DSC 140A

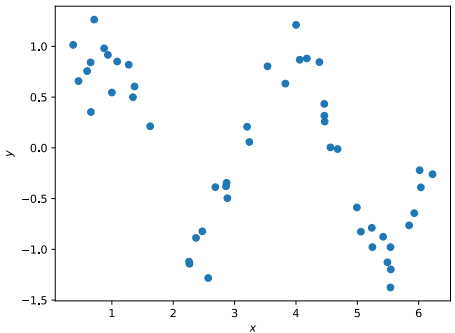
Probabilistic Modeling & Machine Learning

Lecture 8 | Part 1

Model Complexity and Regularization

Linear Regression, Non-linear Trends

- We can use least squares regression to fit non-linear patterns.



Recall: Regression with Basis Functions

- We can fit any function of the form:

$$H(\vec{x}; \vec{w}) = w_0 + w_1\phi_1(\vec{x}) + w_2\phi_2(\vec{x}) + \ldots + w_k\phi_k(\vec{x})$$

- $\phi_i(\vec{x}) : \mathbb{R}^d \to \mathbb{R}$ is called a **basis function**.

Procedure

1. Define $\vec{\phi}(\vec{x}) = (\phi_1(\vec{x}), \phi_2(\vec{x}), \ldots, \phi_k(\vec{x}))^T$

2. Form $n \times k$ **design matrix**:

$$\Phi = \begin{pmatrix} \text{Aug}(\phi(\vec{x}^{(1)})) \longrightarrow \\ \text{Aug}(\phi(\vec{x}^{(2)})) \longrightarrow \\ \vdots \\ \text{Aug}(\phi(\vec{x}^{(n)})) \longrightarrow \end{pmatrix} = \begin{pmatrix} \phi_1(\vec{x}^{(1)}) & \phi_2(\vec{x}^{(1)}) & \ldots & \phi_k(\vec{x}^{(1)}) \\ \phi_1(\vec{x}^{(2)}) & \phi_2(\vec{x}^{(2)}) & \ldots & \phi_k(\vec{x}^{(2)}) \\ \vdots & \vdots & \vdots & \vdots \\ \phi_1(\vec{x}^{(n)}) & \phi_2(\vec{x}^{(n)}) & \ldots & \phi_k(\vec{x}^{(n)}) \end{pmatrix}$$

3. Solve the **normal equations**:

$$\vec{w}^* = (\Phi^T\Phi)^{-1}\Phi^T\vec{y}$$

Example: Polynomial Curve Fitting

- Fit a function of the form:

$$H(x; \vec{w}) = w_0 + w_1 x + w_2 x^2 + w_3 x^3$$

- Use basis functions:

$$\phi_0(x) = 1 \qquad \phi_1(x) = x \qquad \phi_2(x) = x^2 \qquad \phi_3(x) = x^3$$

Example: Polynomial Curve Fitting

- Design matrix becomes:

$$\Phi = \begin{pmatrix} 1 & x_1 & x_1^2 & x_1^3 \\ 1 & x_2 & x_2^2 & x_2^3 \\ \dots & \dots & \ddots & \vdots \\ 1 & x_n & x_n^2 & x_n^3 \end{pmatrix}$$

Gaussian Basis Functions

- **Gaussians** make for useful basis functions.

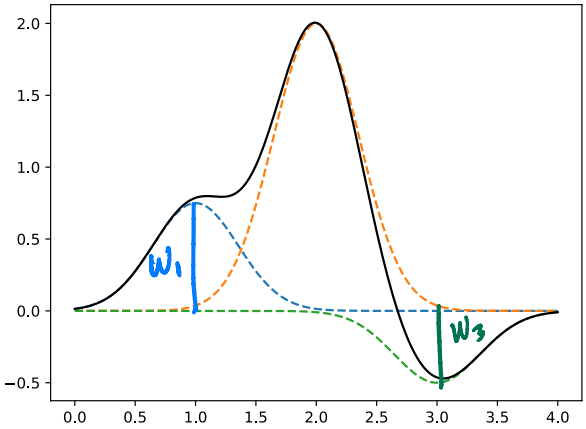
$$\phi_i(x) = \exp\left(-\frac{(x - \mu_i)^2}{\sigma_i^2}\right)$$

- Must specify[1] **center** μ_i and **width** σ_i for each Gaussian basis function.

[1] You pick these; they are not learned!

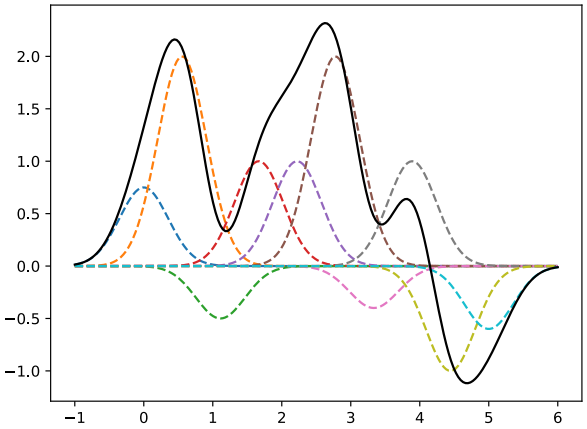
Example: $k = 3$

- A function of the form: $H(x) = w_1\phi_1(x) + w_2\phi_2(x) + w_3\phi_3(x)$, using 3 Gaussian basis functions.



