Overfitting and regularization

Herman Kamper

http://www.kamperh.com/

Linear regression

Examples of overfitting

Herman Kamper

http://www.kamperh.com/

Overfitting example

Suppose we want to fit a regression model with scalar input using basis functions.
Also suppose we have $N = 10$ training items.

Our goal: $\underline{y} \approx \underline{\underline{\Phi}}\, \underline{w}$

If we use 2 basis functions, the shapes will be:

$$\underset{10\times1}{\underline{y}} \approx \underset{10\times2}{\underline{\underline{\Phi}}}\, \underset{2\times1}{\underline{w}}$$

If instead we have 10 basis functions we would have:

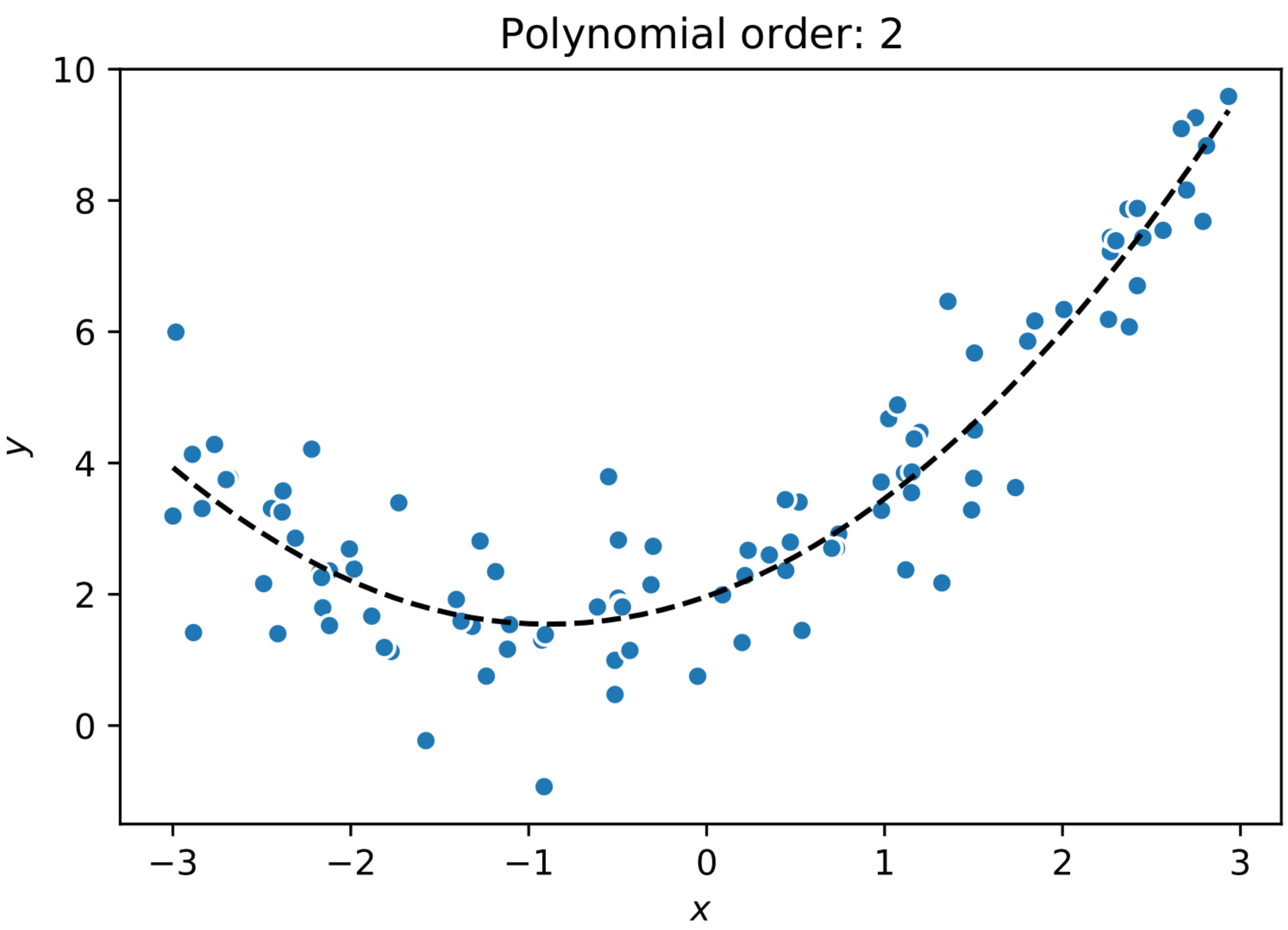
$$\underset{10\times1}{\underline{y}} \approx \underset{10\times10}{\underline{\underline{\Phi}}}\, \underset{10\times1}{\underline{w}}$$

But this is solvable exactly! 10 equations in 10 unknowns (the 10 weights). So we can solve exactly: $\underline{w} = \underline{\underline{\Phi}}^{-1}\underline{y}$ } If $\underline{\underline{\Phi}}$ is invertible.

