Lecture 5: Bias-Variance Tradeoff; Regularization

TTIC 31020: Introduction to Machine Learning

Instructor: Kevin Gimpel

TTI-Chicago

October 15, 2019

Administrivia

- Final Exam: Thursday, Dec 12, 1:30-3:30pm
- Problem set 1 due Friday (Oct 18) 8:00pm
- Office Hours:
    - Mondays 3-4pm (me, room 531)
    - Tuesdays 1-2pm (TA)
    - Wednesdays 3:30-4:30pm (TA)
    - Thursdays 3:30-4:30pm (TA)
    - TA office hours held in 4th floor commons
- Recitations:
    - Tues 3:30-4pm or Thurs 1:20-1:50pm (same material for both)
    - This week: working through problems involving regression and likelihood
    - Next week: working through problems involving optimization and constraints; going over problem set 1

Review: generalized linear regression

- Define a **feature map** $\phi : \mathcal{X} \to \mathbb{R}^{m+1}$, train linear model in the feature space

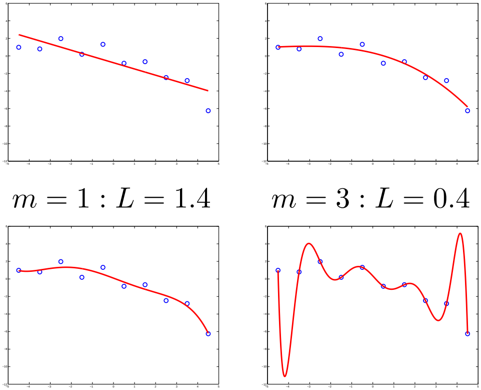
$$f(\mathbf{x}; \mathbf{w}) \;=\; \mathbf{w} \cdot \boldsymbol{\phi}(\mathbf{x})$$

- Can compute feature matrix to represent training data

$$\begin{bmatrix} \phi_0(\mathbf{x}_1) & \phi_1(\mathbf{x}_1) & \phi_2(\mathbf{x}_1) & \ldots & \phi_m(\mathbf{x}_1) \\ \phi_0(\mathbf{x}_2) & \phi_1(\mathbf{x}_2) & \phi_2(\mathbf{x}_2) & \ldots & \phi_m(\mathbf{x}_2) \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ \phi_0(\mathbf{x}_n) & \phi_1(\mathbf{x}_n) & \phi_2(\mathbf{x}_n) & \ldots & \phi_m(\mathbf{x}_n) \end{bmatrix}$$

polynomial

Review: overfitting and model complexity



$m = 1 : L = 1.4$

$m = 3 : L = 0.4$

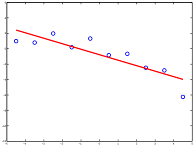
$m = 5 : L = 0.3$

$m = 10 : L = 0$

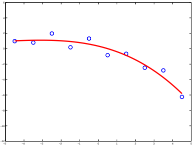
- More complex models (with more parameters) can fit the same number of examples "better"
- But may generalize less well – overfitting!

Review: detecting overfitting

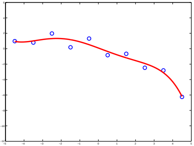
- Can detect overfitting by evaluating models on heldout val set
- Small data regime: cross-validation
  Partition data into k folds;
  For each fold, train on all but one part, test on that part;
  Average test loss over k parts



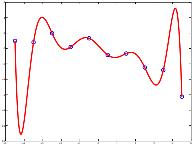
$m = 1 : L = 1.4$
$\hat{L}_{\text{cv}} = 2.6$



$m = 3 : L = 0.4$
$\hat{L}_{\text{cv}} = 1.3$



$m = 5 : L = 0.3$
$\hat{L}_{\text{cv}} = 2.7$



$m = 10 : L = 0$
$\hat{L}_{\text{cv}} = 4 \times 10^4$

simpler model has
lover Loss

Decomposition of error

- $\hat{\mathbf{w}}$: learned from training data $D = (\mathbf{X}, \mathbf{y})$
- $\mathbf{w}^*$: optimal linear regression parameters (generally unknown!),

$$\mathbf{w}^* = \underset{\mathbf{w}}{\operatorname{argmin}} \, \mathbb{E}_{p(\mathbf{x},y)} \left[(y - \mathbf{w} \cdot \mathbf{x})^2 \right]$$

