Regression

Dr. Francis Colas

21.10.2011
Introduction

Regression:
▶ find the relationship between variables,
▶ find the best curve to fit data,
▶ predict the value for a new data point;

Characteristics:
▶ supervised learning,
▶ evaluate using goodness of fit,
▶ problem of overfitting.
Polynomial curve fitting

Example:

- synthetic data,
- generated based on a sine function (green),
- Gaussian noise (red).

Courtesy C. Bishop, PRML.
Polynomial curve fitting

Example:

- synthetic data,
- generated based on a sine function (green),
- Gaussian noise (red).

Courtesy C. Bishop, PRML.
Problem statement

Aim:

- from points \((x_n, t_n)\),
- fit model: \(y(x, w) = \sum_{j=0}^{M} w_j \cdot x^j\),

\[ \Leftrightarrow \text{find weight vector: } w, \]
- minimizing error \(E(w) = \frac{1}{2} \sum_{n=1}^{N} (y(x_n, w) - t_n)^2\).
Solution

Mathematically:

- Looking for: $\arg \min_{\mathbf{w}} E(\mathbf{w})$
- solve: $\frac{dE}{d\mathbf{w}} = 0$

⇒ solve:

$$
\begin{bmatrix}
\sum_{n=1}^{N} x_n^0 \left( \sum_{j=0}^{M} w_j x_n^j - t_n \right)
& \cdots \\
\vdots & \\
\sum_{n=1}^{N} x_n^k \left( \sum_{j=0}^{M} w_j x_n^j - t_n \right)
& \cdots
\end{bmatrix}
= 
\begin{bmatrix}
0 \\
\vdots \\
0
\end{bmatrix}
$$
Solution

Let:

\[
\Phi = \begin{pmatrix}
x_0^0 & \cdots & x_j^0 & \cdots & x_M^0 \\
\vdots & \ddots & \vdots & \ddots & \vdots \\
x_0^n & \cdots & x_j^n & \cdots & x_M^n \\
\vdots & \ddots & \vdots & \ddots & \vdots \\
x_0^N & \cdots & x_j^N & \cdots & x_M^N
\end{pmatrix}
\]

\(N\) rows by \(M + 1\) columns

\[\Phi = (\phi_{n,j} = x_n^j)\]

Then:

\[\frac{dE}{dw} = w^T \Phi^T \Phi - t^T \Phi = 0\]

Finally:

\[w = (\Phi^T \Phi)^{-1} \Phi^T t\]

Where \((\Phi^T \Phi)^{-1} \Phi^T = \Phi^+\) is the pseudoinverse of \(\Phi\).
Solution

Let:

$$\Phi = (\phi_{n,j} = x^j_n)$$

Then:

$$\frac{dE}{dw} = w^T \Phi^T \Phi - t^T \Phi = 0$$

Finally:

$$w = (\Phi^T \Phi)^{-1} \Phi^T t$$

Where $(\Phi^T \Phi)^{-1} \Phi^T = \Phi^+$ is the pseudoinverse of $\Phi$
Polynomial curve fitting

Solve:

- model: \( y(x, w) = \sum_{j=0}^{M} w_j x^j \),
- minimize SSE: \( E(w) = \frac{1}{2} \sum_{n=1}^{N} (y(x_n, w) - t_n)^2 \),
- let: \( \Phi = (x_n^j)_{n,j} \),
- solution: \( w = \Phi^+ t \)

Summary:

- model: linear combination of monomials,
- equation: linear system,
- solution: linear algebra.
Linear regression

Definition:

- model: linear combination of basis functions,
- minimize SSE,
- equation: linear system,
- solution: linear algebra.

