Introduction to Machine Learning

Foundations and Applications

Paul J. Atzberger
University of California Santa Barbara

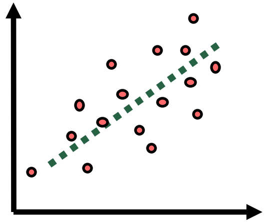
Regression

Regression

Consider

$y_i = f(x_i) + \epsilon_i$, where $f \in \mathcal{F}$ is sampled with $x \sim \mathcal{D}_{\mathcal{X}}$ and ϵ_i is noise with $\mathbb{E}[\epsilon_i] = 0$.

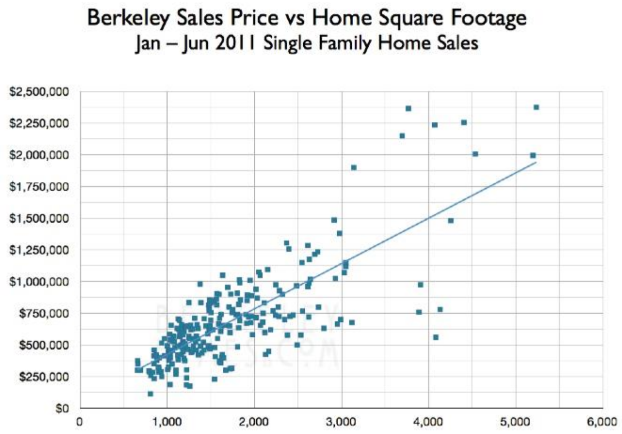
Task: From data samples $\mathcal{S} = \{(x_i, y_i)\}_{i=1}^m$ find model $h \in \mathcal{H}$ so that $y \sim h(x)$.

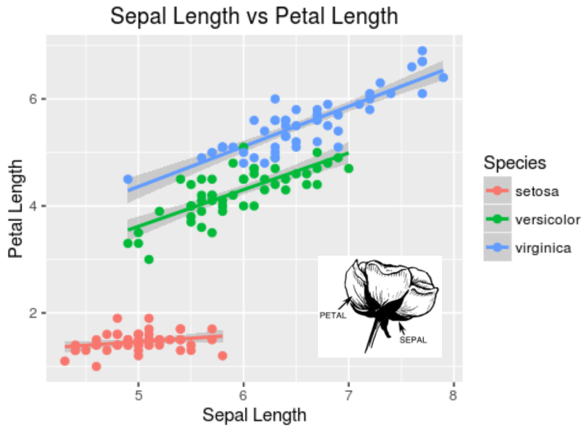


Linear regression: $h(x) = \boldsymbol{w} \cdot \boldsymbol{x} + b$. **Kernel regression:** $h(x) = \boldsymbol{w} \cdot \Phi(\boldsymbol{x}) + b$, with $\mathrm{k}(x_i, x_j) = \langle \Phi(\boldsymbol{x_i}), \Phi(\boldsymbol{x_j}) \rangle$.

Linear regression and variants among the most common.

Insights from weights w into how features $\mathbf{x}_i = (x_i^1, x_i^2, \ldots, x_i^N)$ contribute to y_i.





Regression

Consider
$y_i = f(x_i) + \epsilon_i$, where $f \in \mathcal{F}$ is sampled with $x \sim \mathcal{D}_{\mathcal{X}}$ and ϵ_i is noise with $\mathbb{E}[\epsilon_i] = 0$.

Task: From data samples $\mathcal{S} = \{(x_i, y_i)\}_{i=1}^m$ find model $h \in \mathcal{H}$ so that $y \sim h(x)$.

Loss Function: $L(y', y) : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$.

Examples: L_p**-loss:** $L(y', y) = \|y' - y\|_p^p$, special case L_2-loss (least squares) $L(h(x), f(x)) = \|h(x) - f(x)\|_2^2$.

Generalization Error (Risk):
$R(h) = \mathbb{E}_{x\sim\mathcal{D}}\left[L(h(x), f(x)\right]$.

Empirical Error (Empirical Risk):
$\hat{R}(h) = \frac{1}{m}\sum_{i=1}^m L(h(x_i), f(x_i))$.

Technical Assumption: We may find it useful to bound the loss functions $L(y', y) \leq M$, referred to as **(bounded regression problem)** .

Example: Loss $L(h(x), f(x)) = \min\{\||h(x) - f(x)\||_2^2, M\}$.

Many variants of regression:

- Linear Regression, Kernel Ridge Regression
- Support Vector Regression, LASSO Regression, ...

Regression: Motivation of Least-Squares

Regression: Consider

$y_i = f(x_i) + \eta_i$, with i.i.d. $\eta_i \sim \eta(0, \sigma^2) = \left[\text{Gausssian mean 0, variance } \sigma_*^2\right]$, and $f(x) = w_*^T x$.

Task: From $\mathcal{S} = \{(x_i, y_i)\}_{i=1}^m$ find model $h \in \mathcal{H} = \{h \mid h(x) = w^T x\}$.

Probabilistic Model: Predictions of the data use distribution $\tilde{y}_i = w^T x_i + \eta_i$ with $\eta_i \sim \eta(0, \sigma^2)$.

Probability Densities:

$$\textbf{noise: } \rho(\eta) = \left(2\pi\sigma^2\right)^{-1/2} \exp\left(-\frac{\eta^2}{2\sigma^2}\right) \Rightarrow \textbf{observation: } \rho(y_i \mid x_i, w) = \left(2\pi\sigma^2\right)^{-1/2} \exp\left(-\frac{\left(y_i - w^T x_i\right)^2}{2\sigma^2}\right).$$

For the full data set $\mathcal{S}$ we have

$$\rho(y_1, \ldots, y_m \mid x_1, \ldots, x_m; w) = \prod_{i=1}^m \rho(y_i \mid x_i, w) = \left(2\pi\sigma^2\right)^{-m/2} \exp\left(-\frac{\sum_{i=1}^m \left(y_i - w^T x_i\right)^2}{2\sigma^2}\right) = \underbrace{\mathcal{L}(w|\mathcal{S})}_{\text{Likelihood}}.$$

Maximum Likelihood Method: We can estimate w_* as

$$\tilde{w}^* = \arg\max_w \mathcal{L}(w|\mathcal{S}) \Rightarrow \tilde{w}^* = \arg\min_w \frac{1}{m} \sum_{i=1}^m \left(y_i - w^T x_i\right)^2.$$

This gives **Method of Least-Squares**.

Regression: Bayesian Motivation

Probability of Observations for Model w:

$$\rho(y_1,\ldots,y_m \mid x_1,\ldots,x_m; w) = \prod_{i=1}^{m} \rho(y_i \mid x_i, w) = \left(2\pi\sigma^2\right)^{-m/2} \exp\left(-\frac{\sum_{i=1}^{m}\left(y_i - w^T x_i\right)^2}{2\sigma^2}\right) = \underbrace{\mathcal{L}(w|\mathcal{S})}_{\text{Likelihood}}.$$

Bayes Rule for Posterior Distribution over Models w:

$$\Pr\{w|\mathcal{S}\} = \frac{\Pr\{\mathcal{S}|w\}\Pr\{w\}}{\Pr\{\mathcal{S}\}} = \frac{\overbrace{\mathcal{L}(w|\mathcal{S})}^{\textit{likelihood}}\,\overbrace{\Pr\{w\}}^{\text{prior}}}{\underbrace{\Pr\{\mathcal{S}\}}_{\text{evidence}}}.$$

Maximum A Posteriori (MAP) Estimate : We can estimate w_* as

$$\tilde{w}^* = \arg\min_w -\log\left(\Pr\{w|\mathcal{S}\}\right) \Rightarrow \tilde{w}^* = \arg\min_w \frac{1}{m}\sum_{i=1}^{m}\left(y_i - w^T x_i\right)^2 + \lambda R(w),\ R(w) = -\log\left(\Pr\{w\}\right), \lambda = \frac{2\sigma^2}{m}.$$

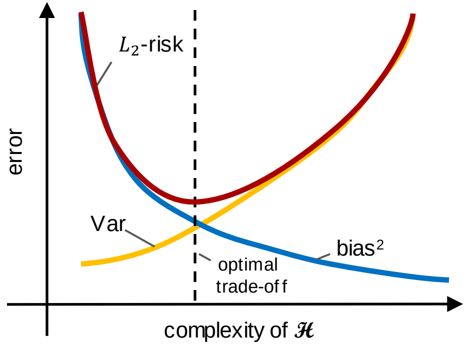
Role of Prior: For $\Pr\{w\}$ with $\rho(w) = \left(2\pi\nu^2\right)^{-1/2}\exp\left(-w^2/2\nu^2\right)$ we can take $R(w) = w^2$, $\lambda = \frac{\sigma^2}{m\nu^2} \in \mathbb{R}_+$.

Bayesian prior provides regularization $R(w)$ for selection of w (related to "ridge regression" methods).

As $\nu \to \infty$ the prior becomes increasingly less informative and $\lambda \to 0$ reducing regularization of least-squares.

Bias-Variance Trade-Off: L_2-Risk

L_2-Risk: $L(h(x), f(x)) = \|h(x) - f(x)\|_2^2$ with
$\mathcal{H} = \{$all measurable functions $x \sim \mathcal{D}\}$, f measurable.



Optimal Solution: $m = \arg\min_{h\in\mathcal{H}} \mathbb{E}_{\mathcal{D}}\left[L(h(X), Y)\right]$ is given by

$$m(x) = \mathbb{E}\left[Y|X = x\right].$$

Recovers $m(x) = f(x)$ except for set of measure zero $\sim \mathcal{D}$.

Regression: Consider $\mathcal{H}$ now more restrictive. Estimate $m_n(x) \in \mathcal{H}$ from n data samples $\mathcal{S}_n = \{(x_i, y_i)\}_{i=1}^n$.

L_2-error can be expressed as

$$\begin{aligned}
\mathbb{E}\left[|m_n(x) - m(x)|^2\right] &= \mathbb{E}\left[m_n^2(x) - 2m_n(x)m(x) + m^2(x)\right] = \mathbb{E}\left[m_n^2(x)\right] - 2\mathbb{E}\left[m_n(x)\right]m(x) + m^2(x) \\
&= \mathbb{E}\left[m_n^2(x)\right] - (\mathbb{E}\left[m_n\right])^2 + (\mathbb{E}\left[m_n\right])^2 - 2\mathbb{E}\left[m_n(x)\right]m(x) + m^2(x) \\
&= \mathrm{Var}\left[m_n(x)\right] + (\mathbb{E}\left[m_n(x)\right] - m(x))^2 \\
&= \mathrm{Var}\left[m_n(x)\right] + (\mathrm{bias}\,(m_n(x)))^2 .
\end{aligned}$$

Bias-Variance Trade-off: As complexity of $\mathcal{H}$ increases bias $\downarrow$ but Var $\uparrow$ since more sensitivity to changes in data samples $\mathcal{S}_n$ drawn.

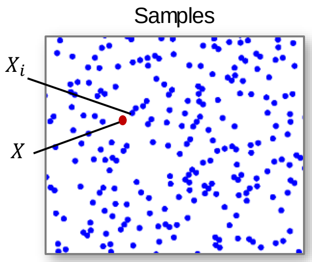
Generalization: Suggests balancing model accuracy on the training set with complexity to help generalization.

Curse of Dimensionality

Sampling on Unit Cube: Consider samples $X, X_1, X_2, \ldots, X_n \in [0,1]^d$ (d-dimensional hypercube).

Minimum Sample Distance: For n samples, denote the minimum distance between X and nearest sample X_i by

$d_\infty(d,n) = \mathbb{E}\left[\min_{i\in[1,n]} \|X - X_i\|_\infty\right]$



We can express in terms of probability as
$d_\infty(d,n) = \int_0^\infty \Pr\{\min_{i\in[1,n]} \|X - X_i\|_\infty > t\}dt = \int_0^\infty 1 - \Pr\{\min_{i\in[1,n]} \|X - X_i\|_\infty \leq t\}dt.$

The probability of being at most t apart in $\|\cdot\|_\infty$-norm is
$\Pr\{\min_{i\in[1,n]} \|X - X_i\|_\infty \leq t\} \leq n(2t)^d.$

Lower Bound on Distance: $$d_\infty(d,n) \geq \int_0^{1/2n^{1/d}} 1 - n(2t)^d dt = \frac{d}{2(d+1)}\frac{1}{n^{1/d}} \sim n^{-1/d}$$

samples:	$n = 10^2$	$n = 10^3$	$n = 10^4$	$n = 10^5$
$d_\infty(1,n)$	≥ 0.0025	≥ 0.00025	≥ 0.000025	≥ 0.0000025
$d_\infty(10,n)$	≥ 0.28	≥ 0.22	≥ 0.18	≥ 0.14
$d_\infty(20,n)$	≥ 0.37	≥ 0.34	≥ 0.30	≥ 0.26

Györfi 2002

Consequence: Shows for n samples, the minimum distance decreases very slowly for large d, $d_\infty \sim n^{-1/d}$.