- Let's work with $\mathbb{E}_{\mathbf{x},y,D} \left[(y - \hat{\mathbf{w}} \cdot \mathbf{x})^2 \right]$:

residuals uncorrelated with data x: but may be dependent

$$\begin{aligned}
\mathbb{E}_{\mathbf{x},y,D} \left[(y - \hat{\mathbf{w}} \cdot \mathbf{x})^2 \right] \; = \; & \mathbb{E}_{\mathbf{x},y,D} \left[(y - \mathbf{w}^* \cdot \mathbf{x})^2 \right] \\
& + \; 2\mathbb{E}_{\mathbf{x},y,D} \left[(y - \mathbf{w}^* \cdot \mathbf{x}) \left(\mathbf{w}^* \cdot \mathbf{x} \; - \; \hat{\mathbf{w}} \cdot \mathbf{x} \right) \right] = 0 \\
& + \; \mathbb{E}_{\mathbf{x},y,D} \left[\left(\mathbf{w}^* \cdot \mathbf{x} \; - \; \hat{\mathbf{w}} \cdot \mathbf{x} \right)^2 \right]
\end{aligned}$$

- Second term vanishes since prediction errors $y - \mathbf{w}^* \cdot \mathbf{x}$ are uncorrelated with *any* linear function of $\mathbf{x}$ including $\mathbf{w}^* \cdot \mathbf{x} \; - \; \hat{\mathbf{w}} \cdot \mathbf{x}$

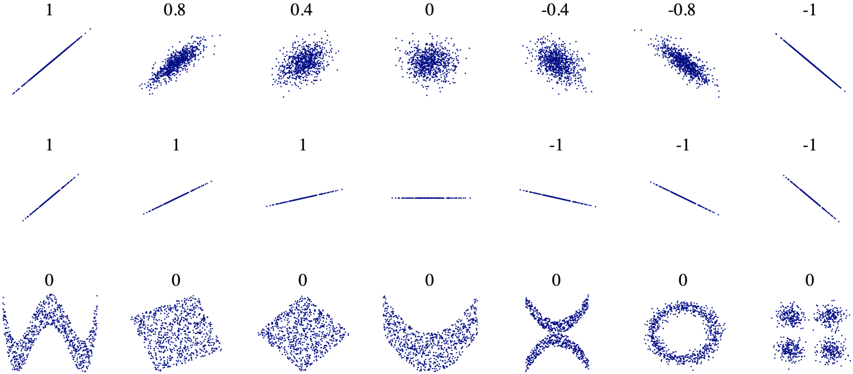
Decomposition of error

$$\mathbb{E}_{\mathbf{x},y,D}\left[(y-\hat{\mathbf{w}}\cdot\mathbf{x})^2\right] = \underbrace{\mathbb{E}_{\mathbf{x},y}\left[(y-\mathbf{w}^*\cdot\mathbf{x})^2\right]}_{\textbf{approximation}} + \underbrace{\mathbb{E}_{\mathbf{x},y,D}\left[(\mathbf{w}^*\cdot\mathbf{x}-\hat{\mathbf{w}}\cdot\mathbf{x})^2\right]}_{\textbf{estimation}}$$

- **Approximation error** $\mathbb{E}_{\mathbf{x},y}\left[(y-\mathbf{w}^*\cdot\mathbf{x})^2\right]$ measures inherent limitations of the chosen hypothesis class (linear function). This error will remain even with infinite training data.
- **Estimation error** $\mathbb{E}_{\mathbf{x},y,D}\left[(\mathbf{w}^*\cdot\mathbf{x}\,-\,\hat{\mathbf{w}}\cdot\mathbf{x})^2\right]$ measures closeness to optimal $\mathbf{w}^*$ of $\hat{\mathbf{w}}$ estimated from (finite) training data.
- Note: since training data $\mathbf{X}, \mathbf{y}$ are random variables drawn from $p(\mathbf{x}, y)$, the estimated $\hat{\mathbf{w}}$ is a random variable as well.

