Regression 2

StatML

20.02.2014

Aasa Feragen
(aasa@diku.dk)

What happens now?

- The TAs have graded your assignments
- General and individual feedback at TA sessions
- **Optional lecture on the Perceptron**
  by Christian Friday 13.30-14.15 in Aud 3 (HCØ)
- **Math Q&A / help session Friday afternoon**
  14.15 - ca 16.00, A103, A104 and A105 at HCØ

About Assignment 1

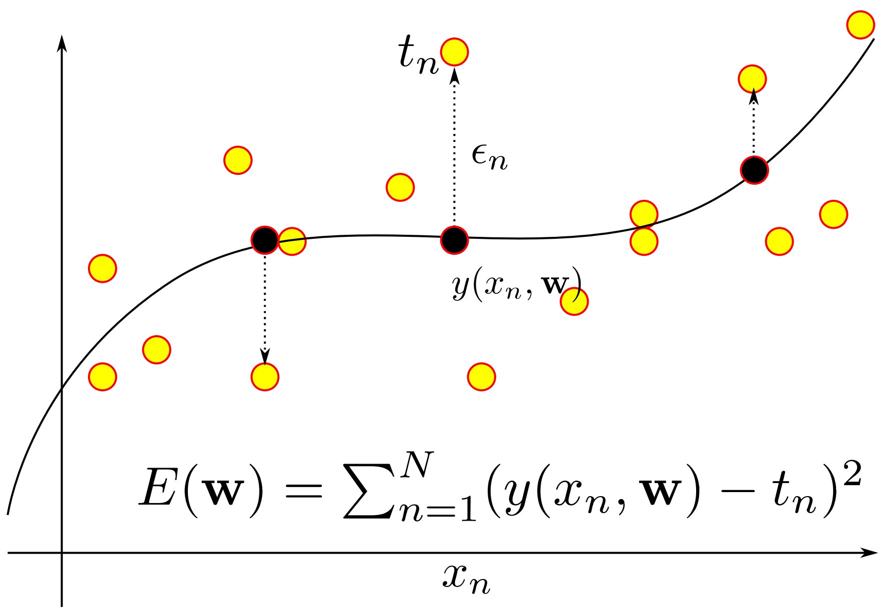
- Deadline for resubmission is Tuesday 25.02.
- There will only be one resubmission round.
- If you are asked to resubmit Exercise 1.3, you may instead choose to resubmit the make-up assignment posted in Absalon.
- This is a one-time only exception.

After today's lecture you should:

- Be able to produce a regularized maximum likelihood solution to a linear regression model
- Be able to produce a maximum a posteriori solution to a linear regression model
- Understand the relation between maximum a posteriori solutions and regularized maximum likelihood solutions
- Be familiar with different choices of regularization of why you would want to use them
- Understand the curse of dimensionality and its impact on solving regression problems
- Understand the effect of choice of prior in MAP estimates for different problems
- Be able to recognize and pose practical regression problems

Last time: Geometric and Probabilistic approaches to Regression

Geometric approach



$$E(\mathbf{w}) = \sum_{n=1}^{N} (y(x_n, \mathbf{w}) - t_n)^2$$

Maximum Likelihood approach

Assume: Gaussian noise model

$$\mathcal{N}(t|y(x, \mathbf{w}), \beta^{-1})$$

Likelihood of data $\mathbf{t}$ under model fixed by $\mathbf{w}, \mathbf{x}$

$$p(\mathbf{t}|\mathbf{x}, \mathbf{w}, \beta) = \prod_{n=1}^{N} \mathcal{N}(t_n|y(x_n, \mathbf{w}), \beta^{-1})$$

Maximizing the likelihood is equivalent to minimizing $\sum_{n=1}^{N} (y(x_n, \mathbf{w}) - t_n)^2$

Last time: Analytic solution to Maximum Likelihood a.k.a. Geometric Least Square regression

Minimizing $\sum_{n=1}^{N}(y(\mathbf{x}_n,\mathbf{w})-t_n)^2$ **when** $y(\mathbf{x},\mathbf{w})=\mathbf{w}^T\boldsymbol{\phi}(\mathbf{x})$.

$$
\begin{aligned}
&\frac{\partial}{\partial w_i}\left[\sum_{n=1}^{N}(\mathbf{w}^T\boldsymbol{\phi}(\mathbf{x}_n)-t_n)^2\right]\\
&=\sum_{n=1}^{N}\frac{\partial}{\partial w_i}\left[(\mathbf{w}^T\boldsymbol{\phi}(\mathbf{x}_n)-t_n)^2\right]\\
&=\sum_{n=1}^{N}2(\mathbf{w}^T\boldsymbol{\phi}(\mathbf{x}_n)-t_n)\cdot\frac{\partial}{\partial w_i}(\mathbf{w}^T\boldsymbol{\phi}(\mathbf{x}_n)-t_n)\\
&=2\sum_{n=1}^{N}(\mathbf{w}^T\boldsymbol{\phi}(\mathbf{x}_n)-t_n)\cdot\phi_i(\mathbf{x}_n)-t_n)=0 \qquad \text{for all } i
\end{aligned}
$$

Since $\boldsymbol{\phi}(\bar{x})^T=(\phi_0(\mathbf{x}),\phi_1(\mathbf{x}),\ldots,\phi_{M-1}(\mathbf{x}))$, we get

$$\sum_{n=1}^{N}\mathbf{w}^T\boldsymbol{\phi}(\mathbf{x}_n)\boldsymbol{\phi}(\mathbf{x}_n)^T-\sum_{n=1}^{N}t_n\boldsymbol{\phi}(\mathbf{x}_n)^T=0,$$

or

$$0=\mathbf{w}^T\sum_{n=1}^{N}\boldsymbol{\phi}(\mathbf{x}_n)\boldsymbol{\phi}(\mathbf{x}_n)^T-\sum_{n=1}^{N}t_n\boldsymbol{\phi}(\mathbf{x}_n)^T \quad (*)$$

Setting $\Phi=\begin{pmatrix}\phi_0(\mathbf{x}_1) & \phi_1(\mathbf{x}_1) & \ldots & \phi_{M-1}(\mathbf{x}_1)\\ \phi_0(\mathbf{x}_2) & \phi_1(\mathbf{x}_2) & \ldots & \phi_{M-1}(\mathbf{x}_2)\\ \vdots & & & \vdots\\ \phi_0(\mathbf{x}_N) & \phi_1(\mathbf{x}_N) & \ldots & \phi_{M-1}(\mathbf{x}_N)\end{pmatrix}$

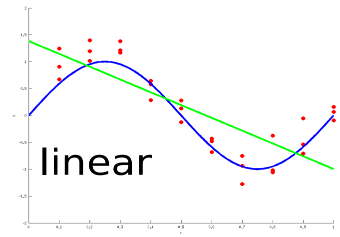
we rewrite $(*)$ as $0=\mathbf{w}^T(\Phi^T\Phi)-\mathbf{t}^T\Phi$

$$
\begin{aligned}
&\Rightarrow \mathbf{w}^T(\Phi^T\Phi)=\mathbf{t}^T\Phi\\
&\Rightarrow (\Phi^T\Phi)^T\mathbf{w}=(\Phi^T\Phi)\mathbf{w}=\Phi^T\mathbf{t} \quad \text{(transpose)}\\
&\Rightarrow \mathbf{w}=(\Phi^T\Phi)^{-1}\Phi^T\mathbf{t}
\end{aligned}
$$

Different basis functions

$y(\mathbf{x}, \mathbf{w}) = \sum_{j=0}^{M-1} w_j \phi_j(\mathbf{x}) = \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x})$ where $\{\phi_j(\mathbf{x})\}$ are **basis functions**

$$\mathbf{w} = \begin{pmatrix} w_0 \\ w_1 \\ \vdots \\ w_{M-1} \end{pmatrix} \text{and } \boldsymbol{\phi} = \begin{pmatrix} \phi_0 \\ \phi_1 \\ \vdots \\ \phi_{M-1} \end{pmatrix}$$

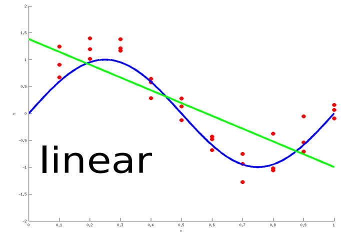


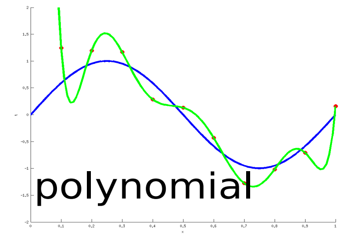
$$y(\mathbf{x}, \mathbf{w}) = w_0 + w_1 x_1 + w_2 x_2 + \ldots + w_D x_D$$

Different basis functions

$y(\mathbf{x},\mathbf{w}) = \sum_{j=0}^{M-1} w_j \phi_j(\mathbf{x}) = \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x})$ where $\{\phi_j(\mathbf{x})\}$ are **basis functions**

$$\mathbf{w} = \begin{pmatrix} w_0 \\ w_1 \\ \vdots \\ w_{M-1} \end{pmatrix} \text{and } \boldsymbol{\phi} = \begin{pmatrix} \phi_0 \\ \phi_1 \\ \vdots \\ \phi_{M-1} \end{pmatrix}$$





$$y(\mathbf{x},\mathbf{w}) = w_0 + w_1 x_1 + w_2 x_2 + \ldots + w_D x_D$$

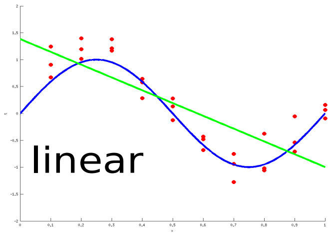
$$y(x,\mathbf{w}) = w_0 + w_1 x^1 + w_2 x^2 + \ldots + w_{M-1} x^{M-1}$$

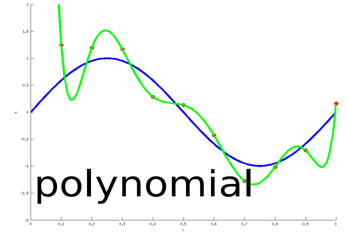
From Bishop

Different basis functions

$y(\mathbf{x}, \mathbf{w}) = \sum_{j=0}^{M-1} w_j \phi_j(\mathbf{x}) = \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x})$ where $\{\phi_j(\mathbf{x})\}$ are **basis functions**

$$\mathbf{w} = \begin{pmatrix} w_0 \\ w_1 \\ \vdots \\ w_{M-1} \end{pmatrix} \text{ and } \boldsymbol{\phi} = \begin{pmatrix} \phi_0 \\ \phi_1 \\ \vdots \\ \phi_{M-1} \end{pmatrix}$$



