression.
- Regularization:
  - Ridge regression.
  - Sparse coding, Lasso, group Lasso.
- Non-linear extension.

Examples

House pricing

- How much is our house worth?

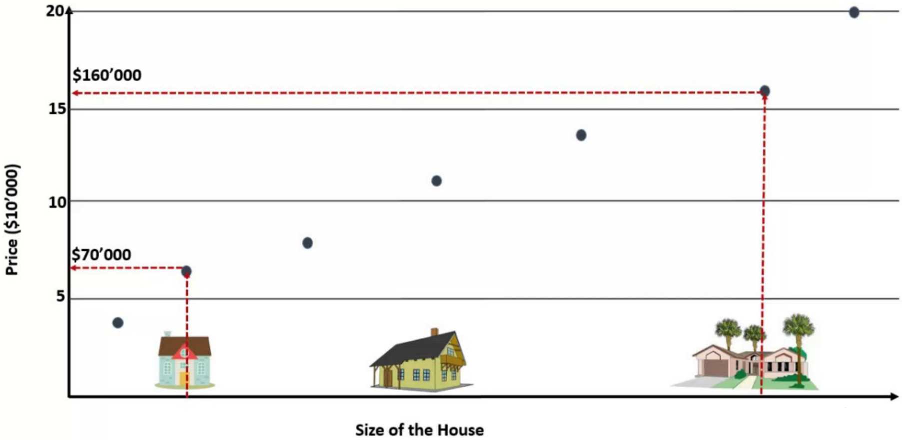
House pricing

- How much is our house worth?
- We can check on the market:
  - various house s, their characteristics , and
  - prices.

Plot

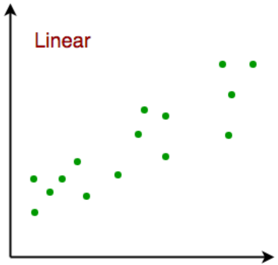
Let us plot: house price vs. size!

House pricing: (size, price) relation = ?



House pricing

- House price: y.
- Feature of the house (square meter): x.
- Dataset: $\{(x_i, y_i)\}_{i=1}^n$.
- Goal: $f = ?$ such that $f(x_i) \approx y_i$; example: $f(x) = b_0 + b_1 x$.



House pricing: multiple features

Probably size itself is not enough for accurate prediction.

$\mathbf{x}$ =

- size (m^2),
- # of bathrooms,
- # of bedrooms,
- year built,
- # of floors,
- parking type,
- heating,
- cooling,
- microwave, ...

Example: $f(\mathbf{x}) = \langle \mathbf{b}, \mathbf{x} \rangle = \sum_{i=1}^{p} b_i x_i$. ($x_0 = 1$: ok)

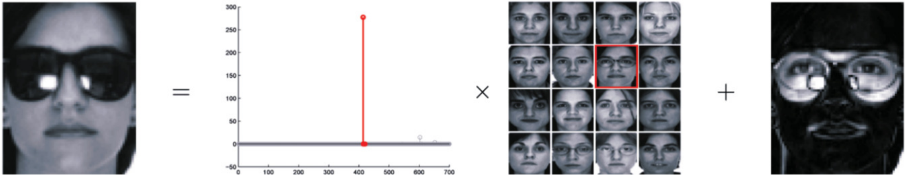
Feature selection: relevant features = ?

Too many features might be hard to interpret / overfitting $\Rightarrow$ simple models.

Goal: find

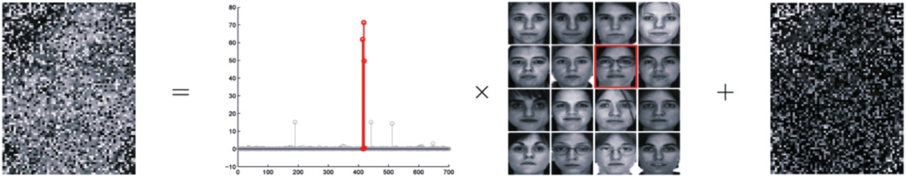
- the feature subset most relevant for house price prediction.

Demo: face recognition.



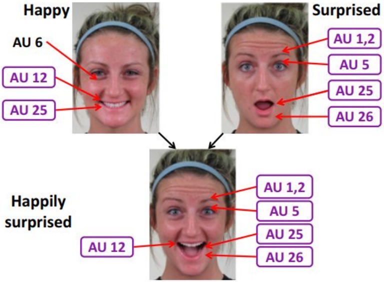
Idea:

- test image = sparse linear combination of the training set + error
- error = corruption/occlusion.



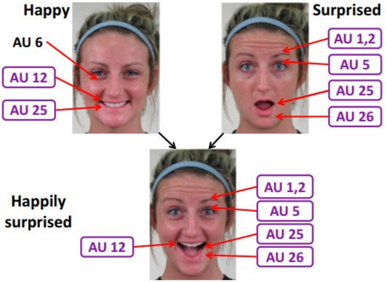
- Nice performance despite severe corruption.

FACS (facial action coding system):



The prototypes can be time series: emotion recognition

FACS (facial action coding system):

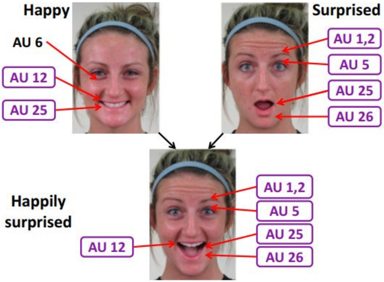


Idea:

- muscle activities $\mapsto$ emotion (happy/surprise/...),

The prototypes can be time series: emotion recognition

FACS (facial action coding system):

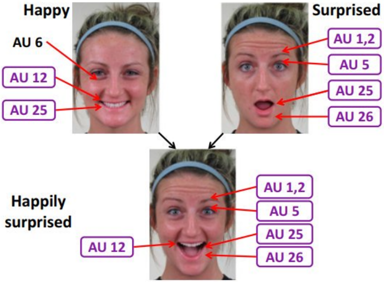


Idea:

- muscle activities $\mapsto$ emotion (happy/surprise/...),
- using time series: prediction is more accurate.

The prototypes can be time series: emotion recognition

FACS (facial action coding system):

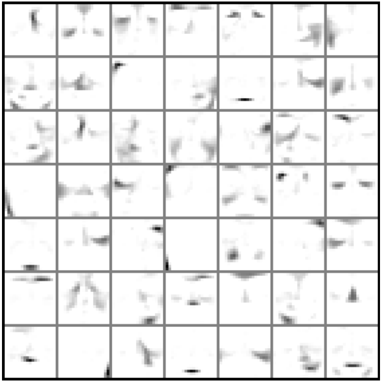


Idea:

- muscle activities $\mapsto$ emotion (happy/surprise/...),
- using time series: prediction is more accurate.
- The same trick helps in action recognition: writing, sports, games (Wii), ...

Additional structure: non-negativity (NMF)

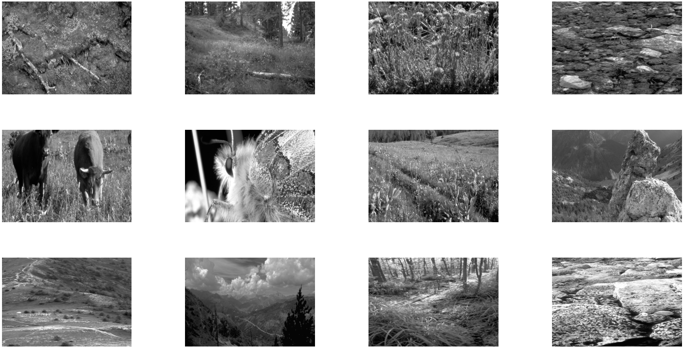
How to impose additional structure?



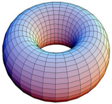
- y_i: i^{th} face image.
- $\mathbf{b}$, $\mathbf{x}$: non-negative; $\mathbf{x}$ is also learned.

Additional structure: structured sparsity

Natural images:

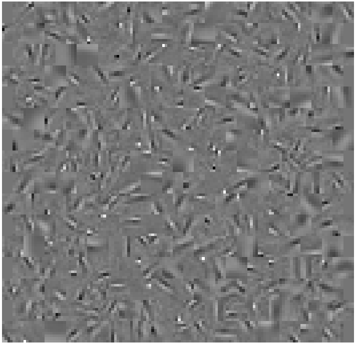


Structure: torus



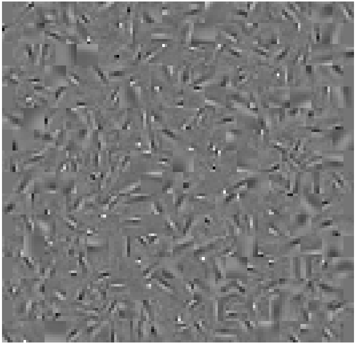
Resulting dictionary elements

Dictionary: sparse

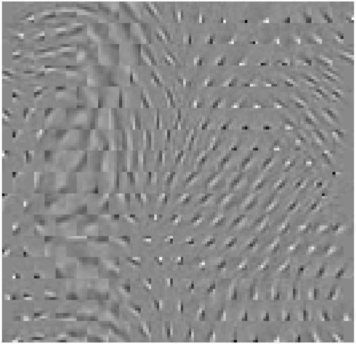


Resulting dictionary elements

Dictionary: sparse



structured sparse



The resulting dictionary: in action

- Inpainting: new image - never seen!

The resulting dictionary: in action

- Inpainting: new image - never seen!

- PSNR (peak signal-to-noise ratio):
  - bigger is better,
  - in wireless communication: $20 - 25$ dB,
  - in image & video compression: 30 dB.

The resulting dictionary: in action

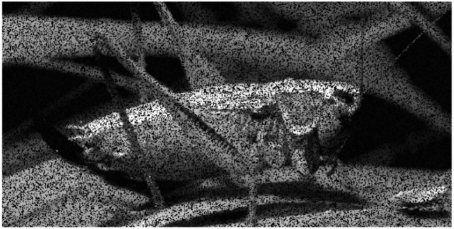
- Inpainting: new image - never seen!

- PSNR (peak signal-to-noise ratio):
  - bigger is better,
  - in wireless communication: $20 - 25$ dB,
  - in image & video compression: 30 dB.

We only show to the algorithm a fraction of the pixels!

Illustration

30% of the pixels is missing



Illustration

30% of the pixels is missing (PSNR $=$ 36 dB):

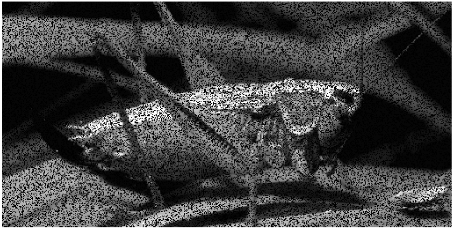


Illustration: continued

70% of the pixels is missing

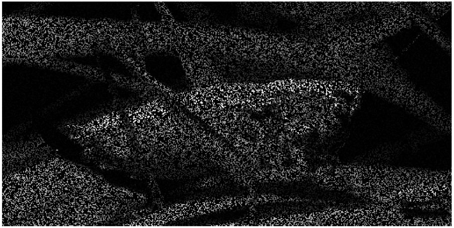
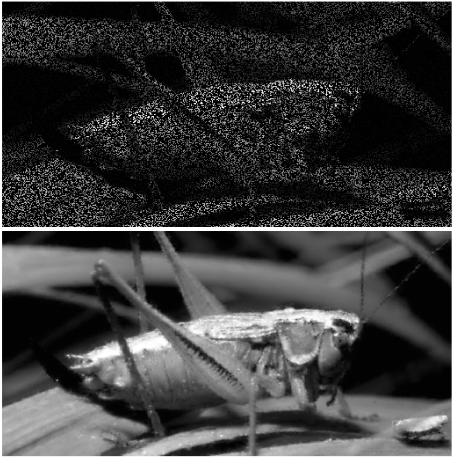
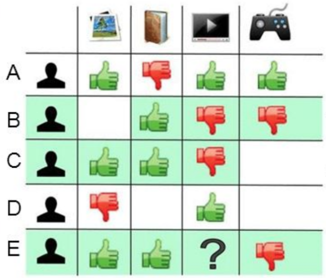


Illustration: continued

70% of the pixels is missing (PSNR $=$ 29 dB):

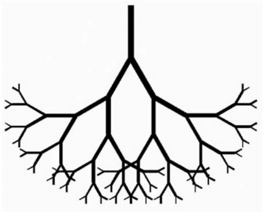


pixel $\leftrightarrow$ (user, movie) rating:



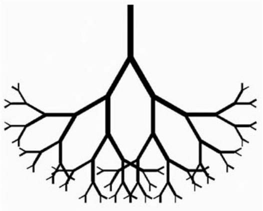
Other structures?

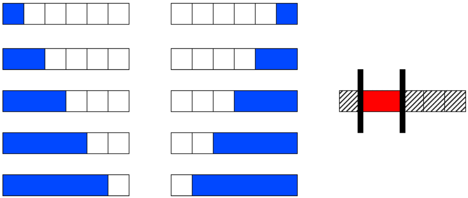
- Left: hierarchical,



Other structures?

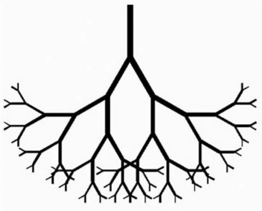
- Left: hierarchical,
- right: continuity on sub-intervals.

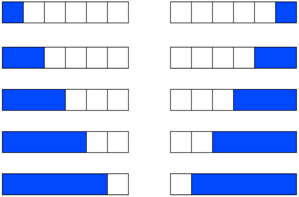


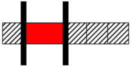


Other structures?

- Left: hierarchical,
- right: continuity on sub-intervals.

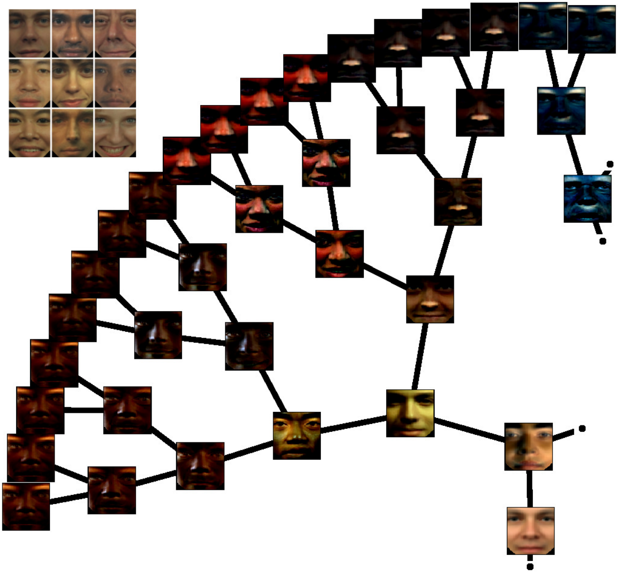






Example: hierarchical dictionary on faces. . .

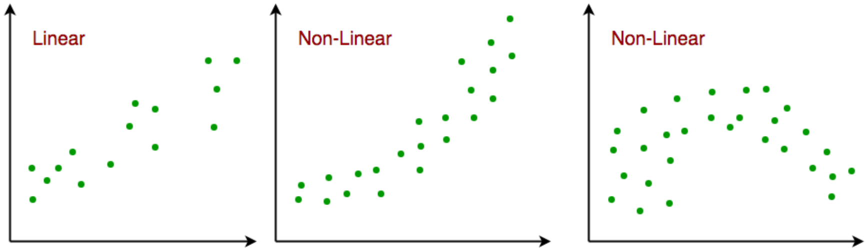
Hierarchically-structured NMF



Regression: non-linear extension

Recall (house pricing): $y \approx b_0 + b_1 x$.

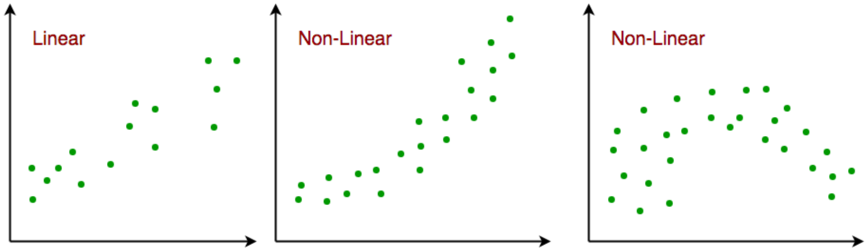
- The (x, y) relation may be highly non-linear: $\boxed{1, x, x^2} \mapsto y$.



Regression: non-linear extension

Recall (house pricing): $y \approx b_0 + b_1 x$.

- The (x, y) relation may be highly non-linear: $\boxed{1, x, x^2} \mapsto y$.

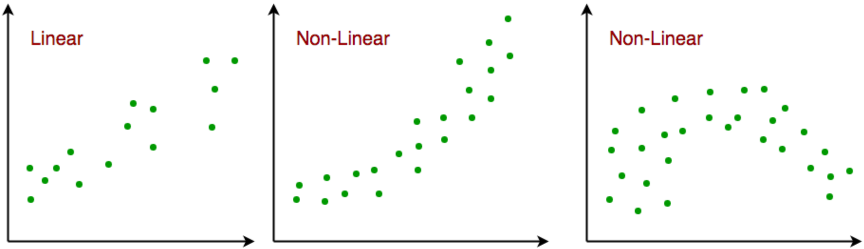


- Interactions might count: $\boxed{x_1 x_2, x_3 x_5} \mapsto y$.

Regression: non-linear extension

Recall (house pricing): $y \approx b_0 + b_1 x$.

- The (x, y) relation may be highly non-linear: $\boxed{1, x, x^2} \mapsto y$.



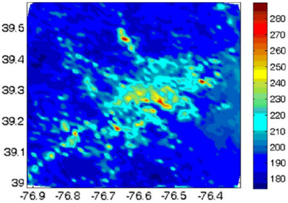
- Interactions might count: $\boxed{x_1 x_2, x_3 x_5} \mapsto y$.

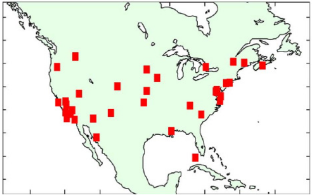
In this case

$f(\mathbf{x}) = \langle \mathbf{b}, \varphi(\mathbf{x}) \rangle$.

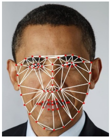
- **Goal**: aerosol prediction = air pollution (climate).

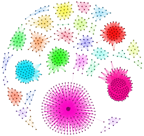
- Prediction using labelled bags:
  - bag := multi-spectral satellite measurements over an area,
  - label := local aerosol value.