Pearson correlations of x, y points

Uncorrelated variables might not be independent (may show complex structural dependencies):



By DenisBoigelot, original uploader was Imagecreator, CC0, `https://commons.wikimedia.org/w/index.php?curid=15165296`

A bit of estimation theory

- An **estimator** $\widehat{\theta}$ of a parameter θ is a function that for data $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ produces estimate (estimated value) $\widehat{\theta}$
- Examples:
  - maximum likelihood (ML) estimator for a Gaussian mean, given X, produces an estimate (vector) $\widehat{\boldsymbol{\mu}}$
  - ML estimator for linear regression parameters $\mathbf{w}$ under Gaussian noise model
- The estimate $\hat{\theta}$ is a random variable since it is based on a randomly drawn dataset X
- We can talk about $\mathbb{E}_X\left[\hat{\theta}\right]$ and $\mathrm{var}(\hat{\theta})$

  (When θ is a vector, we have $\mathrm{Cov}(\hat{\theta})$.)
  - *Analysis done assuming that the data* **is** *distributed according to* $p(\mathbf{x}; \theta)$ *where* θ *is the true parameter value!*

assume we know the true parameter value/ form of the data generating function

Bias of an estimator

- The **bias** of an estimator $\hat{\theta}$ is defined as

  $$\text{bias}(\hat{\theta}) \triangleq \mathbb{E}_X\Big[\hat{\theta} - \theta\Big]$$

  i.e., the expected deviation of the estimate from the true value (taken over all possible sets of n examples)
- An **unbiased** estimator therefore satisfies $\mathbb{E}_X\Big[\hat{\theta}\Big] = \theta$
- Example: ML estimators of 1D Gaussian parameters
  $\hat{\mu}_{ML} = \frac{1}{n}\sum_i x_i \qquad \widehat{\sigma^2}_{ML} = \frac{1}{n}\sum_i (x_i - \hat{\mu})^2$
- Turns out $\hat{\mu}$ is unbiased; however, $\widehat{\sigma^2}_{ML}$ *underestimates* the variance in the data!

  $$\mathbb{E}\left[\widehat{\sigma^2}_{ML}\right] \;=\; \frac{n-1}{n}\sigma^2$$

Consistency of an estimator

- With enough data, bias *may* not be so much of a problem.
- Consider an infinite sequence $\mathbf{x}_1, \ldots$ and define $\hat{\theta}_n$ an estimate obtained on $\mathbf{x}_1, \ldots, \mathbf{x}_n$.
- An estimator $\hat{\theta}$ is **consistent** if

$$\lim_{n \to \infty} \hat{\theta}_n \; = \; \theta$$

  Note: this limit is **in probability**, i.e., the estimator converges in probability to the true value.

- The ML estimator is consistent (under certain conditions)
- So, $\widehat{\sigma^2}_{ML} = \frac{1}{n} \sum_i (x_i - \mu_{ML})^2$, even though biased, is a consistent estimator of σ^2

Bias-variance decomposition

- Consider squared loss: $\left(\hat{\theta} - \theta\right)^2$
- Denote $\bar{\theta} = \mathbb{E}_X\left[\hat{\theta}\right] = \mathbb{E}\left[\hat{\theta}\right]$. Expectations on this slide are taken with respect to distribution over samples X, so we'll drop the "X" subscript. Then, the expected error:

$$\begin{aligned}\mathbb{E}\left[(\hat{\theta} - \theta)^2\right] &= \mathbb{E}\left[(\hat{\theta} - \bar{\theta} + \bar{\theta} - \theta)^2\right] \\ &= \mathbb{E}\left[(\hat{\theta} - \bar{\theta})^2\right] + 2(\bar{\theta} - \theta)\underbrace{\mathbb{E}\left[\hat{\theta} - \bar{\theta}\right]}_{=0} + \mathbb{E}\left[(\bar{\theta} - \theta)^2\right] \\ &= \mathbb{E}\left[(\hat{\theta} - \bar{\theta})^2\right] + (\bar{\theta} - \theta)^2 \\ &= \mathrm{var}(\hat{\theta}) + \mathrm{bias}^2(\hat{\theta})\end{aligned}$$

theta bar = E(theta head)