Basis functions:

- set of functions $\phi = (\phi_j)_j$,
- define the model: $y(x, w) = \sum_{j=0}^{M} w_j \phi_j(x) = w^T \phi(x)$. 
Solution

Build the $N \times M + 1$ design matrix:

$$
\Phi = \begin{pmatrix}
\phi_0(x_1) & \cdots & \phi_j(x_1) & \cdots & \phi_M(x_1) \\
\vdots & \ddots & \vdots & \ddots & \vdots \\
\phi_0(x_n) & \cdots & \phi_j(x_n) & \cdots & \phi_M(x_n) \\
\vdots & \vdots & \ddots & \ddots & \vdots \\
\phi_0(x_N) & \cdots & \phi_j(x_N) & \cdots & \phi_M(x_N)
\end{pmatrix}
$$

Compute the pseudo-inverse and the weights:

$$
w = \Phi^+ t
$$
Basis functions

Different choices:
- monomials (polynomial curve fitting),
- Gaussian functions,
- sigmoidal functions
- ...
Basis functions

Different choices:
- monomials (polynomial curve fitting),
- Gaussian functions,
- sigmoidal functions
- ...
Basis functions

Different choices:
- monomials (polynomial curve fitting),
- Gaussian functions,
- sigmoidal functions
- ...
Basis functions

Different choices:

- monomials (polynomial curve fitting),
- Gaussian functions,
- sigmoidal functions
- ...

![Graph showing basis functions](image-url)
Overfitting

Result depends on the choice and number of basis functions.

With monomials:
Validation

Comparing models:

▶ training set: involved in fitting each model,
▶ test set: to compare the models:

But you need a large enough dataset.
Overfitting

Trade-off between complexity of the model and the data

\[ N = 15 \]

\[ N = 100 \]
Regularization

\begin{align*}
M = 0 & \quad t \quad 0 \quad 1 \\
M = 1 & \quad t \quad 0 \quad 1 \\
M = 3 & \quad t \quad 0 \quad 1 \\
M = 9 & \quad t \quad 0 \quad 1
\end{align*}
## Regularization

### Weight vector:

<table>
<thead>
<tr>
<th></th>
<th>$M = 0$</th>
<th>$M = 1$</th>
<th>$M = 3$</th>
<th>$M = 9$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$w_0$</td>
<td>0.19</td>
<td>0.82</td>
<td>0.31</td>
<td>0.35</td>
</tr>
<tr>
<td>$w_1$</td>
<td>-1.27</td>
<td>7.99</td>
<td></td>
<td>232.37</td>
</tr>
<tr>
<td>$w_2$</td>
<td>-25.43</td>
<td></td>
<td>-5321.83</td>
<td></td>
</tr>
<tr>
<td>$w_3$</td>
<td>17.36</td>
<td></td>
<td>48568.31</td>
<td></td>
</tr>
<tr>
<td>$w_4$</td>
<td></td>
<td></td>
<td>-231639.30</td>
<td></td>
</tr>
<tr>
<td>$w_5$</td>
<td></td>
<td></td>
<td>640042.26</td>
<td></td>
</tr>
<tr>
<td>$w_6$</td>
<td></td>
<td></td>
<td>-1061800.52</td>
<td></td>
</tr>
<tr>
<td>$w_7$</td>
<td></td>
<td></td>
<td>1042400.18</td>
<td></td>
</tr>
<tr>
<td>$w_8$</td>
<td></td>
<td></td>
<td>-557682.99</td>
<td></td>
</tr>
<tr>
<td>$w_9$</td>
<td></td>
<td></td>
<td>125201.43</td>
<td></td>
</tr>
</tbody>
</table>

Overfitting $\rightarrow$ large values!
Regularization

Regularization:
- penalize high values,
- change the error function:

\[ \tilde{E}(w) = \frac{1}{2} \sum_{n=1}^{N} (y(x_n, w) - t_n)^2 + \frac{\lambda}{2} \|w\|^2 \]

- still linear algebra:

\[ w = (\lambda I + \Phi^T \Phi)^{-1} \Phi^T t \]

Choice of \( \lambda \):
- too large: simple model,
- too small: overfitting,
- trade-off between data and model.
Summary

Linear regression:
- linear combination of basis functions,
- find weight minimizing error,
- linear system of equation,
- overfitting;

Regularization:
- penalize big weights,
- change the error,
- still linear system,
- choice of regularization parameter.
Reasoning

Regression:
- estimate parameters,
- based on data,
- given a model;

Last week:
- estimate state,
- based on data,
- given a model.