Regression: Without using assumed structure, regression requires many samples to ensure accuracy.

Generalization Error Bounds

Regression: Rademacher Complexity

Notation and definitions:

$\mathcal{X}$ input space, $\mathcal{Y}$ output space
$\mathcal{C}$ concept class, concept f(x): $\mathcal{X} \to \mathcal{Y}$
$\mathcal{H}$ hypothesis class, hypothesis h(x): $\mathcal{X} \to \mathcal{Y}$.

Theorem: (regression bounds) Consider $\mathcal{H}$ so that $|h(x) - f(x)| \leq M$ for all $x \in \mathcal{X}, h \in \mathcal{H}$, then for any $p \geq 1$ and any $\delta > 0$ we have with probability $1 - \delta$ that the following bounds hold uniformly for $h \in \mathcal{H}$,

$$\mathrm{E}\left[\left|h(x) - f(x)\right|^p\right] \leq \frac{1}{m}\sum_{i=1}^{m}\left|h(x_i) - f(x_i)\right|^p + 2pM^{p-1}\mathfrak{R}_m(H) + M^p\sqrt{\frac{\log\frac{1}{\delta}}{2m}}, \quad \text{(Rademacher bound)}$$

$$\mathrm{E}\left[\left|h(x) - f(x)\right|^p\right] \leq \frac{1}{m}\sum_{i=1}^{m}\left|h(x_i) - f(x_i)\right|^p + 2pM^{p-1}\widehat{\mathfrak{R}}_S(H) + 3M^p\sqrt{\frac{\log\frac{2}{\delta}}{2m}}, \quad \text{(Empirical Rademacher bound)}$$

Significance: The expected value of the loss can be bounded by the observed empirical average. This differs at most by the Rademacher Complexity of regression class $\mathcal{H}$ plus a term vanishing as $m \to \infty$.

We see **complexity of the space of hypothesis functions** used for the regression effects **rate of convergence** of the generalization error as $m \to \infty$.

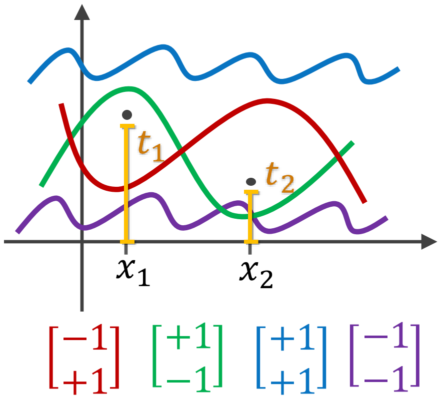
Key is to **find bounds** on the regression space **Rademacher complexity** $\mathcal{R}$(H).

Regression: Pseudo-dimension Bounds and VC-Dimension

Motivation: Are there combinatorial bounds similar in spirit to VC-dimension we can use to characterize complexity of regression spaces $\mathcal{H}$?

Definition: Let G be family of functions $\mathcal{X} \rightarrow \mathbb{R}$. We say a set $\{x_1,x_2,\ldots x_m\}$ is **shattered** by G if there exists $t_1,t_2,\ldots,t_m$ such that

$$\left|\left\{\begin{bmatrix}\operatorname{sgn}\left(g(x_1)-t_1\right)\\ \vdots \\ \operatorname{sgn}\left(g(x_m)-t_m\right)\end{bmatrix} : g \in G\right\}\right| = 2^m$$



We call the threshold values $t_1,t_2,\ldots,t_m$ the **witness** to the shattering.

Definition: For a family of functions G: $\mathcal{X} \rightarrow \mathbb{R}$ we define the **pseudo-dimension** of G denoted Pdim(G) as the largest m so a set of points is shattered.

Remark: This is related to VC-dim by considering corresponding classifiers

$$\mathrm{Pdim}(G) = \mathrm{VCdim}\Big(\left\{(x,t) \mapsto 1_{(g(x)-t)>0} : g \in G\right\}\Big)$$

Lemma (hyperplanes) The pseudo-dimension of hyperplanes in $\mathbb{R}^N$ is given by

$$Pdim(\{\mathbf{x} \mapsto \mathbf{w}\cdot\mathbf{x} + b : \mathbf{w} \in \mathbb{R}^N, b \in \mathbb{R}\}) = N + 1$$

Regression: Pseudo-dimension Bounds

Theorem: If the pseudo-dimension Pdim(G) = d then for any $\delta > 0$ we have with probability $1 - \delta$ that the following bounds hold uniformly for any $h \in \mathcal{H}$

$$R(h) \leq \widehat{R}(h) + M\sqrt{\frac{2d\log\frac{em}{d}}{m}} + M\sqrt{\frac{\log\frac{1}{\delta}}{2m}}.$$

where $G = \{x \to L(h(x), f(x)) \colon h \in H\}$, $L \leq M$.

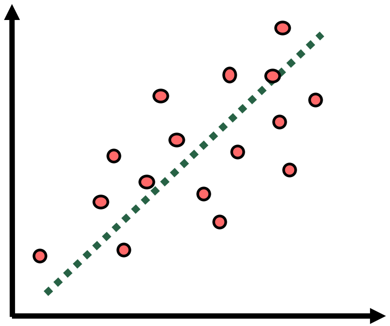
Remark: This gives analogous result as for VC-dimension. This is not tightest bound but gives worst-case guarantees when bounds on Rademacher complexity are difficult.

Remark: Hyperplanes in $\mathbb{R}^N$ (linear regression) $\mathcal{H} = \{h \mid h(x) = w^T x + b\}$ have d = N + 1.

Remark: Note, these bounds are when using only ERM. Alternatively, we also can use regularization and other strategies to select model h(x) (discussed later).

Linear Regression

Linear Regression



Optimization Problem:

$$\min_{\mathbf{w},b} \frac{1}{m}\sum_{i=1}^{m} (\mathbf{w}\cdot\mathbf{\Phi}(x_i) + b - y_i)^2$$

Equivalent Optimization Problem I:

$$\min_W F(W), \quad F(W) = \frac{1}{m}\|X^T W - Y\|^2 \qquad \mathbf{X} = \begin{bmatrix} \Phi(x_1) & \ldots & \Phi(x_m) \\ 1 & \ldots & 1 \end{bmatrix} \qquad \mathbf{W} = \begin{bmatrix} w_1 \\ \vdots \\ w_N \\ b \end{bmatrix} \qquad \mathbf{Y} = \begin{bmatrix} y_1 \\ \vdots \\ y_m \end{bmatrix}$$

Solution: $W = (XX^T)^{\dagger}XY$

$$\nabla_W F = 0, \ \Rightarrow \frac{2}{m} X\left(X^T W - Y\right) = 0 \ \Rightarrow XX^T W = XY \ \Rightarrow W = (XX^T)^{\dagger} XY.$$

Pick W with smallest $\|W\|_2$ when XX^T is non-invertible.

Pseudo-inverse: For matrix A the pseudo-inverse is

$$A^{\dagger} = \lim_{\gamma \downarrow 0} \left(A^T A + \gamma I\right)^{-1} A^T$$

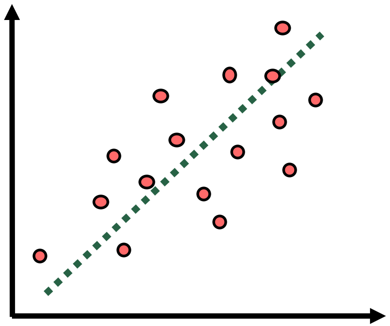
For $Ax = b$, $x = A^{\dagger} b \ \Leftarrow \ x^{\gamma} = \arg\min \|Ax - b\|_2^2 + \gamma\|x\|_2^2, \ \ x = \lim_{\gamma\downarrow 0} x^{\gamma}.$

When A is invertible, $A^{\dagger} = A^{-1}A^{-T}A^T = A^{-1}.$

Linear Regression

Equivalent Optimization Problem I:

$$\min_W F(W), \quad F(W) = \frac{1}{m}\|X^T W - Y\|^2 \qquad \mathbf{X} = \begin{bmatrix} \Phi(x_1) & \ldots & \Phi(x_m) \\ 1 & \ldots & 1 \end{bmatrix} \qquad \mathbf{W} = \begin{bmatrix} w_1 \\ \vdots \\ w_N \\ b \end{bmatrix} \qquad \mathbf{Y} = \begin{bmatrix} y_1 \\ \vdots \\ y_m \end{bmatrix}$$

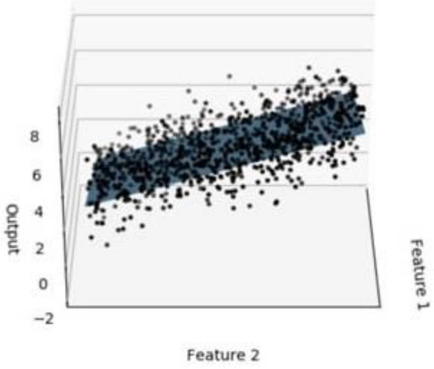


Solution: $W = (XX^T)^{\dagger} XY$

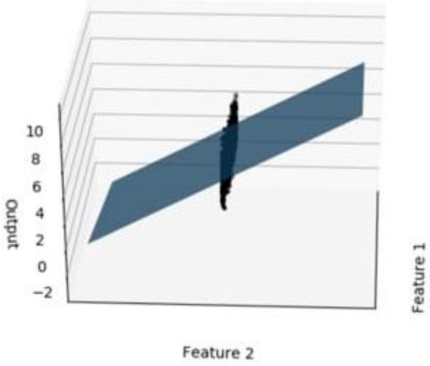
Issues when features x_i^a are strongly correlated with x_i^b , say equal, or one has a fixed value.

Strong correlations or co-linearity can result in XX^T **nearly-singular.** Results very sensitive to noise in data!

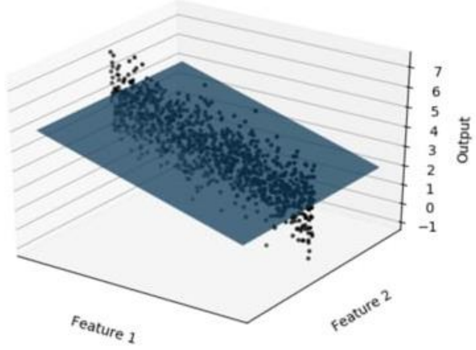
fit with features uncorrelated



fit with features correlated or fixed



pseudo-inverse fit with features correlated or fixed



Kernel Ridge Regression

Kernel Ridge Regression

Theorem: (ridge regression bounds) Consider kernel regression using $\mathcal{H} = \{h(x) = w \cdot \Phi(x) \mid \|w\|_2 \leq \Lambda\}$ with $K(x,x) \leq r^2$ and $|f(x)| \leq \Lambda r$ then for any $\delta > 0$ we have with probability $1-\delta$ that the following bounds hold uniformly for $h \in \mathcal{H}$

$$R(h) \leq \widehat{R}(h) + \frac{8r^2\Lambda^2}{\sqrt{m}}\left(1 + \frac{1}{2}\sqrt{\frac{\log\frac{1}{\delta}}{2}}\right)$$

$$R(h) \leq \widehat{R}(h) + \frac{8r^2\Lambda^2}{\sqrt{m}}\left(\sqrt{\frac{\mathrm{Tr}[\mathbf{K}]}{mr^2}} + \frac{3}{4}\sqrt{\frac{\log\frac{2}{\delta}}{2}}\right)$$

Significance: Provides tighter bounds than the combinatorial approach using pseudo-dimension.

Second bound provides **tighter estimate** since $Tr[K] \leq mr^2$, trace makes use of properties of the kernel.