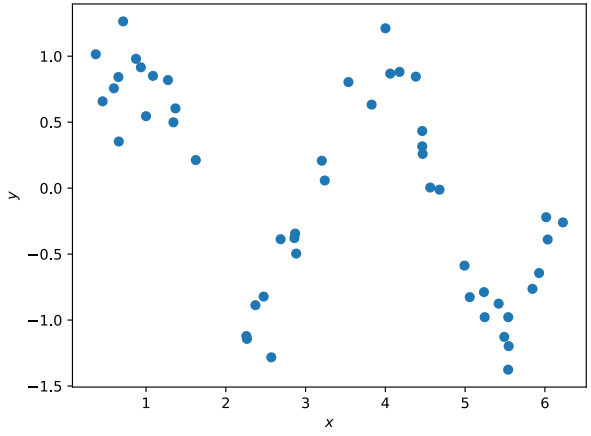
Example: $k = 10$

- The more basis functions, the more complex H can be.



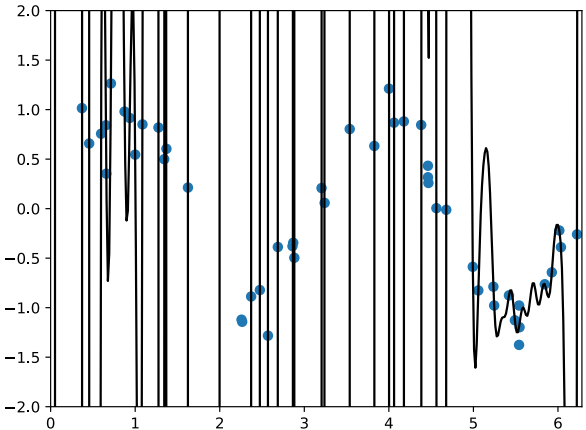
Demo: Sinusoidal Data

- Fit curve to 50 noisy data points.
- Use k = 50 Gaussian basis functions.



Result

- **Overfitting!**



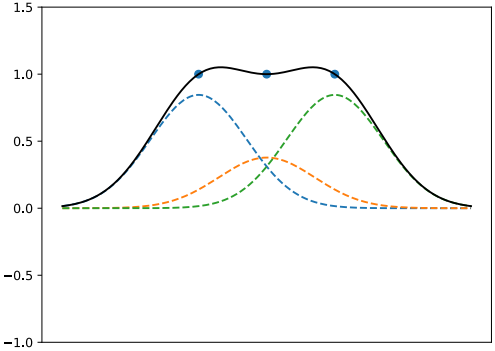
Today's Problem

- We sometimes want to fit complex models.
- **Overfitting** can occur when model is too complex.
- Can decrease complexity by reducing number of basis functions.
- Another way: **regularization.**

Complexity and $\vec{w}$

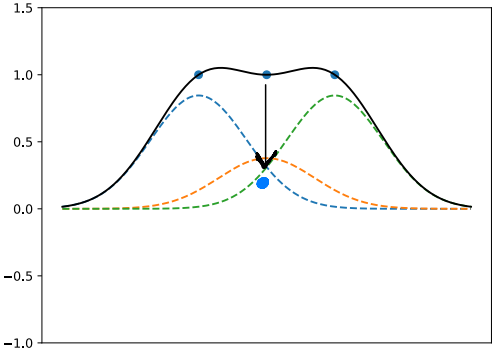
- Consider fitting 3 points with $k = 3$:

$$w_1\phi_1(\vec{x}) + w_2\phi_2(\vec{x}) + w_3\phi_3(\vec{x})$$

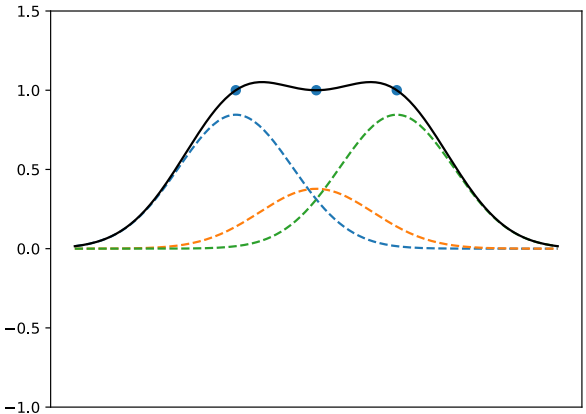


Exercise

What will happen to w_1, w_2, w_3 as the middle point is shifted down towards zero?