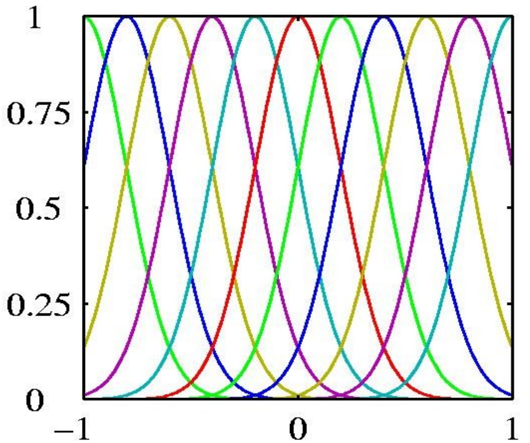


$$y(\mathbf{x}, \mathbf{w}) = w_0 + w_1 x_1 + w_2 x_2 + \ldots + w_D x_D$$

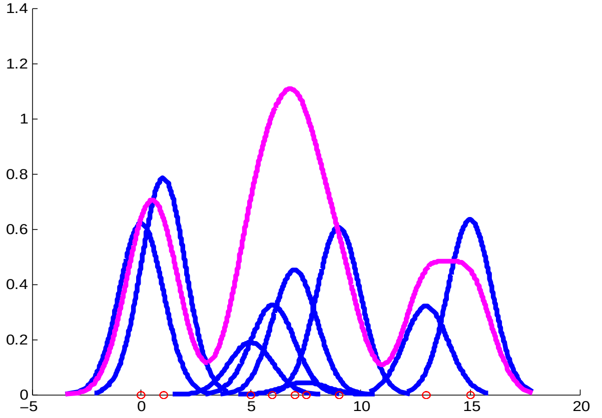
$$y(x, \mathbf{w}) = w_0 + w_1 x^1 + w_2 x^2 + \ldots + w_{M-1} x^{M-1}$$

$$y(x, \mathbf{w}) = w_0 + w_1 e^{-\frac{1}{2s^2}(x-x_1)^2} + \ldots + w_{M-1} e^{-\frac{1}{2s^2}(x-x_{M-1})^2}$$

Radial basis functions (Gaussians)

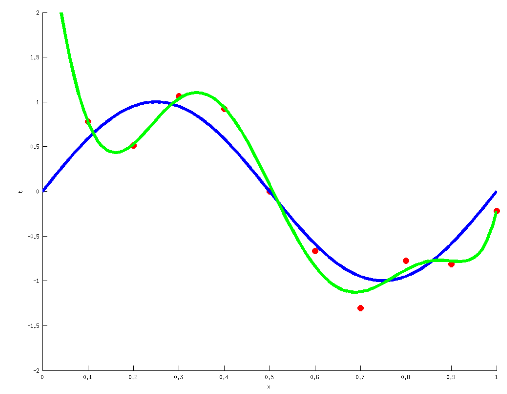


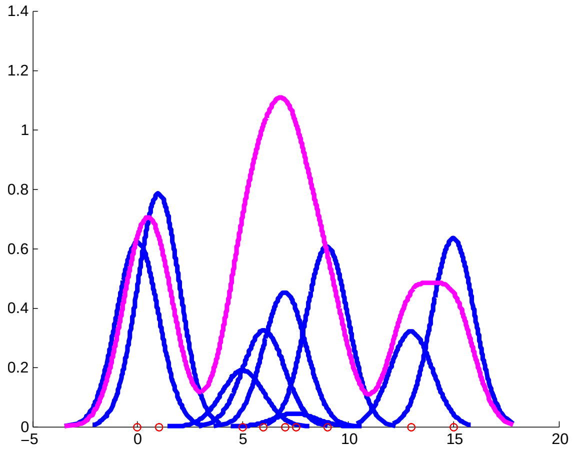
From Bishop



Basis functions: Global versus local effect

- **Polynomials** fits data globally: Change a parameter and it has effect globally by changing the whole curve.
- The **radial basis functions** fits data locally: Changing a parameter changes the basis weight locally and only changes the curve locally. Have infinite support (will cause very small changes far away).
- Splines (piecewise polynomials) fit data locally: Changing a parameter only affects the curve locally (in the region of the local polynomial).





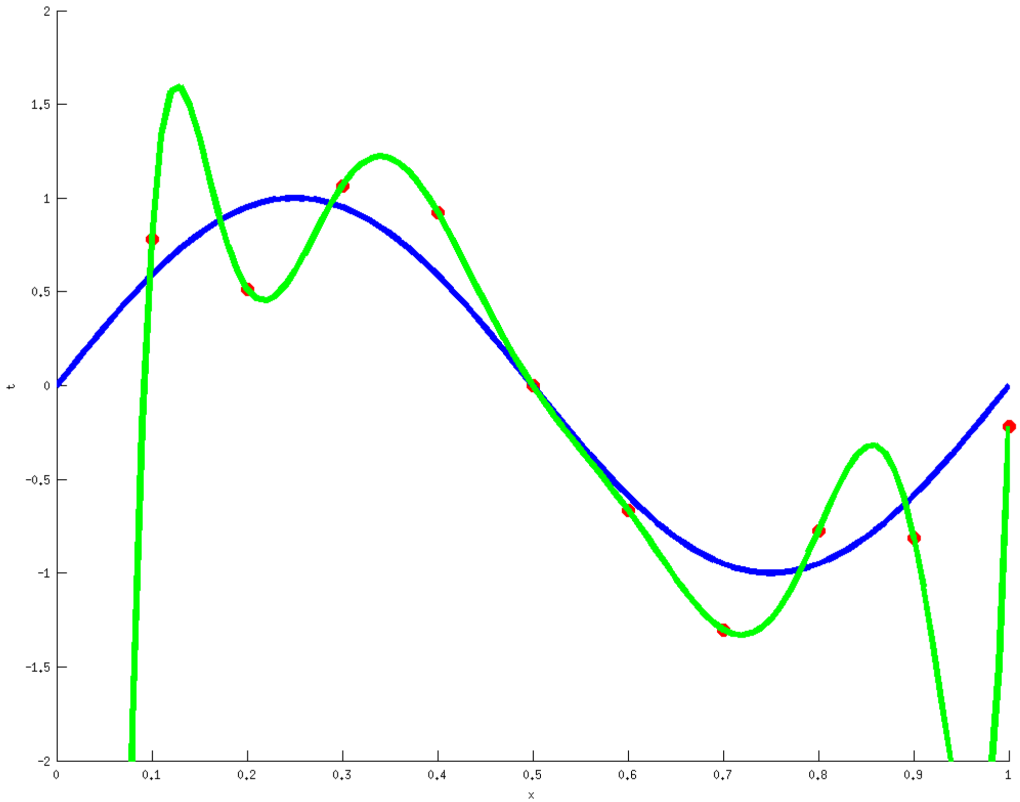
Curse of Dimensionality

- D-dimensional polynomial curve fitting, M = 3:

$$\mathrm{y}(\mathbf{x},\,\mathbf{w}) = \mathrm{w}_0 + \sum_{i=1}^{D} w_i x_i + \sum_{i=1}^{D}\sum_{j=1}^{D} w_{ij} x_i x_j + \sum_{i=1}^{D}\sum_{j=1}^{D}\sum_{k=1}^{D} w_{ijk} x_i x_j x_k$$

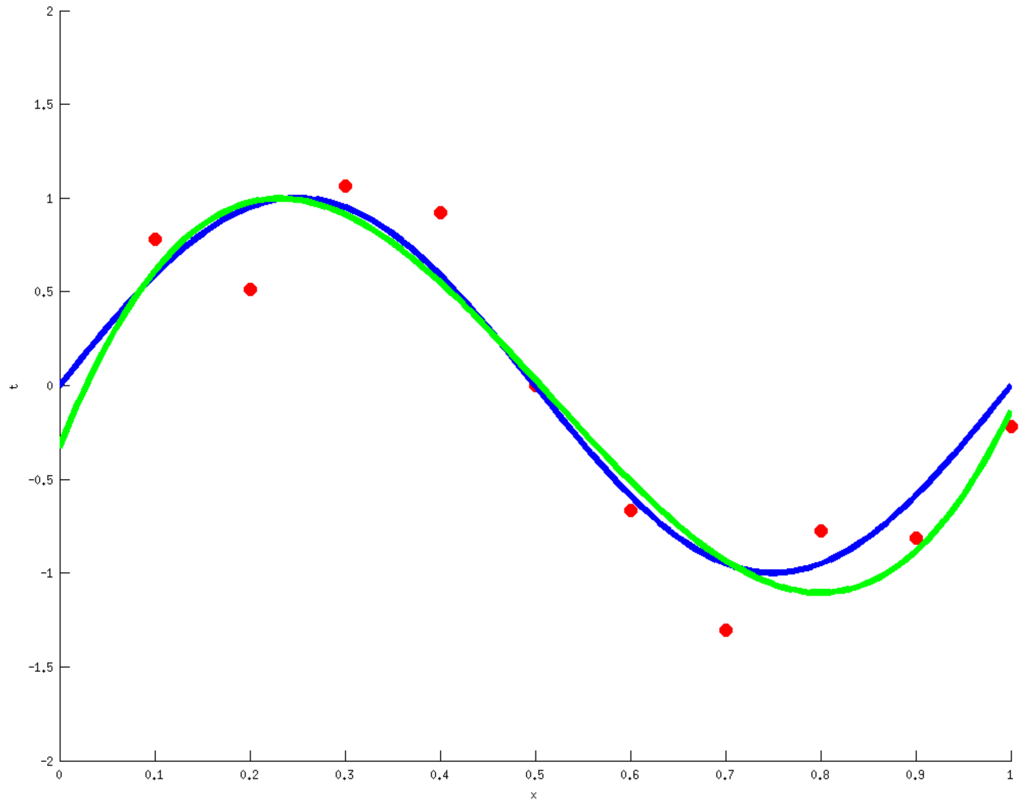
- **In general:**
  - Number of free model parameters grows polynomially in D^M with the dimensionality D.
  - The data set size N should grow polynomially to keep same precision on parameter estimates.

Example from last time:



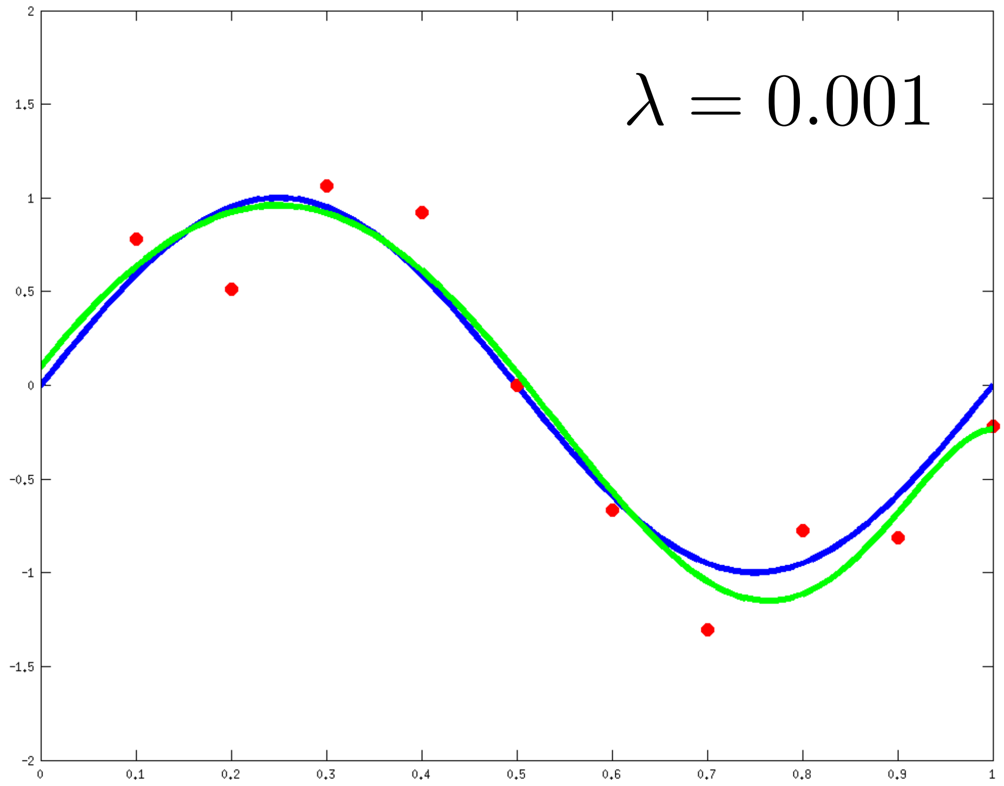
$$\mathrm{y}(\mathrm{x},\,\mathbf{w}) = \mathrm{w}_0 + w_1 x + w_2 x^2 + \ldots + w_9 x^9$$