Tightest bound from minimizing the RHS. This yields an optimization problem.

We need $\|w\|^2 \leq \Lambda^2$ so making Λ^2 as small as possible corresponds to making $\|w\|^2$ small. Can view bound as

$$R(h) \leq \widehat{R}(h) + \lambda\Lambda^2 \quad \text{where} \quad \lambda = \frac{8r^2}{\sqrt{m}}\left(1 + \frac{1}{2}\sqrt{\frac{\log\frac{1}{\delta}}{2}}\right) = O\left(\frac{1}{\sqrt{m}}\right).$$

Yields optimization problem

$$\min_{\mathbf{w}} F(\mathbf{w}) = \lambda\|\mathbf{w}\|^2 + \sum_{i=1}^{m}(\mathbf{w}\cdot\mathbf{\Phi}(x_i) - y_i)^2$$

Kernel Ridge Regression

Optimization Problem:

$$\min_{\mathbf{w}} F(\mathbf{w}), \quad F(\mathbf{w}) = \lambda\|\mathbf{w}\|^2 + \sum_{i=1}^{m} (\mathbf{w}\cdot\mathbf{\Phi}(x_i) - y_i)^2$$

$$\mathbf{X} = \begin{bmatrix} \Phi(x_1) & \dots & \Phi(x_m) \\ 1 & \dots & 1 \end{bmatrix} \qquad \mathbf{W} = \begin{bmatrix} w_1 \\ \vdots \\ w_N \\ b \end{bmatrix} \qquad \mathbf{Y} = \begin{bmatrix} y_1 \\ \vdots \\ y_m \end{bmatrix}$$

Equivalent Problem:

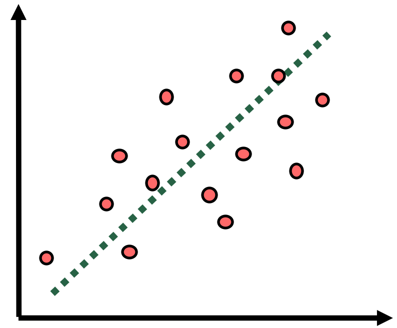
$$\min_w F(w), \quad F(w) = \lambda\|w\|^2 + \|X^T W - Y\|^2$$

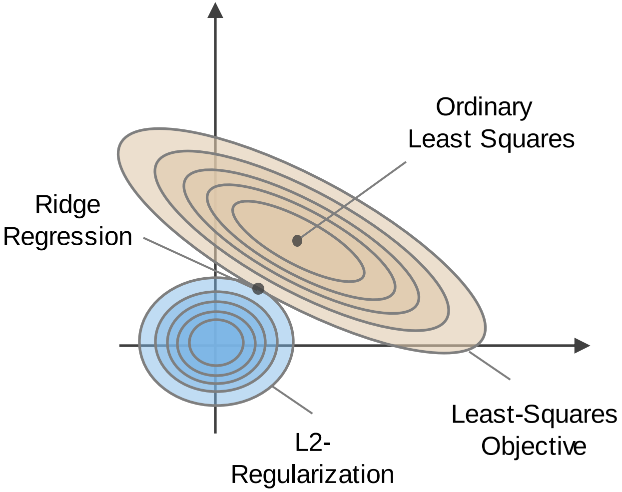
Solution:

$$\nabla_w F(w) = 0 \quad \Rightarrow \quad \left(XX^T + \lambda I\right) w = XY$$

$$\Rightarrow \quad w = \left(XX^T + \lambda I\right)^{-1} XY.$$

Kernelization using the dual formulation.





Kernel Ridge Regression

Primal Problem:

$$\min_{\mathbf{w}} F(\mathbf{w}), \qquad F(\mathbf{w}) = \lambda\|\mathbf{w}\|^2 + \sum_{i=1}^{m} (\mathbf{w}\cdot\mathbf{\Phi}(x_i) - y_i)^2$$

Equivalent optimization problem I:

$$\min_{\mathbf{w}} \sum_{i=1}^{m} (\mathbf{w}\cdot\mathbf{\Phi}(x_i) - y_i)^2 \quad \text{subject to: } \|\mathbf{w}\|^2 \leq \Lambda^2$$

Equivalent optimization problem II:

$$\min_{\mathbf{w}\ \xi_i} \sum_{i=1}^{m} \xi_i^2 \quad \text{subject to: } (\|\mathbf{w}\|^2 \leq \Lambda^2) \wedge \left(\forall i \in [1,m],\ \xi_i = y_i - \mathbf{w}\cdot\mathbf{\Phi}(x_i)\right)$$

Kernelization of the regression makes use of the **dual formulation.**

Lagrangian

$$\mathcal{L}(\boldsymbol{\xi}, \mathbf{w}, \boldsymbol{\alpha}', \lambda) = \sum_{i=1}^{m} \xi_i^2 + \sum_{i=1}^{m} \alpha_i'(y_i - \xi_i - \mathbf{w}\cdot\mathbf{\Phi}(x_i)) + \lambda(\|\mathbf{w}\|^2 - \Lambda^2)$$

Kernel Ridge Regression : Dual Formulation

Lagrangian

$$\mathcal{L}(\boldsymbol{\xi}, \mathbf{w}, \boldsymbol{\alpha}', \lambda) = \sum_{i=1}^{m} \xi_i^2 + \sum_{i=1}^{m} \alpha_i'(y_i - \xi_i - \mathbf{w}\cdot\mathbf{\Phi}(x_i)) + \lambda(\|\mathbf{w}\|^2 - \Lambda^2)$$

$$\mathbf{X} = \begin{bmatrix} \Phi(x_1) & \ldots & \Phi(x_m) \\ 1 & \ldots & 1 \end{bmatrix} \quad \mathbf{Y} = \begin{bmatrix} y_1 \\ \vdots \\ y_m \end{bmatrix}$$

KKT Conditions

$$\nabla_{\mathbf{w}}\mathcal{L} = -\sum_{i=1}^{m} \alpha_i'\mathbf{\Phi}(x_i) + 2\lambda\mathbf{w} = 0 \quad \Longrightarrow \quad \mathbf{w} = \frac{1}{2\lambda}\sum_{i=1}^{m} \alpha_i'\mathbf{\Phi}(x_i)$$

$$\nabla_{\xi_i}\mathcal{L} = 2\xi_i - \alpha_i' = 0 \quad \Longrightarrow \quad \xi_i = \alpha_i'/2$$

$$\forall i \in [1, m], \alpha_i'(y_i - \xi_i - \mathbf{w}\cdot\mathbf{\Phi}(x_i)) = 0$$

$$\lambda(\|\mathbf{w}\|^2 - \Lambda^2) = 0.$$

Solution:
$\alpha = (K + \lambda I)^{-1}\, Y$
$h(x) = w \cdot \Phi(x) = \sum_{i=1}^{m} \alpha_i k(x_i, x)$

Dual Formulation: Substitute w^*, ξ^* so $F(\alpha') = \inf_{w,\xi} \mathcal{L}(\xi, w, \alpha', \lambda) = \mathcal{L}(\xi^*, w^*, \alpha', \lambda)$.

$$F(\alpha') = \sum_{i=1}^{m} \frac{\alpha_i'^{,2}}{4} + \sum_{i=1}^{m} \alpha_i' y_i - \sum_{i=1}^{m} \frac{\alpha_i'^{,2}}{2} - \frac{1}{2\lambda}\sum_{i,j=1}^{m} \alpha_i'^{,2}\alpha_j'^{,2}\Phi(x_i)\cdot\Phi(x_j) + \lambda\left(\frac{1}{4\lambda^2}\left\|\sum_{i=1}^{m} \alpha_i'\Phi(x_i)\right\|^2 - \Lambda^2\right)$$

$$= -\lambda^2\sum_{i=1}^{m} \alpha_i^2 + 2\lambda\sum_{i=1}^{m} \alpha_i y_i - \lambda\sum_{i,j=1}^{m} \alpha_i\alpha_j\Phi(x_i)\cdot\Phi(x_j) - \lambda\Lambda^2, \quad \alpha_i = \alpha_i'/2\lambda.$$

Dual Optimization Problem:

$$\max_{\alpha\in\mathbb{R}} -\lambda\alpha^T\alpha + 2\alpha^T Y - \alpha^T\left(X^T X\right)\alpha \;\rightarrow\; \max_{\alpha\in\mathbb{R}} -\alpha^T(K + \lambda I)\alpha + 2\alpha^T Y. \;\Longrightarrow\; (K + \lambda I)\alpha = Y$$

Regression Examples

- Kernel Ridge Regression
- Support Vector Regression
- LASSO & Tomography Problem

Kernel Ridge Regression Example

Kernel Ridge Regression: Example f(x) = sin(x)

Example: Consider target function $f(x) = \sin(x)$ where data $y_i = f(x_i) + \eta_i$ where η_i is noise. Find $h \in \mathcal{H}_{\text{linear}}$.

Kernel Ridge Regression (KRR): Find minimizer of

$$\min_{\mathbf{w}} F(\mathbf{w}), \quad F(\mathbf{w}) = \lambda\|\mathbf{w}\|^2 + \sum_{i=1}^{m} (\mathbf{w}\cdot\Phi(x_i) - y_i)^2 \implies h(x) = \sum_{i=1}^{m} a_i\, K(x_i, x)$$

Solution: (Radial Basis Function Kernel (RBF), $K(x,y) = e^{-\gamma\|x-y\|^2}$
N = 100, gamma = 10, vary lambda)

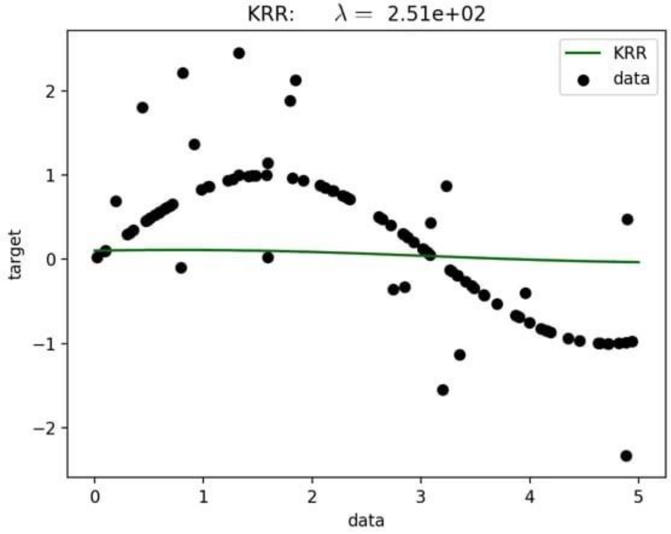
How does fit vary with different choices of the lambda?

How does fit vary with different choices of the RBF gamma width?

Hyperparameter choice is crucial to obtain good fits.

Hyperparameters are tuned through Cross-Validation (CV).

KRR typically use grid-search try to obtain best fit in CV.



$K(x,y) = e^{-\gamma\|x-y\|^2}$ $\gamma = 10$

Kernel Ridge Regression: Example f(x) = sin(x)

Example: Consider target function $f(x) = \sin(x)$ where data $y_i = f(x_i) + \eta_i$ where η_i is noise. Find $h \in \mathcal{H}_{\text{linear}}$.

Kernel Ridge Regression (KRR): Find minimizer of

$$\min_{\mathbf{w}} F(\mathbf{w}), \;\; F(\mathbf{w}) = \lambda\|\mathbf{w}\|^2 + \sum_{i=1}^{m} (\mathbf{w} \cdot \Phi(x_i) - y_i)^2 \implies h(x) = \sum_{i=1}^{m} a_i K(x_i, x)$$

Solution: (Radial Basis Function Kernel (RBF), $K(x,y) = e^{-\gamma\|x-}$
N = 100, gamma = 10, vary lambda)

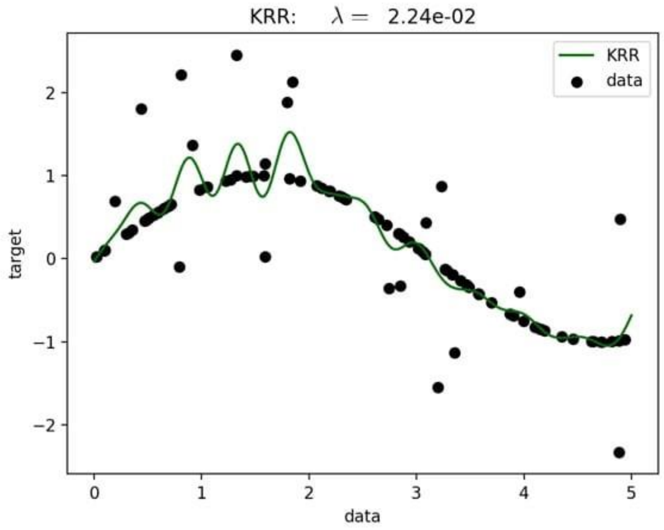
How does fit vary with different choices of the lambda?