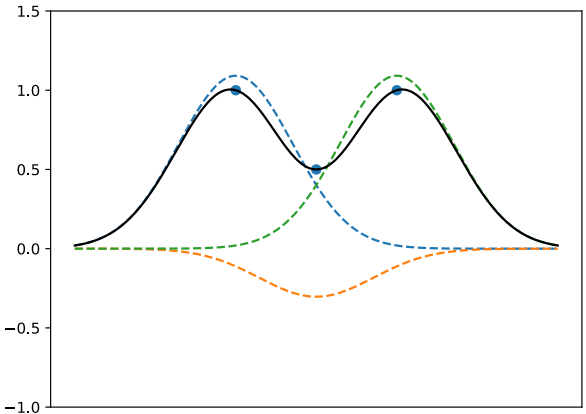
Solution



`w = [0.85 0.38 0.85]`

$\|\vec{w}\| = 1.25$

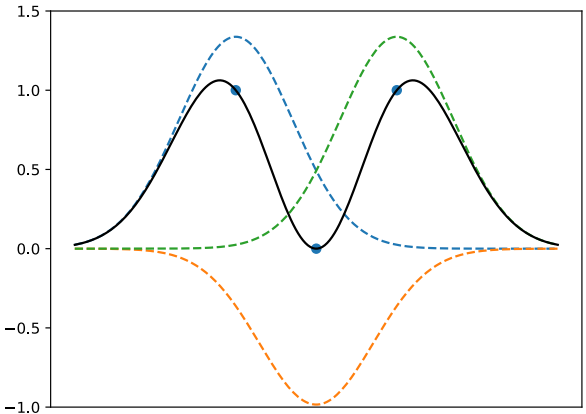
Solution




```
w = [ 1.09 -0.3 1.09]
```

$\|\vec{w}\| = 1.57$

Solution




```
w = [ 1.34 -0.98 1.34]
```

$\|\vec{w}\| = 2.13$

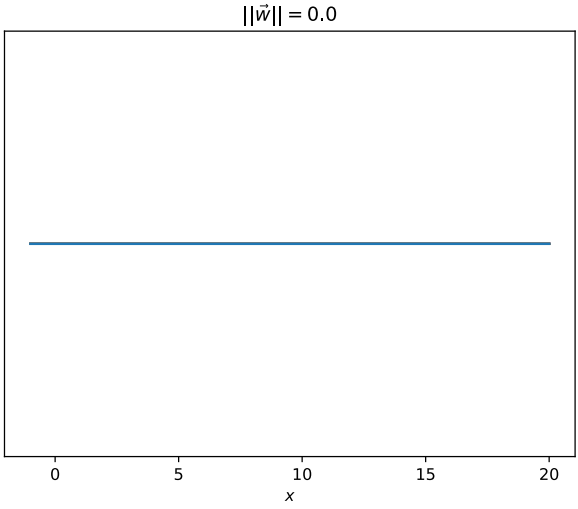
Observations

- As the middle point moves down, H becomes more “complex”.
- The weights grow in magnitude: $\|\vec{w}\|$ increases.
- **Idea:** $\|\vec{w}\|$ measures **complexity** of H.
  - Think: the larger $\|\vec{w}\|$ is, the more complex H may be.

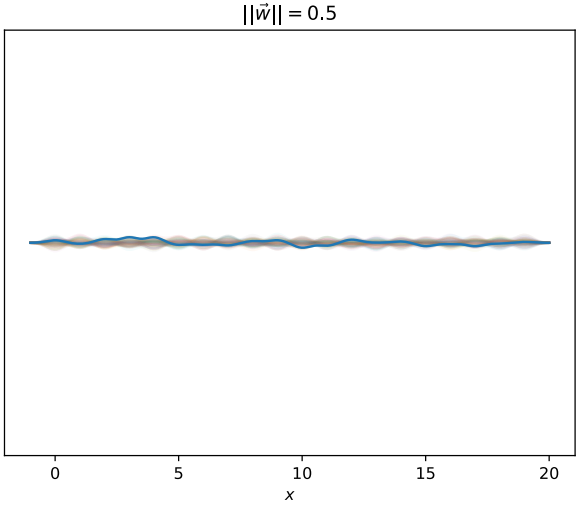
Experiment

- Consider model with $k = 20$ Gaussian basis functions.
- Generate 100 random parameter vectors $\vec{w}$.
- Plot overlapping; observe complexity.