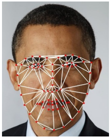
Other examples when x_i-s are distributions (bags)

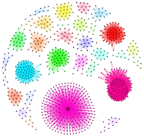




- time-series modelling: user = set of time-series,
- computer vision: image = collection of patch vectors,
- NLP: corpus = bag of documents,
- network analysis: group of people = bag of friendship graphs, ...

Other examples when x_i-s are distributions (bags)





- time-series modelling: user = set of time-series,
- computer vision: image = collection of patch vectors,
- NLP: corpus = bag of documents,
- network analysis: group of people = bag of friendship graphs, ...

Needed: $\boxed{k(x, x') = \langle \varphi(x), \varphi(x') \rangle}$.

Until now

- Goal: $x_i \approx f(x_i)$ given $\{(x_i, y_i)\}_{i=1}^n$ samples.
- Needed:
  - Meaning of '$\approx$': objective function.

Until now

- Goal: $x_i \approx f(x_i)$ given $\{(x_i, y_i)\}_{i=1}^n$ samples.
- Needed:
  - Meaning of '$\approx$': objective function.
  - Hypothesis class: $\mathcal{F} \ni f$.

Until now

- Goal: $x_i \approx f(x_i)$ given $\{(x_i, y_i)\}_{i=1}^n$ samples.
- Needed:
  - Meaning of '$\approx$': objective function.
  - Hypothesis class: $\mathcal{F} \ni f$.
  - Control of model complexity / (structured) sparsity.

Until now

- Goal: $x_i \approx f(x_i)$ given $\{(x_i, y_i)\}_{i=1}^n$ samples.
- Needed:
  - Meaning of '$\approx$': objective function.
  - Hypothesis class: $\mathcal{F} \ni f$.
  - Control of model complexity / (structured) sparsity.
  - Non-linear features.

Until now

- Goal: $x_i \approx f(x_i)$ given $\{(x_i, y_i)\}_{i=1}^n$ samples.
- Needed:
  - Meaning of '$\approx$': objective function.
  - Hypothesis class: $\mathcal{F} \ni f$.
  - Control of model complexity / (structured) sparsity.
  - Non-linear features.
  - Optimization algorithms.

Linear regression

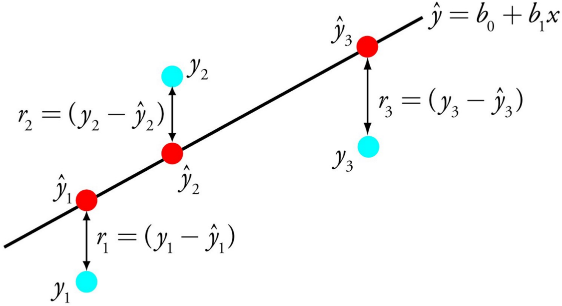
Least squares (LS): univarite case

- Samples: $\{(x_i, y_i)\}_{i=1}^n$, $x_i, y_i \in \mathbb{R}$.
- Prediction: $\hat{y}_i = f(x_i) = b_0 + b_1 x_i$.
- b_0: intercept, b_1: slope.

Least squares (LS): univarite case

- Samples: $\{(x_i, y_i)\}_{i=1}^n$, $x_i, y_i \in \mathbb{R}$.
- Prediction: $\hat{y}_i = f(x_i) = b_0 + b_1 x_i$.
- b_0: intercept, b_1: slope.

Fitting a line:



We minimize the $\{\hat{y}_i - y_i\}_{i=1}^n$ residuals in quadratic sense.

LS objective

$$J(f) = \frac{1}{n}\sum_{i=1}^{n}[f(x_i) - y_i]^2 \to \min_{f \in \boxed{\mathcal{F}}}.$$

Hypothesis class: affine functions, i.e.

$$\boxed{\mathcal{F}} = \{x \mapsto b_0 + b_1 x : b_0, b_1 \in \mathbb{R}\}.$$

Linear regression: multivariate case

- Samples: $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$, $\mathbf{x}_i \in \mathbb{R}^p$, $y_i \in \mathbb{R}$.

Linear regression: multivariate case

- Samples: $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$, $\mathbf{x}_i \in \mathbb{R}^p$, $y_i \in \mathbb{R}$.
- Hypothesis class: $f(\mathbf{z}) = \langle \mathbf{b}, \mathbf{z} \rangle = \sum_{j=1}^p b_j z_j$.

Linear regression: multivariate case

- Samples: $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$, $\mathbf{x}_i \in \mathbb{R}^p$, $y_i \in \mathbb{R}$.
- Hypothesis class: $f(\mathbf{z}) = \langle \mathbf{b}, \mathbf{z} \rangle = \sum_{j=1}^p b_j z_j$.
- Prediction: $\hat{y}_i = f(\mathbf{x}_i) = \langle \mathbf{b}, \mathbf{x}_i \rangle = \mathbf{x}_i^T \mathbf{b}$.

Linear regression: multivariate case

- Samples: $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$, $\mathbf{x}_i \in \mathbb{R}^p$, $y_i \in \mathbb{R}$.
- Hypothesis class: $f(\mathbf{z}) = \langle \mathbf{b}, \mathbf{z} \rangle = \sum_{j=1}^p b_j z_j$.
- Prediction: $\hat{y}_i = f(\mathbf{x}_i) = \langle \mathbf{b}, \mathbf{x}_i \rangle = \mathbf{x}_i^T \mathbf{b}$.
- Objective:

$$\min_{\mathbf{b}} J(\mathbf{b}) := \frac{1}{n} \sum_{i=1}^n [y_i - f(\mathbf{x}_i)]^2$$

Linear regression: multivariate case

- Samples: $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$, $\mathbf{x}_i \in \mathbb{R}^p$, $y_i \in \mathbb{R}$.
- Hypothesis class: $f(\mathbf{z}) = \langle \mathbf{b}, \mathbf{z} \rangle = \sum_{j=1}^p b_j z_j$.
- Prediction: $\hat{y}_i = f(\mathbf{x}_i) = \langle \mathbf{b}, \mathbf{x}_i \rangle = \mathbf{x}_i^T \mathbf{b}$.
- Objective:

$$\min_{\mathbf{b}} J(\mathbf{b}) := \frac{1}{n}\sum_{i=1}^n [y_i - f(\mathbf{x}_i)]^2 = \frac{1}{n}\left\| \underbrace{\begin{bmatrix} \mathbf{x}_1^T \\ \vdots \\ \mathbf{x}_n^T \end{bmatrix}}_{\mathbf{X}} \mathbf{b} - \underbrace{\begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}}_{\mathbf{y}} \right\|_2^2$$

Linear regression: multivariate case

- Samples: $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$, $\mathbf{x}_i \in \mathbb{R}^p$, $y_i \in \mathbb{R}$.
- Hypothesis class: $f(\mathbf{z}) = \langle \mathbf{b}, \mathbf{z} \rangle = \sum_{j=1}^p b_j z_j$.
- Prediction: $\hat{y}_i = f(\mathbf{x}_i) = \langle \mathbf{b}, \mathbf{x}_i \rangle = \mathbf{x}_i^T \mathbf{b}$.
- Objective:

$$\min_{\mathbf{b}} J(\mathbf{b}) := \frac{1}{n}\sum_{i=1}^n [y_i - f(\mathbf{x}_i)]^2 = \frac{1}{n}\left\| \underbrace{\begin{bmatrix} \mathbf{x}_1^T \\ \vdots \\ \mathbf{x}_n^T \end{bmatrix}}_{\mathbf{X}} \mathbf{b} - \underbrace{\begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}}_{\mathbf{y}} \right\|_2^2$$

$$= \boxed{\frac{1}{n}(\mathbf{X}\mathbf{b} - \mathbf{y})^T(\mathbf{X}\mathbf{b} - \mathbf{y})} \in \mathbb{R}.$$

Linear regression: multivariate case

- Samples: $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$, $\mathbf{x}_i \in \mathbb{R}^p$, $y_i \in \mathbb{R}$.
- Hypothesis class: $f(\mathbf{z}) = \langle \mathbf{b}, \mathbf{z} \rangle = \sum_{j=1}^p b_j z_j$.
- Prediction: $\hat{y}_i = f(\mathbf{x}_i) = \langle \mathbf{b}, \mathbf{x}_i \rangle = \mathbf{x}_i^T \mathbf{b}$.
- Objective:

$$\min_{\mathbf{b}} J(\mathbf{b}) := \frac{1}{n} \sum_{i=1}^n [y_i - f(\mathbf{x}_i)]^2 = \frac{1}{n} \left\| \underbrace{\begin{bmatrix} \mathbf{x}_1^T \\ \vdots \\ \mathbf{x}_n^T \end{bmatrix}}_{\mathbf{X}} \mathbf{b} - \underbrace{\begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}}_{\mathbf{y}} \right\|_2^2$$

$$= \boxed{\frac{1}{n} (\mathbf{X}\mathbf{b} - \mathbf{y})^T (\mathbf{X}\mathbf{b} - \mathbf{y})} \in \mathbb{R}.$$

- $\mathbf{X}$ is called the design matrix.

Least squares objective

Objective to minimize:

$$J(\mathbf{b}) = \frac{1}{n}(\mathbf{X}\mathbf{b} - \mathbf{y})^T(\mathbf{X}\mathbf{b} - \mathbf{y}) \to \min_{\mathbf{b}\in\mathbb{R}^p}.$$

Least squares objective

Objective to minimize:

$$J(\mathbf{b}) = \frac{1}{n}(\mathbf{X}\mathbf{b} - \mathbf{y})^T(\mathbf{X}\mathbf{b} - \mathbf{y}) \to \min_{\mathbf{b}\in\mathbb{R}^p}.$$

J is quadratic. $\Rightarrow$ To get $\hat{\mathbf{b}}$:

$$\boxed{\mathbf{0} = \frac{\partial J(\mathbf{b})}{\partial \mathbf{b}}} = \frac{2}{n}\mathbf{X}^T(\mathbf{X}\mathbf{b} - \mathbf{y}).$$

Least squares objective

Objective to minimize:

$$J(\mathbf{b}) = \frac{1}{n}(\mathbf{X}\mathbf{b} - \mathbf{y})^T(\mathbf{X}\mathbf{b} - \mathbf{y}) \to \min_{\mathbf{b}\in\mathbb{R}^p}.$$

J is quadratic. $\Rightarrow$ To get $\hat{\mathbf{b}}$:

$$\boxed{\mathbf{0} = \frac{\partial J(\mathbf{b})}{\partial \mathbf{b}}} = \frac{2}{n}\mathbf{X}^T(\mathbf{X}\mathbf{b} - \mathbf{y}).$$

This means

$$\mathbf{X}^T\mathbf{X}\mathbf{b} = \mathbf{X}^T\mathbf{y} \quad \Rightarrow \quad \mathbf{b} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}.$$

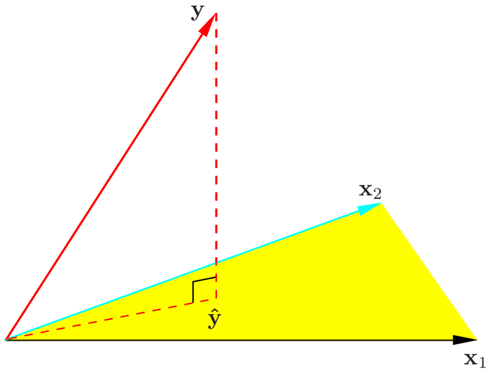
What did we get? – interpretation

Optimal coefficient:

$$\mathbf{b} = (\underbrace{\mathbf{X}^T}_{p\times n}\underbrace{\mathbf{X}}_{n\times p})^{-1}\underbrace{\mathbf{X}^T}_{p\times n}\underbrace{\mathbf{y}}_{n\times 1}.$$

Prediction:

$$\hat{\mathbf{y}} = \mathbf{X}\mathbf{b} = \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}.$$



Linear regression: remarks

$$\hat{\mathbf{y}} = \underbrace{\mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T}_{=:\mathbf{H}}\mathbf{y}$$

H:

- is called hat matrix.
- projector to $span(\mathbf{x}_1, \ldots, \mathbf{x}_n)$.
- $\hat{\mathbf{y}} = \mathbf{H}\mathbf{y}$: linear smoothing.

Linear regression: remarks-2

Recall

$$\mathbf{X}^T\mathbf{X}\mathbf{b} = \mathbf{X}^T\mathbf{y}.$$

What if $\mathbf{X}^T\mathbf{X}$ is not invertible?

Linear regression: remarks-2

Recall

$$\mathbf{X}^T\mathbf{X}\mathbf{b} = \mathbf{X}^T\mathbf{y}.$$

What if $\mathbf{X}^T\mathbf{X}$ is not invertible?

Several options:

- Take pseudo-inverse: $(\mathbf{X}^T\mathbf{X})^-$, i.e. $\mathbf{b} = \left(\mathbf{X}^T\mathbf{X}\right)^- \mathbf{X}^T\mathbf{y}$.

Linear regression: remarks-2

Recall

$$\mathbf{X}^T\mathbf{X}\mathbf{b} = \mathbf{X}^T\mathbf{y}.$$

What if $\mathbf{X}^T\mathbf{X}$ is not invertible?

Several options:

- Take pseudo-inverse: $(\mathbf{X}^T\mathbf{X})^-$, i.e. $\mathbf{b} = \left(\mathbf{X}^T\mathbf{X}\right)^- \mathbf{X}^T\mathbf{y}$.
- Replace $\mathbf{X}^T\mathbf{X}$ with $\mathbf{X}^T\mathbf{X} + c\mathbf{I}$ $(c > 0)$: regularization.

Option-1: pseudo-inverse

- $\mathbf{M} := \mathbf{X}^T\mathbf{X}$
- For any $\mathbf{M} \in \mathbb{R}^{m\times n}$ there is a unique $\mathbf{M}^- \in \mathbb{R}^{n\times m}$:

$$\mathbf{M}\mathbf{M}^-\mathbf{M} = \mathbf{M},$$

Option-1: pseudo-inverse

- $\mathbf{M} := \mathbf{X}^T\mathbf{X}$
- For any $\mathbf{M} \in \mathbb{R}^{m \times n}$ there is a unique $\mathbf{M}^- \in \mathbb{R}^{n \times m}$:

$$\mathbf{M}\mathbf{M}^-\mathbf{M} = \mathbf{M},$$
$$\mathbf{M}^-\mathbf{M}\mathbf{M}^- = \mathbf{M}^-,$$

Option-1: pseudo-inverse

- $\mathbf{M} := \mathbf{X}^T\mathbf{X}$
- For any $\mathbf{M} \in \mathbb{R}^{m\times n}$ there is a unique $\mathbf{M}^- \in \mathbb{R}^{n\times m}$:

$$\begin{aligned}
\mathbf{M}\mathbf{M}^-\mathbf{M} &= \mathbf{M},\\
\mathbf{M}^-\mathbf{M}\mathbf{M}^- &= \mathbf{M}^-,\\
\left(\mathbf{M}\mathbf{M}^-\right)^T &= \mathbf{M}\mathbf{M}^-,
\end{aligned}$$

Option-1: pseudo-inverse

- $\mathbf{M} := \mathbf{X}^T\mathbf{X}$
- For any $\mathbf{M} \in \mathbb{R}^{m\times n}$ there is a unique $\mathbf{M}^- \in \mathbb{R}^{n\times m}$:

$$
\begin{aligned}
\mathbf{M}\mathbf{M}^-\mathbf{M} &= \mathbf{M},\\
\mathbf{M}^-\mathbf{M}\mathbf{M}^- &= \mathbf{M}^-,\\
\left(\mathbf{M}\mathbf{M}^-\right)^T &= \mathbf{M}\mathbf{M}^-,\\
\left(\mathbf{M}^-\mathbf{M}\right)^T &= \left(\mathbf{M}^-\mathbf{M}\right).
\end{aligned}
$$

Option-1: pseudo-inverse

- $\mathbf{M} := \mathbf{X}^T\mathbf{X}$
- For any $\mathbf{M} \in \mathbb{R}^{m\times n}$ there is a unique $\mathbf{M}^- \in \mathbb{R}^{n\times m}$:

$$\begin{aligned}
\mathbf{M}\mathbf{M}^-\mathbf{M} &= \mathbf{M},\\
\mathbf{M}^-\mathbf{M}\mathbf{M}^- &= \mathbf{M}^-,\\
\left(\mathbf{M}\mathbf{M}^-\right)^T &= \mathbf{M}\mathbf{M}^-,\\
\left(\mathbf{M}^-\mathbf{M}\right)^T &= \left(\mathbf{M}^-\mathbf{M}\right).
\end{aligned}$$

Remains

Why is $\mathbf{M}^-$ useful? How to compute it?

Pseudo-inverse: usefulness

Problem: Solution of

$$\mathbf{M}\mathbf{b} = \mathbf{y}$$

might not exist or may not be unique.

Pseudo-inverse: usefulness

Problem: Solution of

$$\mathbf{M}\mathbf{b} = \mathbf{y}$$

might not exist or may not be unique.

The pseudo-inverse based solution $\hat{\mathbf{b}} = \mathbf{M}^{-}\mathbf{y}$ handles this issue and it is 'optimal'.

- smallest error: for any $\mathbf{b} \in \mathbb{R}^n$

$$\|\mathbf{M}\mathbf{b} - \mathbf{y}\|_2 \geqslant \|\mathbf{M}\hat{\mathbf{b}} - \mathbf{y}\|_2.$$

Pseudo-inverse: usefulness

Problem: Solution of

$$\mathbf{M}\mathbf{b} = \mathbf{y}$$

might not exist or may not be unique.

The pseudo-inverse based solution $\hat{\mathbf{b}} = \mathbf{M}^{-}\mathbf{y}$ handles this issue and it is 'optimal'.

- smallest error: for any $\mathbf{b} \in \mathbb{R}^n$

$$\|\mathbf{M}\mathbf{b} - \mathbf{y}\|_2 \geqslant \|\mathbf{M}\hat{\mathbf{b}} - \mathbf{y}\|_2.$$

- smallest norm: among the $\mathbf{b}$ vectors for which '$=$' holds $\hat{\mathbf{b}}$ has minimal Euclidean norm.

Pseudo-inverse: properties

- Generalization of inverse: if $\mathbf{M}^{-1}$ exists, then $\mathbf{M}^{-} = \mathbf{M}^{-1}$.

Pseudo-inverse: properties

- Generalization of inverse: if $\mathbf{M}^{-1}$ exists, then $\mathbf{M}^{-} = \mathbf{M}^{-1}$.
- For $\mathbb{R}^{m\times n} \ni \mathbf{0}$: $\mathbf{0}^{-} = \mathbf{0} \in \mathbb{R}^{n\times m}$.