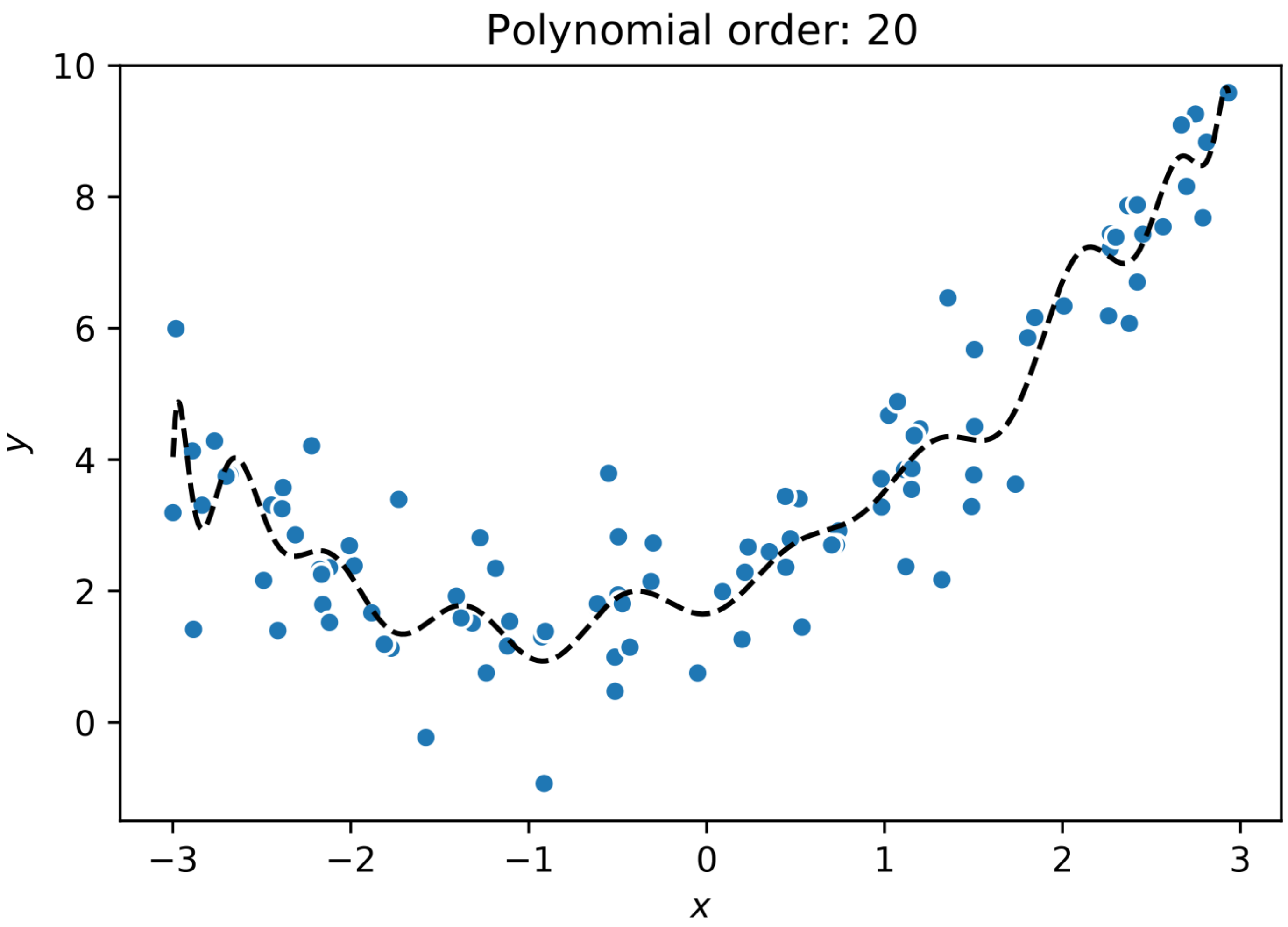
Questions

- What would the value of the loss J be?
- Would this be a good fit for making future predictions?