How does fit vary with different choices of the RBF gamma width?

Hyperparameter choice is crucial to obtain good fits.

Hyperparameters are tuned through Cross-Validation (CV).

KRR typically use grid-search try to obtain best fit in CV.



$K(x,y) = e^{-\gamma\|x-y\|^2}$ $\gamma = 10$

Kernel Ridge Regression: Example f(x) = sin(x)

Example: Consider target function $f(x) = \sin(x)$ where data $y_i = f(x_i) + \eta_i$ where η_i is noise. Find $h \in \mathcal{H}_{\text{linear}}$.

Kernel Ridge Regression (KRR): Find minimizer of

$$\min_{\mathbf{w}} F(\mathbf{w}), \quad F(\mathbf{w}) = \lambda \|\mathbf{w}\|^2 + \sum_{i=1}^{m} (\mathbf{w} \cdot \Phi(x_i) - y_i)^2 \implies h(x) = \sum_{i=1}^{m} a_i \, K(x_i, x)$$

Solution: (Radial Basis Function Kernel (RBF), N = 100, lambda = 0.1, vary gamma)

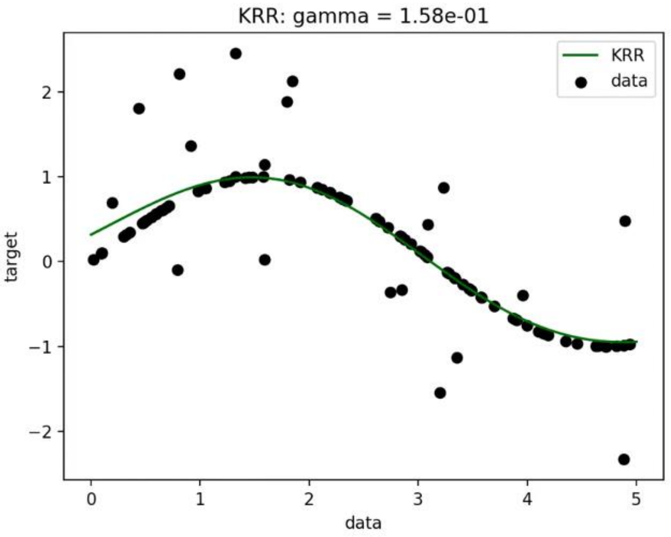
How does fit vary with different choices of the lambda?

How does fit vary with different choices of the RBF gamma width?

Hyperparameter choice is crucial to obtain good fits.

Hyperparameters are tuned through Cross-Validation (CV).

KRR typically use grid-search try to obtain best fit in CV.



$$K(x,y) = e^{-\gamma \|x-y\|^2}$$

Kernel Ridge Regression: Example f(x) = sin(x)

Example: Consider target function $f(x) = \sin(x)$ where data $y_i = f(x_i) + \eta_i$ where η_i is noise. Find $h \in \mathcal{H}_{\text{linear}}$.

Kernel Ridge Regression (KRR): Find minimizer of

$$\min_{\mathbf{w}} F(\mathbf{w}), \quad F(\mathbf{w}) = \lambda \|\mathbf{w}\|^2 + \sum_{i=1}^{m} (\mathbf{w} \cdot \Phi(x_i) - y_i)^2 \implies h(x) = \sum_{i=1}^{m} a_i K(x_i, x)$$

Solution: (Radial Basis Function Kernel (RBF), N = 100, lambda = 0.1, vary gamma)

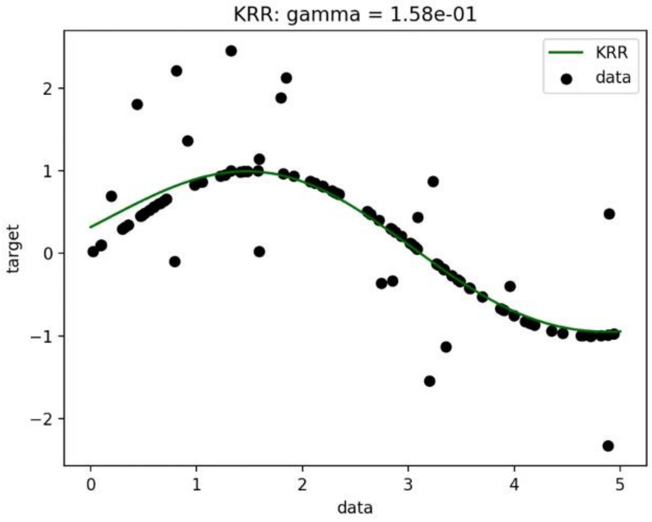
How does fit vary with different choices of the lambda?

How does fit vary with different choices of the RBF gamma width?

Hyperparameter choice is crucial to obtain good fits.

Hyperparameters are tuned through Cross-Validation (CV).

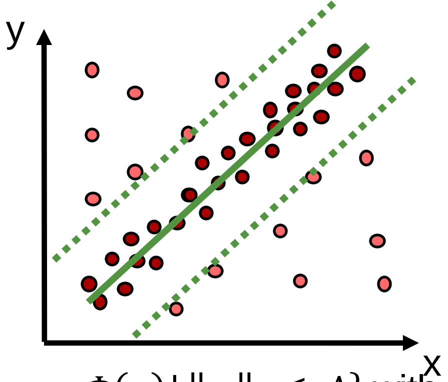
KRR typically use grid-search try to obtain best fit in CV.



$$K(x, y) = e^{-\gamma \|x-y\|^2}$$

Support Vector Regression

Support Vector Regression



Definition: For any $\varepsilon > 0$ we define the **support-limited loss function**

$$|y' - y|_\epsilon = \max(0, |y' - y| - \epsilon)$$

also referred to as the **ε-insensitive loss function**.

Theorem (support vector regression) Consider kernel regression using $\mathcal{H} = \{h(x) = w \cdot \Phi(x) \mid \|w\|_2 \leq \Lambda\}$ with $K(x,x) \leq r^2$ and $|f(x)| \leq \Lambda r$ then for any $\delta > 0$ we have with probability $1 - \delta$ that the following bounds hold uniformly for $h \in \mathcal{H}$

$$\mathop{\mathrm{E}}_{x\sim D}[|h(x) - f(x)|_\epsilon] \leq \mathop{\mathrm{E}}_{x\sim \widehat{D}}[|h(x) - f(x)|_\epsilon] + \frac{2r\Lambda}{\sqrt{m}}\left(1 + \sqrt{\frac{\log \frac{1}{\delta}}{2}}\right)$$

$$\mathop{\mathrm{E}}_{x\sim D}[|h(x) - f(x)|_\epsilon] \leq \mathop{\mathrm{E}}_{x\sim \widehat{D}}[|h(x) - f(x)|_\epsilon] + \frac{2r\Lambda}{\sqrt{m}}\left(\sqrt{\frac{\mathrm{Tr}[\mathbf{K}]}{mr^2}} + 3\sqrt{\frac{\log \frac{2}{\delta}}{2}}\right)$$

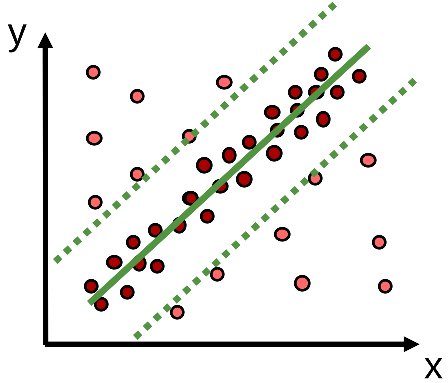
Remark: The bound takes on the form

$$R(h) \leq \widehat{R}(h) + \lambda\Lambda$$

Optimization Problem (Support Vector Regression (SVR))

$$\min_{\mathbf{w},b} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{m}\left|y_i - (\mathbf{w}\cdot\mathbf{\Phi}(x_i) + b)\right|_\epsilon$$

Support Vector Regression



Definition: For any $\varepsilon > 0$ we define the **support-limited loss function**

$$|y' - y|_\epsilon = \max(0, |y' - y| - \epsilon)$$

also referred to as the **ε-insensitive loss function**.

Optimization Problem (Support Vector Regression (SVR))

$$\min_{\mathbf{w},b} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{m} \left|y_i - (\mathbf{w}\cdot\Phi(x_i) + b)\right|_\epsilon$$

Interpretation:

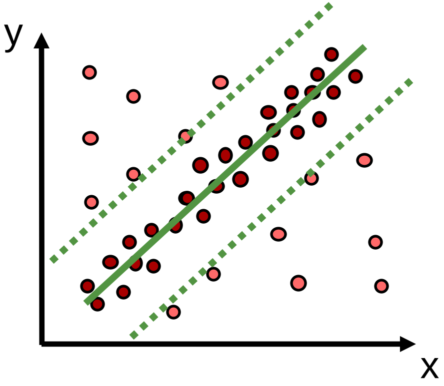
Incurs penalty only when loss exceeds ε. Data with $|y' - y|_\varepsilon > \varepsilon$ are called **Support Vectors**.

Promotes fitting a "tube" that covers large part of the data set.

Helps filter out within data high-frenquency noise, control weighting of outliers, account for density effects.

Shares similarities with Support Vector Machines (SVM).

Support Vector Regression



Equivalent Optimization Problem I:

$$\min_{\mathbf{w},b,\boldsymbol{\xi},\boldsymbol{\xi}'} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{m}(\xi_i + \xi_i')$$

subject $\xi_i \geq 0, \xi_i' \geq 0,\ \ (\mathbf{w}\cdot\boldsymbol{\Phi}(x_i)+b) - y_i \leq \epsilon + \xi_i$

$$y_i - (\mathbf{w}\cdot\boldsymbol{\Phi}(x_i)+b) \leq \epsilon + \xi_i'$$

Dual Formulation:

$$\max_{\boldsymbol{\alpha},\boldsymbol{\alpha}'} -\epsilon(\boldsymbol{\alpha}'+\boldsymbol{\alpha})^\top\mathbf{1} + (\boldsymbol{\alpha}'-\boldsymbol{\alpha})^\top\mathbf{y} - \frac{1}{2}(\boldsymbol{\alpha}'-\boldsymbol{\alpha})^\top\mathbf{K}(\boldsymbol{\alpha}'-\boldsymbol{\alpha})$$

subject to: $(\mathbf{0}\leq\boldsymbol{\alpha}\leq\mathbf{C})\wedge(\mathbf{0}\leq\boldsymbol{\alpha}'\leq\mathbf{C})\wedge((\boldsymbol{\alpha}'-\boldsymbol{\alpha})^\top\mathbf{1}=0)$.

Representation of solution

$$h(x) = \sum_{i=1}^{m}(\alpha_i' - \alpha_i)K(\mathbf{x}_i,\mathbf{x}) + b$$

where b can be determined from any x_j with $0 < \alpha_j < C$ or $0 < \alpha_j' < C$

$$b = -\sum_{i=1}^{m}(\alpha_i' - \alpha_i)K(x_i,x_j) + y_j + \epsilon$$

Complimentary Conditions (KKT)

$$\alpha_i\big((\mathbf{w}\cdot\boldsymbol{\Phi}(x_i)+b) - y_i - \epsilon - \xi_i\big) = 0$$

$$\alpha_i'\big((\mathbf{w}\cdot\boldsymbol{\Phi}(x_i)+b) - y_i + \epsilon + \xi_i'\big) = 0.$$

When we have $\alpha_i' \neq 0$ then

$$y_i - (\mathbf{w}\cdot\boldsymbol{\Phi}(x_i)+b) - \epsilon = \xi_i'.$$

which corresponds to x_i outside of ε-tube.