Example from last time:



$$\mathrm{y}(\mathrm{x},\ \mathbf{w}) = \mathrm{w}_0 + w_1 x + w_2 x^2 + w_3 x^3$$

Example from last time: Regularization



$$y(x,\ \mathbf{w}) = w_0 + w_1 x + w_2 x^2 + \ldots + w_9 x^9$$

$$E(\mathbf{w}) = \sum_{n=1}^{N} (y(x_n, \mathbf{w}) - t_n)^2 + \lambda \|\mathbf{w}\|^2$$

Solving the regularized regression problem

Minimizing $\sum_{n=1}^{N}(y(\mathbf{x}_n,\mathbf{w})-t_n)^2+\lambda\|\mathbf{w}\|^2$ **when** $y(\mathbf{x},\mathbf{w})=\mathbf{w}^T\boldsymbol{\phi}(\mathbf{x})$.

$$\frac{\partial}{\partial w_i}\Big[\sum_{n=1}^{N}(\mathbf{w}^T\boldsymbol{\phi}(\mathbf{x}_n)-t_n)^2+\lambda\mathbf{w}^\mathbf{T}\mathbf{w}\Big]$$
$$=\sum_{n=1}^{N}\frac{\partial}{\partial w_i}\Big[(\mathbf{w}^T\boldsymbol{\phi}(\mathbf{x}_n)-t_n)^2+\lambda\mathbf{w}^\mathbf{T}\mathbf{w}\Big]$$
$$=\sum_{n=1}^{N}2(\mathbf{w}^T\boldsymbol{\phi}(\mathbf{x}_n)-t_n)\cdot\frac{\partial}{\partial w_i}(\mathbf{w}^T\boldsymbol{\phi}(\mathbf{x}_n)-t_n)+2\lambda w_i$$
$$=2\sum_{n=1}^{N}(\mathbf{w}^T\boldsymbol{\phi}(\mathbf{x}_n)-t_n)\cdot\phi_i(\mathbf{x}_n)+2\lambda w_i=0\quad\text{for all } i$$

Since $\boldsymbol{\phi}(\bar{x})^T=(\phi_0(\mathbf{x}),\phi_1(\mathbf{x}),\ldots,\phi_{M-1}(\mathbf{x}))$, we get

$$\sum_{n=1}^{N}\mathbf{w}^T\boldsymbol{\phi}(\mathbf{x}_n)\boldsymbol{\phi}(\mathbf{x}_n)^T-\sum_{n=1}^{N}t_n\boldsymbol{\phi}(\mathbf{x}_n)^T+\lambda\mathbf{w}^T=0,$$

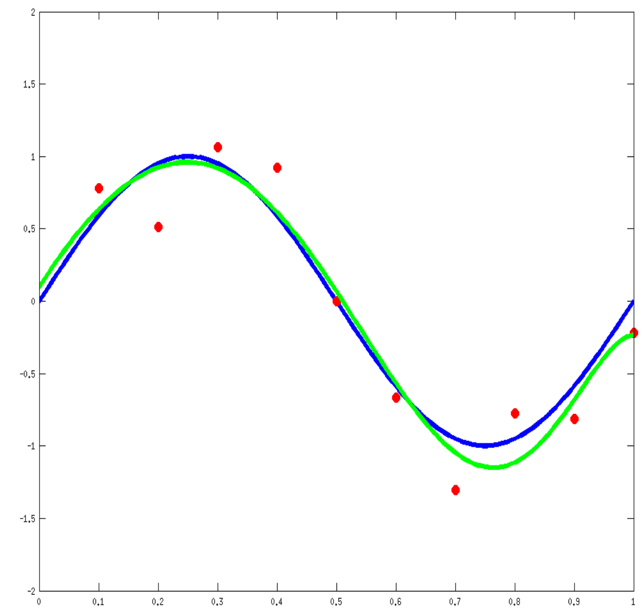
or

$$0=\mathbf{w}^T\sum_{n=1}^{N}\boldsymbol{\phi}(\mathbf{x}_n)\boldsymbol{\phi}(\mathbf{x}_n)^T-\sum_{n=1}^{N}t_n\boldsymbol{\phi}(\mathbf{x}_n)^T+\lambda\mathbf{w}^T(*)$$

$$\text{Setting }\Phi=\begin{pmatrix}\phi_0(\mathbf{x}_1) & \phi_1(\mathbf{x}_1) & \ldots & \phi_{M-1}(\mathbf{x}_1)\\ \phi_0(\mathbf{x}_2) & \phi_1(\mathbf{x}_2) & \ldots & \phi_{M-1}(\mathbf{x}_2)\\ \vdots & & & \vdots\\ \phi_0(\mathbf{x}_N) & \phi_1(\mathbf{x}_N) & \ldots & \phi_{M-1}(\mathbf{x}_N)\end{pmatrix}$$

we rewrite $(*)$ as $0=\mathbf{w}^T(\boldsymbol{\Phi}^T\boldsymbol{\Phi})-\mathbf{t}^T\boldsymbol{\Phi}+\lambda\mathbf{w}^T$

$$\Rightarrow\mathbf{w}^T(\boldsymbol{\Phi}^T\boldsymbol{\Phi}+\lambda I)=\mathbf{t}^T\boldsymbol{\Phi}$$
$$\Rightarrow(\boldsymbol{\Phi}^T\boldsymbol{\Phi}+\lambda\mathbf{I})^T\mathbf{w}=\boldsymbol{\Phi}^T\mathbf{t}$$
$$\Rightarrow(\boldsymbol{\Phi}^T\boldsymbol{\Phi}+\lambda\mathbf{I})\mathbf{w}=\boldsymbol{\Phi}^T\mathbf{t}$$
$$\Rightarrow\mathbf{w}=(\boldsymbol{\Phi}^T\boldsymbol{\Phi}+\lambda\mathbf{I})^{-1}\boldsymbol{\Phi}^T\mathbf{t}$$



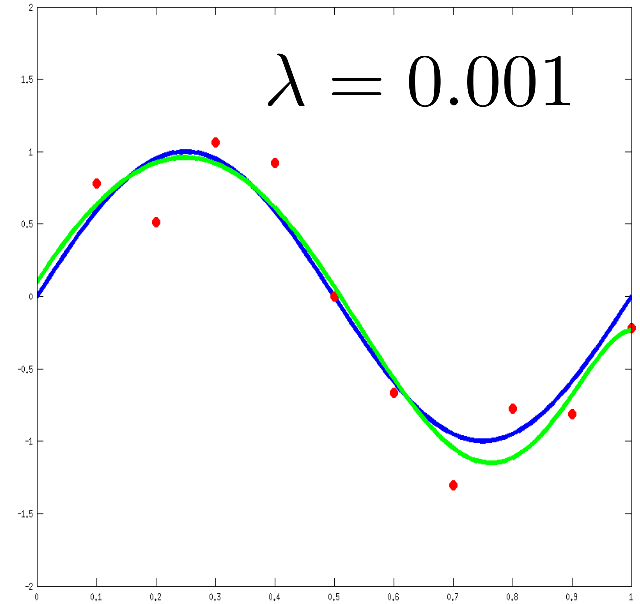
$$\mathrm{y}(\mathrm{x},\mathbf{w})=\mathrm{w}_0+w_1x+w_2x^2+\ldots+w_9x^9$$
$$\mathrm{E}(\mathbf{w})=\sum_{n=1}^{N}(y(x_n,\mathbf{w})-t_n)^2+\lambda\|\mathbf{w}\|^2$$

OBS! Also stabilizes the matrix inversion...

Regularization

- Adding an L2 punishment of the weight vector is referred to as *ridge regression*
- Drives weights towards small norm
- **Interpretation of weights:**

  Tell you about the importance of each basis function for describing the data *(this interpretation is a heuristic!)*



$$y(x, \mathbf{w}) = w_0 + w_1\phi_1(x) + w_2\phi_2(x) + \ldots + w_9\phi_9(x)$$

$$E(\mathbf{w}) = \sum_{n=1}^{N}(y(x_n, \mathbf{w}) - t_n)^2 + \lambda\|\mathbf{w}\|^2$$

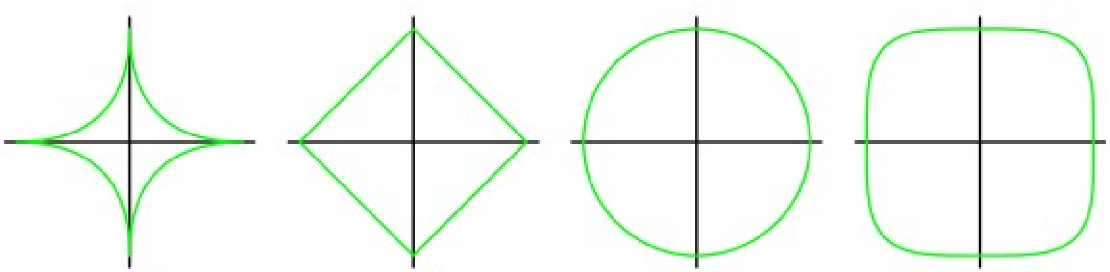
Regularization

- More generally, regularization can be done by adding a term of degree q

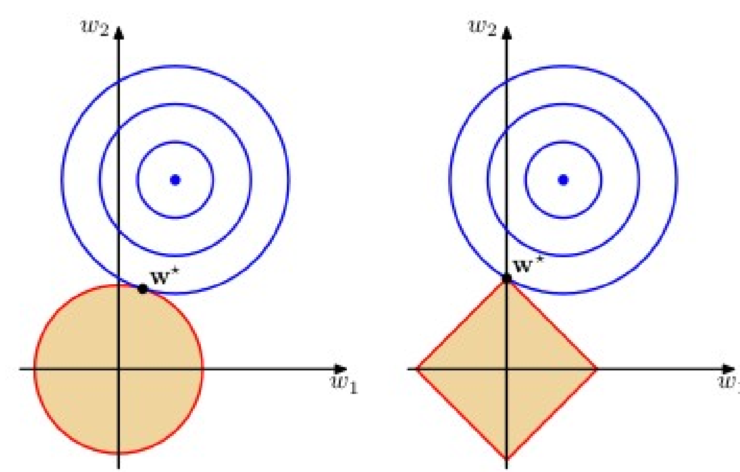
$$\mathrm{E}(\mathbf{w}) = \sum_{n=1}^{N} (y(x_n, \mathbf{w}) - t_n)^2 + \lambda \|\mathbf{w}\|^q$$

- When q = 1, this is called the *lasso*.
- For q = 1 or smaller, minimization will prefer weights = 0
- Interpretation of weights:
  - Supervised **dimensionality reduction / feature selection**
  - Importance of different basis functions