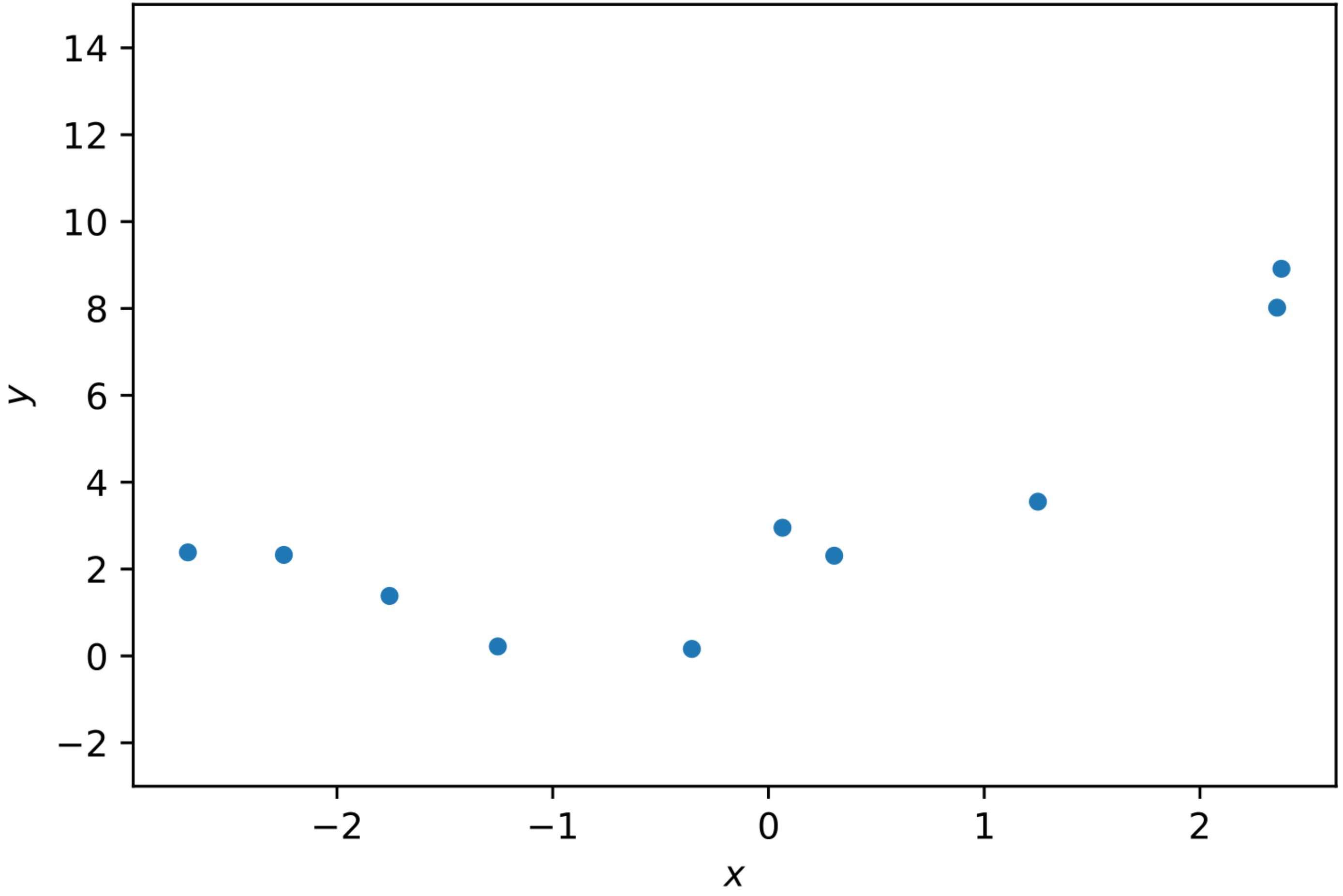
Polynomial regression



Polynomial regression

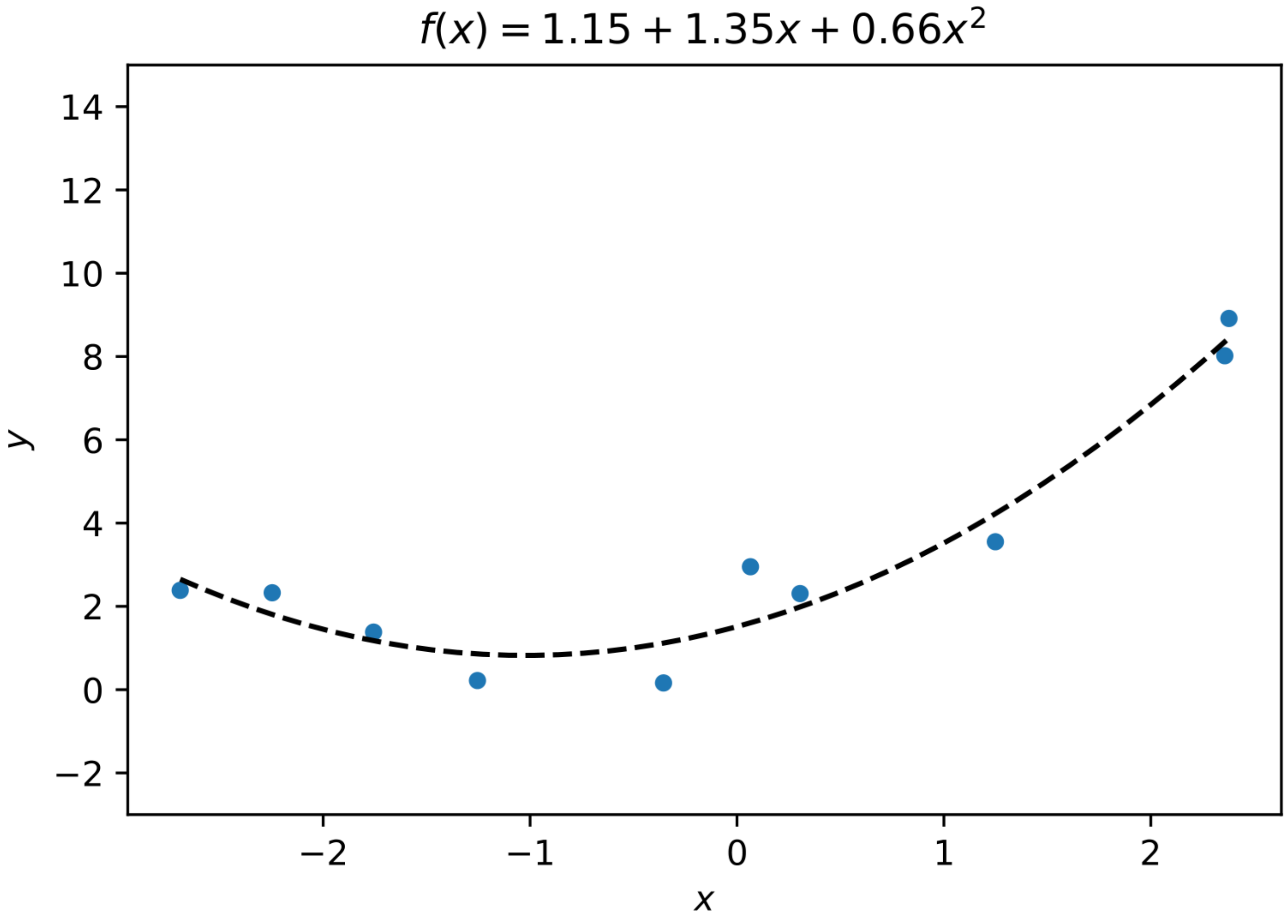


Polynomial regression

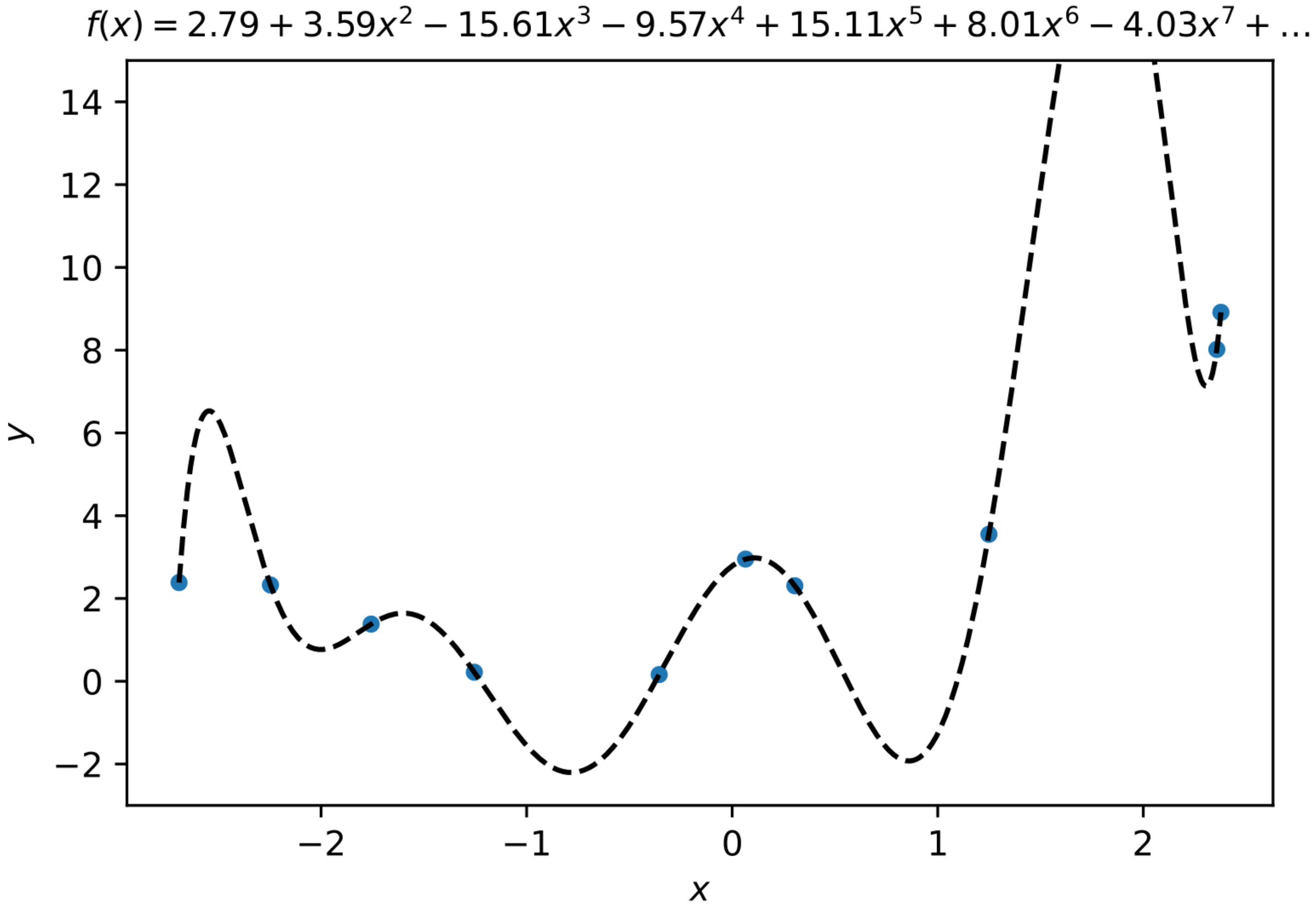