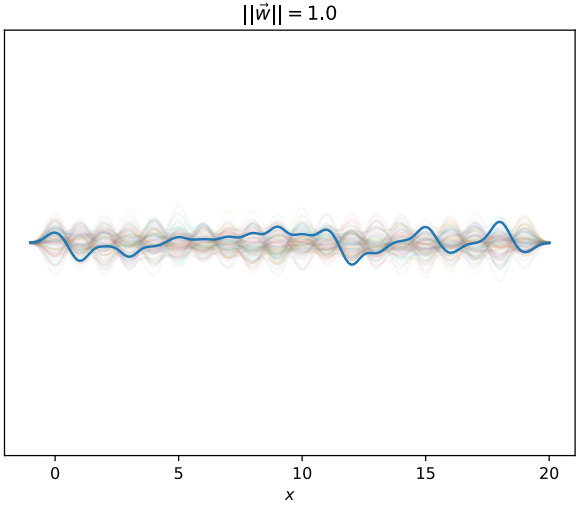
Experiment



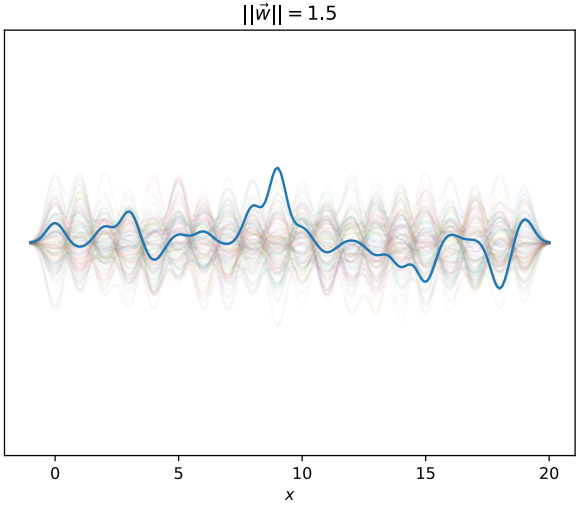
Experiment



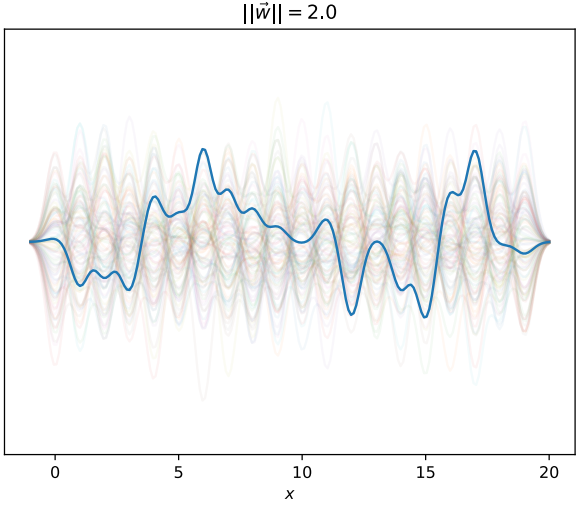
Experiment



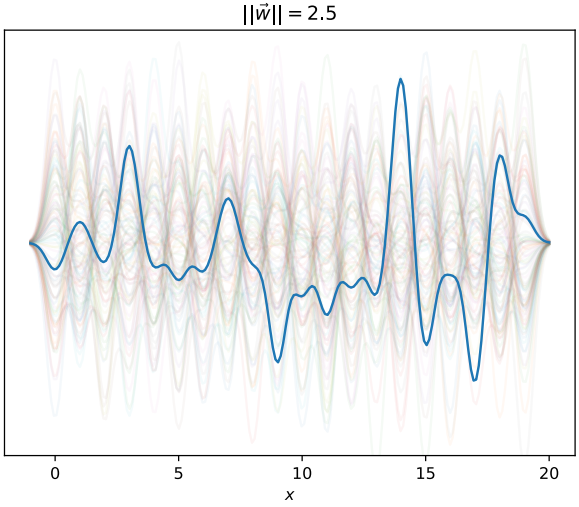
Experiment



Experiment



Experiment



Conclusion

- $\|\vec{w}\|$ is a proxy for model complexity.
  - The larger $\|w\|$, the more complex the model may be.

- **Idea:** find a model with
  - small risk on the data;
  - but also small $\|\vec{w}\|$

DSC 140A

Probabilistic Modeling & Machine Learning

Lecture 8 | Part 2

Regularized Least Squares

Recall: Least Squares Regression

- In **least squares regression**, we minimize the empirical **risk**:

$$R(\vec{w}) = \frac{1}{n}\sum_{i=1}^{n}\left(H(\vec{x}^{(i)}) - y_i\right)^2$$
$$= \frac{1}{n}\sum_{i=1}^{n}\left(\vec{w}\cdot\phi(\vec{x}^{(i)}) - y_i\right)^2$$

- Solution: $\vec{w}^* = (\Phi^T\Phi)^{-1}\Phi^T\vec{y}$

Regularized Least Squares

- **Idea:** penalize large $\|\vec{w}\|$ to control overfitting.
- **Goal:** Minimize the **regularized risk**:

$$\tilde{R}(\vec{w}) = \frac{1}{n}\sum_{i=1}^{n}\left(\vec{w}\cdot\phi(\vec{x}^{(i)}) - y_i\right)^2 + \underbrace{\lambda\|\vec{w}\|^2}_{\text{Regularizer}}$$

- $\lambda\|\vec{w}\|^2$ is a **regularization term**. $\lambda \geq 0$
  - "Tikhonov regularization"
  - λ controls "strength" of regularization.

Ridge Regression

$$\tilde{R}(\vec{w}) = \frac{1}{n}\sum_{i=1}^{n}\left(\vec{w}\cdot\phi(\vec{x}^{(i)}) - y_i\right)^2 + \underbrace{\lambda\|\vec{w}\|^2}_{\text{Regularizer}}$$

- Least squares with $\|w\|^2$ regularization is also known as **ridge regression**.

Why $\|\vec{w}\|^2$?