Curse of dim

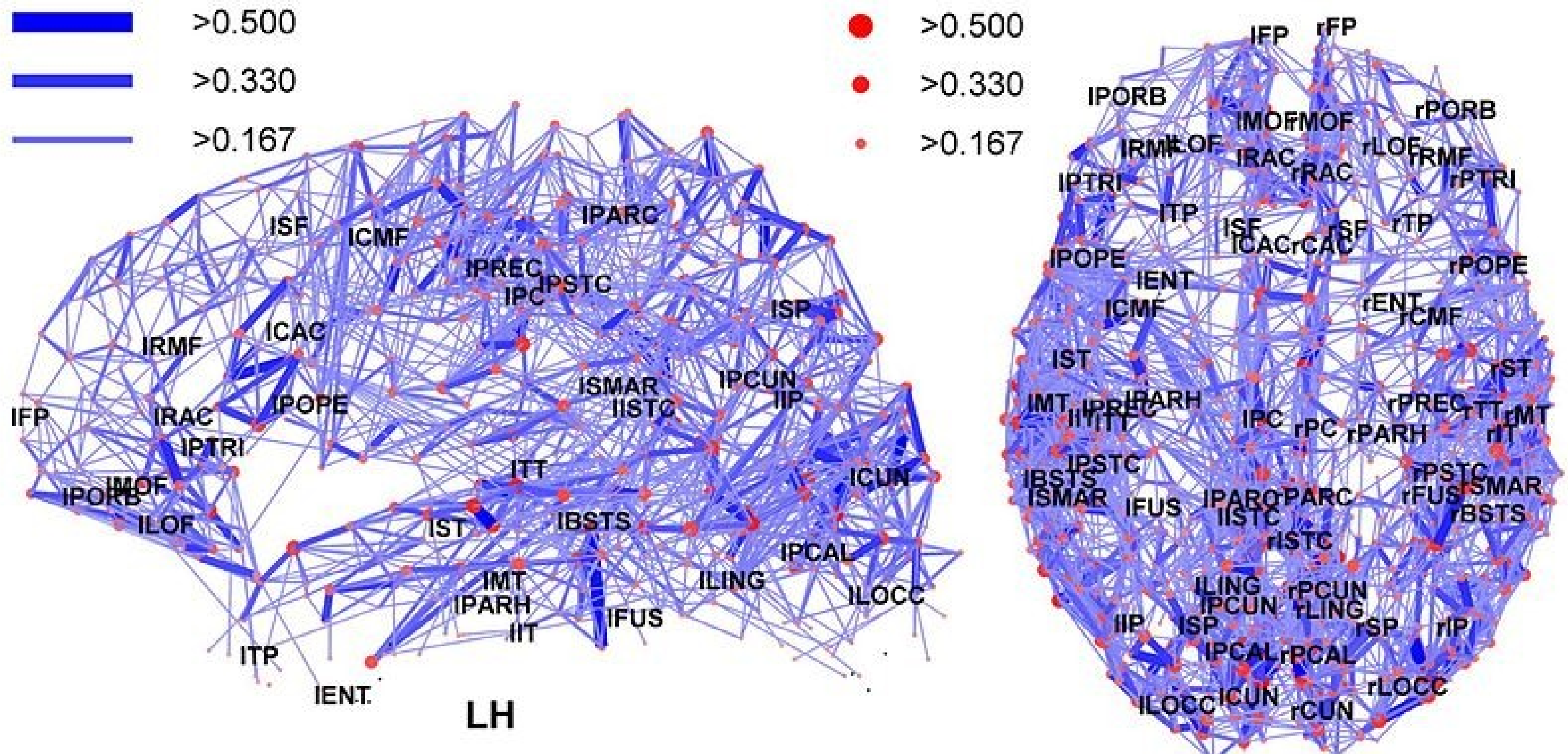


q = 0.5 | q = 1 | q = 2 | q = 4



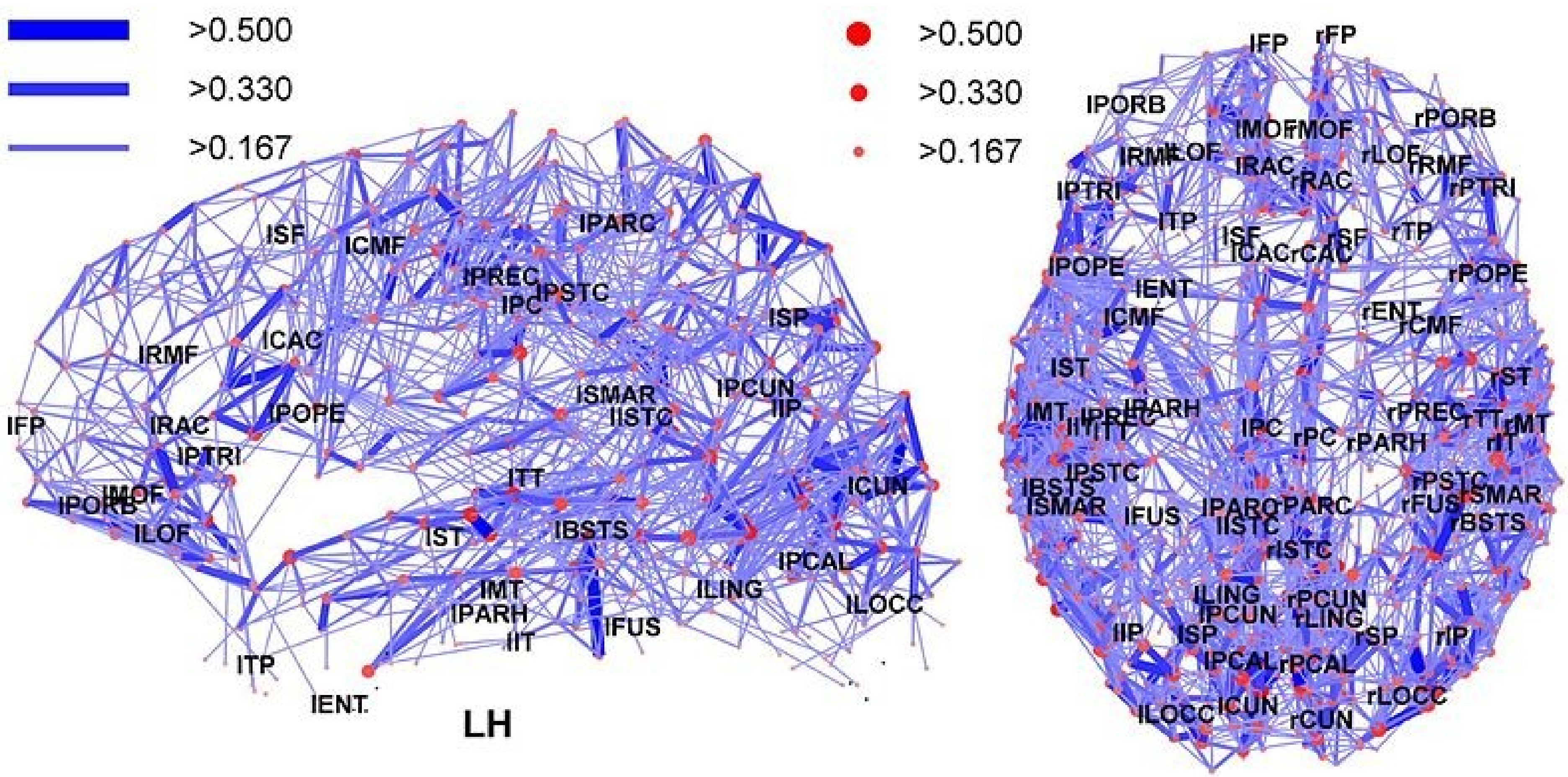
From Bishop

Example: Sparse regression for brain connectivity



$$\mathrm{E}(\mathbf{w}) = \sum_{n=1}^{N} \left(\sum_{e \in E} w_e \phi_e - t_n\right)^2 + \lambda \|\mathbf{w}\|$$

Example: Structured sparse regression for brain connectivity



$$\mathrm{E}(\mathbf{w}) = \sum_{n=1}^{N} \left(\sum_{e \in E} w_e \phi_e - t_n\right)^2 + \lambda \|\mathbf{w}\| + \sum_{G \subset E} \lambda_2 \|\mathbf{w}_G\|^2$$

Image: Hagmann P, Cammoun L, Gigandet X, Meuli R, Honey CJ, Wedeen VJ, Sporns O, CC-BY

Recall from Lecture 2:

- Maximum Likelihood estimates

Find the model parameters

$$\mathbf{w}$$

that maximize the joint probability

$$\mathrm{p}(\mathrm{D} \mid \mathbf{w})$$

of observing the data given the model

- Maximum a posteriori estimates

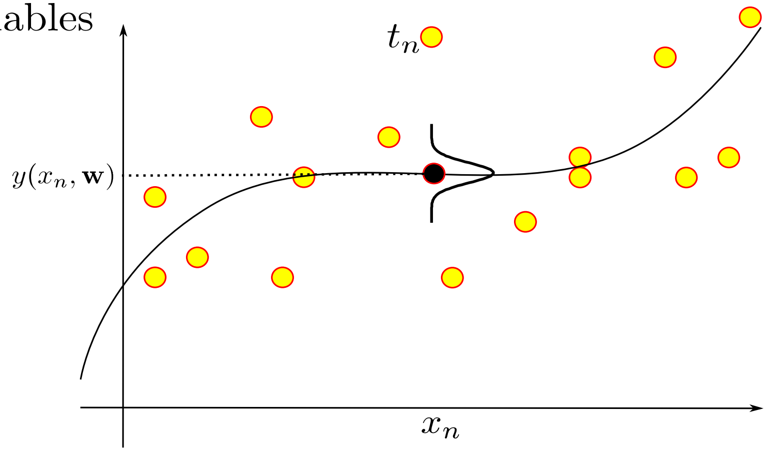
Find the most likely model parameters given the data, that is find the model parameters

$$\mathrm{p}(\mathbf{w} \mid \mathrm{D}) \propto p(D|\mathbf{w})p(\mathbf{w})$$

Bayesian regression: Maximum a Posteriori solution

N input variables with target variables

$$\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix} \qquad \mathbf{t} = \begin{pmatrix} t_1 \\ t_2 \\ \vdots \\ t_N \end{pmatrix}$$



Assume: Gaussian noise model

$$\mathcal{N}(t|y(x, \mathbf{w}), \beta^{-1})$$

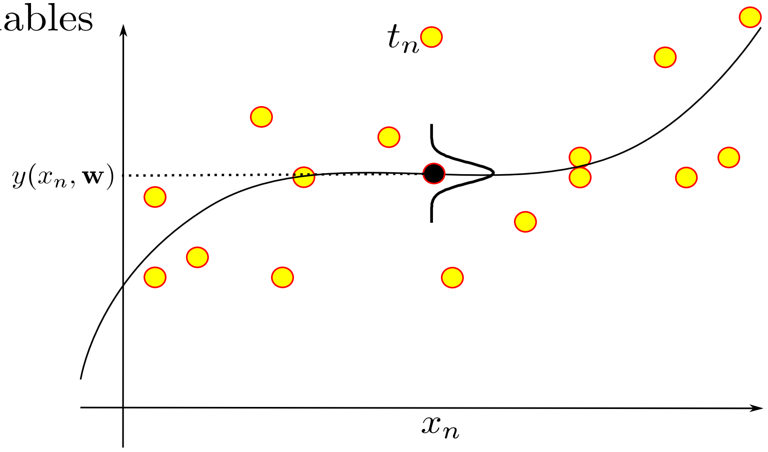
Likelihood of data $\mathbf{t}$ under model fixed by $\mathbf{w}, \mathbf{x}$

$$p(\mathbf{t}|\mathbf{x}, \mathbf{w}, \beta) = \prod_{n=1}^{N} \mathcal{N}(t_n|y(x_n, \mathbf{w}), \beta^{-1})$$

Bayesian regression: Maximum a Posteriori solution

N input variables with target variables

$$\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix} \qquad \mathbf{t} = \begin{pmatrix} t_1 \\ t_2 \\ \vdots \\ t_N \end{pmatrix}$$