Pseudo-inverse: properties

- Generalization of inverse: if $\mathbf{M}^{-1}$ exists, then $\mathbf{M}^{-} = \mathbf{M}^{-1}$.
- For $\mathbb{R}^{m \times n} \ni \mathbf{0}$: $\mathbf{0}^{-} = \mathbf{0} \in \mathbb{R}^{n \times m}$.
- For $\mathbf{\Sigma} = diag(\sigma_i) \in \mathbb{R}^{m \times n}$ diagonal:

$$\mathbf{\Sigma}^{-} = diag(\sigma_i^{-}) \in \mathbb{R}^{n \times m},$$

$$\sigma^{-} = \begin{cases} \frac{1}{\sigma} & \text{if } \sigma \neq 0, \\ 0 & \text{if } \sigma = 0 \end{cases}$$

Pseudo-inverse: properties

- Generalization of inverse: if $\mathbf{M}^{-1}$ exists, then $\mathbf{M}^{-} = \mathbf{M}^{-1}$.
- For $\mathbb{R}^{m\times n} \ni \mathbf{0}$: $\mathbf{0}^{-} = \mathbf{0} \in \mathbb{R}^{n\times m}$.
- For $\mathbf{\Sigma} = diag(\sigma_i) \in \mathbb{R}^{m\times n}$ diagonal:

$$\mathbf{\Sigma}^{-} = diag(\sigma_i^{-}) \in \mathbb{R}^{n\times m},$$
$$\sigma^{-} = \begin{cases} \frac{1}{\sigma} & \text{if } \sigma \neq 0, \\ 0 & \text{if } \sigma = 0 \end{cases}$$

- $(\mathbf{M}^{-})^{-} = \mathbf{M}$.

Pseudo-inverse: properties

- Generalization of inverse: if $\mathbf{M}^{-1}$ exists, then $\mathbf{M}^{-} = \mathbf{M}^{-1}$.
- For $\mathbb{R}^{m\times n} \ni \mathbf{0}$: $\mathbf{0}^{-} = \mathbf{0} \in \mathbb{R}^{n\times m}$.
- For $\mathbf{\Sigma} = diag(\sigma_i) \in \mathbb{R}^{m\times n}$ diagonal:

$$\mathbf{\Sigma}^{-} = diag(\sigma_i^{-}) \in \mathbb{R}^{n\times m},$$
$$\sigma^{-} = \begin{cases} \frac{1}{\sigma} & \text{if } \sigma \neq 0, \\ 0 & \text{if } \sigma = 0 \end{cases}$$

- $(\mathbf{M}^{-})^{-} = \mathbf{M}$.
- It commutes with transposition: $(\mathbf{M}^{T})^{-} = (\mathbf{M}^{-})^{T}$.

Pseudo-inverse: properties

- Generalization of inverse: if $\mathbf{M}^{-1}$ exists, then $\mathbf{M}^{-} = \mathbf{M}^{-1}$.
- For $\mathbb{R}^{m\times n} \ni \mathbf{0}$: $\mathbf{0}^{-} = \mathbf{0} \in \mathbb{R}^{n\times m}$.
- For $\mathbf{\Sigma} = diag(\sigma_i) \in \mathbb{R}^{m\times n}$ diagonal:

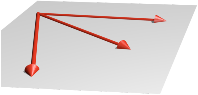
$$\mathbf{\Sigma}^{-} = diag(\sigma_i^{-}) \in \mathbb{R}^{n\times m},$$

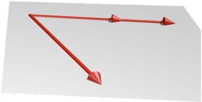
$$\sigma^{-} = \begin{cases} \frac{1}{\sigma} & \text{if } \sigma \neq 0, \\ 0 & \text{if } \sigma = 0 \end{cases}$$

- $(\mathbf{M}^{-})^{-} = \mathbf{M}$.
- It commutes with transposition: $(\mathbf{M}^{T})^{-} = (\mathbf{M}^{-})^{T}$.
- With scalar multiplication: $(c\mathbf{M})^{-} = \frac{1}{c}\mathbf{M}^{-}$ with $c \neq 0$.

Pseudo-inverse: in case of linear independence

Linearly dependent vectors (2 examples):

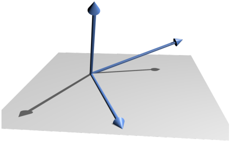




Pseudo-inverse: in case of linear independence

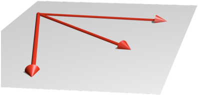
Linearly dependent vectors (2 examples):

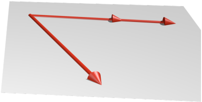
- If $\mathbf{M} \in \mathbb{R}^{m \times n}$ has linearly independent columns (hence $n \leqslant m$):
$$\mathbf{M}^{-} = (\mathbf{M}^T\mathbf{M})^{-1}\mathbf{M}^T.$$



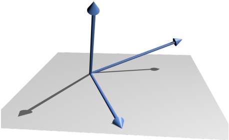
Pseudo-inverse: in case of linear independence

Linearly dependent vectors (2 examples):





- If $\mathbf{M} \in \mathbb{R}^{m \times n}$ has linearly independent columns (hence $n \leqslant m$):
$$\mathbf{M}^{-} = (\mathbf{M}^T\mathbf{M})^{-1}\mathbf{M}^T.$$



- If $\mathbf{M}$ has linearly independent rows:
$$\mathbf{M}^{-} = \mathbf{M}^T(\mathbf{M}\mathbf{M}^T)^{-1}.$$

Pseudo-inverse: computation

One can get $\mathbf{M}^{-}$ from the SVD of $\mathbf{M}$.

For an $\mathbf{M} \in \mathbb{R}^{m \times n}$ there is a

$$\mathbf{M} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T$$

factorization

Pseudo-inverse: computation

One can get $\mathbf{M}^-$ from the SVD of $\mathbf{M}$.

For an $\mathbf{M} \in \mathbb{R}^{m \times n}$ there is a

$$\mathbf{M} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$$

factorization, where

- $\mathbf{U} \in \mathbb{R}^{m \times m}$ is orthogonal: $\mathbf{U}^T\mathbf{U} = \mathbf{I}$.

Pseudo-inverse: computation

One can get $\mathbf{M}^{-}$ from the SVD of $\mathbf{M}$.

For an $\mathbf{M} \in \mathbb{R}^{m \times n}$ there is a

$$\mathbf{M} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T$$

factorization, where

- $\mathbf{U} \in \mathbb{R}^{m \times m}$ is orthogonal: $\mathbf{U}^T\mathbf{U} = \mathbf{I}$.
- $\mathbf{V} \in \mathbb{R}^{n \times n}$ is orthogonal.

Pseudo-inverse: computation

One can get $\mathbf{M}^-$ from the SVD of $\mathbf{M}$.

For an $\mathbf{M} \in \mathbb{R}^{m \times n}$ there is a

$$\mathbf{M} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$$

factorization, where

- $\mathbf{U} \in \mathbb{R}^{m \times m}$ is orthogonal: $\mathbf{U}^T\mathbf{U} = \mathbf{I}$.
- $\mathbf{V} \in \mathbb{R}^{n \times n}$ is orthogonal.
- $\mathbf{\Sigma} = diag(\sigma_i) \in \mathbb{R}^{m \times n}$ is diagonal with non-negative entries, $\searrow$.

Pseudo-inverse: computation

One can get $\mathbf{M}^-$ from the SVD of $\mathbf{M}$.

For an $\mathbf{M} \in \mathbb{R}^{m \times n}$ there is a

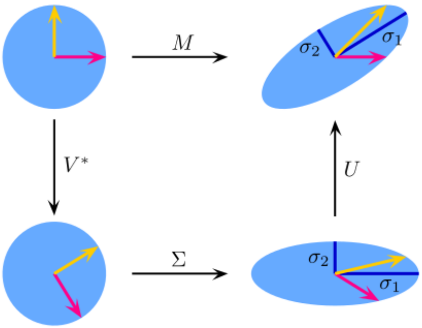
$$\mathbf{M} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T$$

factorization, where

- $\mathbf{U} \in \mathbb{R}^{m \times m}$ is orthogonal: $\mathbf{U}^T\mathbf{U} = \mathbf{I}$.
- $\mathbf{V} \in \mathbb{R}^{n \times n}$ is orthogonal.
- $\boldsymbol{\Sigma} = diag(\sigma_i) \in \mathbb{R}^{m \times n}$ is diagonal with non-negative entries, $\searrow$.
- σ_i: singular values.

Pseudo-inverse: intuition

Let $\mathbf{M} \in \mathbb{R}^{m \times m}$.



$$M = U \cdot \Sigma \cdot V^*$$

rotate $\circ$ scale $\circ$ rotate.

SVD reveals 'everything' on the matrix

Examples:

- $\text{rank}(\mathbf{M}) = \#$ of non-zero σ_i-s.

SVD reveals 'everything' on the matrix

Examples:

- $\text{rank}(\mathbf{M}) = \#$ of non-zero σ_i-s.
- $\|\mathbf{M}\|_F = \sqrt{\sum_i \sigma_i^2(\mathbf{M})}$. (Frobenius / Hilbert-Schmidt norm)

SVD reveals 'everything' on the matrix

Examples:

- $\text{rank}(\mathbf{M}) = \#$ of non-zero σ_i-s.
- $\|\mathbf{M}\|_F = \sqrt{\sum_i \sigma_i^2(\mathbf{M})}$. (Frobenius / Hilbert-Schmidt norm)
- $\|\mathbf{M}\| = \sigma_1(\mathbf{M})$. (operator norm)

SVD reveals 'everything' on the matrix

Examples:

- $\mathrm{rank}(\mathbf{M}) = \#$ of non-zero σ_i-s.
- $\|\mathbf{M}\|_F = \sqrt{\sum_i \sigma_i^2(\mathbf{M})}$. (Frobenius / Hilbert-Schmidt norm)
- $\|\mathbf{M}\| = \sigma_1(\mathbf{M})$. (operator norm)
- Schatter p-norm:

$$\|\mathbf{M}\|_p = \left(\sum_i \sigma_i^p(\mathbf{M})\right)^{\frac{1}{p}}, \quad p \in [1, \infty].$$

Notes:

SVD reveals 'everything' on the matrix

Examples:

- $\text{rank}(\mathbf{M}) = \#$ of non-zero σ_i-s.
- $\|\mathbf{M}\|_F = \sqrt{\sum_i \sigma_i^2(\mathbf{M})}$. (Frobenius / Hilbert-Schmidt norm)
- $\|\mathbf{M}\| = \sigma_1(\mathbf{M})$. (operator norm)
- Schatter p-norm:

$$\|\mathbf{M}\|_p = \left(\sum_i \sigma_i^p(\mathbf{M})\right)^{\frac{1}{p}}, \quad p \in [1, \infty].$$

Notes:

- Spec.: $p = 2$ (Frobenius), $p = \infty$ (operator norm), $p = 1$ (nuclear/trace norm).

SVD reveals 'everything' on the matrix

Examples:

- $\text{rank}(\mathbf{M}) = \#$ of non-zero σ_i-s.
- $\|\mathbf{M}\|_F = \sqrt{\sum_i \sigma_i^2(\mathbf{M})}$. (Frobenius / Hilbert-Schmidt norm)
- $\|\mathbf{M}\| = \sigma_1(\mathbf{M})$. (operator norm)
- Schatter p-norm:

$$\|\mathbf{M}\|_p = \left(\sum_i \sigma_i^p(\mathbf{M})\right)^{\frac{1}{p}}, \quad p \in [1, \infty].$$

Notes:

- Spec.: $p = 2$ (Frobenius), $p = \infty$ (operator norm), $p = 1$ (nuclear/trace norm).
- $\|\mathbf{M}\|_\infty \leqslant \|\mathbf{M}\|_2 \leqslant \|\mathbf{M}\|_1$.

Nuclear norm: intuition

Low-rank view:

$$\mathbf{M} = \sum_i \sigma_i \mathbf{u}_i \mathbf{v}_j^T,$$

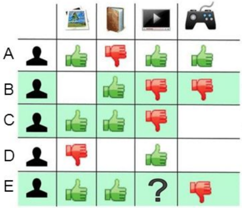
$$\|\mathbf{M}\|_1 = \|\boldsymbol{\sigma}\|_1 = \sum_i \underbrace{|\sigma_i|}_{\sigma_i}.$$

Low-rank view:

$$\mathbf{M} = \sum_i \sigma_i \mathbf{u}_i \mathbf{v}_j^T,$$

$$\|\mathbf{M}\|_1 = \|\boldsymbol{\sigma}\|_1 = \sum_i \underbrace{|\sigma_i|}_{\sigma_i}.$$

$\|\boldsymbol{\sigma}\|_1$ 'captures' the # of nonzero σ_i. Low-rank structures (CF):



Pseudo-inverse from SVD

Option-1:

- For $\mathbf{M} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T$, we have $\mathbf{M}^- = \mathbf{V}\boldsymbol{\Sigma}^-\mathbf{U}^T$.

Option-2:

$\mathbf{X}^T\mathbf{X} + c\mathbf{I}$

This corresponds to a certain form of regularization.

Ridge regression

From least squares to ridge regression

- Least squares:

$$J(\mathbf{b}) = \underbrace{\frac{1}{n}\sum_{i=1}^{n}\left[\langle \mathbf{b}, \mathbf{x}_i\rangle - y_i\right]^2}_{\text{training error}}.$$

From least squares to ridge regression

- Least squares:

$$J(\mathbf{b}) = \underbrace{\frac{1}{n}\sum_{i=1}^{n}\left[\langle \mathbf{b}, \mathbf{x}_i \rangle - y_i\right]^2}_{\text{training error}}.$$

- Ridge regression:

$$J(\mathbf{b}) = \underbrace{\frac{1}{n}\sum_{i=1}^{n}\left[\langle \mathbf{b}, \mathbf{x}_i \rangle - y_i\right]^2}_{\text{training error}} + \underbrace{\lambda \boxed{\|\mathbf{b}\|_2^2}}_{\text{regularization}}.$$

  - $\lambda > 0$: trade-off parameter,
  - $\boxed{\|\mathbf{b}\|_2^2}$: 'complexity control', uniqueness.

Solution of ridge regression

Similarly to least squares:

$$\mathbf{0} = \frac{\partial J(\mathbf{b})}{\partial \mathbf{b}} \xrightarrow{\text{'same' calculation}}$$

Solution of ridge regression

Similarly to least squares:

$$\mathbf{0} = \frac{\partial J(\mathbf{b})}{\partial \mathbf{b}} \xrightarrow{\text{'same' calculation}}$$
$$(\mathbf{X}^T\mathbf{X} + \lambda n\mathbf{I})\mathbf{b} = \mathbf{X}^T\mathbf{y},$$

Solution of ridge regression

Similarly to least squares:

$$\begin{aligned}
\mathbf{0} &= \frac{\partial J(\mathbf{b})}{\partial \mathbf{b}} \xrightarrow{\text{'same' calculation}} \\
(\mathbf{X}^T\mathbf{X} + \lambda n \mathbf{I})\mathbf{b} &= \mathbf{X}^T\mathbf{y}, \\
\mathbf{b} &= \boxed{(\mathbf{X}^T\mathbf{X} + \lambda n \mathbf{I})^{-1}\mathbf{X}^T\mathbf{y}}.
\end{aligned}$$

Solution of ridge regression

Similarly to least squares:

$$\begin{aligned}
\mathbf{0} &= \frac{\partial J(\mathbf{b})}{\partial \mathbf{b}} \xrightarrow{\text{'same' calculation}} \\
(\mathbf{X}^T\mathbf{X} + \lambda n\mathbf{I})\mathbf{b} &= \mathbf{X}^T\mathbf{y}, \\
\mathbf{b} &= \boxed{(\mathbf{X}^T\mathbf{X} + \lambda n\mathbf{I})^{-1}\mathbf{X}^T\mathbf{y}}.
\end{aligned}$$

$\lambda n\mathbf{I}$: regularization.

Estimated coefficient:

$$\mathbf{b} = (\mathbf{X}^T\mathbf{X} + \lambda n\mathbf{I})^{-1}\mathbf{X}^T\mathbf{y}.$$

Prediction:

$$\hat{\mathbf{y}} = \mathbf{X}\mathbf{b}$$

Ridge regression: linear smoother

Estimated coefficient:

$$\mathbf{b} = (\mathbf{X}^T\mathbf{X} + \lambda n\mathbf{I})^{-1}\mathbf{X}^T\mathbf{y}.$$

Prediction:

$$\hat{\mathbf{y}} = \mathbf{X}\mathbf{b} = \underbrace{\mathbf{X}(\mathbf{X}^T\mathbf{X} + \lambda n\mathbf{I})^{-1}\mathbf{X}^T}_{\mathbf{H}}\mathbf{y}.$$

Least squares, ridge regression, next?

- Least squares:

$$J(\mathbf{b}) = \frac{1}{n}\sum_{i=1}^{n}[\langle \mathbf{b}, \mathbf{x}_i \rangle - y_i]^2.$$

- Ridge regression:

$$J(\mathbf{b}) = \frac{1}{n}\sum_{i=1}^{n}[\langle \mathbf{b}, \mathbf{x}_i \rangle - y_i]^2 + \lambda \left\|\mathbf{b}\right\|_2^2.$$

Least squares, ridge regression, next?

- Least squares:

$$J(\mathbf{b}) = \frac{1}{n}\sum_{i=1}^{n}[\langle \mathbf{b}, \mathbf{x}_i\rangle - y_i]^2.$$

- Ridge regression:

$$J(\mathbf{b}) = \frac{1}{n}\sum_{i=1}^{n}[\langle \mathbf{b}, \mathbf{x}_i\rangle - y_i]^2 + \lambda \|\mathbf{b}\|_2^2 .$$

- Non-linear extension: We want to write

$$J(b) = \frac{1}{n}\sum_{i=1}^{n}[\langle b, \varphi(x_i)\rangle - y_i]^2 + \lambda \|b\|^2 .$$

$\varphi(x)$ why?

Recall we are interested in:

- Polynomial / higher order features:

$$\varphi(x) = [1; x; x^2; \dots; x^n],$$
$$\varphi(x) = [x_i x_j]_{(i,j)}.$$

$\varphi(x)$ why?

Recall we are interested in:

- Polynomial / higher order features:

$$\varphi(x) = [1; x; x^2; \ldots; x^n],$$
$$\varphi(x) = [x_i x_j]_{(i,j)}.$$

- Aerosol prediction: $x_i = i^{th}$ distribution / bag.

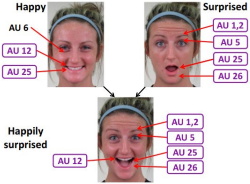
$\varphi(x)$ why?