- We consider $\|\vec{w}\|^2$ instead of $\|\vec{w}\|$ because it will make the calculations cleaner.

Ridge Regression Solution

- Remember, with the unregularized problem, the least squares solution was:

$$\vec{w}^* = (\Phi^T \Phi)^{-1} \Phi^T \vec{y}$$

- Now that we've regularized the risk, this is no longer the solution!

Ridge Regression Solution

- **Goal:** Find $\vec{w}^*$ minimizing the **regularized risk**:

$$\tilde{R}(\vec{w}) = \frac{1}{n}\sum_{i=1}^{n}\left(\vec{w}\cdot\phi(\vec{x}^{(i)}) - y_i\right)^2 + \lambda\|\vec{w}\|^2$$

- Recall:

$$\frac{1}{n}\sum_{i=1}^{n}\left(\vec{w}\cdot\phi(\vec{x}^{(i)}) - y_i\right)^2 = \frac{1}{n}\|\Phi\vec{w} - \vec{y}\|^2$$

- So:

$$\tilde{R}(\vec{w}) = \frac{1}{n}\|\Phi\vec{w} - \vec{y}\|^2 + \lambda\|\vec{w}\|^2$$

Ridge Regression Solution

- **Strategy:** calculate $d\tilde{R}/d\vec{w}$, set to $\vec{0}$, solve.
- **Solution:** $\vec{w}^* = (\Phi^T\Phi + n\lambda I)^{-1}\Phi^T\vec{y}$
- Compare this to solution of unregularized problem: $\vec{w}^* = (\Phi^T\Phi)^{-1}\Phi^T\vec{y}$



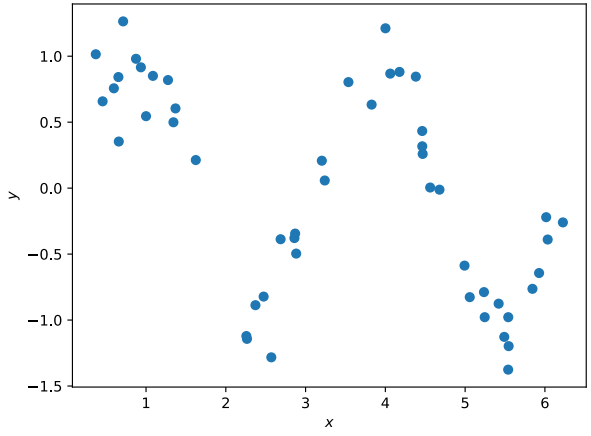
Interpretation

$$\vec{w}^* = (\Phi^T\Phi + n\lambda I)^{-1}\Phi^T\vec{y}$$

- Adds small number λ to diagonal of $\Phi^T\Phi$
- Improves condition number of $\Phi^T\Phi + n\lambda I$
  - Helpful when multicollinearity exists

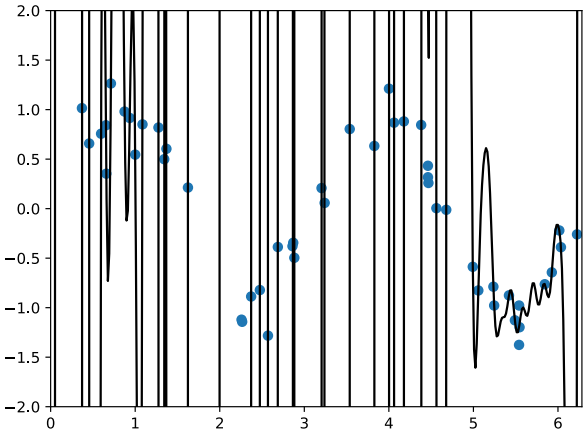
Demo: Sinusoidal Data

- Fit curve to 50 noisy data points.
- Use k = 50 Gaussian basis functions.