Assume: Gaussian noise model

$$\mathcal{N}(t|y(x,\mathbf{w}),\beta^{-1})$$

Likelihood of data $\mathbf{t}$ under model fixed by $\mathbf{w}, \mathbf{x}$

$$p(\mathbf{t}|\mathbf{x},\mathbf{w},\beta) = \prod_{n=1}^{N} \mathcal{N}(t_n|y(x_n,\mathbf{w}),\beta^{-1})$$

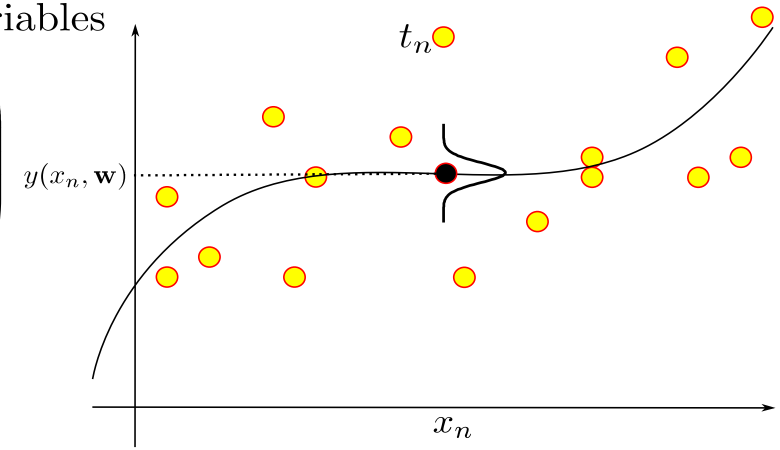
Conjugate prior:

$$p(\mathbf{w}) = \mathcal{N}(\mathbf{w}|\mathbf{m}_0,\mathbf{S}_0)$$

Bayesian regression: Maximum a Posteriori solution

N input variables with target variables

$$\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix} \qquad \mathbf{t} = \begin{pmatrix} t_1 \\ t_2 \\ \vdots \\ t_N \end{pmatrix}$$



Assume: Gaussian noise model

$$\mathcal{N}(t|y(x, \mathbf{w}), \beta^{-1})$$

Likelihood of data $\mathbf{t}$ under model fixed by $\mathbf{w}, \mathbf{x}$

$$p(\mathbf{t}|\mathbf{x}, \mathbf{w}, \beta) = \prod_{n=1}^{N} \mathcal{N}(t_n|y(x_n, \mathbf{w}), \beta^{-1})$$

Conjugate prior:

$$p(\mathbf{w}) = \mathcal{N}(\mathbf{w}|\mathbf{m}_0, \mathbf{S}_0)$$

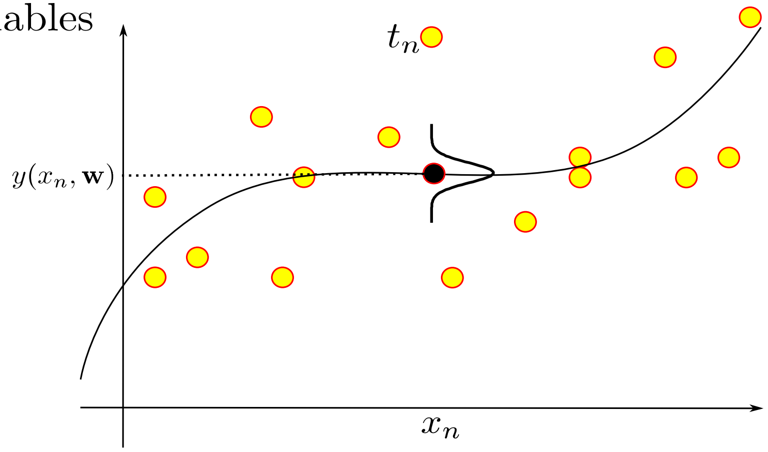
Posterior distribution:

$$p(\mathbf{w}|\mathbf{t}) = p(\mathbf{t}|\mathbf{w})p(\mathbf{w}) = \prod_{n=1}^{N} \mathcal{N}(t_n|y(x_n, \mathbf{w}), \beta^{-1}) \cdot \mathcal{N}(\mathbf{w}|\mathbf{m}_0, \mathbf{S}_0)$$

Bayesian regression: Maximum a Posteriori solution

N input variables with target variables

$$\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix} \qquad \mathbf{t} = \begin{pmatrix} t_1 \\ t_2 \\ \vdots \\ t_N \end{pmatrix}$$



Assume: Gaussian noise model

$$\mathcal{N}(t|y(x, \mathbf{w}), \beta^{-1})$$

Likelihood of data $\mathbf{t}$ under model fixed by $\mathbf{w}, \mathbf{x}$

$$p(\mathbf{t}|\mathbf{x}, \mathbf{w}, \beta) = \prod_{n=1}^{N} \mathcal{N}(t_n|y(x_n, \mathbf{w}), \beta^{-1})$$

Conjugate prior:

$$p(\mathbf{w}) = \mathcal{N}(\mathbf{w}|\mathbf{m}_0, \mathbf{S}_0)$$

Posterior distribution:

$$p(\mathbf{w}|\mathbf{t}) = p(\mathbf{t}|\mathbf{w})p(\mathbf{w}) = \prod_{n=1}^{N} \mathcal{N}(t_n|y(x_n, \mathbf{w}), \beta^{-1}) \cdot \mathcal{N}(\mathbf{w}|\mathbf{m}_0, \mathbf{S}_0)$$

$$= \mathcal{N}(\mathbf{w}|\mathbf{m}_N, \mathbf{S}_N)$$ (product of Gaussians)

where

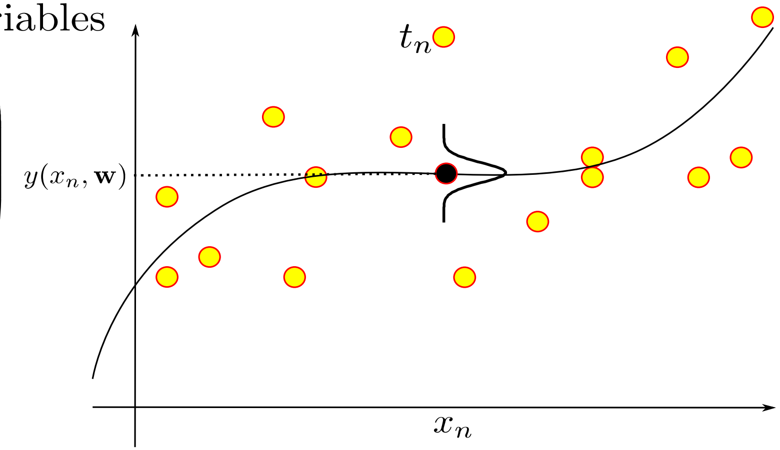
$$\mathbf{m}_N = \mathbf{S}_N(\mathbf{S}_0^{-1}\mathbf{m}_0 + \beta\mathbf{\Phi}^T\mathbf{t}) \qquad \mathbf{S}_N^{-1} = \mathbf{S}_0^{-1} + \beta\mathbf{\Phi}^T\mathbf{\Phi}$$

(See CB for proof)

Bayesian regression: Maximum a Posteriori solution

N input variables with target variables

$$\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix} \qquad \mathbf{t} = \begin{pmatrix} t_1 \\ t_2 \\ \vdots \\ t_N \end{pmatrix}$$



Assume: Gaussian noise model

$$\mathcal{N}(t|y(x, \mathbf{w}), \beta^{-1})$$

Likelihood of data $\mathbf{t}$ under model fixed by $\mathbf{w}, \mathbf{x}$

$$p(\mathbf{t}|\mathbf{x}, \mathbf{w}, \beta) = \prod_{n=1}^{N} \mathcal{N}(t_n|y(x_n, \mathbf{w}), \beta^{-1})$$

Conjugate prior:

$$p(\mathbf{w}) = \mathcal{N}(\mathbf{w}|\mathbf{m}_0, \mathbf{S}_0)$$

Posterior distribution:

$$p(\mathbf{w}|\mathbf{t}) = p(\mathbf{t}|\mathbf{w})p(\mathbf{w}) = \prod_{n=1}^{N} \mathcal{N}(t_n|y(x_n, \mathbf{w}), \beta^{-1}) \cdot \mathcal{N}(\mathbf{w}|\mathbf{m}_0, \mathbf{S}_0)$$

$$= \mathcal{N}(\mathbf{w}|\mathbf{m}_N, \mathbf{S}_N)$$ (product of Gaussians)

where

$$\mathbf{m}_N = \mathbf{S}_N(\mathbf{S}_0^{-1}\mathbf{m}_0 + \beta\mathbf{\Phi}^T\mathbf{t}) \qquad \mathbf{S}_N^{-1} = \mathbf{S}_0^{-1} + \beta\mathbf{\Phi}^T\mathbf{\Phi}$$

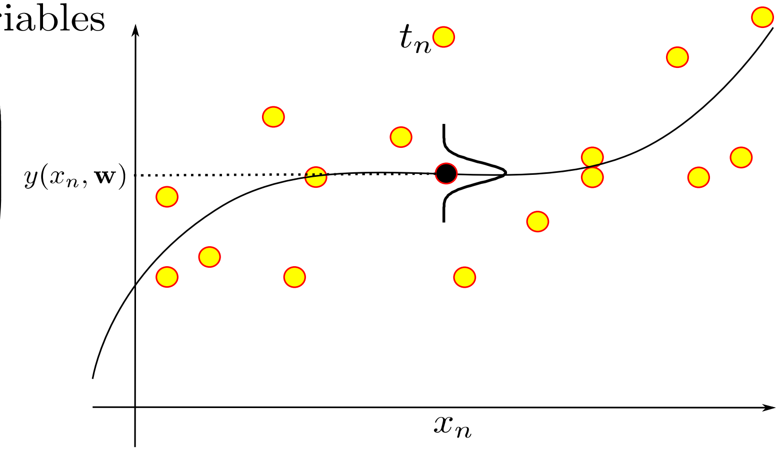
(See CB for proof)

$\mathbf{w}_{MAP} = \mathbf{m}_N$ since $p(\mathbf{w}|\mathbf{t})$ is a (unimodal) Gaussian

Bayesian regression: Maximum a Posteriori solution

N input variables with target variables

$$\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix} \qquad \mathbf{t} = \begin{pmatrix} t_1 \\ t_2 \\ \vdots \\ t_N \end{pmatrix}$$



Assume: Gaussian noise model

$$\mathcal{N}(t|y(x, \mathbf{w}), \beta^{-1})$$

Likelihood of data $\mathbf{t}$ under model fixed by $\mathbf{w}, \mathbf{x}$

$$p(\mathbf{t}|\mathbf{x}, \mathbf{w}, \beta) = \prod_{n=1}^{N} \mathcal{N}(t_n|y(x_n, \mathbf{w}), \beta^{-1})$$

$$\Phi = \begin{pmatrix} \phi_0(\mathrm{x}_1) & \phi_1(\mathrm{x}_1) & \ldots & \phi_{M-1}(\mathrm{x}_1) \\ \phi_0(\mathrm{x}_2) & \phi_1(\mathrm{x}_2) & \ldots & \phi_{M-1}(\mathrm{x}_2) \\ \vdots & & & \vdots \\ \phi_0(\mathrm{x}_N) & \phi_1(\mathrm{x}_N) & \ldots & \phi_{M-1}(\mathrm{x}_N) \end{pmatrix}$$