Similar condition holds for $\alpha_i' \neq 0$.

All x_i inside the ε-tube have $\alpha_i = 0$ and $\alpha_i' = 0$.

Support Vector Regression Example

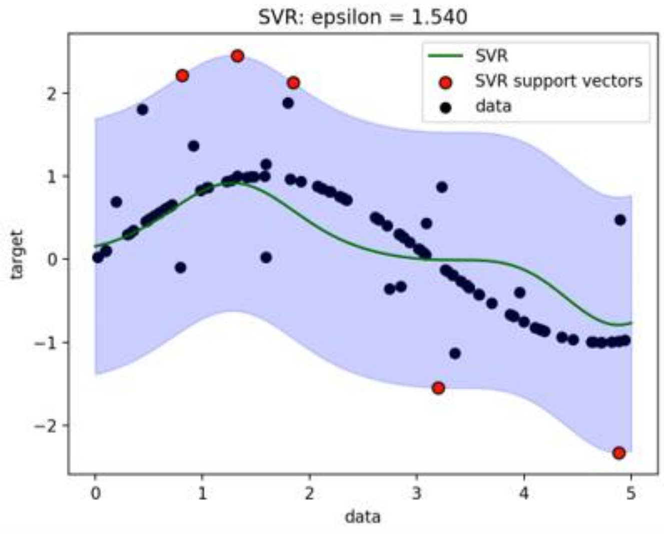
Support Vector Regression: Example f(x) = sin(x)

Example: Consider target function $f(x) = \sin(x)$ where data $y_i = f(x_i) + \eta_i$ where η_i is noise. Find $h \in \mathcal{H}_{\text{linear}}$.

Support Vector Regression (SVR): Find minimizer of

$$\min_{\mathbf{w},b} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{m} \left|y_i - (\mathbf{w}\cdot\boldsymbol{\Phi}(x_i) + b)\right|_\epsilon \quad \Longrightarrow \quad h(x) = \sum_{i=1}^{m} a_i K(x_i, x)$$

Solution: (Radial Basis Function Kernel (RBF), N = 100, epsilon = 0.1, gamma = 1)



How does fit vary with different choices of the ε-tube width?

How does fit vary with different choices of the RBF gamma width?

Hyperparameter choice is crucial to obtain good fits.

Hyperparameters are tuned through Cross-Validation (CV).

SVR typically use grid-search try to obtain best fit in CV.

$$K(x, y) = e^{-\gamma\|x-y\|^2}$$

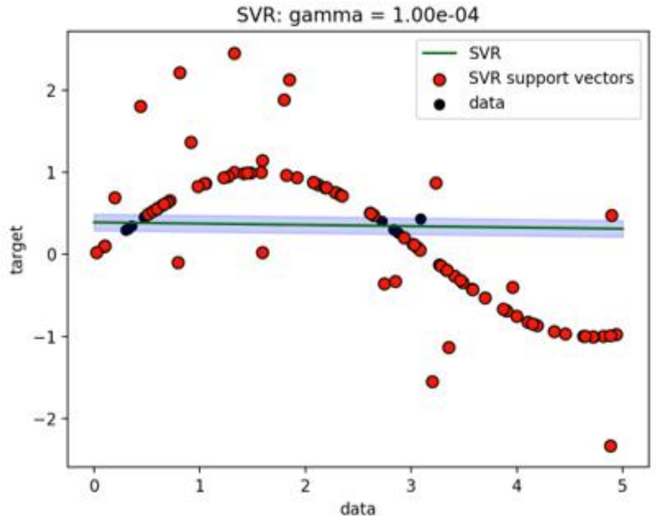
Support Vector Regression: Example f(x) = sin(x)

Example: Consider target function $f(x) = \sin(x)$ where data $y_i = f(x_i) + \eta_i$ where η_i is noise. Find $h \in \mathcal{H}_{\text{linear}}$.

Support Vector Regression (SVR): Find minimizer of

$$\min_{\mathbf{w},b} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{m} \left|y_i - (\mathbf{w}\cdot\mathbf{\Phi}(x_i) + b)\right|_\epsilon \implies h(x) = \sum_{i=1}^{m} a_i K(x_i, x)$$

Solution: (Radial Basis Function Kernel (RBF), N = 100, epsilon = 0.1, gamma = 1)



How does fit vary with different choices of the ε-tube width?

How does fit vary with different choices of the RBF gamma width?

Hyperparameter choice is crucial to obtain good fits.

Hyperparameters are tuned through Cross-Validation (CV).

SVR typically use grid-search try to obtain best fit in CV.

$$K(x, y) = e^{-\gamma\|x-y\|^2}$$

Support Vector Regression: Example f(x) = sin(x)

Example: Consider target function $f(x) = \sin(x)$ where data $y_i = f(x_i) + \eta_i$ where η_i is noise. Find $h \in \mathcal{H}_{\text{linear}}$.

Support Vector Regression (SVR): Find minimizer of

$$\min_{\mathbf{w},b} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{m} \left|y_i - (\mathbf{w}\cdot\mathbf{\Phi}(x_i) + b)\right|_\epsilon \quad \Longrightarrow \quad h(x) = \sum_{i=1}^{m} a_i\, K(x_i, x)$$

Solution: (Radial Basis Function Kernel (RBF), N = 100, epsilon = 0.1, gamma = 1)

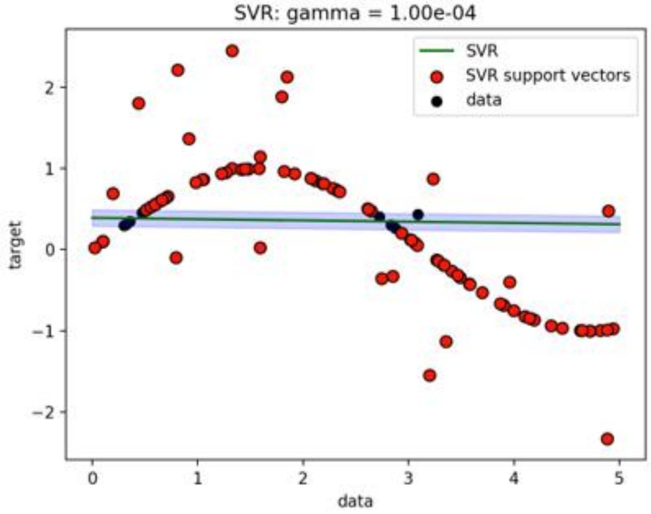
How does fit vary with different choices of the ε-tube width?

How does fit vary with different choices of the RBF gamma width?

Hyperparameter choice is crucial to obtain good fits.

Hyperparameters are tuned through Cross-Validation (CV).

SVR typically use grid-search try to obtain best fit in CV.



$$K(x,y) = e^{-\gamma\|x-y\|^2}$$

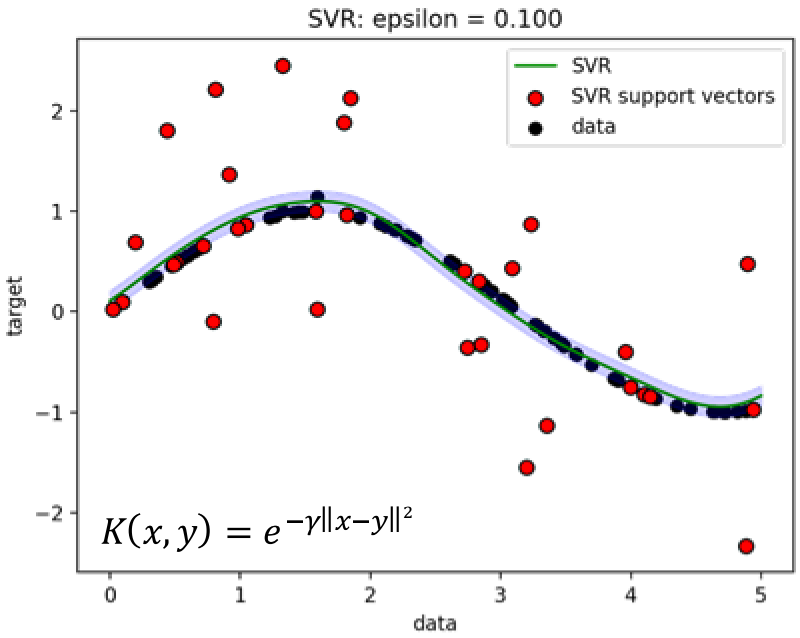
Support Vector Regression: Example f(x) = sin(x)

Example: Consider target function $f(x) = \sin(x)$ where data $y_i = f(x_i) + \eta_i$ where η_i is noise. Find $h \in \mathcal{H}_{\text{linear}}$.

Support Vector Regression (SVR): Find minimizer of

$$\min_{\mathbf{w},b} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{m} \left|y_i - (\mathbf{w}\cdot\mathbf{\Phi}(x_i) + b)\right|_\epsilon \implies h(x) = \sum_{i=1}^{m} a_i K(x_i, x)$$

Solution: (Radial Basis Function Kernel (RBF), N = 100, epsilon = 0.1, gamma = 1)



How does fit vary with different choices of the ε-tube width?

How does fit vary with different choices of the RBF gamma width?

Hyperparameter choice is crucial to obtain good fits.

Hyperparameters are tuned through Cross-Validation (CV).

SVR typically use grid-search try to obtain best fit in CV.

Comparison KRR and SVR

Comparison of KRR and SVR: Example f(x) = sin(x)

Example: Consider target function $f(x) = \sin(x)$ where data $y_i = f(x_i) + \eta_i$ where η_i is noise. Find $h \in \mathcal{H}_{\text{linear}}$.

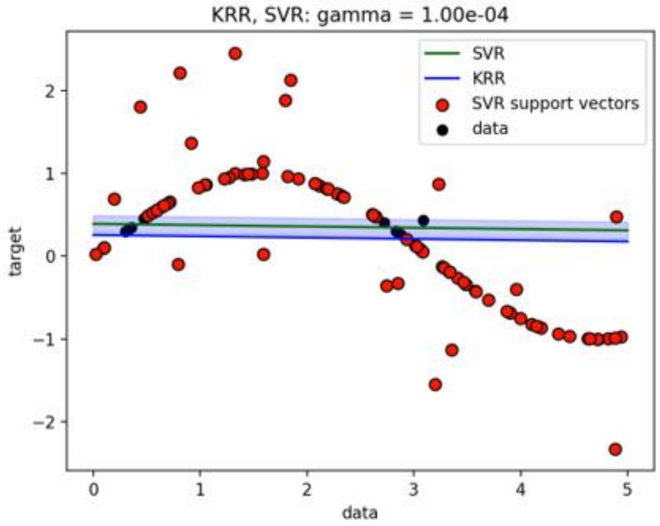
Kernel Ridge Regression (KRR): Find minimizer of

$$\min_{\mathbf{w}} F(\mathbf{w}), \quad F(\mathbf{w}) = \lambda\|\mathbf{w}\|^2 + \sum_{i=1}^{m} (\mathbf{w}\cdot\mathbf{\Phi}(x_i) - y_i)^2 \implies h(x) = \sum_{i=1}^{m} a_i\, K(x_i, x)$$

Support Vector Regression (SVR): Find minimizer of

$$\min_{\mathbf{w},b} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{m} \left|y_i - (\mathbf{w}\cdot\mathbf{\Phi}(x_i) + b)\right|_{\epsilon}$$

$$\implies h(x) = \sum_{i=1}^{m} (\alpha_i' - \alpha_i) K(\mathbf{x}_i, \mathbf{x}) + b$$



Solution: (Radial Basis Function Kernel (RBF), N = 100, epsilon = 0.1, gamma = 1)

Hyperparameter choice is crucial to obtain good fits.

Hyperparameters are tuned through Cross-Validation (CV).

SVR/KRR typically use grid-search try to obtain best fit in CV.

$$K(x, y) = e^{-\gamma\|x-y\|^2}$$

Comparison of KRR and SVR: Example f(x) = sin(x)

Example: Consider target function $f(x) = \sin(x)$ where data $y_i = f(x_i) + \eta_i$ where η_i is noise. Find $h \in \mathcal{H}_{\text{linear}}$.