Polynomial regression

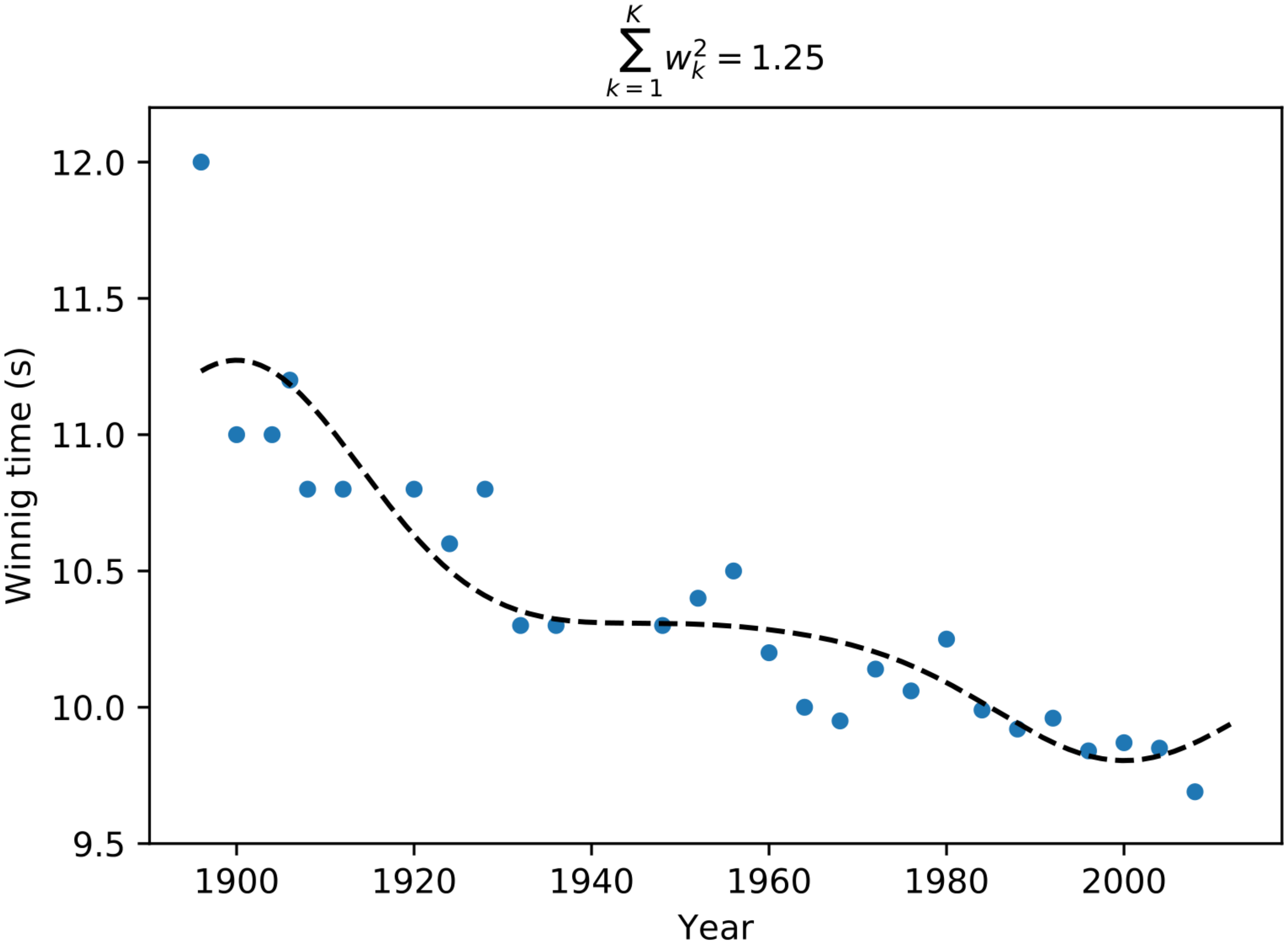
$f(x;\hat{\underline{w}}) =$



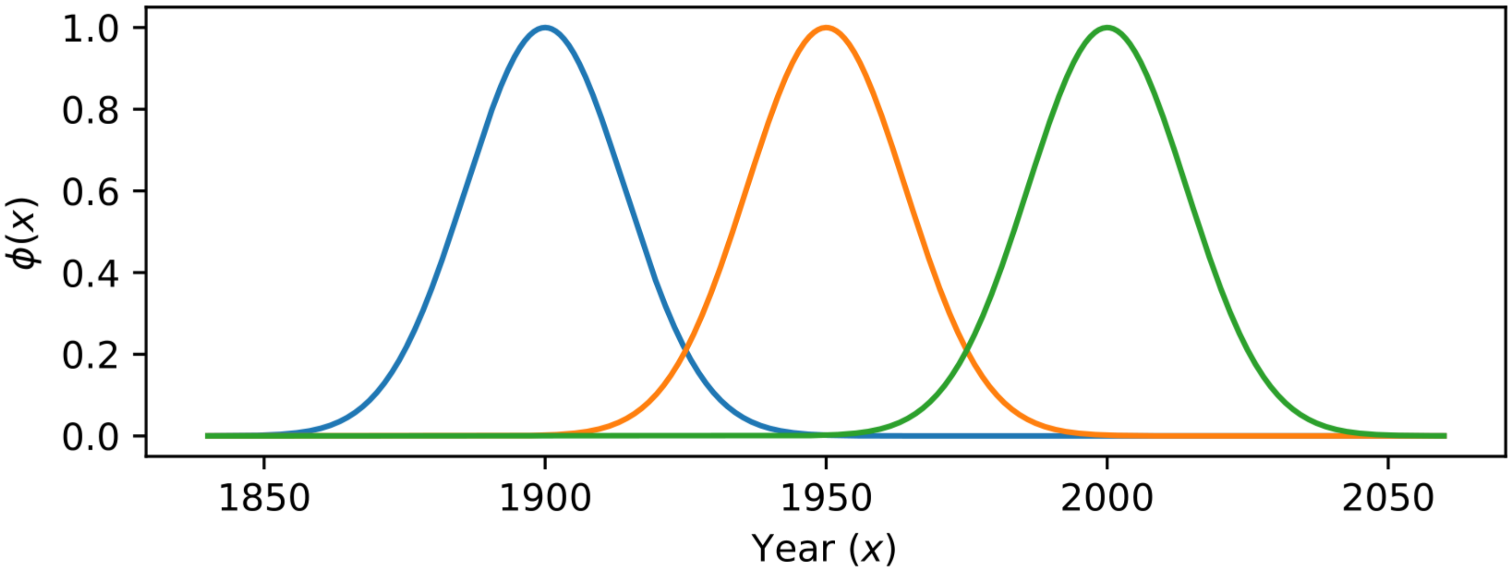
Polynomial regression