Recall we are interested in:

- Polynomial / higher order features:

$$\varphi(x) = [1; x; x^2; \ldots; x^n],$$
$$\varphi(x) = [x_i x_j]_{(i,j)}.$$

- Aerosol prediction: $x_i = i^{th}$ distribution / bag.
- Emotion recognition: $x_i =$ time series of muscle activities.



Quadratic & polynomial features

For simplicity in $\mathbb{R}^2$:

$$\varphi(\mathbf{x}) = \left(x_1^2, \sqrt{2}x_1x_2, x_2^2\right),$$

$$\boxed{\langle \varphi(\mathbf{x}), \varphi(\mathbf{x}') \rangle} = ?$$

Quadratic & polynomial features

For simplicity in $\mathbb{R}^2$:

$$\varphi(\mathbf{x}) = \left(x_1^2, \sqrt{2}x_1x_2, x_2^2\right),$$

$$\boxed{\langle \varphi(\mathbf{x}), \varphi(\mathbf{x}')\rangle} = \left\langle \begin{bmatrix} x_1^2 \\ \sqrt{2}x_1x_2 \\ x_2^2 \end{bmatrix}, \begin{bmatrix} (x_1')^2 \\ \sqrt{2}(x_1')(x_2') \\ (x_2')^2 \end{bmatrix} \right\rangle$$

Quadratic & polynomial features

For simplicity in $\mathbb{R}^2$:

$$\begin{aligned}
\varphi(\mathbf{x}) &= \left(x_1^2, \sqrt{2}x_1x_2, x_2^2\right), \\
\boxed{\langle\varphi(\mathbf{x}), \varphi(\mathbf{x}')\rangle} &= \left\langle \begin{bmatrix} x_1^2 \\ \sqrt{2}x_1x_2 \\ x_2^2 \end{bmatrix}, \begin{bmatrix} (x_1')^2 \\ \sqrt{2}(x_1')(x_2') \\ (x_2')^2 \end{bmatrix} \right\rangle \\
&= x_1^2(x_1')^2 + \underbrace{\sqrt{2}\sqrt{2}}_{2}x_1x_2(x_1')(x_2') + x_2^2(x_2')^2
\end{aligned}$$

Quadratic & polynomial features

For simplicity in $\mathbb{R}^2$:

$$\varphi(\mathbf{x}) = \left(x_1^2, \sqrt{2}x_1x_2, x_2^2\right),$$

$$\boxed{\langle\varphi(\mathbf{x}), \varphi(\mathbf{x}')\rangle} = \left\langle \begin{bmatrix} x_1^2 \\ \sqrt{2}x_1x_2 \\ x_2^2 \end{bmatrix}, \begin{bmatrix} (x_1')^2 \\ \sqrt{2}(x_1')(x_2') \\ (x_2')^2 \end{bmatrix} \right\rangle$$

$$= x_1^2(x_1')^2 + \underbrace{\sqrt{2}\sqrt{2}}_{2} x_1x_2(x_1')(x_2') + x_2^2(x_2')^2$$

$$= \left(x_1x_1' + x_2x_2'\right)^2$$

Quadratic & polynomial features

For simplicity in $\mathbb{R}^2$:

$$\begin{aligned}
\varphi(\mathbf{x}) &= \left(x_1^2, \sqrt{2}x_1x_2, x_2^2\right), \\
\boxed{\langle \varphi(\mathbf{x}), \varphi(\mathbf{x}') \rangle} &= \left\langle \begin{bmatrix} x_1^2 \\ \sqrt{2}x_1x_2 \\ x_2^2 \end{bmatrix}, \begin{bmatrix} (x_1')^2 \\ \sqrt{2}(x_1')(x_2') \\ (x_2')^2 \end{bmatrix} \right\rangle \\
&= x_1^2(x_1')^2 + \underbrace{\sqrt{2}\sqrt{2}}_{2} x_1x_2(x_1')(x_2') + x_2^2(x_2')^2 \\
&= \left(x_1x_1' + x_2x_2'\right)^2 \\
&= \left\langle \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \begin{bmatrix} x_1' \\ x_2' \end{bmatrix} \right\rangle^2 = \boxed{\langle \mathbf{x}, \mathbf{x}' \rangle^2} =: k(\mathbf{x}, \mathbf{x}').
\end{aligned}$$

Quadratic & polynomial features

For simplicity in $\mathbb{R}^2$:

$$
\begin{aligned}
\varphi(\mathbf{x}) &= \left(x_1^2, \sqrt{2}x_1x_2, x_2^2\right), \\
\boxed{\langle \varphi(\mathbf{x}), \varphi(\mathbf{x}')\rangle} &= \left\langle \begin{bmatrix} x_1^2 \\ \sqrt{2}x_1x_2 \\ x_2^2 \end{bmatrix}, \begin{bmatrix} (x_1')^2 \\ \sqrt{2}(x_1')(x_2') \\ (x_2')^2 \end{bmatrix} \right\rangle \\
&= x_1^2(x_1')^2 + \underbrace{\sqrt{2}\sqrt{2}}_{2} x_1x_2(x_1')(x_2') + x_2^2(x_2')^2 \\
&= \left(x_1x_1' + x_2x_2'\right)^2 \\
&= \left\langle \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \begin{bmatrix} x_1' \\ x_2' \end{bmatrix} \right\rangle^2 = \boxed{\langle \mathbf{x}, \mathbf{x}' \rangle^2} =: k(\mathbf{x}, \mathbf{x}').
\end{aligned}
$$

$\langle \mathbf{x}, \mathbf{x}' \rangle^d = \langle \varphi(\mathbf{x}), \varphi(\mathbf{x}') \rangle$: $\varphi(\mathbf{x}) = d$-order polynomial. $\Rightarrow$

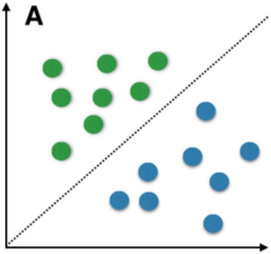
For simplicity in $\mathbb{R}^2$:

$$
\begin{aligned}
\varphi(\mathbf{x}) &= \left(x_1^2, \sqrt{2}x_1x_2, x_2^2\right), \\
\boxed{\langle\varphi(\mathbf{x}), \varphi(\mathbf{x}')\rangle} &= \left\langle \begin{bmatrix} x_1^2 \\ \sqrt{2}x_1x_2 \\ x_2^2 \end{bmatrix}, \begin{bmatrix} (x_1')^2 \\ \sqrt{2}(x_1')(x_2') \\ (x_2')^2 \end{bmatrix} \right\rangle \\
&= x_1^2(x_1')^2 + \underbrace{\sqrt{2}\sqrt{2}}_{2}x_1x_2(x_1')(x_2') + x_2^2(x_2')^2 \\
&= \left(x_1x_1' + x_2x_2'\right)^2 \\
&= \left\langle \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \begin{bmatrix} x_1' \\ x_2' \end{bmatrix} \right\rangle^2 = \boxed{\langle\mathbf{x}, \mathbf{x}'\rangle^2} =: k(\mathbf{x}, \mathbf{x}').
\end{aligned}
$$

$\langle\mathbf{x}, \mathbf{x}'\rangle^d = \langle\varphi(\mathbf{x}), \varphi(\mathbf{x}')\rangle$: $\varphi(\mathbf{x}) = d$-order polynomial. $\Rightarrow$ Explicit computation would be heavy!

Classification motivation: linear separability

Idealized situation

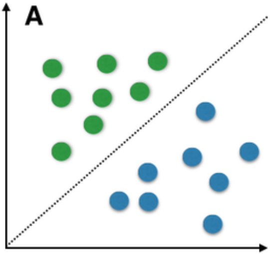


Decision surface:

$$\{\mathbf{x} : \langle \mathbf{w}, \mathbf{x} \rangle = 0\}$$

Classification motivation: linear separability

Idealized situation



Decision surface:

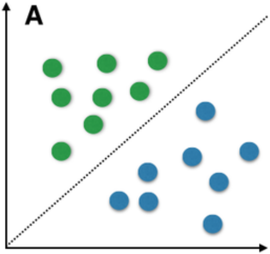
$$\{\mathbf{x} : \langle \mathbf{w}, \mathbf{x} \rangle = 0\} \ \Rightarrow$$

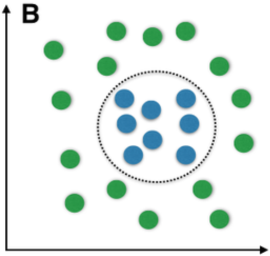
classes:

$$\{\mathbf{x} : \langle \mathbf{w}, \mathbf{x} \rangle \geqslant 0\} \qquad \{\mathbf{x} : \langle \mathbf{w}, \mathbf{x} \rangle < 0\}$$

Classification motivation: non-linear separability

Idealized situation　　　　Real world





Decision surface (left):

$$\{\mathbf{x} : \langle \mathbf{w}, \mathbf{x} \rangle = 0\} \Rightarrow$$

classes:

$$\{\mathbf{x} : \langle \mathbf{w}, \mathbf{x} \rangle \geqslant 0\} \qquad \{\mathbf{x} : \langle \mathbf{w}, \mathbf{x} \rangle < 0\}.$$

Non-linear separability – continued

On the ellipse

$$\left\{ \mathbf{x} : \frac{(x_1 - c_1)^2}{a^2} + \frac{(x_2 - c_2)^2}{b^2} = 1 \right\}$$

Non-linear separability – continued

On the ellipse, outside

$$\left\{\mathbf{x} : \frac{(x_1 - c_1)^2}{a^2} + \frac{(x_2 - c_2)^2}{b^2} = 1\right\},$$

$$\left\{\mathbf{x} : \frac{(x_1 - c_1)^2}{a^2} + \frac{(x_2 - c_2)^2}{b^2} > 1\right\}$$

Non-linear separability – continued

On the ellipse, outside, inside:

$$\left\{\mathbf{x} : \frac{(x_1 - c_1)^2}{a^2} + \frac{(x_2 - c_2)^2}{b^2} = 1\right\},$$

$$\left\{\mathbf{x} : \frac{(x_1 - c_1)^2}{a^2} + \frac{(x_2 - c_2)^2}{b^2} > 1\right\},$$

$$\left\{\mathbf{x} : \frac{(x_1 - c_1)^2}{a^2} + \frac{(x_2 - c_2)^2}{b^2} < 1\right\}.$$

Non-linear separability – continued

On the ellipse, outside, inside:

$$\left\{\mathbf{x} : \frac{(x_1-c_1)^2}{a^2} + \frac{(x_2-c_2)^2}{b^2} = 1\right\},$$
$$\left\{\mathbf{x} : \frac{(x_1-c_1)^2}{a^2} + \frac{(x_2-c_2)^2}{b^2} > 1\right\},$$
$$\left\{\mathbf{x} : \frac{(x_1-c_1)^2}{a^2} + \frac{(x_2-c_2)^2}{b^2} < 1\right\}.$$

With polynomial feature: $\boxed{\varphi(\mathbf{x}) = (x_1^2, x_1, 1, x_2^2, x_2)}$:

- Decision surface: $\{\mathbf{x} : \langle \mathbf{w}, \varphi(\mathbf{x})\rangle = 0\}$.

Non-linear separability – continued

On the ellipse, outside, inside:

$$\left\{\mathbf{x} : \frac{(x_1 - c_1)^2}{a^2} + \frac{(x_2 - c_2)^2}{b^2} = 1\right\},$$
$$\left\{\mathbf{x} : \frac{(x_1 - c_1)^2}{a^2} + \frac{(x_2 - c_2)^2}{b^2} > 1\right\},$$
$$\left\{\mathbf{x} : \frac{(x_1 - c_1)^2}{a^2} + \frac{(x_2 - c_2)^2}{b^2} < 1\right\}.$$

With polynomial feature: $\boxed{\varphi(\mathbf{x}) = (x_1^2, x_1, 1, x_2^2, x_2)}$:

- Decision surface: $\{\mathbf{x} : \langle \mathbf{w}, \varphi(\mathbf{x})\rangle = 0\}$.
- Classes: $\{\mathbf{x} : \langle \mathbf{w}, \varphi(\mathbf{x})\rangle > 0\}$, $\{\mathbf{x} : \langle \mathbf{w}, \varphi(\mathbf{x})\rangle < 0\}$.

Kernel: similarity between features

- Given: x and x' objects (images/texts/time series/distributions).

Kernel: similarity between features

- Given: x and x' objects (images/texts/time series/distributions).
- Question: how similar they are?

Kernel: similarity between features

- Given: x and x' objects (images/texts/time series/distributions).
- Question: how similar they are?
- Define features of the objects:

$$\varphi(x) : \text{features of } x,$$
$$\varphi(x') : \text{features of } x'.$$

- Kernel: inner product of these features

$$k(x, x') := \langle \varphi(x), \varphi(x') \rangle .$$

It is a simple extension of $k(\mathbf{x}, \mathbf{x}') = \mathbf{x}^T \mathbf{x}'$; in that case $\varphi(\mathbf{x}) = \mathbf{x}$.

Kernel examples on $\mathbb{R}^d$ ($\gamma > 0, p \in \mathbb{Z}^+$)

- Polynomial kernel:

$$k(\mathbf{x}, \mathbf{x}') = \left(\left\langle \mathbf{x}, \mathbf{x}' \right\rangle + \gamma\right)^p.$$

- Gaussian kernel:

$$k(\mathbf{x}, \mathbf{x}') = e^{-\gamma \|\mathbf{x} - \mathbf{x}'\|_2^2}.$$

A few other examples:

Kernels exist on:

- Trees, time series, strings,
- mixture models, hidden Markov models or linear dynamical systems,
- sets, fuzzy domains, distributions,
- groups $\xrightarrow{\text{spec.}}$ permutations,
- graphs.

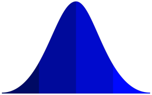
Kernels, RKHS: Definition-2

- Def-1 $=$ feature space point of view, $k(a,b) = \langle \varphi(a), \varphi(b) \rangle_{\mathcal{H}}$.

Kernels, RKHS: Definition-2

- Def-1 = feature space point of view, $k(a,b) = \langle \varphi(a), \varphi(b) \rangle_{\mathcal{H}}$.
- Def-2 = constructive. $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a **reproducing kernel** of a $\mathcal{H}$(ilbert) $\subset \mathbb{R}^{\mathcal{X}}$

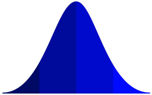
$$\underbrace{k(\cdot,x)} \in \mathcal{H}, \qquad \underbrace{f(x) = \langle f, k(\cdot,x) \rangle_{\mathcal{H}}}_{\text{reproducing property}}.$$



Kernels, RKHS: Definition-2

- Def-1 = feature space point of view, $k(a,b) = \langle \varphi(a), \varphi(b) \rangle_{\mathcal{H}}$.
- Def-2 = constructive. $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a reproducing kernel of a $\mathcal{H}(\text{ilbert}) \subset \mathbb{R}^{\mathcal{X}}$

$$\underbrace{k(\cdot,x)} \in \mathcal{H}, \qquad \underbrace{f(x) = \langle f, k(\cdot,x) \rangle_{\mathcal{H}}}_{\text{reproducing property}}.$$

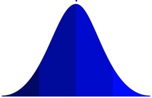


$$\xrightarrow{\text{spec.}} k(x',x) = \langle k(\cdot,x'), k(\cdot,x) \rangle_{\mathcal{H}}.$$

Kernels, RKHS: Definition-2

- Def-1 = feature space point of view, $k(a,b) = \langle \varphi(a), \varphi(b) \rangle_{\mathcal{H}}$.
- Def-2 = constructive. $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a reproducing kernel of a $\mathcal{H}(\text{ilbert}) \subset \mathbb{R}^{\mathcal{X}}$

$$\underbrace{k(\cdot,x)}\in \mathcal{H}, \qquad \underbrace{f(x) = \langle f, k(\cdot,x) \rangle_{\mathcal{H}}}_{\text{reproducing property}}.$$

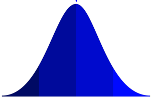


$$\xrightarrow{\text{spec.}} k(x',x) = \langle k(\cdot,x'), k(\cdot,x) \rangle_{\mathcal{H}}. \quad \boxed{\mathcal{H}_k = \overline{\left\{\sum_{i=1}^n \alpha_i k(\cdot,x_i)\right\}}}.$$

Kernels, RKHS: Definition-2

- Def-1 = feature space point of view, $k(a,b) = \langle \varphi(a), \varphi(b) \rangle_{\mathcal{H}}$.
- Def-2 = constructive. $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a reproducing kernel of a $\mathcal{H}(\text{ilbert}) \subset \mathbb{R}^{\mathcal{X}}$

$$\underbrace{k(\cdot, x)} \in \mathcal{H}, \qquad \underbrace{f(x) = \langle f, k(\cdot, x) \rangle_{\mathcal{H}}}_{\text{reproducing property}}.$$



$$\xrightarrow{\text{spec.}} k(x', x) = \langle k(\cdot, x'), k(\cdot, x) \rangle_{\mathcal{H}}. \quad \boxed{\mathcal{H}_k = \overline{\left\{\sum_{i=1}^{n} \alpha_i k(\cdot, x_i)\right\}}}.$$

This is what we will use in non-linear ridge regression!

$f(\mathbf{x}) = \langle \mathbf{b}, \mathbf{x} \rangle \leftrightarrow f(x) = \langle f, \varphi(x) \rangle_{\mathcal{H}}$.