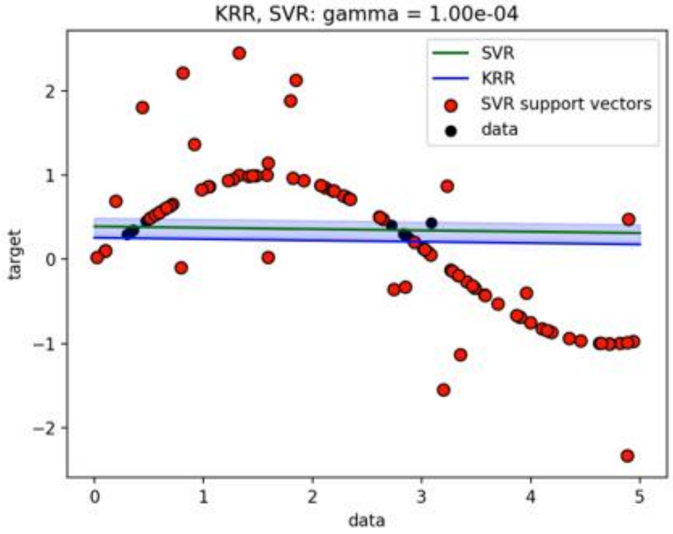
Kernel Ridge Regression (KRR): Find minimizer of

$$\min_{\mathbf{w}} F(\mathbf{w}), \quad F(\mathbf{w}) = \lambda\|\mathbf{w}\|^2 + \sum_{i=1}^{m} (\mathbf{w}\cdot\Phi(x_i) - y_i)^2 \implies h(x) = \sum_{i=1}^{m} a_i K(x_i, x)$$

Support Vector Regression (SVR): Find minimizer of

$$\min_{\mathbf{w},b} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{m} \left|y_i - (\mathbf{w}\cdot\Phi(x_i) + b)\right|_{\epsilon}$$

$$\implies h(x) = \sum_{i=1}^{m} (\alpha_i' - \alpha_i) K(\mathbf{x}_i, \mathbf{x}) + b$$



Solution: (Radial Basis Function Kernel (RBF), N = 100, epsilon = 0.1, gamma = 1)

Hyperparameter choice is crucial to obtain good fits.

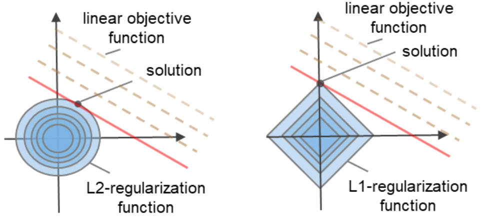
Hyperparameters are tuned through Cross-Validation (CV).

SVR/KRR typically use grid-search try to obtain best fit in CV.

$$K(x, y) = e^{-\gamma\|x-y\|^2}$$

LASSO Regression

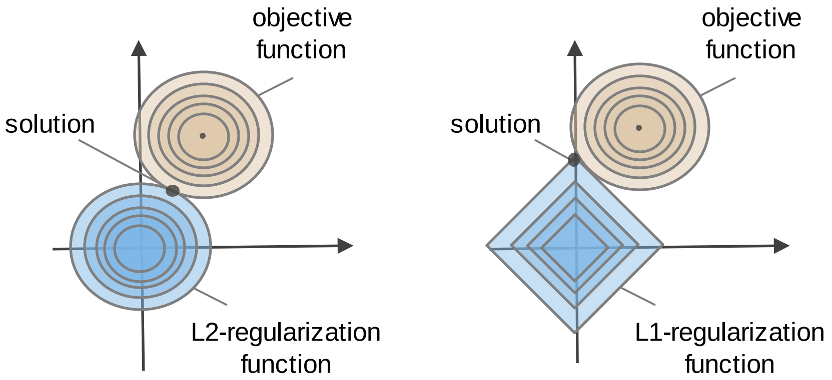
LASSO: Least Absolute Shrinkage and Selection Operator



L1-Norm Regularization: Tends to result in weights that are more sparse than L2-Regularization ($\min\|w\|_2$ vs $\min\|w\|_1$).

Theorem (LASSO regression) Consider kernel regression using $\mathcal{H} = \{h(x) = w \cdot x \mid \|w\|_1 \leq \Lambda_1\}$ with $\|x\| \leq r_\infty$ and $|f(x)| \leq \Lambda_1 r_\infty$ then for any $\delta > 0$ we have with probability $1-\delta$ that the following bounds hold uniformly for $h \in \mathcal{H}$

$$R(h) \leq \widehat{R}(h) + \frac{8r_\infty^2\Lambda_1^2}{\sqrt{m}}\left(\sqrt{\log(2N)} + \frac{1}{2}\sqrt{\frac{\log\frac{1}{\delta}}{2}}\right)$$



Optimization Problem:

$$\min_{\mathbf{w},b} F(\mathbf{w},b),\quad F(\mathbf{w},b) = \lambda\|\mathbf{w}\|_1 + \sum_{i=1}^{m}(\mathbf{w}\cdot\mathbf{x}_i + b - y_i)^2$$

Equivalent Problem I:

$$\min_{\mathbf{w},b} \sum_{i=1}^{m}(\mathbf{w}\cdot\mathbf{x}_i + b - y_i)^2 \quad \text{subject to: } \|\mathbf{w}\|_1 \leq \Lambda_1$$

Kernelization trick not available for L1 so would need to compute inner-products in new feature space.

High-dimensional regression problems especially useful to promote **sparsity**.

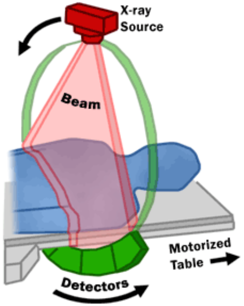
LASSO Regression: Computed Tomography (CT) & Compressed Sensing

Computed Tomography (CT) and Radon Transform:

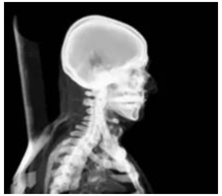
$$(x(z), y(z)) = \left((z\sin\alpha + s\cos\alpha), (-z\cos\alpha + s\sin\alpha)\right)$$

$$Rf(\alpha, s) = \int_{-\infty}^{\infty} f(x(z), y(z))\, dz$$

fda.gov



fda.gov



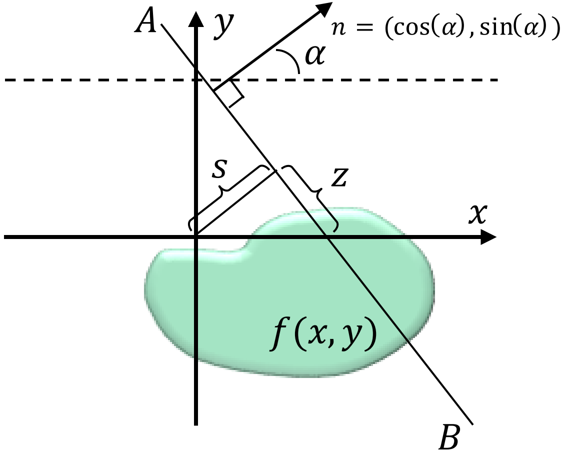
Arielinson

fda.gov

Inverse Problem: Reconstruct density f(x,y) based on projection data Rf.

Optimization Problem: Over the hypothesis class $\mathcal{H}$ of images $h(x_l,y_l)$ minimize error in matching projection data

$$min_{h\in\mathcal{H}}\, \lambda\,\|h\|_1 + \|Rf - Rh\|_2^2$$

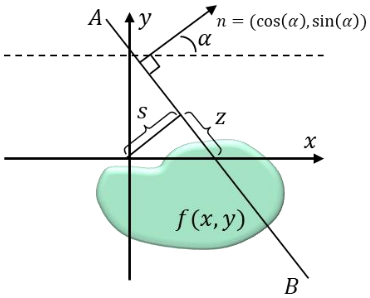


Sparse solutions desirable to reduce ghost artifacts.

Sparse density maps inherent in many cases
(scientific imaging, engineering characterization, industrial applications).

L1-regularization → sparse reconstructions → compressed sensing.

LASSO Regression: Computed Tomography (CT) & Compressed Sensing

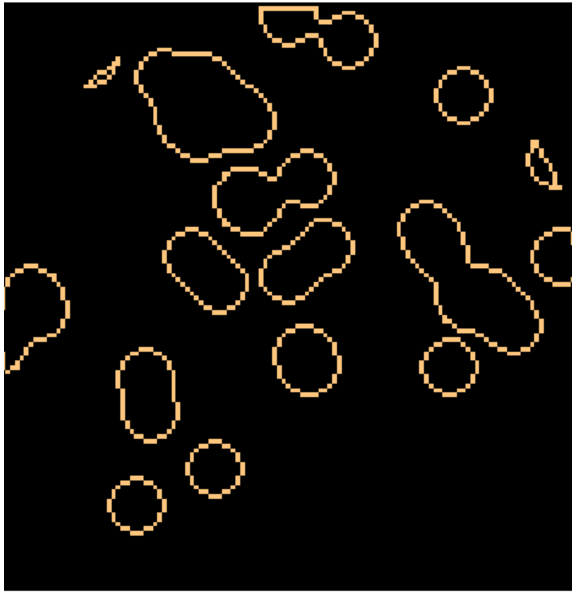


Example: Consider 2D density with data from 1D projections. (N = 36 angles).

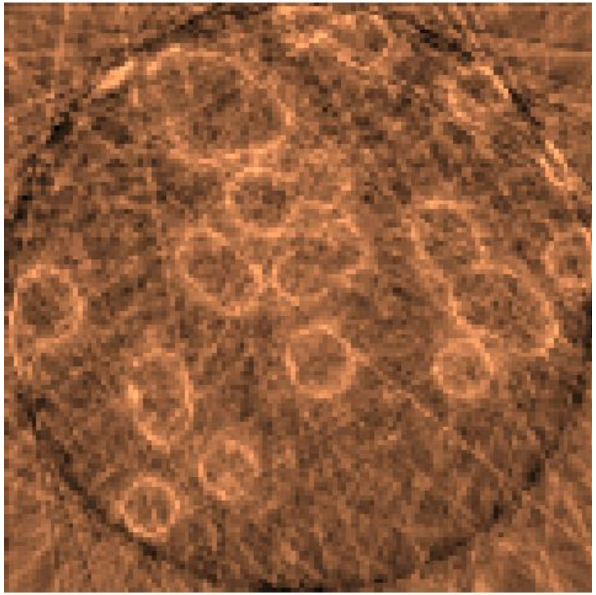
Density sparsely localized only on boundaries.

Task: Reconstruct the density map from the projection data. Compare KRR vs LASSO.