Conjugate prior:

$$p(\mathbf{w}) = \mathcal{N}(\mathbf{w}|\mathbf{m}_0, \mathbf{S}_0)$$

Posterior distribution:

$$p(\mathbf{w}|\mathbf{t}) = p(\mathbf{t}|\mathbf{w})p(\mathbf{w}) = \prod_{n=1}^{N} \mathcal{N}(t_n|y(x_n, \mathbf{w}), \beta^{-1}) \cdot \mathcal{N}(\mathbf{w}|\mathbf{m}_0, \mathbf{S}_0)$$

$$= \mathcal{N}(\mathbf{w}|\mathbf{m}_N, \mathbf{S}_N)$$ (product of Gaussians)

Design matrix

where

$$\mathbf{m}_N = \mathbf{S}_N(\mathbf{S}_0^{-1}\mathbf{m}_0 + \beta\mathbf{\Phi}^T\mathbf{t}) \qquad \mathbf{S}_N^{-1} = \mathbf{S}_0^{-1} + \beta\mathbf{\Phi}^T\mathbf{\Phi}$$

(See CB for proof)

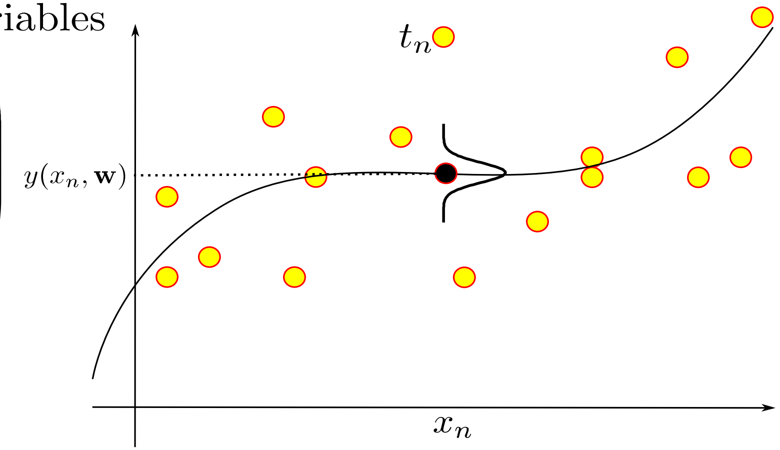
$\mathbf{w}_{MAP} = \mathbf{m}_N$ since $p(\mathbf{w}|\mathbf{t})$ is a (unimodal) Gaussian

Analytic solution

Effect of the prior

N input variables with target variables

$$\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix} \qquad \mathbf{t} = \begin{pmatrix} t_1 \\ t_2 \\ \vdots \\ t_N \end{pmatrix}$$



Assume: Gaussian noise model

$$\mathcal{N}(t|y(x,\mathbf{w}), \beta^{-1})$$

Likelihood of data $\mathbf{t}$ under model fixed by $\mathbf{w}, \mathbf{x}$

$$p(\mathbf{t}|\mathbf{x},\mathbf{w},\beta) = \prod_{n=1}^{N} \mathcal{N}(t_n|y(x_n,\mathbf{w}), \beta^{-1})$$

Conjugate prior:

$$p(\mathbf{w}) = \mathcal{N}(\mathbf{w}|\mathbf{m}_0, \mathbf{S}_0)$$

Posterior distribution: $\mathcal{N}(\mathbf{w}|\mathbf{m}_N, \mathbf{S}_N)$ where

$$\mathbf{m}_N = \mathbf{S}_N(\mathbf{S}_0^{-1}\mathbf{m}_0 + \beta\boldsymbol{\Phi}^T\mathbf{t}) \qquad \mathbf{S}_N^{-1} = (\mathbf{S}_0^{-1} + \beta\boldsymbol{\Phi}^T\boldsymbol{\Phi})$$

$$\Phi = \begin{pmatrix} \phi_0(\mathrm{x}_1) & \phi_1(\mathrm{x}_1) & \dots & \phi_{M-1}(\mathrm{x}_1) \\ \phi_0(\mathrm{x}_2) & \phi_1(\mathrm{x}_2) & \dots & \phi_{M-1}(\mathrm{x}_2) \\ \vdots & & & \vdots \\ \phi_0(\mathrm{x}_N) & \phi_1(\mathrm{x}_N) & \dots & \phi_{M-1}(\mathrm{x}_N) \end{pmatrix}$$

Design matrix

Effect of prior:

If $\mathbf{S}_0 = \alpha^{-1}I$ with $\alpha \to 0$, then $\mathbf{m}_N \to \mathbf{w}_{ML} = (\boldsymbol{\phi}^T\boldsymbol{\phi})^{-1}\boldsymbol{\phi}^T\mathbf{t}$

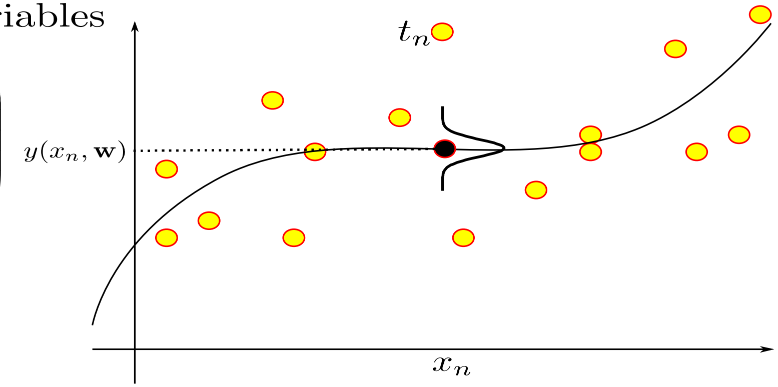
If $N = 0$, then $\mathbf{m}_N = \mathbf{m}_0$

Analytic solution

Relation to Maximum Likelihood

N input variables with target variables

$$\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix} \qquad \mathbf{t} = \begin{pmatrix} t_1 \\ t_2 \\ \vdots \\ t_N \end{pmatrix}$$



Assume: Gaussian noise model

$$\mathcal{N}(t|y(x, \mathbf{w}), \beta^{-1})$$

Likelihood of data $\mathbf{t}$ under model fixed by $\mathbf{w}, \mathbf{x}$

$$p(\mathbf{t}|\mathbf{x}, \mathbf{w}, \beta) = \prod_{n=1}^{N} \mathcal{N}(t_n|y(x_n, \mathbf{w}), \beta^{-1})$$

Conjugate prior:

$$p(\mathbf{w}) = \mathcal{N}(\mathbf{w}|\mathbf{0}, \alpha\mathbf{I})$$

Maximize the posterior:

$$\begin{aligned}
&\underset{\mathbf{w}}{\operatorname{argmax}}\ p(\mathbf{w}|\mathbf{x}, \mathbf{t}, \alpha, \beta) \\
&= \underset{\mathbf{w}}{\operatorname{argmin}}\ -\ln(p(\mathbf{w}|\mathbf{x}, \mathbf{t}, \alpha, \beta)) \\
&= \underset{\mathbf{w}}{\operatorname{argmin}}\ -\ln(p(\mathbf{t}|\mathbf{w}, \mathbf{x}, \alpha, \beta)p(\mathbf{w})) \\
&= \underset{\mathbf{w}}{\operatorname{argmin}}\ [-\ln(p(\mathbf{t}|\mathbf{w}, \mathbf{x}, \alpha, \beta)) - \ln(p(\mathbf{w}))] \\
&= \underset{\mathbf{w}}{\operatorname{argmin}}\ [-\sum_{n=1}^{N}(y(x_n, \mathbf{w}) - t_n)^2 + \frac{\alpha}{2}\|\mathbf{w}\|^2 + const\ ]
\end{aligned}$$

So adding a prior in the MAP estimate is equivalent to adding a regularizer in the ML estimate

Sequential learning

Assume: measurements are arriving sequentially

$$(\mathbf{x}_1, t_1), (\mathbf{x}_2, t_2), \ldots, (\mathbf{x}_N, t_N)$$

Want to learn "on the go" – e.g.
- tracking
- personalized models
- etc

Sequential learning

Assume: measurements are arriving sequentially

$$(\mathbf{x}_1, t_1), (\mathbf{x}_2, t_2), \ldots, (\mathbf{x}_N, t_N)$$

Want to learn "on the go" – e.g.

- tracking
- personalized models
- etc

$$p(\mathbf{w}|\mathbf{t}, \mathbf{x}, \beta) \propto \underbrace{p(\mathbf{t}|\mathbf{x}, \mathbf{w}, \beta)}_{\text{likelihood}} \underbrace{p(\mathbf{w})}_{\text{prior}} \quad \text{(Bayes)}$$

Sequential learning

Assume: measurements are arriving sequentially

$$(\mathbf{x}_1, t_1), (\mathbf{x}_2, t_2), \ldots, (\mathbf{x}_N, t_N)$$

Want to learn "on the go" – e.g.

tracking
personalized models
etc

$$p(\mathbf{w}|\mathbf{t}, \mathbf{x}, \beta) \propto \underbrace{p(\mathbf{t}|\mathbf{x}, \mathbf{w}, \beta)}_{\text{likelihood}} \underbrace{p(\mathbf{w})}_{\text{prior}} \quad \text{(Bayes)}$$

$$= \prod_{n=1}^{N} \mathcal{N}(t_n|\mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}_n), \beta^{-1}) \mathcal{N}(\mathbf{w}|\mathbf{m}_0, \mathbf{S}_0)$$

Sequential learning

Assume: measurements are arriving sequentially

$$(\mathbf{x}_1, t_1), (\mathbf{x}_2, t_2), \ldots, (\mathbf{x}_N, t_N)$$

Want to learn "on the go" – e.g. tracking
personalized models
etc

$$p(\mathbf{w}|\mathbf{t}, \mathbf{x}, \beta) \propto \underbrace{p(\mathbf{t}|\mathbf{x}, \mathbf{w}, \beta)}_{\text{likelihood}} \underbrace{p(\mathbf{w})}_{\text{prior}} \quad \text{(Bayes)}$$

$$= \prod_{n=1}^{N} \mathcal{N}(t_n|\mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}_n), \beta^{-1}) \mathcal{N}(\mathbf{w}|\mathbf{m}_0, \mathbf{S}_0)$$

$$= \mathcal{N}(t_N|\mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}_n), \beta^{-1}) \prod_{n=1}^{N-1} \mathcal{N}(t_n|\mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}_n), \beta^{-1}) \mathcal{N}(\mathbf{w}|\mathbf{m}_0, \mathbf{S}_0)$$

Sequential learning

Assume: measurements are arriving sequentially

$$(\mathbf{x}_1, t_1), (\mathbf{x}_2, t_2), \ldots, (\mathbf{x}_N, t_N)$$

Want to learn "on the go" – e.g.

tracking
personalized models
etc

$$p(\mathbf{w}|\mathbf{t}, \mathbf{x}, \beta) \propto \underbrace{p(\mathbf{t}|\mathbf{x}, \mathbf{w}, \beta)}_{\text{likelihood}} \underbrace{p(\mathbf{w})}_{\text{prior}} \quad \text{(Bayes)}$$

$$= \prod_{n=1}^{N} \mathcal{N}(t_n|\mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}_n), \beta^{-1}) \mathcal{N}(\mathbf{w}|\mathbf{m}_0, \mathbf{S}_0)$$

$$= \mathcal{N}(t_N|\mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}_n), \beta^{-1}) \prod_{n=1}^{N-1} \mathcal{N}(t_n|\mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}_n), \beta^{-1}) \mathcal{N}(\mathbf{w}|\mathbf{m}_0, \mathbf{S}_0)$$

$$= \underbrace{p(t_N|x_N, \mathbf{w}, \beta)}_{\text{likelihood for } t_N} \qquad \underbrace{p(\mathbf{w})}_{\text{original prior}} \qquad \underbrace{\prod_{n=1}^{N-1} p(t_n|\mathbf{x}_n, \mathbf{w}, \beta)}_{\text{likelihood of observing } t_1, \ldots, t_{N-1}}$$