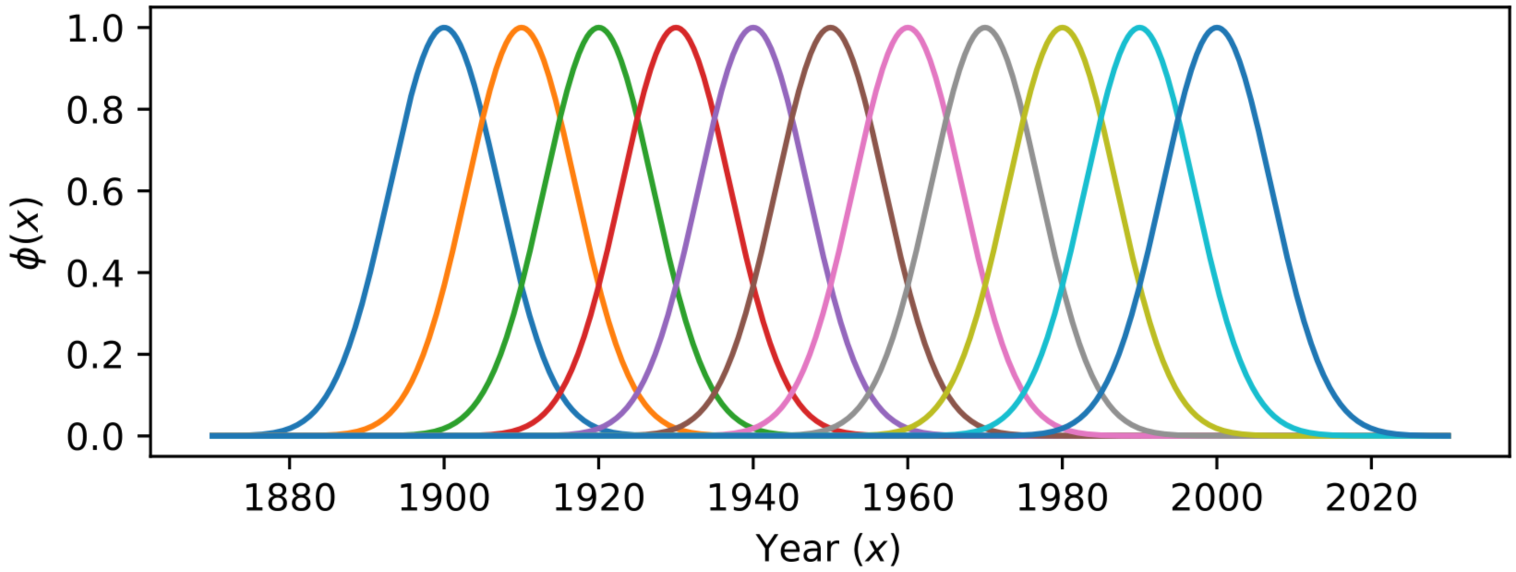
RBF with $c = [1900, 1950, 2000]$ and $h = 20$



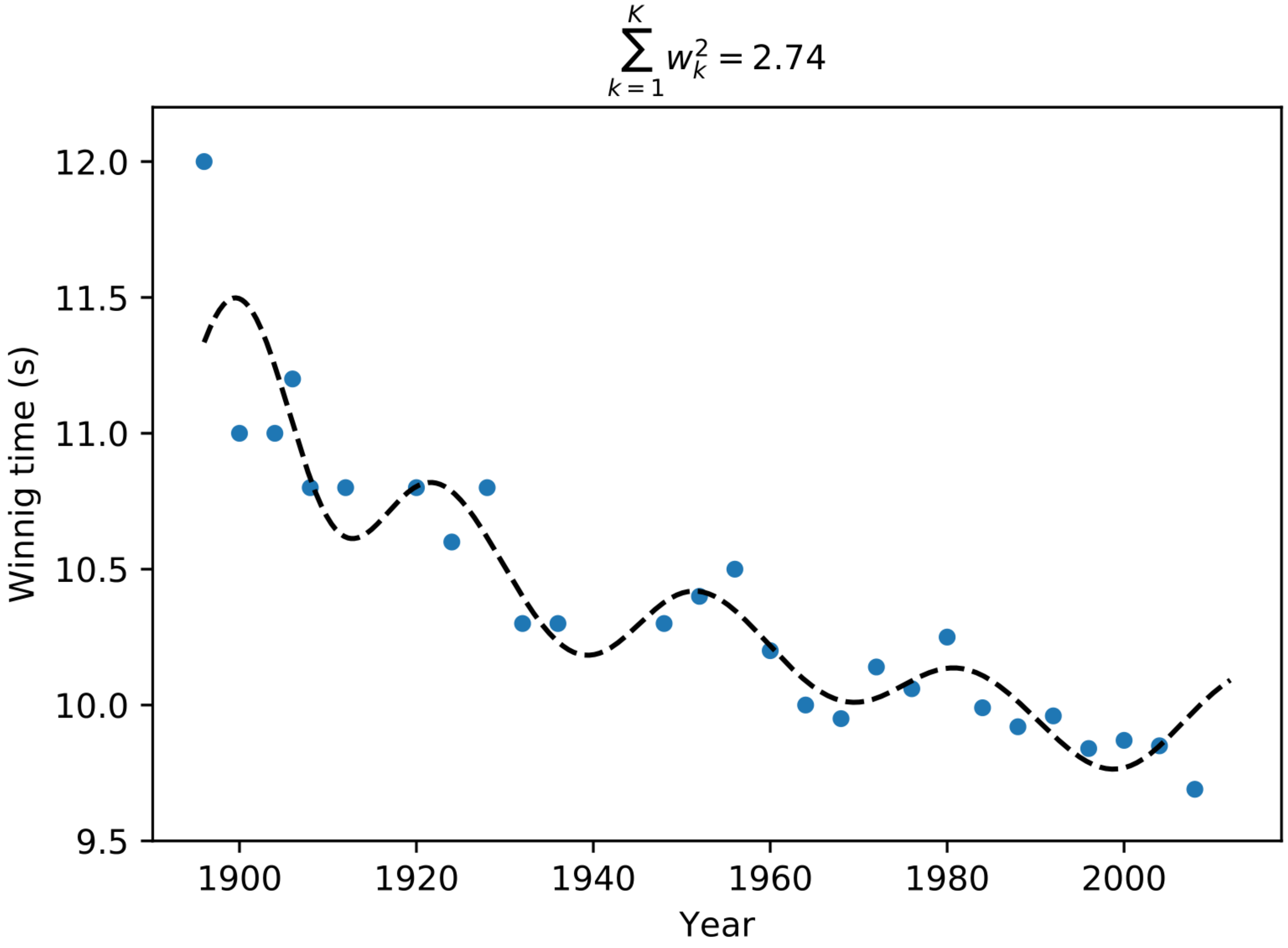
RBF with $c = [1900, 1950, 2000]$ and $h = 20$