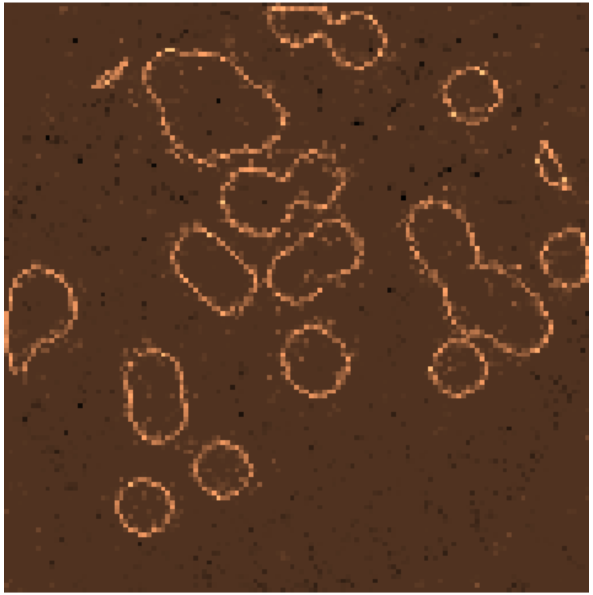
original image



L2 penalization $\lambda = 0.2$

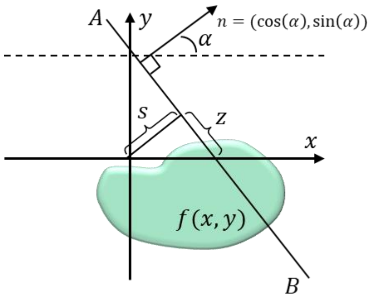


L1 penalization $\lambda = 0.00001$



Gouillart 2018

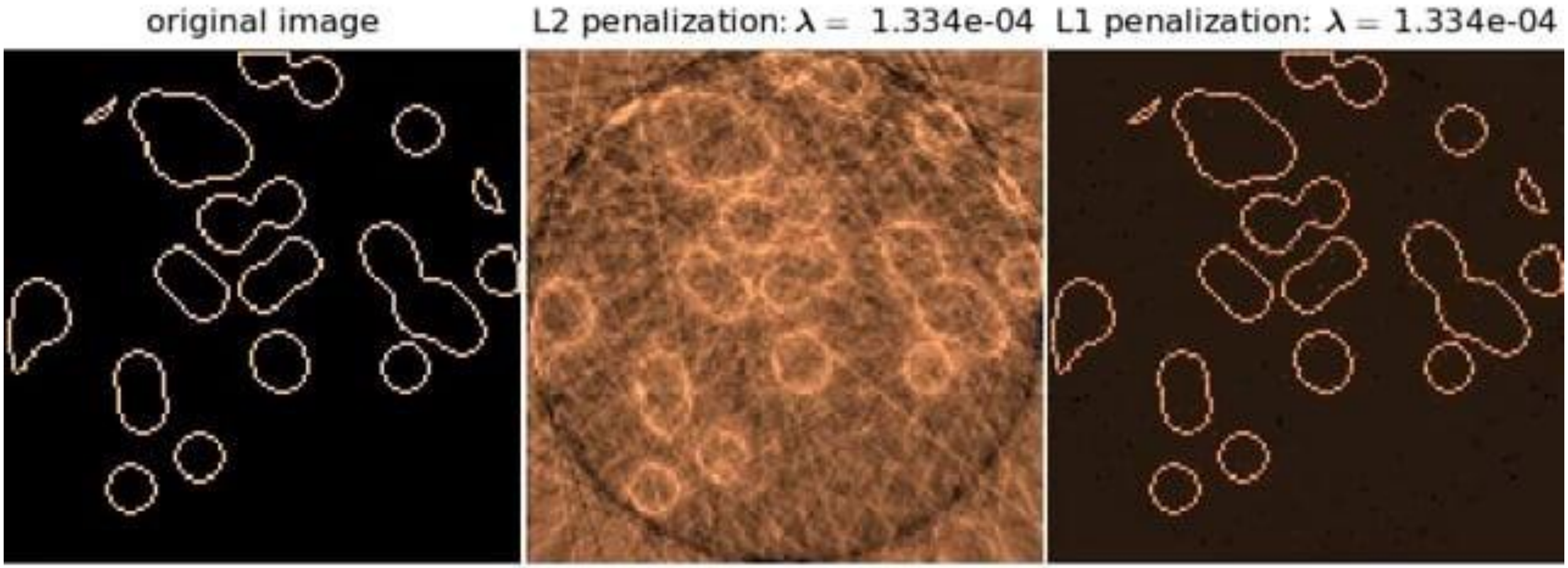
LASSO Regression: Computed Tomography (CT) & Compressed Sensing



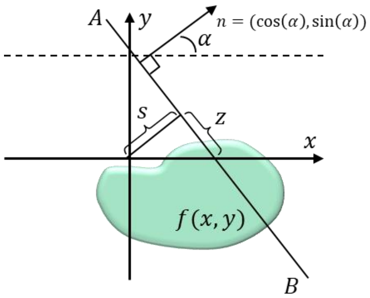
Example: Consider 2D density with data from 1D projections. (N = 36 angles).

Density sparsely localized only on boundaries.

Task: Reconstruct the density map from the projection data. Compare KRR vs LASSO.



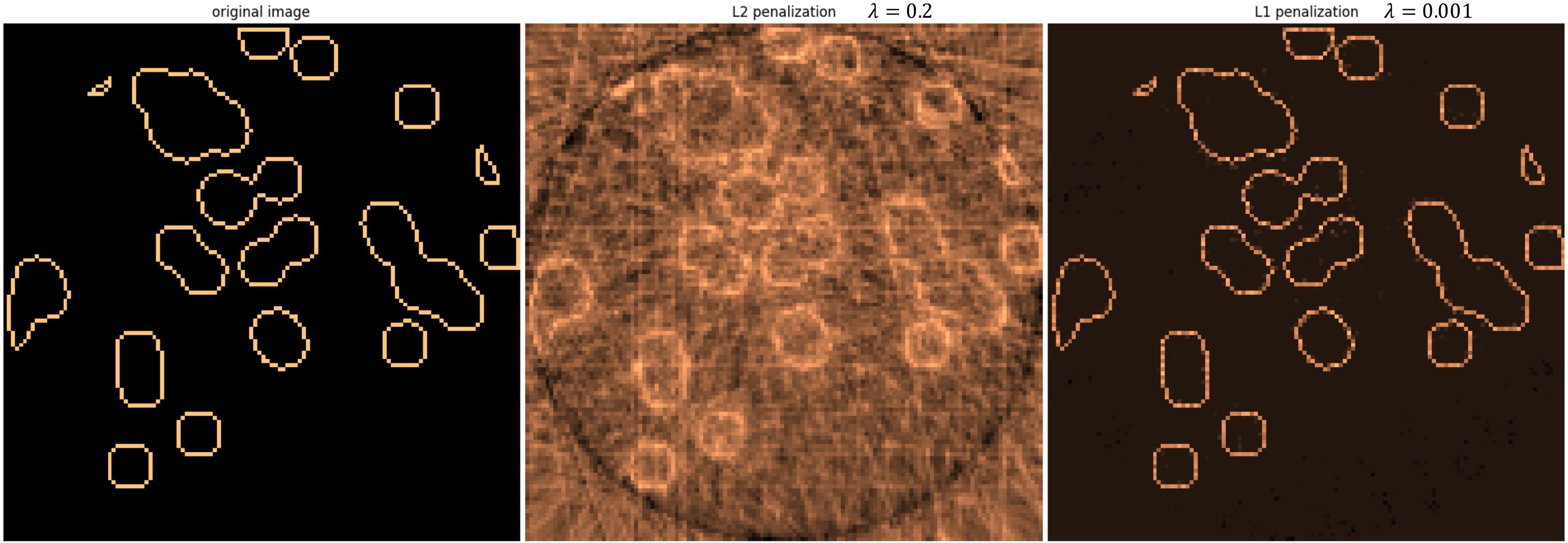
LASSO Regression: Computed Tomography (CT) & Compressed Sensing



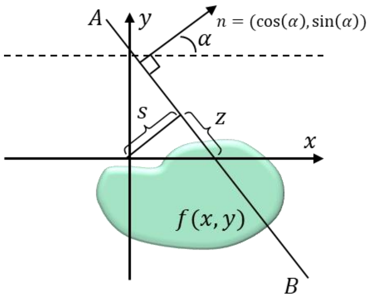
Example: Consider 2D density with data from 1D projections. (N = 36 angles).

Density sparsely localized only on boundaries.

Task: Reconstruct the density map from the projection data. Compare KRR vs LASSO.



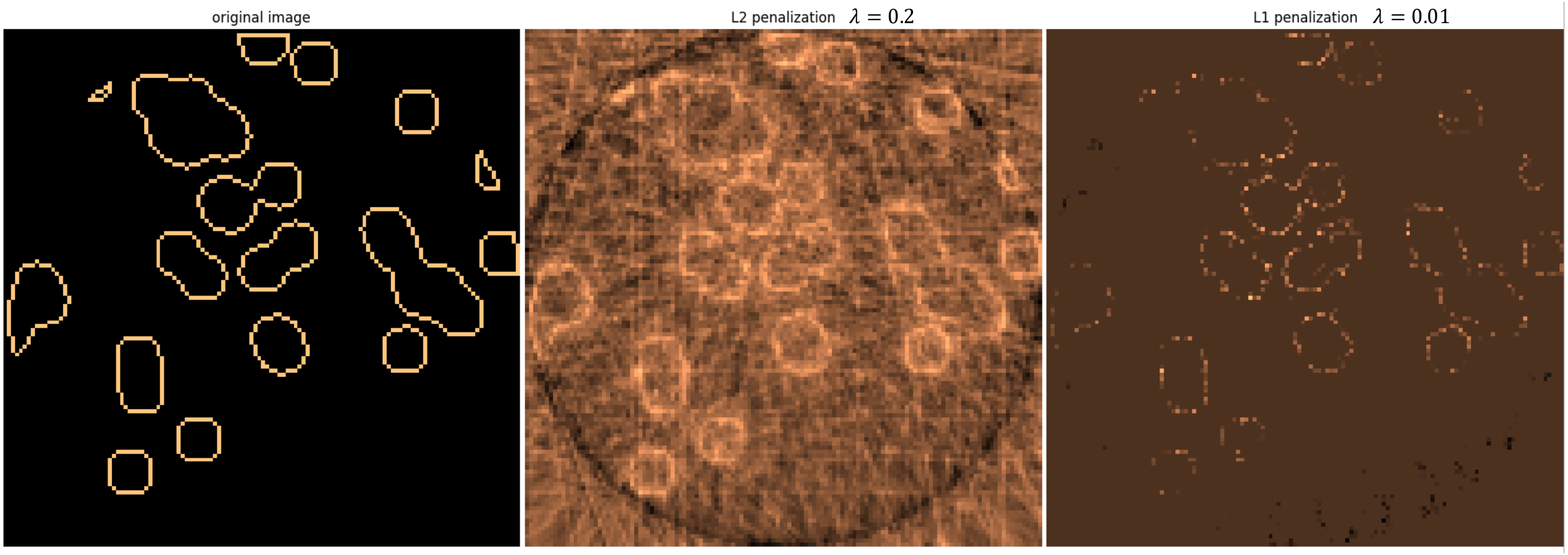
LASSO Regression: Computed Tomography (CT) & Compressed Sensing



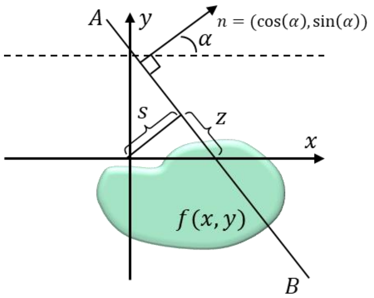
Example: Consider 2D density with data from 1D projections. (N = 36 angles).

Density sparsely localized only on boundaries.

Task: Reconstruct the density map from the projection data. Compare KRR vs LASSO.



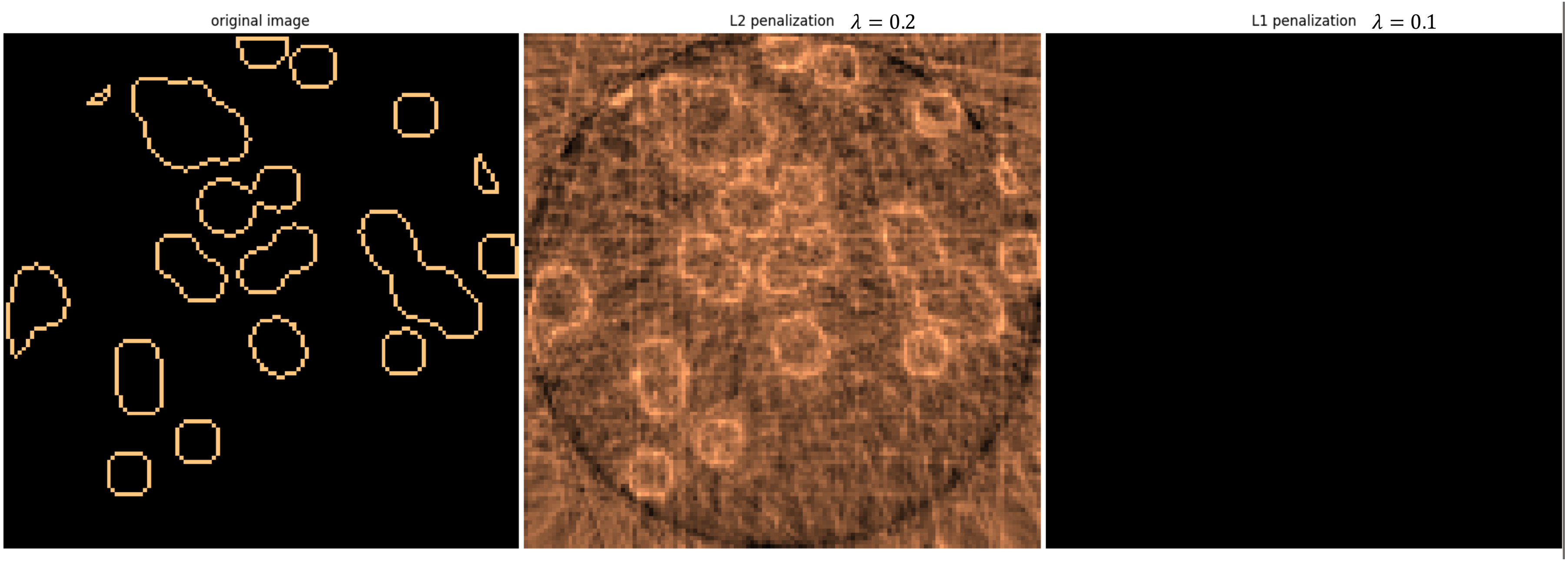
LASSO Regression: Computed Tomography (CT) & Compressed Sensing



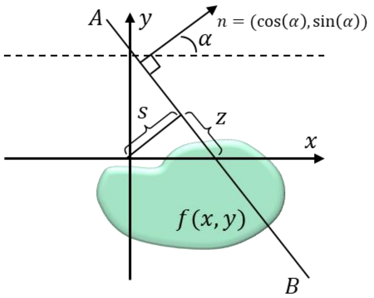
Example: Consider 2D density with data from 1D projections. (N = 36 angles).