Kernels: Definition-3

- Def-3: Gram matrix, optimization point of view.
- Intuition: $\mathcal{X} := \mathbb{R}^d$, data matrix $\mathbf{X} = [x_1, \ldots, x_n] \in \mathbb{R}^{d \times n}$, then

$$\mathbf{G} := \mathbf{X}^T\mathbf{X} = \left[\langle x_i, x_j \rangle_2\right]_{i,j=1}^n \succeq 0.$$

- Def-3: Gram matrix, optimization point of view.
- Intuition: $\mathcal{X} := \mathbb{R}^d$, data matrix $\mathbf{X} = [x_1, \ldots, x_n] \in \mathbb{R}^{d\times n}$, then

$$\mathbf{G} := \mathbf{X}^T\mathbf{X} = \left[\langle x_i, x_j\rangle_2\right]_{i,j=1}^n \succeq 0.$$

i.e.

$$\mathbf{G}^T = \mathbf{G} \qquad \text{(symmetry)},$$

Kernels: Definition-3

- Def-3: Gram matrix, optimization point of view.
- Intuition: $\mathcal{X} := \mathbb{R}^d$, data matrix $\mathbf{X} = [x_1, \ldots, x_n] \in \mathbb{R}^{d \times n}$, then

$$\mathbf{G} := \mathbf{X}^T \mathbf{X} = \left[\langle x_i, x_j \rangle_2\right]_{i,j=1}^n \succeq 0.$$

i.e.

$$\begin{aligned} \mathbf{G}^T &= \mathbf{G} && \text{(symmetry)}, \\ \mathbf{v}^T \mathbf{G} \mathbf{v} &= \mathbf{v}^T \mathbf{X}^T \mathbf{X} \mathbf{v} = \|\mathbf{X}\mathbf{v}\|_2^2 \geqslant 0 && (\forall \mathbf{v} \in \mathbb{R}^d). \end{aligned}$$

Kernels: Definition-3

- Def-3: Gram matrix, optimization point of view.
- Intuition: $\mathcal{X} := \mathbb{R}^d$, data matrix $\mathbf{X} = [x_1, \ldots, x_n] \in \mathbb{R}^{d \times n}$, then

$$\mathbf{G} := \mathbf{X}^T \mathbf{X} = \left[\langle x_i, x_j \rangle_2\right]_{i,j=1}^n \succeq 0.$$

i.e.

$$\begin{aligned} \mathbf{G}^T &= \mathbf{G} && \text{(symmetry)}, \\ \mathbf{v}^T \mathbf{G} \mathbf{v} &= \mathbf{v}^T \mathbf{X}^T \mathbf{X} \mathbf{v} = \|\mathbf{X}\mathbf{v}\|_2^2 \geqslant 0 && (\forall \mathbf{v} \in \mathbb{R}^d). \end{aligned}$$

- $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ symmetric is positive definite if

$$\mathbf{G} = [k(x_i, x_j)]_{i,j=1}^n \succeq 0 \quad \forall n \in \mathbb{Z}^+, \forall \{x_i\}_{i=1}^n.$$

Kernels: Definition-3

- Def-3: Gram matrix, optimization point of view.
- Intuition: $\mathcal{X} := \mathbb{R}^d$, data matrix $\mathbf{X} = [x_1, \ldots, x_n] \in \mathbb{R}^{d \times n}$, then

$$\mathbf{G} := \mathbf{X}^T \mathbf{X} = \left[\langle x_i, x_j \rangle_2 \right]_{i,j=1}^n \succeq 0.$$

  i.e.

$$\begin{aligned} \mathbf{G}^T &= \mathbf{G} && \text{(symmetry)}, \\ \mathbf{v}^T \mathbf{G} \mathbf{v} &= \mathbf{v}^T \mathbf{X}^T \mathbf{X} \mathbf{v} = \|\mathbf{X}\mathbf{v}\|_2^2 \geqslant 0 && (\forall \mathbf{v} \in \mathbb{R}^d). \end{aligned}$$

- $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ symmetric is positive definite if

$$\mathbf{G} = [k(x_i, x_j)]_{i,j=1}^n \succeq 0 \quad \forall n \in \mathbb{Z}^+, \forall \{x_i\}_{i=1}^n.$$

Importance

In non-linear ridge regression we will invert $\mathbf{G} + \lambda n \mathbf{I} \succ \mathbf{0}$!

Kernels: Definition-4 – motivation

- Def-4 intuition: We want

$$\boxed{(f_n)_{n\in\mathbb{N}} \xrightarrow{\|\cdot\|} f \quad \Rightarrow \quad (f_n(x))_{n\in\mathbb{N}} \to f(x) \quad \forall x.}$$

Kernels: Definition-4 – motivation

- Def-4 intuition: We want

$$\boxed{(f_n)_{n\in\mathbb{N}} \xrightarrow{\|\cdot\|} f \quad \Rightarrow \quad (f_n(x))_{n\in\mathbb{N}} \to f(x) \quad \forall x.}$$

- Example-1: For $\mathcal{H} := (C[0,1], \|\cdot\|_\infty)$

$$|f_n(x) - f(x)| \leqslant \max_{x\in[0,1]} |f_n(x) - f(x)| = \|f_n - f\|_\infty \xrightarrow{n\to\infty} 0,$$

Kernels: Definition-4 – motivation

- Def-4 intuition: We want

$$\boxed{(f_n)_{n\in\mathbb{N}} \xrightarrow{\|\cdot\|} f \quad \Rightarrow \quad (f_n(x))_{n\in\mathbb{N}} \to f(x) \quad \forall x.}$$

- Example-1: For $\mathcal{H} := (C[0,1], \|\cdot\|_\infty)$

$$|f_n(x) - f(x)| \leqslant \max_{x\in[0,1]} |f_n(x) - f(x)| = \|f_n - f\|_\infty \xrightarrow{n\to\infty} 0,$$

but no inner product in $C[0,1]$ (parallelogram rule: violated).

Let us now try a Hilbert space: $\mathcal{H} = L^2[0,1] \ni f_n(x) = x^n$ (simple).

Let us now try a Hilbert space: $\mathcal{H} = L^2[0,1] \ni f_n(x) = x^n$ (simple).

- $f_n \xrightarrow{n\to\infty} 0 := f^* \in \mathcal{H}$ since

$$\lim_{n\to\infty} \|f_n - 0\|_{L^2[0,1]} = \lim_{n\to\infty} \left(\int_0^1 x^{2n} \mathrm{d}x \right)^{1/2}$$

Kernels: Definition-4 – continued

Let us now try a Hilbert space: $\mathcal{H} = L^2[0,1] \ni f_n(x) = x^n$ (simple).

- $f_n \xrightarrow{n\to\infty} 0 := f^* \in \mathcal{H}$ since

$$\lim_{n\to\infty} \|f_n - 0\|_{L^2[0,1]} = \lim_{n\to\infty} \left(\int_0^1 x^{2n} \mathrm{d}x\right)^{1/2} = \lim_{n\to\infty} \sqrt{\left[\frac{x^{2n+1}}{2n+1}\right]_{x=0}^{x=1}}$$

Kernels: Definition-4 – continued

Let us now try a Hilbert space: $\mathcal{H} = L^2[0,1] \ni f_n(x) = x^n$ (simple).

- $f_n \xrightarrow{n\to\infty} 0 := f^* \in \mathcal{H}$ since

$$\lim_{n\to\infty} \|f_n - 0\|_{L^2[0,1]} = \lim_{n\to\infty} \left(\int_0^1 x^{2n} \mathrm{d}x\right)^{1/2} = \lim_{n\to\infty} \sqrt{\left[\frac{x^{2n+1}}{2n+1}\right]_{x=0}^{x=1}}$$
$$= \lim_{n\to\infty} \frac{1}{\sqrt{2n+1}} = 0,$$

Kernels: Definition-4 – continued

Let us now try a Hilbert space: $\mathcal{H} = L^2[0,1] \ni f_n(x) = x^n$ (simple).

- $f_n \xrightarrow{n\to\infty} 0 := f^* \in \mathcal{H}$ since

$$\lim_{n\to\infty} \|f_n - 0\|_{L^2[0,1]} = \lim_{n\to\infty} \left(\int_0^1 x^{2n} \mathrm{d}x\right)^{1/2} = \lim_{n\to\infty} \sqrt{\left[\frac{x^{2n+1}}{2n+1}\right]_{x=0}^{x=1}}$$
$$= \lim_{n\to\infty} \frac{1}{\sqrt{2n+1}} = 0,$$

- but $f_n(1) = 1 \nrightarrow f^*(1) = 0$.

In L^2: norm convergence $\not\Rightarrow$ pointwise convergence.

Kernels: Definition-4

- Evaluation functional: $\delta_x(f) := f(x)$ is linear

$$\delta_x(f+g)$$

Kernels: Definition-4

- Evaluation functional: $\delta_x(f) := f(x)$ is linear

$$\delta_x(f+g) = (f+g)(x)$$

- Evaluation functional: $\delta_x(f) := f(x)$ is linear

$$\delta_x(f+g) = (f+g)(x) = f(x) + g(x)$$

Kernels: Definition-4

- Evaluation functional: $\delta_x(f) := f(x)$ is linear

$$\delta_x(f+g) = (f+g)(x) = f(x) + g(x) = \delta_x(f) + \delta_x(g),$$

Kernels: Definition-4

- Evaluation functional: $\delta_x(f) := f(x)$ is linear

$$\delta_x(f+g) = (f+g)(x) = f(x) + g(x) = \delta_x(f) + \delta_x(g),$$
$$\delta_x(\lambda f)$$

Kernels: Definition-4

- Evaluation functional: $\delta_x(f) := f(x)$ is linear

$$\begin{aligned}\delta_x(f+g) &= (f+g)(x) = f(x) + g(x) = \delta_x(f) + \delta_x(g),\\ \delta_x(\lambda f) &= (\lambda f)(x)\end{aligned}$$

Kernels: Definition-4

- Evaluation functional: $\delta_x(f) := f(x)$ is linear

$$\delta_x(f+g) = (f+g)(x) = f(x) + g(x) = \delta_x(f) + \delta_x(g),$$
$$\delta_x(\lambda f) = (\lambda f)(x) = \lambda f(x)$$

Kernels: Definition-4

- Evaluation functional: $\delta_x(f) := f(x)$ is linear

$$\delta_x(f+g) = (f+g)(x) = f(x) + g(x) = \delta_x(f) + \delta_x(g),$$
$$\delta_x(\lambda f) = (\lambda f)(x) = \lambda f(x) = \lambda \delta_x(f).$$

Kernels: Definition-4

- Evaluation functional: $\delta_x(f) := f(x)$ is linear

$$\delta_x(f+g) = (f+g)(x) = f(x) + g(x) = \delta_x(f) + \delta_x(g),$$
$$\delta_x(\lambda f) = (\lambda f)(x) = \lambda f(x) = \lambda \delta_x(f).$$

- Def-4 (evaluation point of view): $\mathcal{H} \subset \mathbb{R}^{\mathcal{X}}$ Hilbert space,

$$\delta_x : f \in \mathcal{H} \mapsto f(x) \in \mathbb{R}$$

is continuous for all $x \in \mathcal{X}$.

Relation of Definition 1-4

- Def-1 (feature space):

$$k(a,b) = \langle \varphi(a), \varphi(b) \rangle_{\mathcal{H}}.$$

- Def-2 (reproducing kernel, constructive):

$$k(\cdot,x) \in \mathcal{H}, \qquad f(x) = \langle f, k(\cdot,x) \rangle_{\mathcal{H}}.$$

- Def-3 (Gram matrix): $\mathbf{G} = [k(x_i,x_j)] \succeq 0$.
- Def-4 (evaluation): $\delta_x(f) = f(x)$ is continuous for all x.

Relation of Definition 1-4

- Def-1 (feature space):

$$k(a,b) = \langle \varphi(a), \varphi(b) \rangle_{\mathcal{H}}.$$

- Def-2 (reproducing kernel, constructive):

$$k(\cdot, x) \in \mathcal{H}, \qquad f(x) = \langle f, k(\cdot, x) \rangle_{\mathcal{H}}.$$

- Def-3 (Gram matrix): $\mathbf{G} = [k(x_i, x_j)] \succeq 0$.
- Def-4 (evaluation): $\delta_x(f) = f(x)$ is continuous for all x.

- All these definitions are equivalent, $k \overset{1:1}{\leftrightarrow} \mathcal{H}_k$.
- Trickiest direction (Moore-Aronszajn theorem):

$$k \text{ positive-definite function} \xrightarrow{\text{construction}} \text{RKHS}.$$

Kernel puzzle

Let

$$\mathcal{X} = \{0, 1\},$$
$$k(x, x') = \left\{ \begin{array}{ll} 1, & \text{if } x \neq x' \\ -1, & \text{if } x = x' \end{array} \right\}.$$

Kernel puzzle

Let

$$\mathcal{X} = \{0, 1\},$$

$$k(x, x') = \left\{ \begin{array}{ll} 1, & \text{if } x \neq x' \\ -1, & \text{if } x = x' \end{array} \right\}.$$

Puzzle

Is k a kernel?

Kernel puzzle

Let

$$\mathcal{X} = \{0,1\},$$
$$k(x,x') = \begin{Bmatrix} 1, & \text{if } x \neq x' \\ -1, & \text{if } x = x' \end{Bmatrix}.$$

Puzzle

Is k a kernel?

No!

$$k(x,x) = \langle \varphi(x), \varphi(x) \rangle_{\mathcal{H}} \qquad ,$$

Kernel puzzle

Let

$$\mathcal{X} = \{0,1\},$$
$$k(x,x') = \left\{ \begin{array}{ll} 1, & \text{if } x \neq x' \\ -1, & \text{if } x = x' \end{array} \right\}.$$

Puzzle

Is k a kernel?

No!

$$k(x,x) = \langle \varphi(x), \varphi(x) \rangle_{\mathcal{H}} = \|\varphi(x)\|_{\mathcal{H}}^2 \geqslant 0 \quad (\text{Gram with } n = 1),$$

Kernel puzzle

Let

$$\mathcal{X} = \{0,1\},$$
$$k(x,x') = \left\{ \begin{array}{ll} 1, & \text{if } x \neq x' \\ -1, & \text{if } x = x' \end{array} \right\}.$$

Puzzle

Is k a kernel?

No!

$$k(x,x) = \langle \varphi(x), \varphi(x) \rangle_{\mathcal{H}} = \|\varphi(x)\|_{\mathcal{H}}^2 \geqslant 0 \quad \text{(Gram with } n = 1\text{)},$$
$$k(0,0) = k(1,1) = -1 \quad \text{(in our case)}.$$

Kernel ridge regression

Kernel ridge regression

- Given: $\{(x_i, y_i)\}_{i=1}^n$, $\mathcal{H} := \mathcal{H}_k$, $y_i \in \mathbb{R}$.
- Task ($\lambda > 0$):

$$J(f) = \frac{1}{n}\sum_{i=1}^{n}[y_i - f(x_i)]^2 + \lambda \|f\|_{\mathcal{H}}^2 \to \min_{f\in\mathcal{H}}.$$

Kernel ridge regression

- Given: $\{(x_i, y_i)\}_{i=1}^n$, $\mathcal{H} := \mathcal{H}_k$, $y_i \in \mathbb{R}$.
- Task ($\lambda > 0$):

$$J(f) = \frac{1}{n}\sum_{i=1}^n [y_i - f(x_i)]^2 + \lambda \|f\|_{\mathcal{H}}^2 \to \min_{f \in \mathcal{H}}.$$

- Analytical solution:

$$f(x) = \underbrace{[k(x_1, x), \ldots, k(x_n, x)]}_{1 \times n}(\underbrace{\mathbf{G} + \lambda n I}_{n \times n})^{-1} \underbrace{[y_1; \ldots; y_n]}_{n \times 1},$$

$$\mathbf{G} = [k(x_i, x_j)]_{i,j=1}^n.$$

Kernel ridge regression

- Given: $\{(x_i, y_i)\}_{i=1}^n$, $\mathcal{H} := \mathcal{H}_k$, $y_i \in \mathbb{R}$.
- Task ($\lambda > 0$):

$$J(f) = \frac{1}{n}\sum_{i=1}^n [y_i - f(x_i)]^2 + \lambda \|f\|_{\mathcal{H}}^2 \to \min_{f \in \mathcal{H}}.$$

- Analytical solution:

$$f(x) = \underbrace{[k(x_1,x),\ldots,k(x_n,x)]}_{1\times n}(\underbrace{\mathbf{G} + \lambda n I}_{n\times n})^{-1}\underbrace{[y_1;\ldots;y_n]}_{n\times 1},$$

$$\mathbf{G} = [k(x_i,x_j)]_{i,j=1}^n.$$

Question

How do we get this solution?

Kernel ridge regression

By the representer theorem

$$f(x) = \sum_{i=1}^{n} a_i k(\cdot, x_i).$$

Kernel ridge regression

By the representer theorem

$$f(x) = \sum_{i=1}^{n} a_i k(\cdot, x_i).$$

Multiplying the objective by n, using the reproducing property:

$$J(f) = \sum_{j=1}^{n} [y_j - \langle f, k(\cdot, x_j)\rangle_{\mathcal{H}}]^2 + \lambda n \|f\|_{\mathcal{H}}^2$$

Kernel ridge regression

By the representer theorem

$$f(x) = \sum_{i=1}^{n} a_i k(\cdot, x_i).$$

Multiplying the objective by n, using the reproducing property:

$$\begin{aligned} J(f) &= \sum_{j=1}^{n} [y_j - \langle f, k(\cdot, x_j) \rangle_{\mathcal{H}}]^2 + \lambda n \|f\|_{\mathcal{H}}^2 \\ &= \|\mathbf{y} - \mathbf{G}\mathbf{a}\|_2^2 + (\lambda n) \mathbf{a}^T \mathbf{G}\mathbf{a} \end{aligned}$$

Kernel ridge regression

By the representer theorem

$$f(x) = \sum_{i=1}^{n} a_i k(\cdot, x_i).$$

Multiplying the objective by n, using the reproducing property:

$$\begin{aligned} J(f) &= \sum_{j=1}^{n} [y_j - \langle f, k(\cdot, x_j)\rangle_{\mathcal{H}}]^2 + \lambda n \|f\|_{\mathcal{H}}^2 \\ &= \|\mathbf{y} - \mathbf{G}\mathbf{a}\|_2^2 + (\lambda n)\mathbf{a}^T\mathbf{G}\mathbf{a} \\ &= \mathbf{y}^T\mathbf{y} - 2\mathbf{y}^T\mathbf{G}\mathbf{a} + \mathbf{a}^T\left[\mathbf{G}^2 + (\lambda n)\mathbf{G}\right]\mathbf{a}. \end{aligned}$$

Kernel ridge regression

By the representer theorem

$$f(x) = \sum_{i=1}^{n} a_i k(\cdot, x_i).$$

Multiplying the objective by n, using the reproducing property:

$$\begin{aligned} J(f) &= \sum_{j=1}^{n} [y_j - \langle f, k(\cdot, x_j)\rangle_{\mathcal{H}}]^2 + \lambda n \|f\|_{\mathcal{H}}^2 \\ &= \|\mathbf{y} - \mathbf{G}\mathbf{a}\|_2^2 + (\lambda n)\mathbf{a}^T\mathbf{G}\mathbf{a} \\ &= \mathbf{y}^T\mathbf{y} - 2\mathbf{y}^T\mathbf{G}\mathbf{a} + \mathbf{a}^T\left[\mathbf{G}^2 + (\lambda n)\mathbf{G}\right]\mathbf{a}. \end{aligned}$$

Solving $\mathbf{0} = \frac{\partial J}{\partial \mathbf{a}}$, one gets $\mathbf{a}^* = (\mathbf{G} + \lambda n \mathbf{I})^{-1}\mathbf{y}$

Kernel ridge regression

By the representer theorem

$$f(x) = \sum_{i=1}^{n} a_i k(\cdot, x_i).$$

Multiplying the objective by n, using the reproducing property:

$$\begin{aligned} J(f) &= \sum_{j=1}^{n} [y_j - \langle f, k(\cdot, x_j) \rangle_{\mathcal{H}}]^2 + \lambda n \|f\|_{\mathcal{H}}^2 \\ &= \|\mathbf{y} - \mathbf{G}\mathbf{a}\|_2^2 + (\lambda n)\mathbf{a}^T \mathbf{G}\mathbf{a} \\ &= \mathbf{y}^T\mathbf{y} - 2\mathbf{y}^T\mathbf{G}\mathbf{a} + \mathbf{a}^T \left[\mathbf{G}^2 + (\lambda n)\mathbf{G}\right]\mathbf{a}. \end{aligned}$$

Solving $\mathbf{0} = \frac{\partial J}{\partial \mathbf{a}}$, one gets $\mathbf{a}^* = (\mathbf{G} + \lambda n \mathbf{I})^{-1}\mathbf{y}$ by

$$\frac{\partial \mathbf{a}^T \mathbf{B} \mathbf{a}}{\partial \mathbf{a}} = \left(\mathbf{B} + \mathbf{B}^T\right)\mathbf{a}, \quad \frac{\partial \mathbf{c}^T \mathbf{a}}{\partial \mathbf{a}} = \mathbf{c}.$$

Representer theorem

- Given: $\{(x_i, y_i)\}_{i=1}^n$, say classification/regression.
- Goal:

$$J(f) = V(x_1, y_1, f(x_1), \ldots, x_n, y_n, f(x_n)) + r\left(\|f\|^2_{\mathcal{H}_k}\right) \to \min_{\mathcal{H}_k},$$

r : monotonically increasing.