- Recall expected squared loss decomposition:
  - bias^2 term ⇔ approximation error
  - variance ⇔ estimation error due to finite data

Bias-variance tradeoff

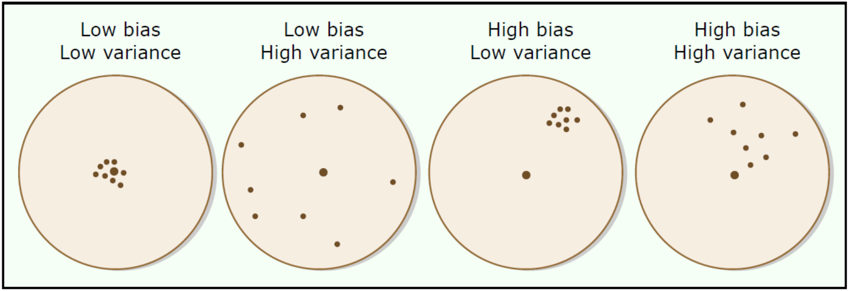


Image by MIT OpenCourseWare.

- Ideally, want to minimize bias and variance;
  turns out there is a tradeoff: lower bias generally corresponds to higher variance (cf. Cramer-Rao inequality)
- The objective in ML: find the "sweet spot"
- Major component in this struggle: model complexity

Estimation and regression

- The true model: $y = F(\mathbf{x}) + \nu$, zero-mean additive noise ν
- We approximate F by $\hat{f}_D \in \mathcal{F}$, with $\hat{f}_D$ estimated from data D
- We have:

$$\begin{aligned}
F &= \operatorname*{argmin}_{f} \mathbb{E}_{p(\mathbf{x},y)}\left[(y - f(\mathbf{x}))^2\right] && \text{best predictor} \\
f^* &= \operatorname*{argmin}_{f \in \mathcal{F}} \mathbb{E}_{p(\mathbf{x},y)}\left[(y - f(\mathbf{x}))^2\right] && \text{best predictor in } \mathcal{F} \\
\hat{f}_D &= \operatorname*{argmin}_{f \in \mathcal{F}} \sum_{(\mathbf{x}_i, y_i) \in D} (y_i - f(\mathbf{x}_i))^2 && \text{predictor learned from } D \\
\bar{f} &= \mathbb{E}_D\left[\hat{f}_D\right] && \text{average predictor} \\
\bar{f}(\mathbf{x}_0) &= \mathbb{E}_D\left[\hat{f}_D(\mathbf{x}_0)\right] && \text{applying } \bar{f} \text{ to } \mathbf{x}_0
\end{aligned}$$

Bias-variance in regression

- For a single $\mathbf{x}_0$ with true output $\mathbf{y}_0$ and prediction $\hat{f}(\mathbf{x}_0)$, $\mathbb{E}_D\left[(y_0 - \hat{f}(\mathbf{x}_0))^2\right]$ can be decomposed (shown without proof):

$$\mathbb{E}_D\left[(y_0 - \hat{f}(\mathbf{x}_0))^2\right] \;=\; (y_0 - \bar{f}(\mathbf{x}_0))^2 \;+\; \underbrace{\mathbb{E}_D\left[(\hat{f}(\mathbf{x}_0) - \bar{f}(\mathbf{x}_0))^2\right]}_{variance}$$

- The first term can be further decomposed (shown without proof):

$$(y_0 - \bar{f}(\mathbf{x}_0))^2 = \underbrace{(y_0 - F(\mathbf{x}_0))^2}_{\text{noise}} + \underbrace{(F(\mathbf{x}_0) - \bar{f}(\mathbf{x}_0))^2}_{\mathrm{bias}^2}$$

- Can integrate all of this over $\mathbf{x}_0, y_0$ to get the *expected* bias and variance.