Density sparsely localized only on boundaries.

Task: Reconstruct the density map from the projection data. Compare KRR vs LASSO.



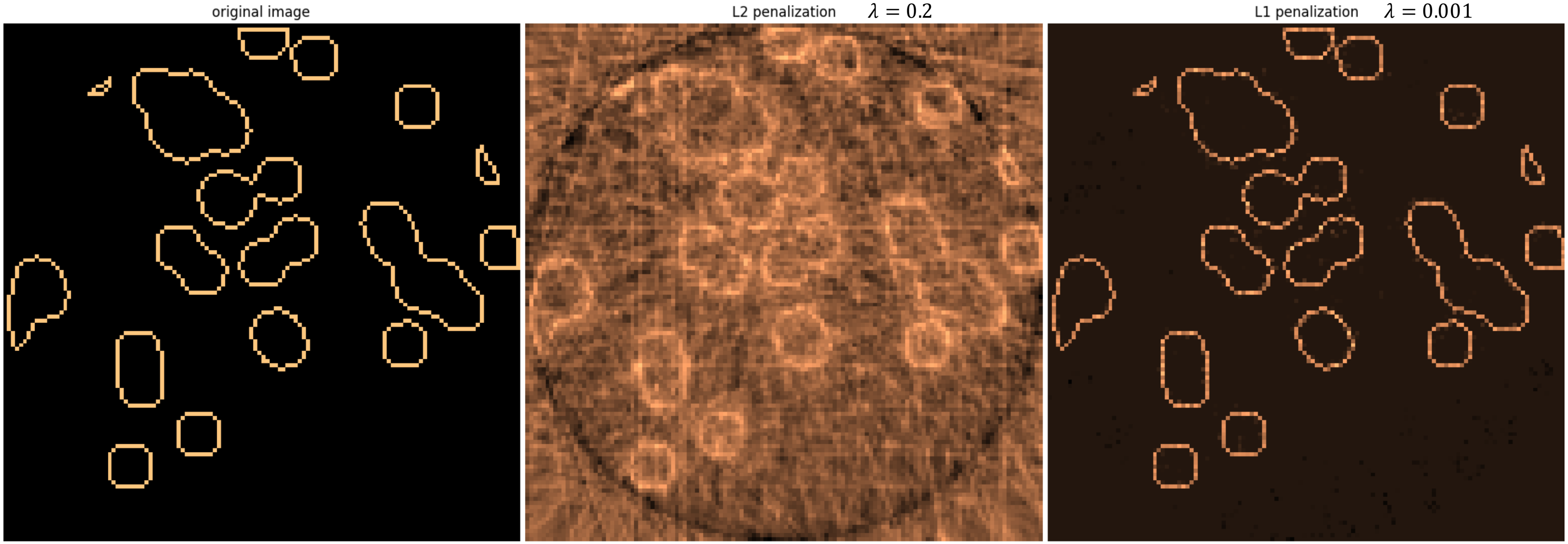
LASSO Regression: Computed Tomography (CT) & Compressed Sensing



Example: Consider 2D density with data from 1D projections. (N = 36 angles).

Density sparsely localized only on boundaries.

Task: Reconstruct the density map from the projection data. Compare KRR vs LASSO.



Curse of Dimensionality
and
Regression

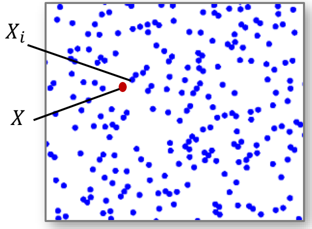
Curse of Dimensionality

Sampling on Unit Cube: Consider samples $X, X_1, X_2, \ldots, X_n \in [0,1]^d$ (d-dimensional hypercube).

Minimum Sample Distance: For n samples, denote the minimum distance between X and nearest sample X_i by

$$d_\infty(d,n) = \mathbb{E}\left[\min_{i\in[1,n]} \|X - X_i\|_\infty\right]$$

Samples



We can express in terms of probability as
$d_\infty(d,n) = \int_0^\infty \Pr\{\min_{i\in[1,n]} \|X - X_i\|_\infty > t\} dt = \int_0^\infty 1 - \Pr\{\min_{i\in[1,n]} \|X - X_i\|_\infty \leq t\} dt.$

The probability of being at most t apart in $\|\cdot\|_\infty$-norm is
$\Pr\{\min_{i\in[1,n]} \|X - X_i\|_\infty \leq t\} \leq n(2t)^d$.

Lower Bound on Distance: $$d_\infty(d,n) \geq \int_0^{1/2n^{1/d}} 1 - n(2t)^d dt = \frac{d}{2(d+1)} \frac{1}{n^{1/d}} \sim n^{-1/d}$$

samples:	$n = 10^2$	$n = 10^3$	$n = 10^4$	$n = 10^5$
$d_\infty(1,n)$	≥ 0.0025	≥ 0.00025	≥ 0.000025	≥ 0.0000025
$d_\infty(10,n)$	≥ 0.28	≥ 0.22	≥ 0.18	≥ 0.14
$d_\infty(20,n)$	≥ 0.37	≥ 0.34	≥ 0.30	≥ 0.26

Györfi 2002

Consequence: Shows for n samples, the minimum distance decreases very slowly for large d, $d_\infty \sim n^{-1/d}$.

Regression: Without using assumed structure, regression requires many samples to ensure accuracy.

Curse of Dimensionality and Generalization Bounds for Regression

Regression Task: From data samples $\mathcal{S} = \{(x_i, y_i)\}_{i=1}^n$ find model $f \in \mathcal{F}$ so that $y \sim f(x)$.

$$\hat{R}(f) = \frac{1}{n}\sum_{i=1}^n \ell(y_i, f(x_i)), \quad R(f) = \mathbb{E}_{(x,y)\sim\mathcal{D}}\left[\ell(y, f(x))\right], \quad \ell(y, f(x)) = \frac{1}{2}\left(y - f(x)\right)^2.$$

Approach: Regularized Loss Minimization (RLM), $\tilde{f} = \arg\min_{f\in\mathcal{F}}\left(\hat{R}(f) + \lambda\gamma(f)\right)$.

$$\gamma(f) = \inf_{\mu\in\mathcal{M}_f} |\mu|(\mathcal{V}), \quad \mathcal{M}_f = \{\mu \mid f(x) = \int_{\mathcal{V}} \phi_v(x)d\mu(v)\}, \quad \mathcal{V} \text{ compact}, \quad \mu \text{ Radon measure.}$$

$$|\mu|(\mathcal{V}) = \sup_{g\in\mathcal{G}} \int_{\mathcal{V}} g(v)d\mu(v), \quad \mathcal{G} = \{g \mid g \text{ continuous}, g(x) \in [-1, 1]\}.$$

related to: $\tilde{f} = \arg\min_{f\in\mathcal{F}^\delta} \hat{R}(f)$, $\mathcal{F}^\delta\{f \in \mathcal{F} \mid \gamma(f) \leq \delta\}$ (appropriate choice of δ).

Generalization Bound:

$$\underbrace{R(\hat{f}) - \inf_{f\in\mathcal{F}} R(f)}_{\text{generalization error}} \leq \underbrace{\left[\inf_{f\in\mathcal{F}^\delta} R(f) - \inf_{f\in\mathcal{F}} R(f)\right]}_{\text{approximation error}} + 2\underbrace{\inf_{f\in\mathcal{F}^\delta} |\hat{R}(f) - R(f)|}_{\text{estimation error}} + \underbrace{|\hat{R}(\hat{f}) - \inf_{f\in\mathcal{F}^\delta} \hat{R}(f)|}_{\text{optimization error}}.$$

Bach 2017

Curse of Dimensionality and Generalization Bounds for Regression

Regression Task: From data samples $\mathcal{S} = \{(x_i, y_i)\}_{i=1}^n$ find model $f \in \mathcal{F}$ so that $y \sim f(x)$.

$\tilde{f} = \arg\min_{f \in \mathcal{F}^\delta} \hat{R}(f), \quad \mathcal{F}^\delta \{f \in \mathcal{F} \mid \gamma(f) \leq \delta\}$.

Generalization Bound:

$$\underbrace{R(\hat{f}) - \inf_{f \in \mathcal{F}} R(f)}_{\text{generalization error}} \leq \underbrace{\left[\inf_{f \in \mathcal{F}^\delta} R(f) - \inf_{f \in \mathcal{F}} R(f)\right]}_{\text{approximation error}} + 2 \underbrace{\inf_{f \in \mathcal{F}^\delta} |\hat{R}(f) - R(f)|}_{\text{estimation error}} + \underbrace{|\hat{R}(\hat{f}) - \inf_{f \in \mathcal{F}^\delta} \hat{R}(f)|}_{\text{optimization error}}.$$

Bach 2017

Scaling in (n, d): When assuming the target function's form,

Case	Functional Form	L_2-risk generalization error
general	—	$n^{-1/(d+3)} \log(n)$
affine	$w^T x + b$	$d^{1/2} n^{-1/2}$
neural network (single layer)	$\sum_{j=1}^k \eta_j (w_j^T x + b_j)_+$	$k d^{1/2} n^{-1/2}$
projection pursuit	$\sum_{j=1}^k f_j(w_j^T x), \; w_j \in \mathbb{R}^d$	$k d^{1/2} n^{-1/4} \log(n)$
subspace projection	$\sum_{j=1}^k f_j(W_j^T x), \; W_j \in \mathbb{R}^{d \times s}$	$k d^{1/2} n^{-1/(s+3)} \log(n)$

Bach 2017

Summary: **General case** has exponential scaling in d! **However,** assumed structure → improves to polynomial in d! If target function approximated well by above form → even high dimensional d may be tractable.

In practice: Many functions in ML empirically appear well approximated by above (modest k, s).

Deep architectures (not case above) seem empirically to provide even better representations for many ML tasks.

Summary

Regression Summary

Task: Find function $h \in \mathcal{H}$ that models in data the relationship of y_i to x_i as $y_i \sim h(x_i)$.

Ordinary Least-Squares (OLS): Fits considering only least-squared deviations of y_i with $h(x_i)$. Can become overly sensitive to noise if features x_i^a and x_i^b are strongly correlated or co-linear.

Kernel Ridge Regression (KRR): Fits using L2-penalty in addition to least-squares loss. The penalty helps "shrink" weights yielding smaller values in directions where features x_i^a and x_i^b are strongly correlated or co-linear.

Support Vector Regression (SVR): Fits using ϵ-insensitive least-squares loss (ϵ-tube) and L2-penalty. The ϵ-tube helps filter localized variations without incurring loss and L2-penalty results in "shrinkage" as in KRR.

Least Absolute Shrinkage and Selection Operator (LASSO): Fits using L1-penalty in addition to least-squares loss. The penalty further helps "shrink" weights in many cases resulting in zero weight components giving a sparse representation (very helpful in high-dimensional regression).

Many other forms of regression: Elastic Net, LARS, Bayesian Regression, Neural-Network Methods.