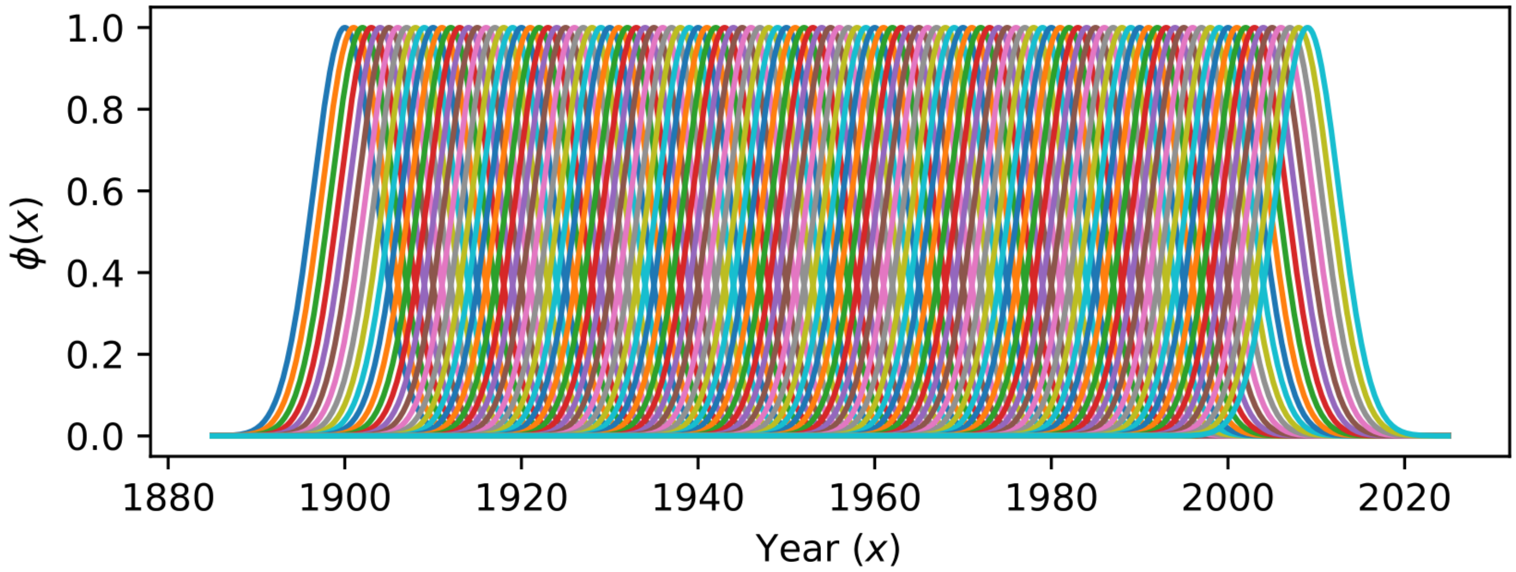
RBF with $c = [1900, 1910, \ldots, 2000]$ and $h = 10$