Probabilistic reasoning applied:
- regression,
- machine learning.
Probabilistic formulation

We have:
- \( x \) values,
- predictions \( y \) for \( x \) given \( w \),
- \( t \) values that should be close to \( y \).

We want:
- \( w \).

More formally:
- \( t_n = y(x_n, w) + \epsilon \)
- \( P(t_n|x_n, w) = \mathcal{N}(y(x_n, w), \sigma^2) \).
Probabilistic formulation

Assumptions:
- points $x_n$ are all independent,
- values $t_n$ are independent from everything given $x_n$ and $w$,
- Gaussian noise with identical variance.

\[ P(w, x_1, t_1, \ldots, x_N, t_N) = P(w) \prod_{n=1}^{N} P(x_n)P(t_n|x_n, w) \]
Inference

Maximum likelihood estimation (MLE):

$$\arg\max_w P(t|x, w)$$

Product $\rightarrow$ taking log:

$$\ln P(t|x, w) = \sum_{n=1}^{N} \ln P(t_n|x_n, w)$$

$$= \sum_{n=1}^{N} \left( -\frac{1}{2} \ln (2\pi \sigma^2) - \frac{1}{2\sigma^2} (t_n - y(x_n, w))^2 \right)$$

$$= -\frac{N}{2} \ln (2\pi \sigma^2) - \frac{1}{\sigma^2} \frac{1}{2} \sum_{n=1}^{N} (y(x_n, w) - t_n)^2$$

$$= \beta - \frac{1}{\sigma^2} E(w)$$

Maximum likelihood $\Leftrightarrow$ regression!
Inference

Maximum likelihood estimation (MLE):

\[ \arg \max_w P(t|x, w) \]

Product \(\rightarrow\) taking log:

\[
\ln P(t|x, w) = \sum_{n=1}^{N} \ln P(t_n|x_n, w) \\
= \sum_{n=1}^{N} \left( -\frac{1}{2} \left( \ln (2\pi\sigma^2) \right) - \frac{1}{2\sigma^2} \left( t_n - y(x_n, w) \right)^2 \right) \\
= -\frac{N}{2} \ln (2\pi\sigma^2) - \frac{1}{\sigma^2} \frac{1}{2} \sum_{n=1}^{N} \left( y(x_n, w) - t_n \right)^2 \\
= \beta - \frac{1}{\sigma^2} E(w)
\]

Maximum likelihood \(\leftrightarrow\) regression!
Inference

Maximum likelihood estimation (MLE):

\[ \arg \max_w P(t|x, w) \]

Product \(\rightarrow\) taking log:

\[
\ln P(t|x, w)
= \sum_{n=1}^{N} \ln P(t_n|x_n, w)
= \sum_{n=1}^{N} \left( -\frac{1}{2} \ln (2\pi\sigma^2) - \frac{1}{2\sigma^2} (t_n - y(x_n, w))^2 \right)
= -\frac{N}{2} \ln (2\pi\sigma^2) - \frac{1}{\sigma^2} \frac{1}{2} \sum_{n=1}^{N} (y(x_n, w) - t_n)^2
= \beta - \frac{1}{\sigma^2} E(w)
\]

Maximum likelihood \(\Leftrightarrow\) regression!
Inference

Maximum a posteriori (MAP):

$$\arg \max_w P(w|x, t)$$

Assuming Gaussian prior over $w$:

$$\ln P(w|x, t)$$

$$= \alpha + \ln P((t|x, w) + \ln P(w)$$

$$= \alpha + \beta - \frac{1}{\sigma^2} E(w) + \gamma - \frac{1}{2\sigma^2_w} w^T w$$

$$= \delta - \frac{1}{\sigma^2} \left( E(w) + \frac{\sigma^2}{2\sigma^2_w} \|w\|^2 \right)$$