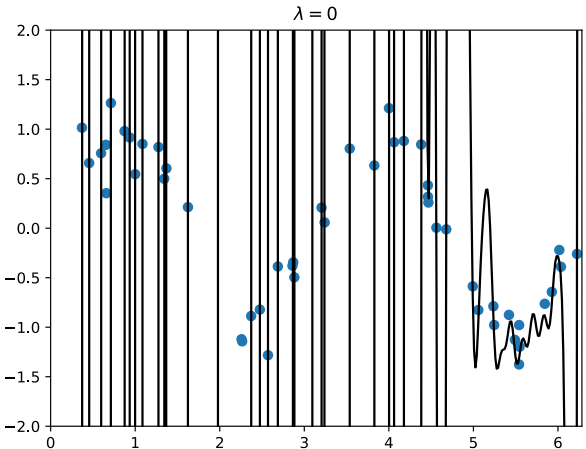


Result: no regularization

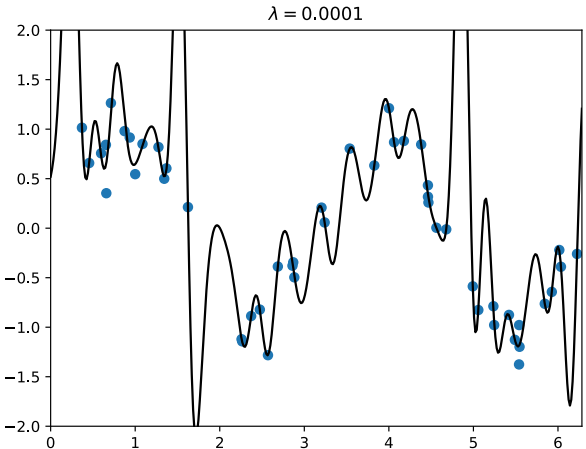
- **Overfitting!**



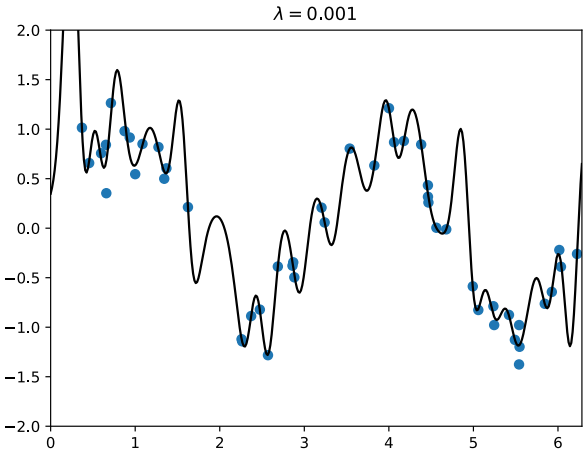
Result: regularization



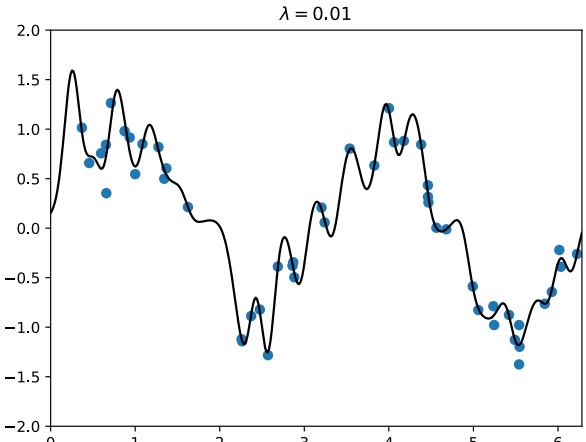
Result: regularization



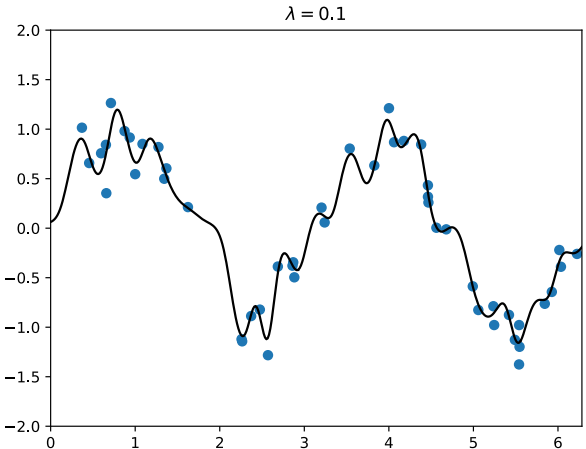
Result: regularization



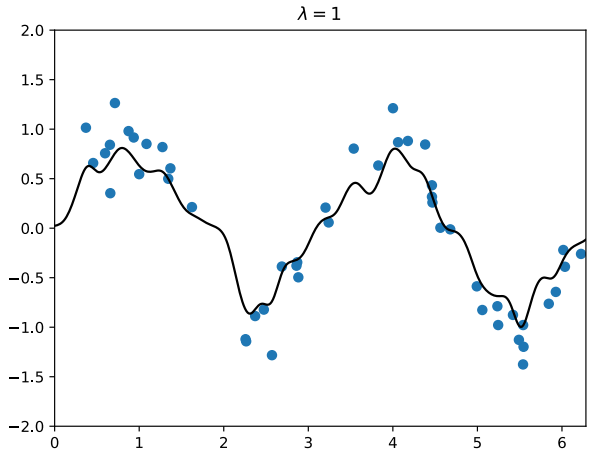
Result: regularization



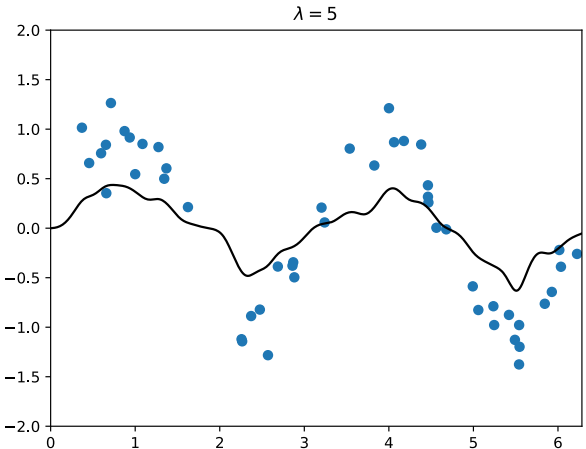
Result: regularization



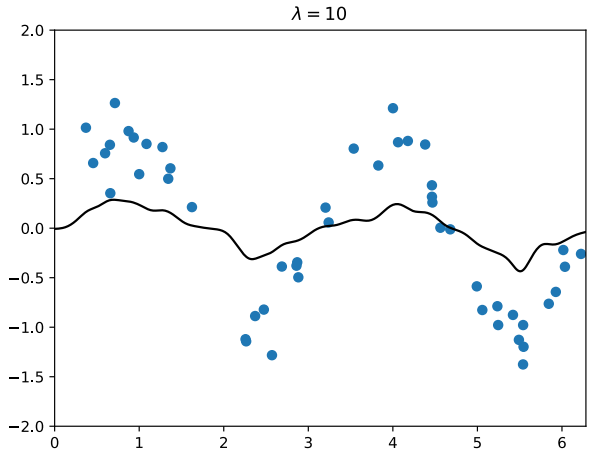
Result: regularization



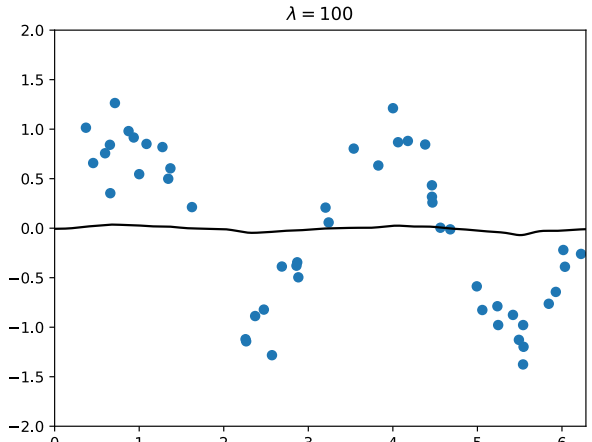
Result: regularization



Result: regularization



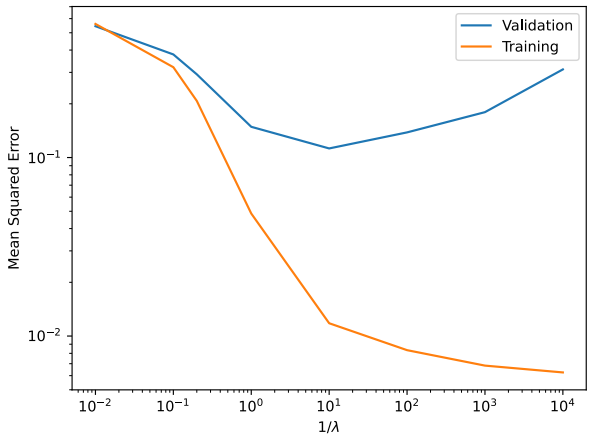
Result: regularization



Picking λ

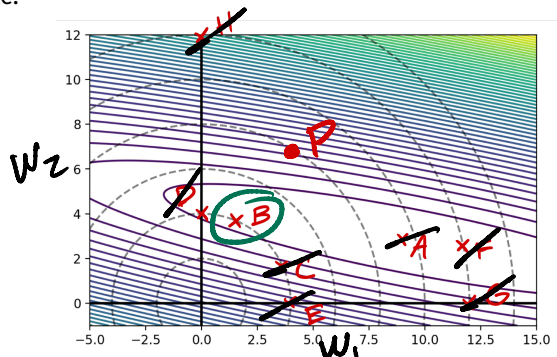
- λ controls strength of penalty
  - Larger λ: penalize complexity more
  - Smaller λ: allow more complexity

- To choose, use **cross-validation**.


Validation
Training
Mean Squared Error
10^{-1}
10^{-2}
10^{-2}
10^{-1}
10^{0}
10^{1}
10^{2}
10^{3}
10^{4}
$1/\lambda$

Exercise

Shown below are the contours of the unregularized empirical risk, $R(\vec{w})$. The dashed lines show places where $\|\vec{w}\|_2$ is 2, 4, 6, etc.



Which of the marked points could be a minimizer of the regularized risk, $\tilde{R}(\vec{w}) = R(\vec{w}) + \lambda\|\vec{w}\|_2^2$, where $\lambda > 0$?

Main Idea

Regularization trades an **increase** in risk for a **decrease** in complexity.

DSC 140A

Probabilistic Modeling & Machine Learning

Lecture 8 | Part 3

The LASSO

p norm regularization

- In the last section, we minimized:

$$\tilde{R}(\vec{w}) = \frac{1}{n} \sum_{i=1}^{n} \left(\vec{w} \cdot \phi(\vec{x}^{(i)}) - y_i \right)^2 + \lambda \|\vec{w}\|^2$$

- What is special about $\|\vec{w}\|$?