Representer theorem

- Given: $\{(x_i, y_i)\}_{i=1}^n$, say classification/regression.
- Goal:

$$J(f) = V(x_1, y_1, f(x_1), \ldots, x_n, y_n, f(x_n)) + r\left(\|f\|_{\mathcal{H}_k}^2\right) \to \min_{\mathcal{H}_k},$$

r : monotonically increasing.

- Example:

$$V(\ldots) = \frac{1}{n}\sum_{i=1}^{n}[f(x_i) - y_i]^2 \qquad \text{(regression)},$$

$$V(\ldots) = \frac{1}{n}\sum_{i=1}^{n}\max(1 - y_i f(x_i), 0) \quad \text{(soft classification)}.$$

Representer theorem – continued

…then

- $\boxed{\exists \text{ solution}}$ in the form:

$$\boxed{f^* = \sum_{i=1}^n \alpha_i k(\cdot, x_i)}, \quad \alpha_i \in \mathbb{R}.$$

- r: strictly increasing $\Rightarrow$ $\forall$ solution is of this form.
- Example: $r(z) = \lambda z$, $\lambda > 0$.

Representer theorem – proof

Objective

$$J(f) = V(x_1, y_1, f(x_1), \ldots, x_n, y_n, f(x_n)) + r\left(\|f\|_{\mathcal{H}_k}^2\right) \to \min_{\mathcal{H}_k}.$$

Decompose & Pythagorean theorem:

$$\begin{aligned} S &= span\,(k(\cdot, x_i), i = 1, \ldots, n)\,, \\ f &= f_S + f_\perp, \\ \|f\|_{\mathcal{H}_k}^2 &= \|f_S\|_{\mathcal{H}_k}^2 + \underbrace{\|f_\perp\|_{\mathcal{H}_k}^2}_{\geqslant 0} \geqslant \|f_S\|_{\mathcal{H}_k}^2\,. \end{aligned}$$

Representer theorem – proof

Objective

$$J(f) = V(x_1, y_1, f(x_1), \ldots, x_n, y_n, f(x_n)) + r\left(\|f\|^2_{\mathcal{H}_k}\right) \to \min_{\mathcal{H}_k}.$$

Decompose & Pythagorean theorem:

$$\begin{aligned} S &= span\left(k(\cdot, x_i), i = 1, \ldots, n\right), \\ f &= f_S + f_\perp, \\ \|f\|^2_{\mathcal{H}_k} &= \|f_S\|^2_{\mathcal{H}_k} + \underbrace{\|f_\perp\|^2_{\mathcal{H}_k}}_{\geqslant 0} \geqslant \|f_S\|^2_{\mathcal{H}_k}. \end{aligned}$$

In J

- 1st term: depends on f_S only, $f(x_i) = \langle f, k(\cdot, x_i)\rangle_{\mathcal{H}_k} = \langle f_S + f_\perp, k(\cdot, x_i)\rangle_{\mathcal{H}_k} = \langle f_S, k(\cdot, x_i)\rangle_{\mathcal{H}_k}$.

Representer theorem – proof

Objective

$$J(f) = V\left(x_1, y_1, f(x_1), \ldots, x_n, y_n, f(x_n)\right) + r\left(\|f\|^2_{\mathcal{H}_k}\right) \to \min_{\mathcal{H}_k}.$$

Decompose & Pythagorean theorem:

$$\begin{aligned} S &= span\left(k(\cdot, x_i), i = 1, \ldots, n\right), \\ f &= f_S + f_\perp, \\ \|f\|^2_{\mathcal{H}_k} &= \|f_S\|^2_{\mathcal{H}_k} + \underbrace{\|f_\perp\|^2_{\mathcal{H}_k}}_{\geqslant 0} \geqslant \|f_S\|^2_{\mathcal{H}_k}. \end{aligned}$$

In J

- 1st term: depends on f_S only, $f(x_i) = \langle f, k(\cdot, x_i)\rangle_{\mathcal{H}_k} = \langle f_S + f_\perp, k(\cdot, x_i)\rangle_{\mathcal{H}_k} = \langle f_S, k(\cdot, x_i)\rangle_{\mathcal{H}_k}$.
- 2nd term: can only decrease by neglecting $f_\perp$ ($r \nearrow$).

Regression on labelled bags

- Given:
  - labelled bags: $\left\{\left(\hat{\mathbb{P}}_i, y_i\right)\right\}_{i=1}^n$, $\hat{\mathbb{P}}_i$: bag from $\mathbb{P}_i$, $N := |\hat{\mathbb{P}}_i|$.
  - test bag: $\hat{\mathbb{P}}$.

Regression on labelled bags

- Given:
  - labelled bags: $\left\{\left(\hat{\mathbb{P}}_i, y_i\right)\right\}_{i=1}^n$, $\hat{\mathbb{P}}_i$: bag from $\mathbb{P}_i$, $N := |\hat{\mathbb{P}}_i|$.
  - test bag: $\hat{\mathbb{P}}$.
- Estimator:

$$\hat{f} = \arg\min_{f \in \mathcal{H}} \frac{1}{n} \sum\nolimits_{i=1}^n \left[f\big(\underbrace{\mu_{\hat{\mathbb{P}}_i}}_{\text{feature of } \hat{\mathbb{P}}_i}\big) - y_i \right]^2 + \lambda \left\| f \right\|_{\mathcal{H}}^2 .$$

Regression on labelled bags

- Given:
  - labelled bags: $\left\{\left(\hat{\mathbb{P}}_i, y_i\right)\right\}_{i=1}^n$, $\hat{\mathbb{P}}_i$: bag from $\mathbb{P}_i$, $N := |\hat{\mathbb{P}}_i|$.
  - test bag: $\hat{\mathbb{P}}$.
- Estimator:

$$\hat{f} = \arg\min_{f \in \mathcal{H}_K} \frac{1}{n} \sum_{i=1}^n \left[f\left(\mu_{\hat{\mathbb{P}}_i}\right) - y_i \right]^2 + \lambda \left\| f \right\|_{\mathcal{H}_K}^2 .$$

- Prediction:

$$\begin{aligned} \hat{y}\left(\hat{\mathbb{P}}\right) &= \hat{f}\left(\mu_{\hat{\mathbb{P}}}\right) = \mathbf{g}^T (\mathbf{G} + n\lambda \mathbf{I})^{-1} \mathbf{y}, \\ \mathbf{g} &= \left[K\left(\mu_{\hat{\mathbb{P}}}, \mu_{\hat{\mathbb{P}}_i}\right)\right], \mathbf{G} = \left[K\left(\mu_{\hat{\mathbb{P}}_i}, \mu_{\hat{\mathbb{P}}_j}\right)\right], \mathbf{y} = [y_i]. \end{aligned}$$

Regression on labelled bags

- Given:
  - labelled bags: $\left\{\left(\hat{\mathbb{P}}_i, y_i\right)\right\}_{i=1}^n$, $\hat{\mathbb{P}}_i$: bag from $\mathbb{P}_i$, $N := |\hat{\mathbb{P}}_i|$.
  - test bag: $\hat{\mathbb{P}}$.
- Estimator:

$$\hat{f} = \arg\min_{f \in \mathcal{H}_K} \frac{1}{n} \sum_{i=1}^n \left[f\left(\mu_{\hat{\mathbb{P}}_i}\right) - y_i \right]^2 + \lambda \|f\|_{\mathcal{H}_K}^2 .$$

- Prediction:

$$\hat{y}\left(\hat{\mathbb{P}}\right) = \hat{f}\left(\mu_{\hat{\mathbb{P}}}\right) = \mathbf{g}^T(\mathbf{G} + n\lambda\mathbf{I})^{-1}\mathbf{y},$$

$$\mathbf{g} = \left[K\left(\mu_{\hat{\mathbb{P}}}, \mu_{\hat{\mathbb{P}}_i}\right)\right], \mathbf{G} = \left[K\left(\mu_{\hat{\mathbb{P}}_i}, \mu_{\hat{\mathbb{P}}_j}\right)\right], \mathbf{y} = [y_i].$$

Inner product of distributions

$K\left(\mu_{\hat{\mathbb{P}}_i}, \mu_{\hat{\mathbb{P}}_j}\right) = ?$

Distribution representation via functions

$$\boxed{\mathbb{P} \mapsto \mu_{\mathbb{P}} = \int_{\mathcal{X}} \varphi(x) \mathrm{d}\mathbb{P}(x)}.$$

Distribution representation via functions

$$\boxed{\mathbb{P} \mapsto \mu_{\mathbb{P}} = \int_{\mathcal{X}} \varphi(x) \mathrm{d}\mathbb{P}(x)}.$$

- Cdf:

$$\mathbb{P} \mapsto F_{\mathbb{P}}(z) = \mathbb{E}_{x \sim \mathbb{P}} \chi_{(-\infty,z)}(x).$$

Distribution representation via functions

$$\boxed{\mathbb{P} \mapsto \mu_{\mathbb{P}} = \int_{\mathcal{X}} \varphi(x) \mathrm{d}\mathbb{P}(x)}.$$

- Cdf:

$$\mathbb{P} \mapsto F_{\mathbb{P}}(z) = \mathbb{E}_{x \sim \mathbb{P}} \chi_{(-\infty,z)}(x).$$

- Characteristic function:

$$\mathbb{P} \mapsto c_{\mathbb{P}}(z) = \int e^{i\langle z,x \rangle} \mathrm{d}\mathbb{P}(x).$$

Distribution representation via functions

$$\boxed{\mathbb{P} \mapsto \mu_{\mathbb{P}} = \int_{\mathcal{X}} \varphi(x) \mathrm{d}\mathbb{P}(x)}.$$

- Cdf:

$$\mathbb{P} \mapsto \mathsf{F}_{\mathbb{P}}(z) = \mathbb{E}_{x \sim \mathbb{P}} \chi_{(-\infty, z)}(x).$$

- Characteristic function:

$$\mathbb{P} \mapsto \mathsf{c}_{\mathbb{P}}(z) = \int e^{i\langle z, x \rangle} \mathrm{d}\mathbb{P}(x).$$

- Moment generating function:

$$\mathbb{P} \mapsto \mathsf{M}_{\mathbb{P}}(z) = \int e^{\langle z, x \rangle} \mathrm{d}\mathbb{P}(x).$$

Distribution representation via functions

$$\boxed{\mathbb{P} \mapsto \mu_{\mathbb{P}} = \int_{\mathcal{X}} \varphi(x) \mathrm{d}\mathbb{P}(x)}.$$

- Cdf:

$$\mathbb{P} \mapsto \mathsf{F}_{\mathbb{P}}(z) = \mathbb{E}_{x\sim\mathbb{P}} \chi_{(-\infty,z)}(x).$$

- Characteristic function:

$$\mathbb{P} \mapsto \mathsf{c}_{\mathbb{P}}(z) = \int e^{i\langle z,x\rangle} \mathrm{d}\mathbb{P}(x).$$

- Moment generating function:

$$\mathbb{P} \mapsto \mathsf{M}_{\mathbb{P}}(z) = \int e^{\langle z,x\rangle} \mathrm{d}\mathbb{P}(x).$$

Trick

φ: on any kernel-endowed domain! $\varphi(x) := k(\cdot,x)$, $\mu_{\mathbb{P}} \in \mathcal{H}_k$.

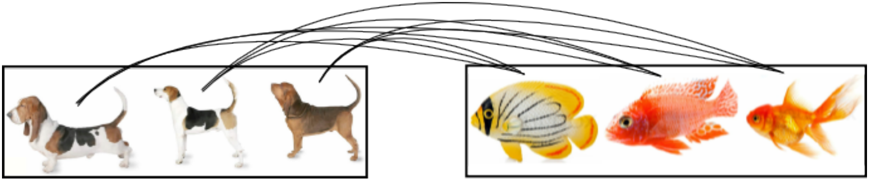
How to compute $K(\mu_{\hat{\mathbb{P}}}, \mu_{\hat{\mathbb{Q}}})$?

$$\sim P$$

$$\sim Q$$

Computation:

$$K(\mathbb{P}_m, \mathbb{Q}_n) := \langle \mu_{\mathbb{P}_m}, \mu_{\mathbb{Q}_n} \rangle_{\mathcal{H}_k} = \frac{1}{mn} \sum_{i=1}^{m} \sum_{j=1}^{n} k(x_i, y_j).$$



Local wrap-up

- Least squares ($\lambda = 0$):

$$J(\mathbf{b}) = \frac{1}{n}\sum_{i=1}^{n}[\langle \mathbf{b}, \mathbf{x}_i\rangle - y_i]^2 \to \min_{\mathbf{b}\in\mathbb{R}^p}.$$

- Ridge regression ($k(\mathbf{x}, \mathbf{x}') = \langle \mathbf{x}, \mathbf{x}'\rangle$):

$$J(\mathbf{b}) = \frac{1}{n}\sum_{i=1}^{n}[\langle \mathbf{b}, \mathbf{x}_i\rangle - y_i]^2 + \lambda \|\mathbf{b}\|_2^2 \to \min_{\mathbf{b}\in\mathbb{R}^p}.$$

- Kernel ridge regression:

$$J(f) = \frac{1}{n}\sum_{i=1}^{n}[\underbrace{\langle f, \varphi(x_i)\rangle_{\mathcal{H}_k}}_{f(x_i)} - y_i]^2 + \lambda \|f\|_{\mathcal{H}_k}^2 \to \min_{f\in\mathcal{H}_k}.$$

Sparsity

Sparse coding

- Least squares (minor rescaling):

$$J(\mathbf{b}) = \frac{1}{2}\|\mathbf{y} - \mathbf{X}\mathbf{b}\|_2^2 \to \min_{\mathbf{b}\in\mathbb{R}^p}.$$

- Sparse coding:

$$J(\mathbf{b}) = \frac{1}{2}\|\mathbf{y} - \mathbf{X}\mathbf{b}\|_2^2 \to \min_{\mathbf{b}\in\mathbb{R}^p:\|\mathbf{b}\|_0\le B}.$$

$B \in \mathbb{Z}^+$: max # of non-zero coordinates in $\mathbf{b}$.

Motivation

interpretability /computation / JPEG

Sparse coding: solution

All subset method:

- Easy: try all $S \subset \{1, \ldots, p\}$, $|S| \leqslant B$.

Sparse coding: solution

All subset method:

- Easy: try all $S \subset \{1, \ldots, p\}$, $|S| \leqslant B$.
- Solve: $\mathbf{X}_S$ restriction of $\mathbf{X}$ to the columns in S,

$$J(\mathbf{b}_S) = \frac{1}{2} \|\mathbf{y} - \mathbf{X}_S \mathbf{b}_S\|_2^2 \to \min_{\mathbf{b}_S \in \mathbb{R}^p}.$$

Sparse coding: solution

All subset method:

- Easy: try all $S \subset \{1, \dots, p\}$, $|S| \leqslant B$.
- Solve: $\mathbf{X}_S$ restriction of $\mathbf{X}$ to the columns in S,

$$J(\mathbf{b}_S) = \frac{1}{2} \|\mathbf{y} - \mathbf{X}_S \mathbf{b}_S\|_2^2 \to \min_{\mathbf{b}_S \in \mathbb{R}^p}.$$

- Pick the S for which $J(\mathbf{b}_S)$ is minimal!

Sparse coding: solution

All subset method:

- Easy: try all $S \subset \{1,\dots,p\}$, $|S| \leqslant B$.
- Solve: $\mathbf{X}_S$ restriction of $\mathbf{X}$ to the columns in S,

$$J(\mathbf{b}_S) = \frac{1}{2}\|\mathbf{y} - \mathbf{X}_S\mathbf{b}_S\|_2^2 \to \min_{\mathbf{b}_S \in \mathbb{R}^p}.$$

- Pick the S for which $J(\mathbf{b}_S)$ is minimal!

The end?

Number of subsets

- $|S| = 1$: $\binom{p}{1} = p$.

Number of subsets

- $|S| = 1$: $\binom{p}{1} = p$.
- $|S| = 2$: $\binom{p}{2} = \frac{p(p-1)}{2} = \mathcal{O}(p^2)$.

Number of subsets

- $|S| = 1$: $\binom{p}{1} = p$.
- $|S| = 2$: $\binom{p}{2} = \frac{p(p-1)}{2} = \mathcal{O}(p^2)$.
- $\vdots$
- $|S| = B$: $\binom{p}{B} = \mathcal{O}(p^B)$.

Number of subsets

- $|S| = 1$: $\binom{p}{1} = p$.
- $|S| = 2$: $\binom{p}{2} = \frac{p(p-1)}{2} = \mathcal{O}(p^2)$.
- $\vdots$
- $|S| = B$: $\binom{p}{B} = \mathcal{O}(p^B)$.

In total: $\sum_{i=1}^{B} \binom{p}{i} \geqslant \boxed{\mathcal{O}(p^B)}$.

'Almost' scalable!