Bias-variance tradeoff

- We have

$$\mathbb{E}_D\left[(y-\hat{f}(\mathbf{x}))^2\right] = \underbrace{\overset{\text{True}}{(y-F(\mathbf{x}))^2}}_{\text{noise}} + \underbrace{\overset{\text{average predictor}}{(F(\mathbf{x})-\bar{f}(\mathbf{x}))^2}}_{\text{bias}^2} + \underbrace{\overset{\text{predictor learned}}{\mathbb{E}_D\left[(\hat{f}(\mathbf{x})-\bar{f}(\mathbf{x}))^2\right]}}_{variance}$$

- The noise term is *irreducible* (independent of data/model) will be there even if we know $p\,(y\,|\,\mathbf{x})$
- The bias^2 term is due to difference between f and F; can address by changing f
- The variance is due to finite data; can address by getting more data
- Ideally, want to minimize bias and variance; can we drive both to zero?

Bias-variance tradeoff: theory

- **Cramer-Rao inequality**: for an unbiased estimator $\hat{\theta}_n$ and true parameter value θ,

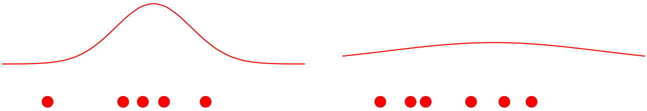
produce more information, var will be smaller

$$\mathrm{var}(\hat{\theta}_n) \;\geq\; \frac{1}{\mathbb{E}_{\mathbf{X}}\left[\left(\frac{\partial}{\partial\theta}\log p(\mathbf{X};\theta)\right)^2\right]}$$

have to assume we have model of X

- The **Fisher information** $\mathcal{I}(\theta) = \mathbb{E}_{\mathbf{X}}\left[\left(\frac{\partial}{\partial\theta}\log p(\mathbf{X};\theta)\right)^2\right]$ is related to the shape of $p(\mathbf{x};\theta)$. Intuitively, it measures the amount of information the data $\mathbf{X}$ provides about a parameter with true value θ

$\mathcal{N}\left(x;\, \mu, \sigma^2\right)$
Known σ^2
Estimating μ



large $\mathcal{I}(\mu)$ small $\mathcal{I}(\mu)$

Model complexity - theory

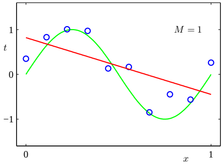
- Remember: we are talking about complexity of *model class*, not individual model!
- Basic intuition: model complexity = number of models in the class (assuming finite model class!)
- For model f, can measure empirical loss $L(f) = \frac{1}{n}\sum_i \ell(f; \mathbf{x}_i, y_i)$ on n examples
- Interested in risk $R(f) = \mathbb{E}_{\mathbf{x},y}\left[\ell(f; \mathbf{x}, y)\right]$
- Learning theory provides **generalization bounds** of the form

$$\Pr\left(\max_{f \in \mathcal{F}} |L(f) - R(f)| > \epsilon\right) \leq 2|\mathcal{F}|e^{-2n\epsilon^2}$$

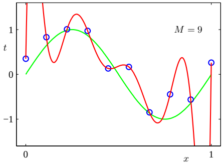
- If $\mathcal{F}$ is infinite: $|\mathcal{F}|$ is replaced with another measure of complexity, e.g., VC-dimension
- Caution: bounds often very loose, hard to compute for interesting $\mathcal{F}$

Model complexity - intuition

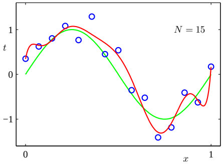
- Intuitively, the complexity of the model can be measured by the number of "degrees of freedom" (independent parameters).
- The more complex the model, the more data needed to fit
  $\Rightarrow$ For a given number of points, a more complex model more likely to overfit.
- Example from Bishop: m-degree polynomial fit to n points