Sequential learning

Assume: measurements are arriving sequentially

$$(\mathbf{x}_1, t_1), (\mathbf{x}_2, t_2), \ldots, (\mathbf{x}_N, t_N)$$

Want to learn "on the go" – e.g. tracking
personalized models
etc

$$p(\mathbf{w}|\mathbf{t}, \mathbf{x}, \beta) \propto \underbrace{p(\mathbf{t}|\mathbf{x}, \mathbf{w}, \beta)}_{\text{likelihood}} \underbrace{p(\mathbf{w})}_{\text{prior}} \quad \text{(Bayes)}$$

$$= \prod_{n=1}^{N} \mathcal{N}(t_n|\mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}_n), \beta^{-1}) \mathcal{N}(\mathbf{w}|\mathbf{m}_0, \mathbf{S}_0)$$

$$= \mathcal{N}(t_N|\mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}_n), \beta^{-1}) \prod_{n=1}^{N-1} \mathcal{N}(t_n|\mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}_n), \beta^{-1}) \mathcal{N}(\mathbf{w}|\mathbf{m}_0, \mathbf{S}_0)$$

$$= \underbrace{p(t_N|x_N, \mathbf{w}, \beta)}_{\text{likelihood for } t_N} \quad \underbrace{\underbrace{p(\mathbf{w})}_{\text{original prior}} \quad \underbrace{\prod_{n=1}^{N-1} p(t_n|\mathbf{x}_n, \mathbf{w}, \beta)}_{\text{likelihood of observing } t_1, \ldots, t_{N-1}}}_{\text{posterior for } N-1 \,=\, \text{prior for } N}$$

Sequential learning

Assume: measurements are arriving sequentially

$$(\mathbf{x}_1, t_1), (\mathbf{x}_2, t_2), \ldots, (\mathbf{x}_N, t_N)$$

Want to learn "on the go" – e.g.
tracking
personalized models
etc

$$p(\mathbf{w}|\mathbf{t}, \mathbf{x}, \beta) \propto \underbrace{p(\mathbf{t}|\mathbf{x}, \mathbf{w}, \beta)}_{\text{likelihood}} \underbrace{p(\mathbf{w})}_{\text{prior}} \quad \text{(Bayes)}$$

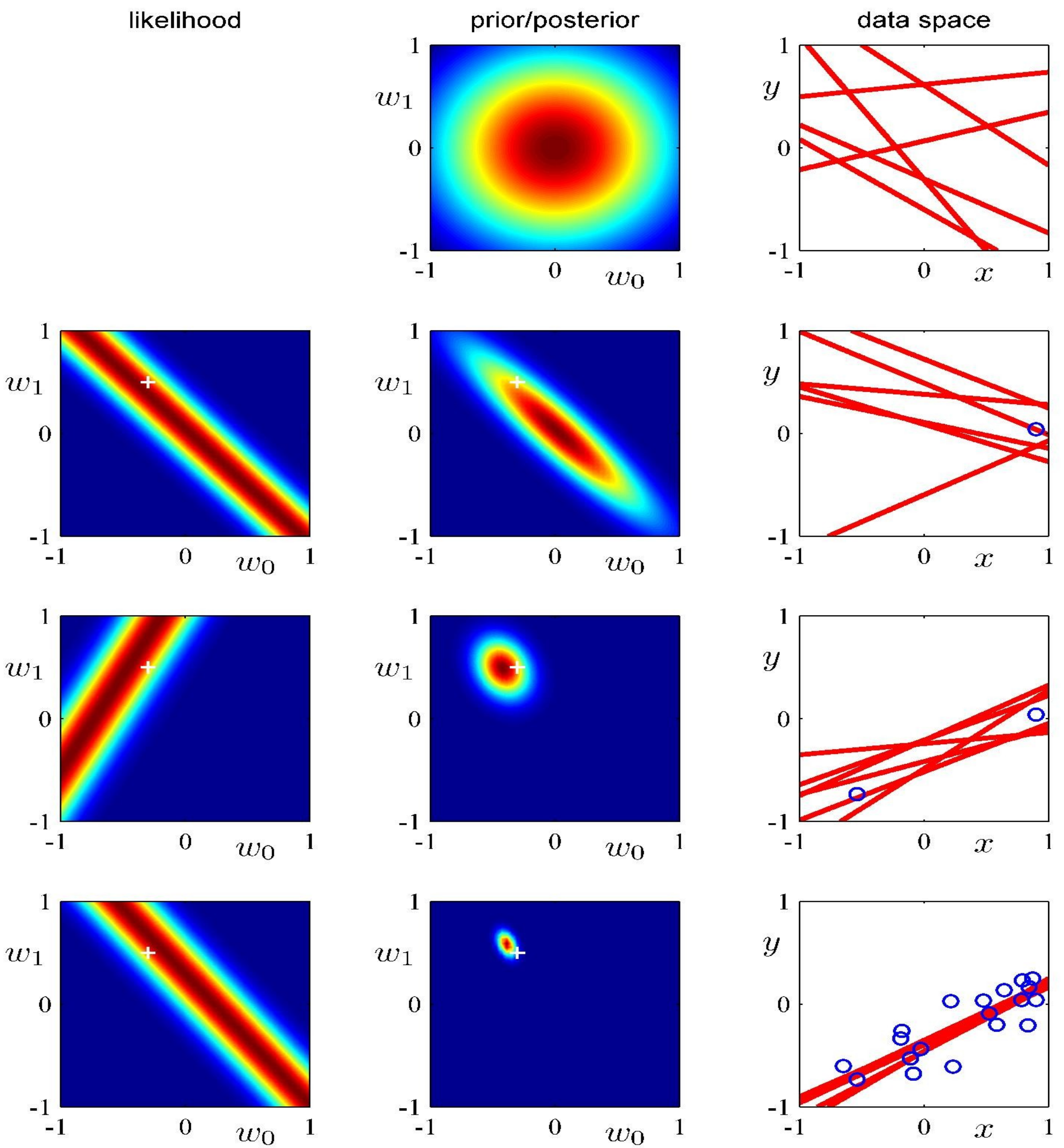
$$= \prod_{n=1}^{N} \mathcal{N}(t_n|\mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}_n), \beta^{-1}) \mathcal{N}(\mathbf{w}|\mathbf{m}_0, \mathbf{S}_0)$$

$$= \mathcal{N}(t_N|\mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}_n), \beta^{-1}) \prod_{n=1}^{N-1} \mathcal{N}(t_n|\mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}_n), \beta^{-1}) \mathcal{N}(\mathbf{w}|\mathbf{m}_0, \mathbf{S}_0)$$

$$= \underbrace{p(t_N|x_N, \mathbf{w}, \beta)}_{\text{likelihood for } t_N} \quad \underbrace{\underbrace{p(\mathbf{w})}_{\text{original prior}} \quad \underbrace{\prod_{n=1}^{N-1} p(t_n|\mathbf{x}_n, \mathbf{w}, \beta)}_{\text{likelihood of observing } t_1, \ldots, t_{N-1}}}_{\text{posterior for } N-1 \,=\, \text{prior for } N}$$

Posterior for $N-1$ acts as prior for N

Sequential learning
Example from CB



Summary

Three views on regression

Geometric least squares

Minimize $\sum_{n=1}^{N}(y(\mathbf{x}_n, \mathbf{w}) - t_n)^2$

(Summary of ML/MAP under assumption of i.i.d. Gaussian noise model)

Summary

Three views on regression

Geometric least squares	Maximum Likelihood
Minimize $\sum_{n=1}^{N}(y(\mathbf{x}_n, \mathbf{w}) - t_n)^2$	Find $\mathbf{w}$ that maximizes $p(\mathbf{t}\vert\mathbf{x}, \mathbf{w})$

(Summary of ML/MAP under assumption of i.i.d. Gaussian noise model)

Summary

Three views on regression

Geometric least squares $\Leftrightarrow$ Maximum Likelihood

Minimize $\sum_{n=1}^{N}(y(\mathbf{x}_n, \mathbf{w}) - t_n)^2$ Find $\mathbf{w}$ that maximizes $p(\mathbf{t}|\mathbf{x}, \mathbf{w})$

(Summary of ML/MAP under assumption of i.i.d. Gaussian noise model)

Summary

Three views on regression

Geometric least squares	$\Leftrightarrow$	Maximum Likelihood	Maximum a Posteriori
Minimize $\sum_{n=1}^{N}(y(\mathbf{x}_n, \mathbf{w}) - t_n)^2$		Find $\mathbf{w}$ that maximizes $p(\mathbf{t}\|\mathbf{x}, \mathbf{w})$	Find $\mathbf{w}$ that maximizes $p(\mathbf{w}\|\mathbf{x}, \mathbf{t})$ $= p(\mathbf{t}\|\mathbf{w}, \mathbf{x})p(\mathbf{w})$

(Summary of ML/MAP under assumption of i.i.d. Gaussian noise model)

Summary

Three views on regression

Geometric least squares $\Leftrightarrow$ Maximum Likelihood   Maximum a Posteriori

Minimize $\sum_{n=1}^{N}(y(\mathbf{x}_n, \mathbf{w}) - t_n)^2$ Find $\mathbf{w}$ that maximizes $p(\mathbf{t}|\mathbf{x}, \mathbf{w})$ Find $\mathbf{w}$ that maximizes $p(\mathbf{w}|\mathbf{x}, \mathbf{t})$

$= p(\mathbf{t}|\mathbf{w}, \mathbf{x})p(\mathbf{w})$

Tend to overfit $\rightsquigarrow$ regularization

Minimize $\sum_{n=1}^{N}(y(\mathbf{x}_n, \mathbf{w}) - t_n)^2 + \|\mathbf{w}\|^2$

(Summary of ML/MAP under assumption of i.i.d. Gaussian noise model)

Summary

Three views on regression

Geometric least squares $\Leftrightarrow$ Maximum Likelihood   Maximum a Posteriori

Minimize $\sum_{n=1}^{N}(y(\mathbf{x}_n, \mathbf{w}) - t_n)^2$ Find $\mathbf{w}$ that maximizes $p(\mathbf{t}|\mathbf{x}, \mathbf{w})$ Find $\mathbf{w}$ that maximizes $p(\mathbf{w}|\mathbf{x}, \mathbf{t})$

$= p(\mathbf{t}|\mathbf{w}, \mathbf{x})p(\mathbf{w})$

Tend to overfit $\rightsquigarrow$ regularization

Minimize $\sum_{n=1}^{N}(y(\mathbf{x}_n, \mathbf{w}) - t_n)^2 \; + \|\mathbf{w}\|^2$ $\rightleftarrows$ MAP $\Leftrightarrow$ ML + L_2 regularizer

(Summary of ML/MAP under assumption of i.i.d. Gaussian noise model)

Summary

Three views on regression

Geometric least squares $\Leftrightarrow$ Maximum Likelihood | Maximum a Posteriori

Minimize $\sum_{n=1}^{N}(y(\mathbf{x}_n, \mathbf{w}) - t_n)^2$

Find $\mathbf{w}$ that maximizes $p(\mathbf{t}|\mathbf{x}, \mathbf{w})$

Find $\mathbf{w}$ that maximizes $p(\mathbf{w}|\mathbf{x}, \mathbf{t})$
$= p(\mathbf{t}|\mathbf{w}, \mathbf{x})p(\mathbf{w})$

Tend to overfit $\rightsquigarrow$ regularization

Minimize $\sum_{n=1}^{N}(y(\mathbf{x}_n, \mathbf{w}) - t_n)^2 + \|\mathbf{w}\|^2$