Maximum a posteriori $\Leftrightarrow$ regularization!
Inference

Maximum a posteriori (MAP):

$$\arg \max_w P(w|x, t)$$

Assuming Gaussian prior over $w$:

$$\ln P(w|x, t) = \alpha + \ln P(t|x, w) + \ln P(w) = \alpha + \beta - \frac{1}{\sigma^2} E(w) + \gamma - \frac{1}{2\sigma^2_w} w^T w$$

$$= \delta - \frac{1}{\sigma^2} \left( E(w) + \frac{\sigma^2}{2\sigma^2_w} \|w\|^2 \right)$$

Maximum a posteriori $\iff$ regularization!
Inference

Maximum a posteriori (MAP):

\[
\arg \max_w P(w|\mathbf{x}, \mathbf{t})
\]

Assuming Gaussian prior over \( w \):

\[
\ln P(w|\mathbf{x}, \mathbf{t}) = \alpha + \ln P((\mathbf{t}|\mathbf{x}, w) + \ln P(w) = \alpha + \beta - \frac{1}{\sigma^2}E(w) + \gamma - \frac{1}{2\sigma_w^2}w^Tw
\]

\[
= \delta - \frac{1}{\sigma^2} \left( E(w) + \frac{\sigma^2}{2\sigma_w^2} \| \mathbf{w} \|^2 \right)
\]

Maximum a posteriori \( \Leftrightarrow \) regularization!
Example

Fitting a line incrementally:

\[ P(t_n|\mathbf{x}_n, \mathbf{w}) \quad P(\mathbf{w}|...) \quad \text{data space} \]
Example

Fitting a line incrementally:

\[ P(t_n|x_n, w) \quad P(w|...) \quad \text{data space} \]
Example

Fitting a line incrementally:

\[ P(t_n | x_n, \mathbf{w}) \quad P(\mathbf{w} | \ldots) \quad \text{data space} \]
Example

Fitting a line incrementally:

\[ P(t_n | x_n, w) \]

\[ P(w | ... \) \]

data space

\[ w_0 \]

\[ w_1 \]

\[ y \]

\[ x \]
Example

Fitting a line incrementally:

\[ P(t_n | x_n, w) \quad P(w | ...) \quad \text{data space} \]
Example

Fitting a line incrementally:

\[ P(t_n|x_n, w) \quad P(w|\ldots) \quad \text{data space} \]
Example

Fitting a line incrementally:

\[ P(t_n|x_n, w) \]  
\[ P(w|\ldots) \]

data space
Example

Fitting a line incrementally:

\[ P(t_n|x_n, w) \quad P(w|...) \]

data space

\[ w_1 \]

\[ w_0 \]

\[ x \]

\[ y \]
Prediction

Predict the value for a new point:

\[ P(\tilde{t}|\tilde{x}, x, t) = \int_w P(\tilde{t}|\tilde{x}, w)P(w|x, t) \]
Prediction

Predict the value for a new point:

\[ P(\tilde{t} | \tilde{x}, x, t) = \int_w P(\tilde{t} | \tilde{x}, w) P(w | x, t) \]
Prediction

Predict the value for a new point:

\[ P(\tilde{t}|\tilde{x}, x, t) = \int_{w} P(\tilde{t}|\tilde{x}, w) P(w|x, t) \]
Prediction

Predict the value for a new point:

\[ P(\tilde{t}|\tilde{x}, x, t) = \int_w P(\tilde{t}|\tilde{x}, w)P(w|x, t) \]
Prediction

Predict the value for a new point:

\[ P(\tilde{t}|\tilde{x}, x, t) = \int_w P(\tilde{t}|\tilde{x}, w) P(w|x, t) \]
Summary

Linear regression:
- linear combination of basis functions,
- linear system of equation,
- overfitting;

Regularization:
- penalize big weights,
- still linear system,
- choice of regularization parameter;

Probabilistic formulation:
- Gaussian noise,
- MLE is regression,
- Gaussian prior,
- MAP is regularization.