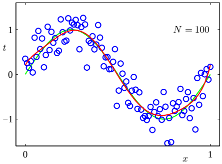
$m = 1$
$n = 10$



$m = 9$
$n = 10$



$m = 9$
$n = 15$



$m = 9$
$n = 100$

Penalizing model complexity

- Idea 1: restrict model complexity based on amount of data
  - Rule of thumb: approx. 10 examples per parameter
- Idea 2: directly penalize by the number of parameters
  Akaike information criterion (AIC): maximize

$$\log p\left(\mathbf{y}, \mathbf{X};\ \widehat{\mathbf{w}}\right) - \#\text{params}$$

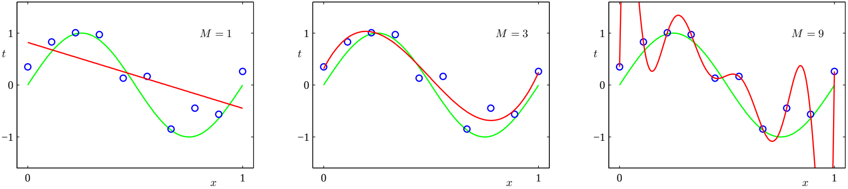
- But: Definition of model complexity as a number of parameters is a bit too simplistic. Consider feature vector

$$\boldsymbol{\phi}(x) = \begin{bmatrix} 1 & x & -2x & 2x & x^2 & \frac{1}{2}x^2 \end{bmatrix}$$

  Does linear regression $\boldsymbol{\phi}(x) \to y$ really have 6 parameters?
- Idea: look at the behavior of the values of $\mathbf{w}^*$

Linear regression complexity

- Example: polynomial regression [from Bishop, Ch. 1]



- Value of the optimal (ML) regression coefficients:

	$m = 0$	$m = 1$	$m = 3$	$m = 9$
w_0^*	0.19	0.82	0.31	0.35
w_1^*		-1.27	7.99	232.37
w_2^*			-25.43	-5321.83
w_3^*			17.37	48568.31
w_4^*				-231639.30
w_5^*				640042.26
w_6^*				-1061800.52
w_7^*				1042400.18
w_8^*				-557682.99
w_9^*				125201.43

The weight are too large. cannot generalized well

Description length

- Intuition: should penalize not the parameters, but the number of bits required to encode the parameters
- We can limit the effective number of degrees of freedom by restricting the values of the parameters
- Note: this argument assumes finite precision (e.g., in a computer)
- Then we have penalized log-likelihood:

$$\mathbf{w}^* = \operatorname*{argmax}_{\mathbf{w}} \left\{ \frac{1}{n} \sum_{i=1}^{n} \log p(\mathsf{data}_i;\, \mathbf{w}) - \mathsf{penalty}(\mathbf{w}) \right\}$$

- Equivalently, penalized ERM:

$$\mathbf{w}^* = \operatorname*{argmin}_{\mathbf{w}} \left\{ -\frac{1}{n} \sum_{i=1}^{n} \log p(\mathsf{data}_i;\, \mathbf{w}) + \mathsf{penalty}(\mathbf{w}) \right\}$$

Shrinkage methods

- Shrinkage methods impose penalty on the size of $\mathbf{w}$ The value
- Can measure "size" in different ways. Let us start with L_2 norm:

$$\mathbf{w}^*_{\text{ridge}} = \underset{\mathbf{w}}{\operatorname{argmin}} \left\{ -\sum_{i=1}^{n} \log p(\text{data}_i;\, \mathbf{w}) \,+\, \lambda \|\mathbf{w}\|^2 \right\}$$

in regression "data$_i$" $= y_i | \mathbf{x}_i$

- This is **ridge regression**; λ is the **regularization** parameter
- Does it matter that log-likelihood is not averaged?

can be differentiated directly, so this is more widely used

$$\min_{\mathbf{w}} \left\{ -\sum_{i=1}^{n} \log p(\text{data}_i; \mathbf{w}) + \lambda \|\mathbf{w}\|^2 \right\}$$

vs. $$\min_{\mathbf{w}} \left\{ -\frac{1}{n} \sum_{i=1}^{n} \log p(\text{data}_i; \mathbf{w}) + \lambda \|\mathbf{w}\|^2 \right\}$$