MAP $\Leftrightarrow$ ML + L_2 regularizer

Alternative regularizers:

Minimize $\sum_{n=1}^{N}(y(\mathbf{x}_n, \mathbf{w}) - t_n)^2 + \|\mathbf{w}\|^q$

Built-in dimensionality reduction / feature selection

(Summary of ML/MAP under assumption of i.i.d. Gaussian noise model)

Summary

Three views on regression

Geometric least squares $\Leftrightarrow$ Maximum Likelihood   Maximum a Posteriori

Minimize $\sum_{n=1}^{N}(y(\mathbf{x}_n, \mathbf{w}) - t_n)^2$ Find $\mathbf{w}$ that maximizes $p(\mathbf{t}|\mathbf{x}, \mathbf{w})$ Find $\mathbf{w}$ that maximizes $p(\mathbf{w}|\mathbf{x}, \mathbf{t})$

$= p(\mathbf{t}|\mathbf{w}, \mathbf{x})p(\mathbf{w})$

Tend to overfit $\rightsquigarrow$ regularization

Minimize $\sum_{n=1}^{N}(y(\mathbf{x}_n, \mathbf{w}) - t_n)^2 + \|\mathbf{w}\|^2$ $\Longleftrightarrow$ MAP $\Leftrightarrow$ ML + L_2 regularizer

Alternative regularizers:

Minimize $\sum_{n=1}^{N}(y(\mathbf{x}_n, \mathbf{w}) - t_n)^2 + \|\mathbf{w}\|^q$

Built-in dimensionality reduction / feature selection

No trivial Bayesian counterpart

(Summary of ML/MAP under assumption of i.i.d. Gaussian noise model)

Summary

Three views on regression

Geometric least squares $\Leftrightarrow$ Maximum Likelihood   Maximum a Posteriori

Minimize $\sum_{n=1}^{N}(y(\mathbf{x}_n,\mathbf{w})-t_n)^2$ Find $\mathbf{w}$ that maximizes $p(\mathbf{t}|\mathbf{x},\mathbf{w})$ Find $\mathbf{w}$ that maximizes $p(\mathbf{w}|\mathbf{x},\mathbf{t})$

$= p(\mathbf{t}|\mathbf{w},\mathbf{x})p(\mathbf{w})$

Tend to overfit $\rightsquigarrow$ regularization

Minimize $\sum_{n=1}^{N}(y(\mathbf{x}_n,\mathbf{w})-t_n)^2 + \|\mathbf{w}\|^2$ MAP $\Leftrightarrow$ ML + L_2 regularizer

Alternative regularizers:

Minimize $\sum_{n=1}^{N}(y(\mathbf{x}_n,\mathbf{w})-t_n)^2 + \|\mathbf{w}\|^q$

Built-in dimensionality reduction / feature selection

No trivial Bayesian counterpart

Frequentist

(Summary of ML/MAP under assumption of i.i.d. Gaussian noise model)

Summary

Three views on regression

Geometric least squares $\Leftrightarrow$ Maximum Likelihood   Maximum a Posteriori

Minimize $\sum_{n=1}^{N}(y(\mathbf{x}_n,\mathbf{w})-t_n)^2$ Find $\mathbf{w}$ that maximizes $p(\mathbf{t}|\mathbf{x},\mathbf{w})$

Find $\mathbf{w}$ that maximizes $p(\mathbf{w}|\mathbf{x},\mathbf{t})$

$= p(\mathbf{t}|\mathbf{w},\mathbf{x})p(\mathbf{w})$

Bayesian

Tend to overfit $\rightsquigarrow$ regularization

Minimize $\sum_{n=1}^{N}(y(\mathbf{x}_n,\mathbf{w})-t_n)^2 \ +\|\mathbf{w}\|^2$ MAP $\Leftrightarrow$ ML + L_2 regularizer

Alternative regularizers:

Minimize $\sum_{n=1}^{N}(y(\mathbf{x}_n,\mathbf{w})-t_n)^2 \ +\|\mathbf{w}\|^q$

Built-in dimensionality reduction / feature selection

No trivial Bayesian counterpart

Frequentist

(Summary of ML/MAP under assumption of i.i.d. Gaussian noise model)

Summary

Three views on regression

Geometric least squares $\Leftrightarrow$ Maximum Likelihood Maximum a Posteriori

Minimize $\sum_{n=1}^{N}(y(\mathbf{x}_n, \mathbf{w}) - t_n)^2$ Find $\mathbf{w}$ that maximizes $p(\mathbf{t}|\mathbf{x}, \mathbf{w})$

Tend to overfit $\rightsquigarrow$ regularization

Minimize $\sum_{n=1}^{N}(y(\mathbf{x}_n, \mathbf{w}) - t_n)^2 + \|\mathbf{w}\|^2$

Alternative regularizers:

Minimize $\sum_{n=1}^{N}(y(\mathbf{x}_n, \mathbf{w}) - t_n)^2 + \|\mathbf{w}\|^q$

Built-in dimensionality reduction / feature selection

No trivial Bayesian counterpart

Frequentist

Find $\mathbf{w}$ that maximizes $p(\mathbf{w}|\mathbf{x}, \mathbf{t})$

$= p(\mathbf{t}|\mathbf{w}, \mathbf{x})p(\mathbf{w})$

Bayesian

MAP $\Leftrightarrow$ ML + L_2 regularizer

For solving real problems, ask yourself questions like:

- Do I have a good prior (prior know-ledge can be better than standard regularizer)?
- Is my data normally (or similarly nicely) distributed?
- Do I need a sparse regularizer?

(Summary of ML/MAP under assumption of i.i.d. Gaussian noise model)

Summary

Three views on regression

Geometric least squares $\Leftrightarrow$ Maximum Likelihood — Maximum a Posteriori

Minimize $\sum_{n=1}^{N}(y(\mathbf{x}_n,\mathbf{w})-t_n)^2$ — Find $\mathbf{w}$ that maximizes $p(\mathbf{t}|\mathbf{x},\mathbf{w})$

Tend to overfit $\rightsquigarrow$ regularization

Minimize $\sum_{n=1}^{N}(y(\mathbf{x}_n,\mathbf{w})-t_n)^2 + \|\mathbf{w}\|^2$

Alternative regularizers:

Minimize $\sum_{n=1}^{N}(y(\mathbf{x}_n,\mathbf{w})-t_n)^2 + \|\mathbf{w}\|^q$

Built-in dimensionality reduction / feature selection

No trivial Bayesian counterpart

Frequentist

Find $\mathbf{w}$ that maximizes $p(\mathbf{w}|\mathbf{x},\mathbf{t})$

$= p(\mathbf{t}|\mathbf{w},\mathbf{x})p(\mathbf{w})$

Bayesian

MAP $\Leftrightarrow$ ML + L_2 regularizer

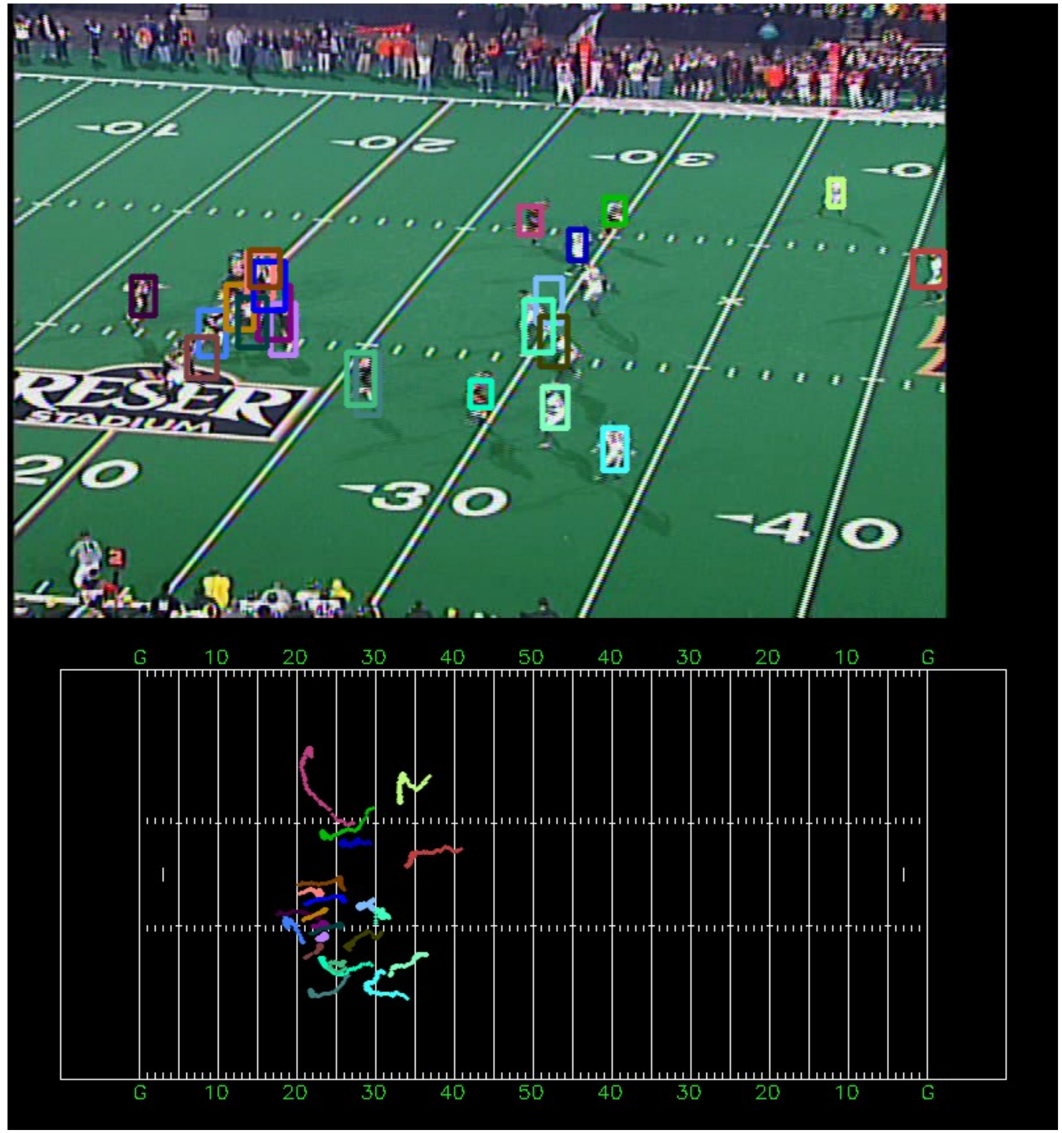
For solving real problems, ask yourself questions like:

- Do I have a good prior (prior knowledge can be better than standard regularizer)?
- Is my data normally (or similarly nicely) distributed?
- Do I need a sparse regularizer?

Scientist

(Summary of ML/MAP under assumption of i.i.d. Gaussian noise model)

Case: Tracking humans in video



http://eecs.oregonstate.edu/football/tracking

After today's lecture you should:

- Be able to produce a regularized maximum likelihood solution to a linear regression model
- Be able to produce a maximum a posteriori solution to a linear regression model
- Understand the relation between maximum a posteriori solutions and regularized maximum likelihood solutions
- Be familiar with different choices of regularization of why you would want to use them
- Understand the curse of dimensionality and its impact on solving regression problems
- Understand the effect of choice of prior in MAP estimates for different problems
- Be able to recognize and pose practical regression problems
- Reading material: CB 138-147, 152-156.

Next time!

- Neural networks (Christian)
- CB sections 5.1 - 5.3.3