Sparse coding

- Sparse coding-1:

$$J(\mathbf{b}) = \frac{1}{2}\left\|\mathbf{y} - \mathbf{X}\mathbf{b}\right\|_2^2 \to \min_{\mathbf{b}\in\mathbb{R}^p : \|\mathbf{b}\|_0 \leqslant B}.$$

- Sparse coding-2 ($B \in \mathbb{R}^+$):

$$J(\mathbf{b}) = \frac{1}{2}\left\|\mathbf{y} - \mathbf{X}\mathbf{b}\right\|_2^2 \to \min_{\mathbf{b}\in\mathbb{R}^p : \|\mathbf{b}\|_1 \leqslant B}, \tag{1}$$

where $\|\mathbf{b}\|_1 = \sum_{i=1}^p |b_i|$.

Sparse coding

- Sparse coding-1:

$$J(\mathbf{b}) = \frac{1}{2}\|\mathbf{y} - \mathbf{X}\mathbf{b}\|_2^2 \to \min_{\mathbf{b}\in\mathbb{R}^p : \|\mathbf{b}\|_0 \leqslant B}.$$

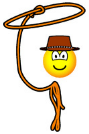
- Sparse coding-2 ($B \in \mathbb{R}^+$):

$$J(\mathbf{b}) = \frac{1}{2}\|\mathbf{y} - \mathbf{X}\mathbf{b}\|_2^2 \to \min_{\mathbf{b}\in\mathbb{R}^p : \|\mathbf{b}\|_1 \leqslant B}, \tag{1}$$

  where $\|\mathbf{b}\|_1 = \sum_{i=1}^p |b_i|$.

- Lasso: (1) is equivalent (for some $\lambda > 0$) to

$$J(\mathbf{b}) = \boxed{\frac{1}{2}\|\mathbf{y} - \mathbf{X}\mathbf{b}\|_2^2 + \lambda\|\mathbf{b}\|_1} \to \min_{\mathbf{b}\in\mathbb{R}^p}.$$



Rationale

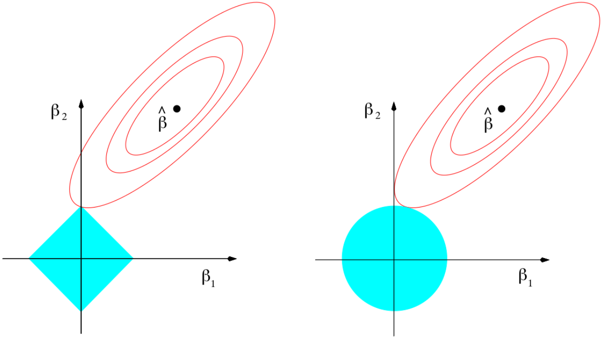
- $\|\mathbf{b}\|_1$: convex relaxation of $\|\mathbf{b}\|_0$.

Rationale

- $\|\mathbf{b}\|_1$: convex relaxation of $\|\mathbf{b}\|_0$.
- If $\mathbf{b}$ is sparse and $\mathbf{X}$ is RIP: $\|\cdot\|_1$ minimization is equivalent $\|\cdot\|_0$.

Rationale

- $\|\mathbf{b}\|_1$: convex relaxation of $\|\mathbf{b}\|_0$.
- If $\mathbf{b}$ is sparse and $\mathbf{X}$ is RIP: $\|\cdot\|_1$ minimization is equivalent $\|\cdot\|_0$.
- In terms of objective:



Lasso solver: ISTA/FISTA

$$J(\mathbf{b}) = \underbrace{\frac{1}{2}\left\|\mathbf{y} - \mathbf{X}\mathbf{b}\right\|_2^2}_{=:f(\mathbf{b})} + \underbrace{\lambda\left\|\mathbf{b}\right\|_1}_{=:g(\mathbf{b})} \to \min_{\mathbf{b}}.$$

Lasso solver: ISTA/FISTA

$$J(\mathbf{b}) = \underbrace{\frac{1}{2}\|\mathbf{y} - \mathbf{X}\mathbf{b}\|_2^2}_{=:f(\mathbf{b})} + \underbrace{\lambda\|\mathbf{b}\|_1}_{=:g(\mathbf{b})} \to \min_{\mathbf{b}}.$$

- f: smooth convex,

$$\|\nabla f(\mathbf{a}) - \nabla f(\mathbf{b})\|_2 \leqslant \underbrace{L}_{>0}\|\mathbf{a} - \mathbf{b}\|_2 \quad \forall \mathbf{a}, \mathbf{b}.$$

Lasso solver: ISTA/FISTA

$$J(\mathbf{b}) = \underbrace{\frac{1}{2}\|\mathbf{y} - \mathbf{X}\mathbf{b}\|_2^2}_{=:f(\mathbf{b})} + \underbrace{\lambda \|\mathbf{b}\|_1}_{=:g(\mathbf{b})} \to \min_{\mathbf{b}}.$$

- f: smooth convex,

$$\|\nabla f(\mathbf{a}) - \nabla f(\mathbf{b})\|_2 \leqslant \underbrace{L}_{>0} \|\mathbf{a} - \mathbf{b}\|_2 \quad \forall \mathbf{a}, \mathbf{b}.$$

Example: $f(\mathbf{b}) = \frac{1}{2}\|\mathbf{y} - \mathbf{X}\mathbf{b}\|_2^2$, smallest $L = \lambda_{max}\left(\mathbf{X}^T\mathbf{X}\right)$.

Lasso solver: ISTA/FISTA

$$J(\mathbf{b}) = \underbrace{\frac{1}{2}\left\|\mathbf{y}-\mathbf{X}\mathbf{b}\right\|_2^2}_{=:f(\mathbf{b})} + \underbrace{\lambda\left\|\mathbf{b}\right\|_1}_{=:g(\mathbf{b})} \to \min_{\mathbf{b}}.$$

- f: smooth convex,

$$\left\|\nabla f(\mathbf{a}) - \nabla f(\mathbf{b})\right\|_2 \leqslant \underbrace{L}_{>0}\left\|\mathbf{a}-\mathbf{b}\right\|_2 \quad \forall \mathbf{a}, \mathbf{b}.$$

  Example: $f(\mathbf{b}) = \frac{1}{2}\left\|\mathbf{y}-\mathbf{X}\mathbf{b}\right\|_2^2$, smallest $L = \lambda_{max}\left(\mathbf{X}^T\mathbf{X}\right)$.

- g: continuous, convex, often nonsmooth.

ISTA '=' gradient descent

- Gradient descent ($\delta_t > 0$):

$$\mathbf{b}_t = \mathbf{b}_{t-1} - \delta_t \nabla f(\mathbf{b}_{t-1}) \Leftrightarrow$$

$$\mathbf{b}_t = \arg\min_{\mathbf{b}} \left[f(\mathbf{b}_{t-1}) + \langle \mathbf{b} - \mathbf{b}_{t-1}, \nabla f(\mathbf{b}_{t-1}) \rangle + \frac{1}{2\delta_t} \left\| \mathbf{b} - \mathbf{b}_{t-1} \right\|_2^2 \right].$$

ISTA '=' gradient descent

- Gradient descent ($\delta_t > 0$):

$$\mathbf{b}_t = \mathbf{b}_{t-1} - \delta_t \nabla f(\mathbf{b}_{t-1}) \Leftrightarrow$$

$$\mathbf{b}_t = \arg\min_{\mathbf{b}} \left[f(\mathbf{b}_{t-1}) + \langle \mathbf{b} - \mathbf{b}_{t-1}, \nabla f(\mathbf{b}_{t-1}) \rangle + \frac{1}{2\delta_t} \|\mathbf{b} - \mathbf{b}_{t-1}\|_2^2 \right].$$

- Quadratic approximation of $f + g$ at $\mathbf{y}$:

$$\left(\widehat{f+g}\right)_L(\mathbf{b}, \mathbf{y}) := f(\mathbf{y}) + \langle \mathbf{b} - \mathbf{y}, \nabla f(\mathbf{y}) \rangle + \frac{L}{2} \|\mathbf{b} - \mathbf{y}\|_2^2 + g(\mathbf{b}). \Rightarrow$$

ISTA '=' gradient descent

- Gradient descent ($\delta_t > 0$):

$$\mathbf{b}_t = \mathbf{b}_{t-1} - \delta_t \nabla f(\mathbf{b}_{t-1}) \Leftrightarrow$$

$$\mathbf{b}_t = \arg\min_{\mathbf{b}} \left[f(\mathbf{b}_{t-1}) + \langle \mathbf{b} - \mathbf{b}_{t-1}, \nabla f(\mathbf{b}_{t-1}) \rangle + \frac{1}{2\delta_t} \|\mathbf{b} - \mathbf{b}_{t-1}\|_2^2 \right].$$

- Quadratic approximation of $f + g$ at $\mathbf{y}$:

$$\left(\widehat{f + g}\right)_L(\mathbf{b}, \mathbf{y}) := f(\mathbf{y}) + \langle \mathbf{b} - \mathbf{y}, \nabla f(\mathbf{y}) \rangle + \frac{L}{2} \|\mathbf{b} - \mathbf{y}\|_2^2 + g(\mathbf{b}). \Rightarrow$$

$$p_L(\mathbf{y}) := \arg\min_{\mathbf{b}} \left(\widehat{f + g}\right)_L(\mathbf{b}, \mathbf{y})$$

ISTA '=' gradient descent

- Gradient descent ($\delta_t > 0$):

$$\mathbf{b}_t = \mathbf{b}_{t-1} - \delta_t \nabla f(\mathbf{b}_{t-1}) \Leftrightarrow$$

$$\mathbf{b}_t = \arg\min_{\mathbf{b}} \left[f(\mathbf{b}_{t-1}) + \langle \mathbf{b} - \mathbf{b}_{t-1}, \nabla f(\mathbf{b}_{t-1}) \rangle + \frac{1}{2\delta_t} \|\mathbf{b} - \mathbf{b}_{t-1}\|_2^2 \right].$$

- Quadratic approximation of $f + g$ at $\mathbf{y}$:

$$(\widehat{f+g})_L(\mathbf{b}, \mathbf{y}) := f(\mathbf{y}) + \langle \mathbf{b} - \mathbf{y}, \nabla f(\mathbf{y}) \rangle + \frac{L}{2} \|\mathbf{b} - \mathbf{y}\|_2^2 + g(\mathbf{b}). \Rightarrow$$

$$p_L(\mathbf{y}) := \arg\min_{\mathbf{b}} (\widehat{f+g})_L(\mathbf{b}, \mathbf{y})$$

$$= \arg\min_{\mathbf{b}} \left[g(\mathbf{b}) + \frac{L}{2} \left\| \mathbf{b} - \left(\mathbf{y} - \frac{1}{L} \nabla f(\mathbf{y}) \right) \right\|_2^2 \right]$$

ISTA '=' gradient descent

- Gradient descent ($\delta_t > 0$):

$$\mathbf{b}_t = \mathbf{b}_{t-1} - \delta_t \nabla f(\mathbf{b}_{t-1}) \Leftrightarrow$$

$$\mathbf{b}_t = \arg\min_{\mathbf{b}} \left[f(\mathbf{b}_{t-1}) + \langle \mathbf{b} - \mathbf{b}_{t-1}, \nabla f(\mathbf{b}_{t-1}) \rangle + \frac{1}{2\delta_t} \|\mathbf{b} - \mathbf{b}_{t-1}\|_2^2 \right].$$

- Quadratic approximation of $f + g$ at $\mathbf{y}$:

$$\widehat{(f+g)}_L(\mathbf{b}, \mathbf{y}) := f(\mathbf{y}) + \langle \mathbf{b} - \mathbf{y}, \nabla f(\mathbf{y}) \rangle + \frac{L}{2} \|\mathbf{b} - \mathbf{y}\|_2^2 + g(\mathbf{b}). \Rightarrow$$

$$\begin{aligned} p_L(\mathbf{y}) &:= \arg\min_{\mathbf{b}} \widehat{(f+g)}_L(\mathbf{b}, \mathbf{y}) \\ &= \arg\min_{\mathbf{b}} \left[g(\mathbf{b}) + \frac{L}{2} \left\| \mathbf{b} - \left(\mathbf{y} - \frac{1}{L} \nabla f(\mathbf{y}) \right) \right\|_2^2 \right] \\ &= prox_{\frac{1}{L} g} \left(\mathbf{y} - \frac{1}{L} \nabla f(\mathbf{y}) \right). \end{aligned}$$

ISTA: L given

Init: $\mathbf{b}_0$

for all $t = 1 : T$ **do**

  $\mathbf{b}_t = p_L\left(\mathbf{b}_{t-1}\right) \leftrightarrow$ gradient descent + 'projection'

ISTA: L given

1: Init: $\mathbf{b}_0$
2: **for all** $t = 1 : T$ **do**
3: $\quad\mathbf{b}_t = p_L\left(\mathbf{b}_{t-1}\right) \leftrightarrow$ gradient descent + 'projection'

Notes:

- L: does not have to be known – backtracking.

ISTA: L given

Init: $\mathbf{b}_0$
for all $t = 1 : T$ **do**
    $\mathbf{b}_t = p_L\left(\mathbf{b}_{t-1}\right) \leftrightarrow$ gradient descent $+$ 'projection'

Notes:

- L: does not have to be known – backtracking.
- Convergence: $O\left(\frac{1}{T}\right)$ in J-sense.

FISTA: L given

Init: $\mathbf{y}_1 = \mathbf{b}_0$, $\delta_1 = 1$
for all $t = 1 : T$ **do**
    $\mathbf{b}_t = p_L\left(\mathbf{y}_t\right)$
    $\delta_{t+1} = \frac{1+\sqrt{1+4\delta_t^2}}{2}$
    $\mathbf{y}_{t+1} = \mathbf{b}_t + \frac{\delta_t - 1}{\delta_{t+1}}\left(\mathbf{b}_t - \mathbf{b}_{t-1}\right)$

FISTA: L given

Init: $\mathbf{y}_1 = \mathbf{b}_0$, $\delta_1 = 1$
for all $t = 1 : T$ **do**
    $\mathbf{b}_t = p_L(\mathbf{y}_t)$
    $\delta_{t+1} = \frac{1+\sqrt{1+4\delta_t^2}}{2}$
    $\mathbf{y}_{t+1} = \mathbf{b}_t + \frac{\delta_t - 1}{\delta_{t+1}}(\mathbf{b}_t - \mathbf{b}_{t-1})$

Notes:

- L: not needed – backtracking.

FISTA: L given

1: Init: $\mathbf{y}_1 = \mathbf{b}_0$, $\delta_1 = 1$
2: **for all** $t = 1 : T$ **do**
3: $\quad \mathbf{b}_t = p_L(\mathbf{y}_t)$
4: $\quad \delta_{t+1} = \frac{1+\sqrt{1+4\delta_t^2}}{2}$
5: $\quad \mathbf{y}_{t+1} = \mathbf{b}_t + \frac{\delta_t - 1}{\delta_{t+1}}(\mathbf{b}_t - \mathbf{b}_{t-1})$

Notes:

- L: not needed – backtracking.
- Convergence: $O\left(\frac{1}{T^2}\right)$ in J-sense.

Local summary: ISTA/FISTA

We can solve

$$J(\mathbf{b}) = \underbrace{\frac{1}{2}\|\mathbf{y} - \mathbf{X}\mathbf{b}\|_2^2}_{=:f(\mathbf{b})} + \underbrace{\lambda\|\mathbf{b}\|_1}_{=:g(\mathbf{b})} \to \min_{\mathbf{b}}$$

type sparse coding problems quickly if

$$\nabla_f : \checkmark,$$

$$prox_g(\mathbf{v}) = \arg\min_{\mathbf{y}}\left[g(\mathbf{y}) + \frac{1}{2}\|\mathbf{y} - \mathbf{v}\|_2^2\right] \checkmark.$$

Prox: generalization of projection

$prox_g$ = Euclidean projection onto C if

$$g(\mathbf{y}) = I_C(\mathbf{y}) = \begin{cases} 0 & \mathbf{y} \in C, \\ \infty & \mathbf{y} \notin C. \end{cases}$$

Prox: properties

Our case: $g(\mathbf{y}) = \sum_m |y_m|$.

- Separable g: for $g(\mathbf{y}) = \sum_{m=1}^{M} g_m(\mathbf{y}_m)$

$$prox_g(\mathbf{y}_1, \ldots, \mathbf{y}_M) = \left[prox_{g_1}(\mathbf{y}_1); \ldots; prox_{g_M}(\mathbf{y}_M)\right].$$

Prox: properties

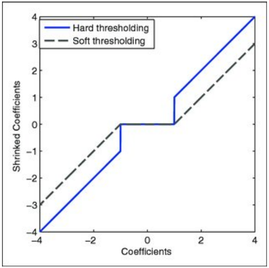
Our case: $g(\mathbf{y}) = \sum_m |y_m|$.

- Separable g: for $g(\mathbf{y}) = \sum_{m=1}^{M} g_m(\mathbf{y}_m)$

$$prox_g(\mathbf{y}_1, \ldots, \mathbf{y}_M) = \left[prox_{g_1}(\mathbf{y}_1); \ldots; prox_{g_M}(\mathbf{y}_M)\right].$$

- For $g(y) = |y|$

$$prox_{\kappa g}(y) = \begin{cases} y - \kappa & y \geqslant \kappa, \\ 0 & |y| \leqslant \kappa \\ y + \kappa & y \leqslant -\kappa. \end{cases}$$



Lasso optimization: coordinate descent

- Objective:

$$J(\mathbf{b}) = \frac{1}{2}\left\|\mathbf{y} - \mathbf{X}\mathbf{b}\right\|_2^2 + \lambda\left\|\mathbf{b}\right\|_1.$$

- Coordinate descent:

1: Init: $\hat{\mathbf{b}}$.
2: **repeat**
3: **for all** $j = 1 : p$ **do**
4: $\boxed{b_j \leftarrow \arg\min_{b_j} J(\mathbf{b})}$.
5: **until** convergence

Lasso: coordinate descent

- Objective: $J(\mathbf{b}) = \frac{1}{2}\left\|\mathbf{y} - \mathbf{X}\mathbf{b}\right\|_2^2 + \lambda \left\|\mathbf{b}\right\|_1$.
- $b_j^* = \arg\min_{b_j} J(\mathbf{b})$:

$$\boxed{b_j^* \text{ optimal} \Leftrightarrow 0 \in \frac{\partial[b_j \mapsto J(\mathbf{b})]}{\partial b_j}(b_j^*).}$$

Lasso: coordinate descent

- Objective: $J(\mathbf{b}) = \frac{1}{2}\|\mathbf{y} - \mathbf{X}\mathbf{b}\|_2^2 + \lambda\|\mathbf{b}\|_1$.
- $b_j^* = \arg\min_{b_j} J(\mathbf{b})$:

$$\boxed{b_j^* \text{ optimal} \Leftrightarrow 0 \in \frac{\partial[b_j \mapsto J(\mathbf{b})]}{\partial b_j}(b_j^*).}$$

- This gives

$$\rho_j = \sum_{i=1}^n x_{ij}\left(y_i - \sum_{a=1;a\neq j}^n b_a x_{ia}\right),$$
$$z_j = \sum_{i=1}^n (x_{ij})^2,$$
$$b_j^* = \frac{1}{z_j} S(\rho_j, \lambda),$$

where $S(\cdot, \lambda)$ is the soft-thresholding at λ.