Ridge regression

$$\mathbf{w}^*_{\text{ridge}} = \underset{\mathbf{w}}{\operatorname{argmin}} \left\{ \sum_{i=1}^{n} (y_i - \mathbf{w} \cdot \mathbf{x}_i)^2 + \lambda \sum_{j=1}^{m} w_j^2 \right\}$$

problem:
setting lambda

- Recall: $\mathbf{w} = [w_0, w_1, \ldots, w_m]$
- Usually do not include w_0 in regularization (why?)
- Closed form solution:

XTX

$$\widehat{\mathbf{w}}^*_{\text{ridge}} = \left(\lambda \mathbf{I} + \mathbf{X}^\top \mathbf{X}\right)^{-1} \mathbf{X}^\top \mathbf{y}.$$

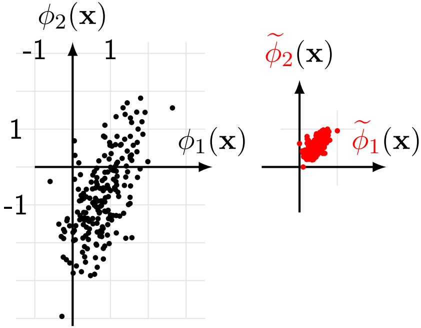
- Careful: solution *not* invariant to scaling! Should normalize input before solving.

Digression: feature normalization

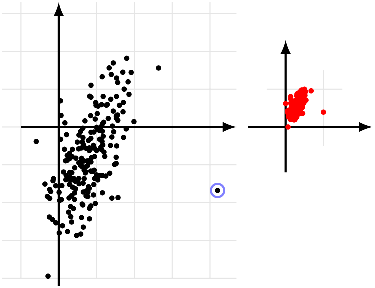
- Feature normalization: bring features to common scale
- Often a good idea to ensure good numerical behavior (even if not stricly needed mathematically)
- Box normalization: $\forall\, j : \quad \widetilde{\phi}_j(\mathbf{x}) \in [0,1]$. Procedure:

limit them in a comparable space

$$\widetilde{\phi}_j(\mathbf{x}) = \frac{\phi_j(\mathbf{x}) - \min_i \phi_j(\mathbf{x}_i)}{\max_i \phi_j(\mathbf{x}_i) - \min_i \phi_j(\mathbf{x}_i)}$$



this normalization is sensible to outliers



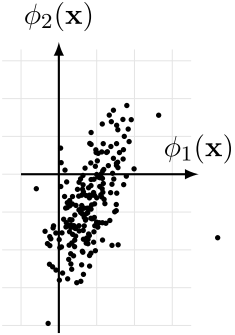
Digression: feature normalization

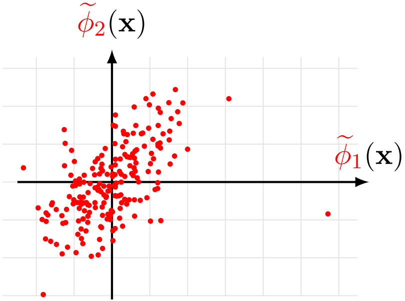
- z-scoring: $\forall\, j$, $\widetilde{\phi}_j(\mathbf{x})$ is zero-mean, unit variance over i

$$\mu_j = \frac{1}{n}\sum_i \phi_j(\mathbf{x}_i),\ \sigma_j^2 = \frac{1}{n}\sum_i(\phi_j(\mathbf{x}_i) - \mu_j)^2,$$

$$\widetilde{\phi}_j(\mathbf{x}) \ = \ \frac{\phi_j(\mathbf{x}) - \mu_j}{\sqrt{\sigma_j^2}}$$

better


$\phi_2(\mathbf{x})$
$\phi_1(\mathbf{x})$


$\widetilde{\phi}_2(\mathbf{x})$
$\widetilde{\phi}_1(\mathbf{x})$