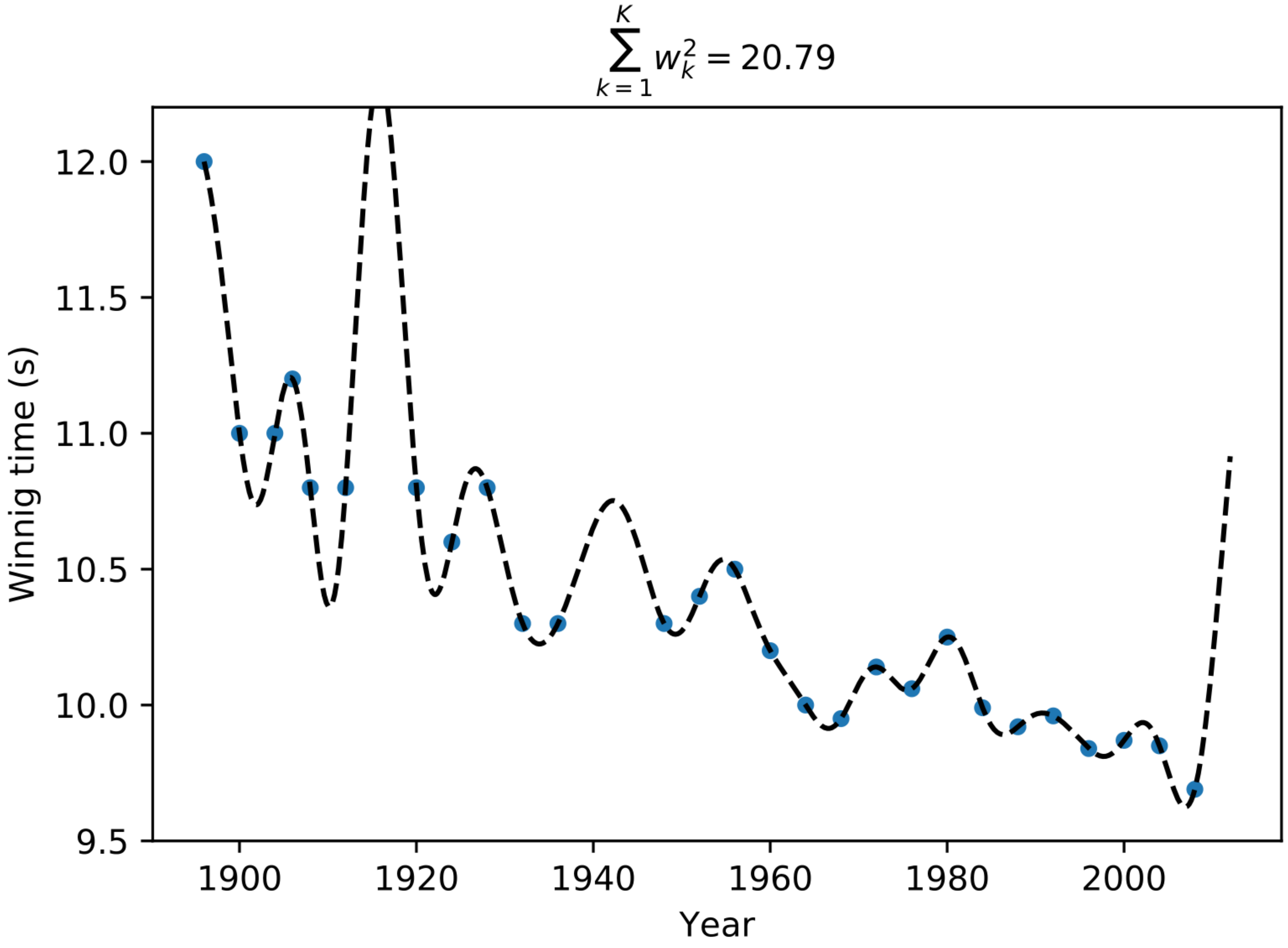
RBF with $c = [1900, 1910, \ldots, 2000]$ and $h = 10$



RBF with $c = [1900, 1901, \ldots, 2000]$ and $h = 1$



RBF with $c = [1900, 1901, \ldots, 2000]$ and $h = 1$



Regularization

Combatting overfitting

Herman Kamper

http://www.kamperh.com/

Regularization:

Might want to fit higher-order models, but then it would be useful to be able to control their "complexity" in some way.

Idea:

Standard loss (e.g. squared)

$$\hat{\underline{w}} = \arg\min_{\underline{w}} \left\{ J(\underline{w}) + \text{penalty}(\underline{w}) \right\}$$

Penalty functions that constrains $\underline{w}$ to be small are sometimes called "shrinkage" methods. We consider two penalty methods:

- Ridge (L_2) regularization
- Lasso (L_1) regularization

Ridge (L_2) regularization:

Need to choose (see later)

$$J_\lambda(\underline{w}) = \sum_{n=1}^{N} \left(y^{(n)} - f(\underline{x}^{(n)}; \underline{w}) \right)^2 + \lambda \sum_{k=1}^{K} w_k^2 \quad \ldots (1)$$

We normally don't regularize w_0. Why not? An easy hack is to zero-mean your data beforehand, i.e. the colums of $\underline{X}$ (or $\underline{\Phi}$) normalized to have a mean of $\underline{0}$.

$$J_\lambda(\underline{w}) = \sum_{n=1}^{N} \left(y^{(n)} - f(\underline{x}^{(n)}; \underline{w}) \right)^2 + \lambda \underline{w}^T \underline{w}$$

Can find closed-form solution exactly as before:

D-dimensional identity matrix

$$\underline{\hat{w}}_\lambda = \left(\underline{X}^T \underline{X} + \lambda \underline{I} \right)^{-1} \underline{X}^T y$$

Note that anywhere we have an $\underline{X}$ we can always replace that with basis function design martrix $\underline{\Phi}$.

Lasso ($L1$) regularization:

$$J_\lambda(\underline{w}) = \sum_{n=1}^{N} \left(y^{(n)} - f(\underline{x}^{(n)}; \underline{w}) \right)^2 + \lambda \sum_{k=1}^{K} |w_k| \quad \ldots (2)$$

Still convex (unique minimum) but not "smooth" (differentiable). But other methods exist to solve (e.g. gradient descent instead of closed form method – later). L_1 regularization has the effect of pushing weights to 0. This can be useful for interpreting data/a model (but be careful!)

Why does L_1 do this but not L_2?