$\vec{u} = (u_1, u_2, \cdots, u_d)^T$

p norms

- For any $p \in [0, \infty)$, the p **norm** of a vector $\vec{u}$ is defined as

$$\|\vec{u}\|_p = \left(\sum_{i=1}^{d} |u_i|^p \right)^{1/p}$$

Special Case: $p = 2$

- When $p = 2$, we have the familiar **Euclidean norm**:

$$\|\vec{u}\|_2 = \left(\sum_{i=1}^{d} u_i^2\right)^{1/2} = \|\vec{u}\|$$

$$= \sqrt{u_1^2 + u_2^2 + \cdots + u_d^2}$$

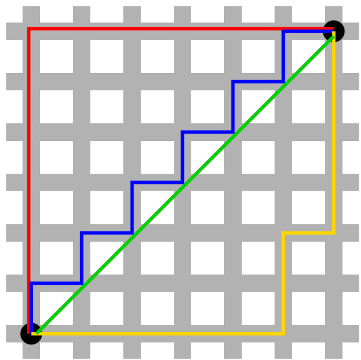
Special Case: $p = 1$

- When $p = 1$, we have the "taxicab norm"

$$\|\vec{u}\|_1 = \sum_{i=1}^{d} |u_i|$$

$= |u_1| + |u_2| + \cdots + |u_d|$

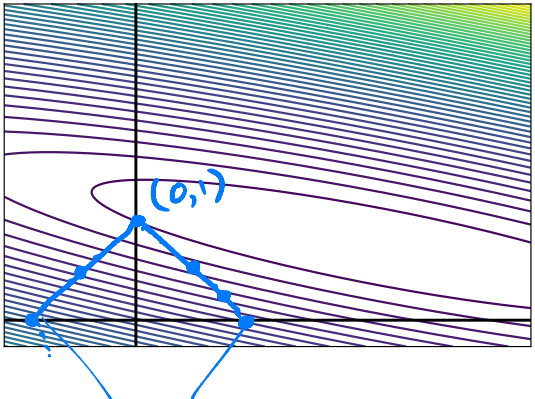
→ Manhattan Norm



Exercise

Draw the “unit circle” for the 1-norm. That is, draw where $\|\vec{w}\|_1 = 1$.

$\|\vec{w}\|_1 = 1$



$(\frac{1}{2}, \frac{1}{2})$

$(-\frac{1}{2}, \frac{1}{2})$

$(\frac{3}{4}, \frac{1}{4})$

1-norm Regularization

- Consider the 1-norm regularized risk:

$$\tilde{R}(\vec{w}) = \frac{1}{n}\sum_{i=1}^{n}\left(\vec{w}\cdot\phi(\vec{x}^{(i)}) - y_i\right)^2 + \lambda\|\vec{w}\|_1$$

- Least squares regression with 1-norm regularization is called the **LASSO**.

Solving the LASSO

- No longer differentiable.
- No exact solution, unlike ridge regression.[2]
- Can solve with subgradient descent.

[2] Except in special cases, such as orthonormal Φ

1-norm Regularization

- The 1-norm encourages **sparse** solutions.
  - That is, solutions where many entries of $\vec{w}$ are zero.

- **Interpretation:** feature selection.

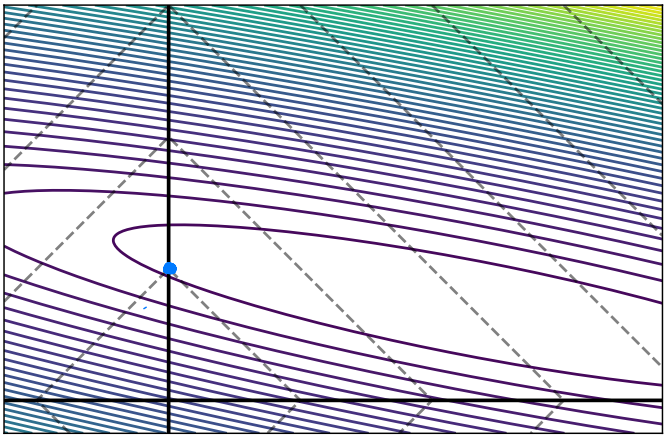
Example

- Randomly-generated data:

$$y = 3x_1 + 0.2x_2 - 4x_3 + \mathcal{N}(0, .2)$$

	w_1	w_2	w_3
Unreg.	2.33	-0.08	-4.77
2-norm	2.29	-0.10	-4.73
1-norm	2.72	0	-3.76

Why?



DSC 140A

Probabilistic Modeling & Machine Learning

Lecture 8 | Part 4

Regularized Risk Minimization

Regularized ERM

- We have seen regularization in the context of least squares regression.
- However, it is generally useful with other risks.
- E.g., hinge loss + 2-norm regularization = soft-SVM

General Regularization

- Let $R(\vec{w})$ be a risk function.
- Let $\rho(\vec{w})$ be a regularization function.
- The regularized risk is:

$$\tilde{R}(\vec{w}) = R(\vec{w}) + \rho(\vec{w})$$

- **Goal:** minimized regularized empirical risk.

Regularized Linear Models

Loss	Regularization	Name
square	2-norm	ridge regression
square	1-norm	LASSO
square	1-norm + 2-norm	elastic net
hinge	2-norm	soft-SVM