Structured sparse coding ($\lambda > 0$):

$$J(\mathbf{b}) = \frac{1}{2}\|\mathbf{y} - \mathbf{X}\mathbf{b}\|_2^2 + \lambda \left\| \left(\|\mathbf{b}_G\|_2\right)_{G \in \mathcal{G}} \right\|_1 \to \min_{\mathbf{b}},$$

$$\mathcal{G} : \text{group structure on } \{1, \ldots, p\} = \cup_{G \in \mathcal{G}}.$$

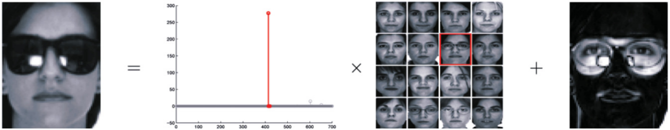
Structured sparse coding ($\lambda > 0$):

$$J(\mathbf{b}) = \frac{1}{2}\left\|\mathbf{y} - \mathbf{X}\mathbf{b}\right\|_2^2 + \lambda \left\| \left(\|\mathbf{b}_G\|_2\right)_{G\in\mathcal{G}} \right\|_1 \to \min_{\mathbf{b}},$$

$$\mathcal{G} : \text{group structure on } \{1,\ldots,p\} = \cup_{G\in\mathcal{G}}.$$

Non-overlapping group Lasso:

- $\mathcal{G}$ = partition: face recognition.



Lasso: extensions – FISTA handles them

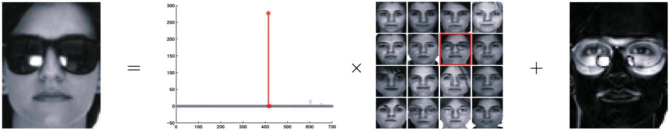
Structured sparse coding ($\lambda > 0$):

$$J(\mathbf{b}) = \frac{1}{2}\|\mathbf{y} - \mathbf{X}\mathbf{b}\|_2^2 + \lambda \left\| \left(\|\mathbf{b}_G\|_2 \right)_{G \in \mathcal{G}} \right\|_1 \to \min_{\mathbf{b}},$$

$\mathcal{G}$: group structure on $\{1, \ldots, p\} = \cup_{G \in \mathcal{G}}$.

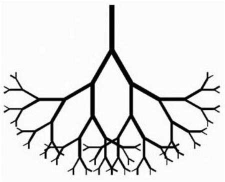
Non-overlapping group Lasso:

- $\mathcal{G}$ = partition: face recognition.



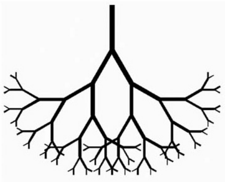
- prox: block soft-thresholding.

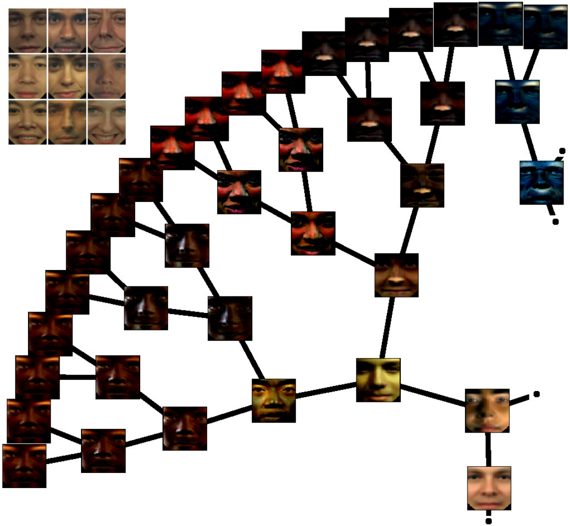
Overlapping $\mathcal{G}$ example:



$\mathcal{G}$: sub-trees & **X** learning

Overlapping $\mathcal{G}$ example:





Example: time series,

$$J(\mathbf{b}) = \frac{1}{2}\left\|y - \mathbf{X}\mathbf{b}\right\|_{\mathcal{H}_k}^2 + \lambda \left\| \left(\left\| \mathbf{b}_G \right\|_2 \right)_{G \in \mathcal{G}} \right\|_1 \to \min_{\mathbf{b} \in \mathbb{R}^n}.$$

Idea:

- $\mathbf{X} = [\varphi(x_1), \ldots, \varphi(x_n)]$.
- x_i samples:
  - evolution of facial muscle activities,
  - possibly different length.

Structured sparse coding in feature space

Example: time series,

$$J(\mathbf{b}) = \frac{1}{2}\left\|y - \mathbf{X}\mathbf{b}\right\|_{\mathcal{H}_k}^2 + \lambda \left\| \left(\|\mathbf{b}_G\|_2 \right)_{G \in \mathcal{G}} \right\|_1 \to \min_{\mathbf{b} \in \mathbb{R}^n}.$$

Idea:

- $\mathbf{X} = [\varphi(x_1), \ldots, \varphi(x_n)]$.
- x_i samples:
  - evolution of facial muscle activities,
  - possibly different length.
- G_j: indices of samples from the j^{th} emotion.

Structured sparse coding in feature space

Example: time series,

$$J(\mathbf{b}) = \frac{1}{2}\left\|y - \mathbf{X}\mathbf{b}\right\|_{\mathcal{H}_k}^2 + \lambda \left\| \left(\left\|\mathbf{b}_G\right\|_2\right)_{G\in\mathcal{G}} \right\|_1 \to \min_{\mathbf{b}\in\mathbb{R}^n}.$$

Idea:

- $\mathbf{X} = [\varphi(x_1), \ldots, \varphi(x_n)]$.
- x_i samples:
  - evolution of facial muscle activities,
  - possibly different length.
- G_j: indices of samples from the j^{th} emotion.
- $y = \varphi(x)$: $x =$ test time series.

Example: time series,

$$J(\mathbf{b}) = \frac{1}{2}\left\|y - \mathbf{X}\mathbf{b}\right\|_{\mathcal{H}_k}^2 + \lambda \left\| \left(\left\| \mathbf{b}_G \right\|_2 \right)_{G \in \mathcal{G}} \right\|_1 \to \min_{\mathbf{b}\in\mathbb{R}^n}.$$

Idea:

- $\mathbf{X} = [\varphi(x_1), \ldots, \varphi(x_n)]$.
- x_i samples:
  - evolution of facial muscle activities,
  - possibly different length.
- G_j: indices of samples from the j^{th} emotion.
- $y = \varphi(x)$: $x =$ test time series.
- $k(x, x') = \langle \varphi(x), \varphi(x') \rangle_{\mathcal{H}_k}$.

Rewriting

- Reformulation:

$$J(\mathbf{b}) = \frac{1}{2}\left\|\varphi(x) - \sum_{j=1}^{n} b_j\varphi(x_j)\right\|_{\mathcal{H}_k}^2 + \lambda\left\|\left(\|\mathbf{b}_G\|_2\right)_{G\in\mathcal{G}}\right\|_1 \to \min_{\mathbf{b}} \Leftrightarrow$$

$$\tilde{J}(\mathbf{b}) = \underbrace{\left(\frac{1}{2}\mathbf{b}^T\mathbf{G}\mathbf{b} - \mathbf{k}^T\mathbf{b}\right)}_{f(\mathbf{b})} + \underbrace{\lambda\left\|\left(\|\mathbf{b}_G\|_2\right)_{G\in\mathcal{G}}\right\|_1}_{g(\mathbf{b})} \to \min_{\mathbf{b}},$$

where $\mathbf{G} = [k(x_i, x_j)] \in \mathbb{R}^{n\times n}$, $\mathbf{k} = [k(x, x_1); \ldots ; k(x, x_n)] \in \mathbb{R}^n$.

Rewriting

- Reformulation:

$$J(\mathbf{b}) = \frac{1}{2}\left\|\varphi(x) - \sum_{j=1}^{n} b_j\varphi(x_j)\right\|_{\mathcal{H}_k}^2 + \lambda\left\|\left(\|\mathbf{b}_G\|_2\right)_{G\in\mathcal{G}}\right\|_1 \to \min_{\mathbf{b}} \Leftrightarrow$$

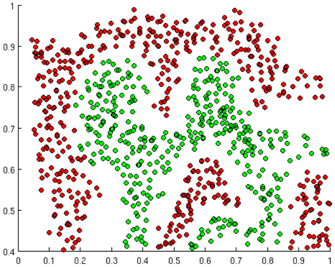
$$\tilde{J}(\mathbf{b}) = \underbrace{\left(\frac{1}{2}\mathbf{b}^T\mathbf{G}\mathbf{b} - \mathbf{k}^T\mathbf{b}\right)}_{f(\mathbf{b})} + \underbrace{\lambda\left\|\left(\|\mathbf{b}_G\|_2\right)_{G\in\mathcal{G}}\right\|_1}_{g(\mathbf{b})} \to \min_{\mathbf{b}},$$

where $\mathbf{G} = [k(x_i, x_j)] \in \mathbb{R}^{n\times n}$, $\mathbf{k} = [k(x, x_1); \ldots; k(x, x_n)] \in \mathbb{R}^n$.
- Optimization: FISTA. Classification:

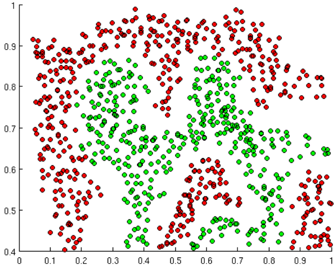
$$\hat{c} = \arg\max_c \|\mathbf{b}_{G_c}\|_2 .$$

Classification: original motivation of kernels, large margin

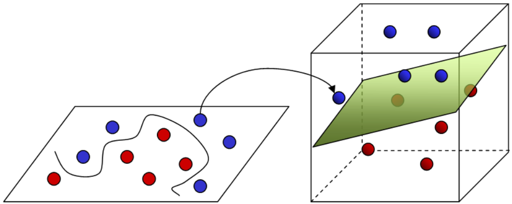
- Typical task:



- Typical task:



- Trick: the problem can be easier in the feature space



Classification

- Given: $\{(x_i, y_i)\}_{i=1}^n$ samples, $y_i \in \{-1, 1\}$.

Idea

$\hat{y}_i := f(x_i)$ and y_i should have the same sign!

Classification

- Given: $\{(x_i, y_i)\}_{i=1}^n$ samples, $y_i \in \{-1, 1\}$.

Idea

$\hat{y}_i := f(x_i)$ and y_i should have the same sign!

- Objective: non-linear SVMC ($C > 0$)

$$\min_{f \in \mathcal{H}_k, \boldsymbol{\xi}} C \underbrace{\sum_{i=1}^n \xi_i}_{\text{misclassification error}} + \frac{1}{2} \|f\|_{\mathcal{H}_k}^2, \quad \text{s.t. } y_i f(x_i) \geqslant 1 - \xi_i,\ \xi_i \geqslant 0,\ \forall i.$$

Classification

- Given: $\{(x_i, y_i)\}_{i=1}^n$ samples, $y_i \in \{-1, 1\}$.

Idea

$\hat{y}_i := f(x_i)$ and y_i should have the same sign!

- Objective: non-linear SVMC ($C > 0$, primal)

$$\min_{f \in \mathcal{H}_k, \boldsymbol{\xi}} C \underbrace{\sum_{i=1}^n \xi_i}_{\text{misclassification error}} + \frac{1}{2} \|f\|_{\mathcal{H}_k}^2, \quad \text{s.t. } y_i f(x_i) \geqslant 1 - \xi_i, \ \xi_i \geqslant 0, \ \forall i.$$

- Non-linear SVM (dual), still QP:

$$\max_{\boldsymbol{\alpha}} \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j=1}^n \alpha_i \alpha_j y_i y_j k(x_i, x_j), \ \text{s.t. } \sum_{i=1}^n \alpha_i y_i = 0, \ 0 \leqslant \alpha_i \leqslant C \ (\forall i).$$

Soft SVMC

Equivalent form:

$$\min_{f\in\mathcal{H}_k,\boldsymbol{\xi}} C\sum_{i=1}^n \xi_i + \frac{1}{2}\|f\|^2_{\mathcal{H}_k}, \text{ s.t. } y_i f(x_i) \geqslant 1-\xi_i,\ \xi_i \geqslant 0,\ \forall i. \Leftrightarrow$$

$$\min_{f\in\mathcal{H}_k} C\sum_{i=1}^n \underbrace{\max\left(1-y_i f(\mathbf{x}_i), 0\right)}_{=:h(y_i f(x_i))} + \frac{1}{2}\|f\|^2_{\mathcal{H}_k},$$

Equivalent form:

$$\min_{f\in\mathcal{H}_k,\boldsymbol{\xi}} C\sum_{i=1}^{n}\xi_i + \frac{1}{2}\|f\|_{\mathcal{H}_k}^2, \text{ s.t. } y_i f(x_i) \geqslant 1-\xi_i,\ \xi_i \geqslant 0,\ \forall i. \Leftrightarrow$$

$$\min_{f\in\mathcal{H}_k} C\sum_{i=1}^{n}\underbrace{\max\left(1-y_i f(\mathbf{x}_i), 0\right)}_{=:h(y_i f(x_i))} + \frac{1}{2}\|f\|_{\mathcal{H}_k}^2,$$

where $h(u) = \max(1-u, 0)$ is the hinge loss.

We use hinge loss in classification instead of squared.

Hard vs soft-SVM classification

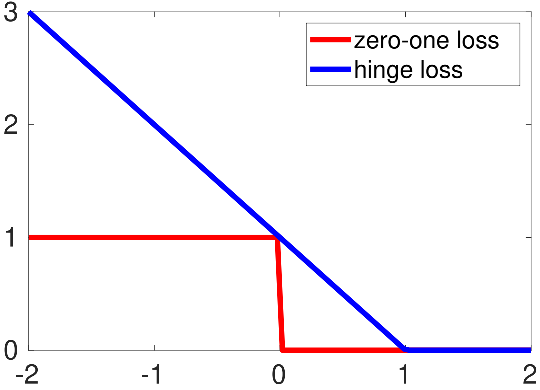
The hinge loss is the convex envelope of the zero-one loss:

$$z(u) = \mathbb{I}_{u<0}, \qquad u = y_i f(x_i),$$
$$h(u) = \max(1-u, 0).$$

Hard vs soft-SVM classification

The hinge loss is the convex envelope of the zero-one loss:

$$z(u) = \mathbb{I}_{u<0}, \qquad u = y_i f(x_i),$$
$$h(u) = \max(1-u, 0).$$



Studied task: line fitting

- Least squares, ridge regression, kernel ridge regression / classification:

$$J(\mathbf{b}) = \frac{1}{n}\sum_{i=1}^{n}\left[\langle \mathbf{b}, \mathbf{x}_i\rangle - y_i\right]^2, \quad J(\mathbf{b}) = \frac{1}{n}\sum_{i=1}^{n}\left[\langle \mathbf{b}, \mathbf{x}_i\rangle - y_i\right]^2 + \lambda \left\|\mathbf{b}\right\|_2^2,$$

$$J(f) = \frac{1}{n}\sum_{i=1}^{n}\left[f(x_i) - y_i\right]^2 + \lambda \left\|f\right\|_{\mathcal{H}_k}^2,$$

$$J(f) = \frac{1}{n}\sum_{i=1}^{n} h(y_i f(x_i)) + \lambda \left\|f\right\|_{\mathcal{H}_k}^2.$$

Studied task: line fitting

- Least squares, ridge regression, kernel ridge regression / classification:

$$J(\mathbf{b}) = \frac{1}{n}\sum_{i=1}^{n}[\langle \mathbf{b}, \mathbf{x}_i\rangle - y_i]^2, \quad J(\mathbf{b}) = \frac{1}{n}\sum_{i=1}^{n}[\langle \mathbf{b}, \mathbf{x}_i\rangle - y_i]^2 + \lambda \left\|\mathbf{b}\right\|_2^2,$$

$$J(f) = \frac{1}{n}\sum_{i=1}^{n}[f(x_i) - y_i]^2 + \lambda \left\|f\right\|_{\mathcal{H}_k}^2,$$

$$J(f) = \frac{1}{n}\sum_{i=1}^{n} h(y_i f(x_i)) + \lambda \left\|f\right\|_{\mathcal{H}_k}^2.$$

- Lasso, group Lasso, kernel group Lasso:

$$J(\mathbf{b}) = \frac{1}{2}\left\|\mathbf{y} - \mathbf{X}\mathbf{b}\right\|_2^2 + \lambda \left\|\mathbf{b}\right\|_1,$$

$$J(\mathbf{b}) = \frac{1}{2}\left\|\mathbf{y} - \mathbf{X}\mathbf{b}\right\|_2^2 + \lambda \left\| \left(\left\|\mathbf{b}_G\right\|_2\right)_{G\in\mathcal{G}} \right\|_1,$$

$$J(\mathbf{b}) = \frac{1}{2}\left\|\varphi(x) - \sum_{j=1}^{n} b_j\varphi(x_j)\right\|_{\mathcal{H}_k}^2 + \lambda \left\| \left(\left\|\mathbf{b}_G\right\|_2\right)_{G\in\mathcal{G}} \right\|_1.$$

Summary

- Optimization:
  - Least squares, (kernel) ridge regression: closed form.

Summary

- Optimization:
  - Least squares, (kernel) ridge regression: closed form.
  - Lasso, (kernel) group Lasso:
    - coordinate descent,
    - ISTA $\xrightarrow{\text{acceleration}}$ FISTA.

Summary

- Optimization:
  - Least squares, (kernel) ridge regression: closed form.
  - Lasso, (kernel) group Lasso:
    - coordinate descent,
    - ISTA $\xrightarrow{\text{acceleration}}$ FISTA.
  - SVMC: QP.

Summary

- Optimization:
  - Least squares, (kernel) ridge regression: closed form.
  - Lasso, (kernel) group Lasso:
    - coordinate descent,
    - ISTA $\xrightarrow{\text{acceleration}}$ FISTA.
  - SVMC: QP.
- Applications: house pricing, feature selection, face recognition, inpainting, collaborative filtering, aerosol prediction, emotion classification.

Thank you for the attention!