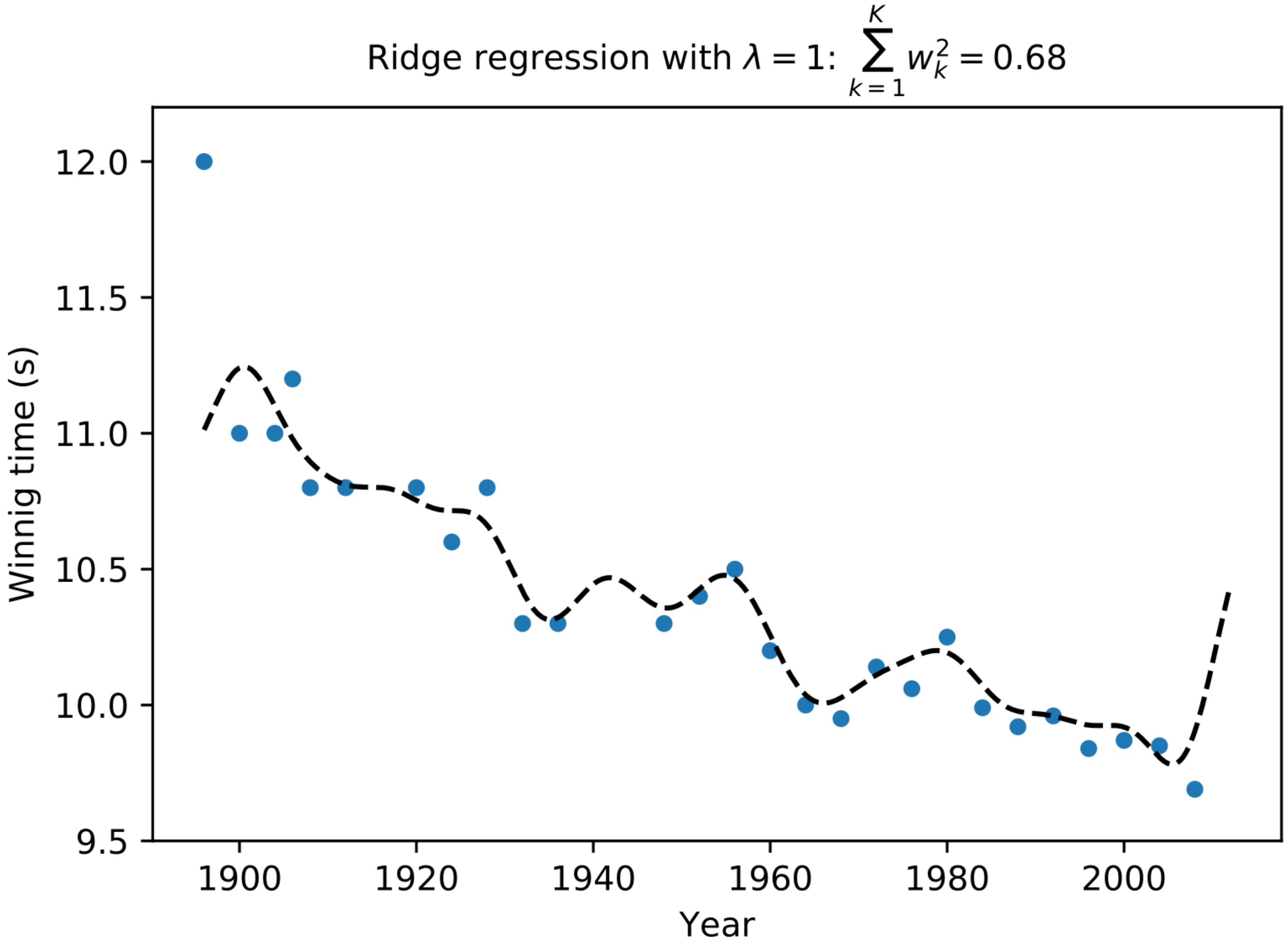
(Just intuitively from (1) and (2))

$$\frac{\partial J_\lambda}{\partial w_k}$$

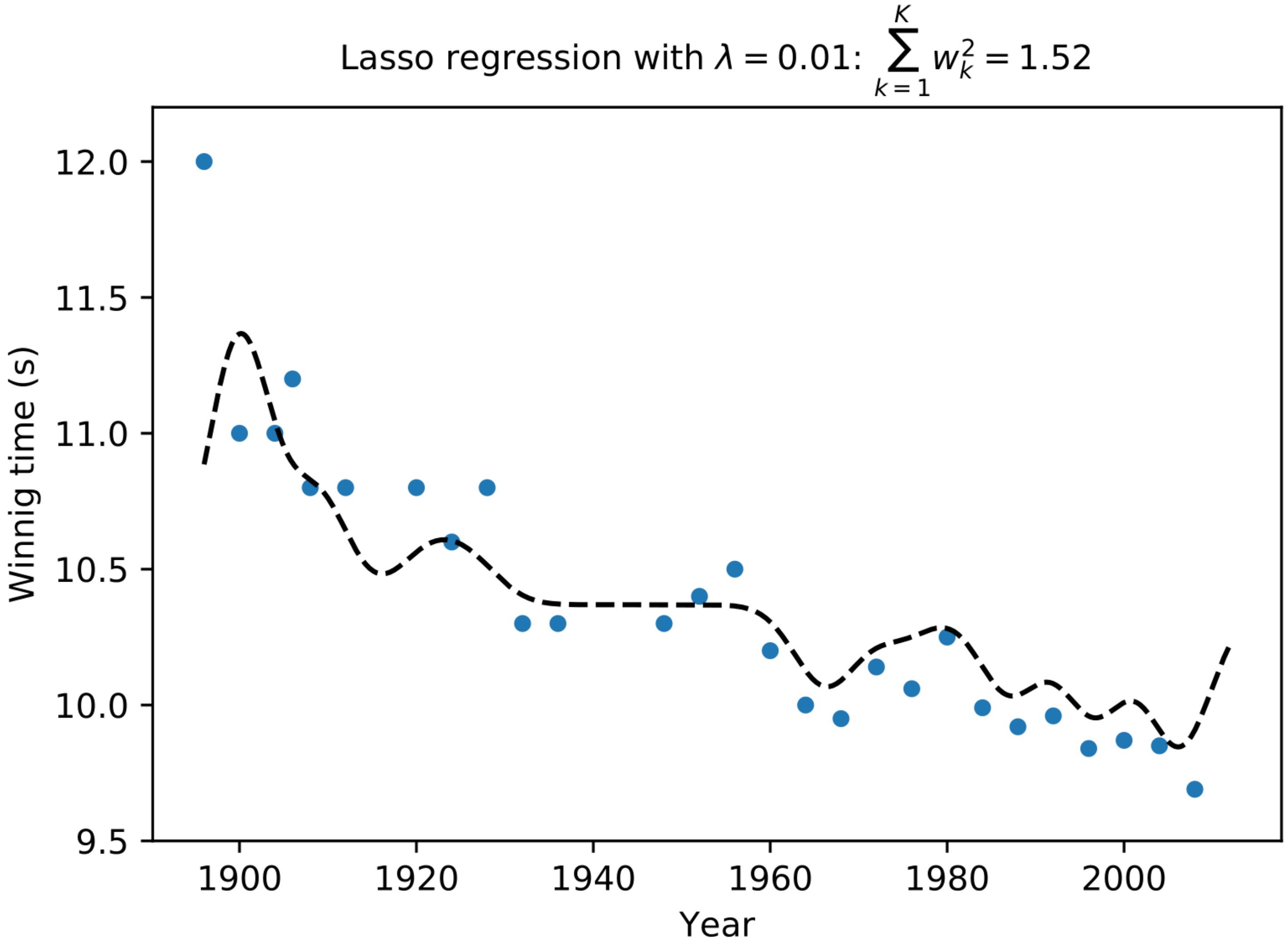
L_2:

L_1:

RBF with $c = [1900, 1901, \ldots, 2000]$ and $h = 1$



RBF with $c = [1900, 1901, \ldots, 2000]$ and $h = 1$



Lasso and ridge regression on diabetes data

L_1 L_2