Regression

Practical Machine Learning
Fabian Wauthier
09/10/2009

Adapted from slides by Kurt Miller and Romain Thibaux
Outline

• Ordinary Least Squares Regression
  - Online version
  - Normal equations
  - Probabilistic interpretation

• Overfitting and Regularization

• Overview of additional topics
  - $L_1$ Regression
  - Quantile Regression
  - Generalized linear models
  - Kernel Regression and Locally Weighted Regression
Outline

• Ordinary Least Squares Regression
  - Online version
  - Normal equations
  - Probabilistic interpretation

• Overfitting and Regularization

• Overview of additional topics
  - $L_1$ Regression
  - Quantile Regression
  - Generalized linear models
  - Kernel Regression and Locally Weighted Regression
Regression vs. Classification:

Classification

\[ X \Rightarrow Y \]

Anything:
- continuous \((\mathbb{R}, \mathbb{R}^d, \ldots)\)
- discrete \((\{0,1\}, \{1,\ldots,k\}, \ldots)\)
- structured (tree, string, \ldots)
- \ldots

Discrete:
- \{0,1\} \hspace{1cm} \text{binary}
- \{1,\ldots,k\} \hspace{1cm} \text{multi-class}
- tree, etc. \hspace{1cm} \text{structured}
Regression vs. Classification:

Classification

\[ X \rightarrow Y \]

Anything:

- Continuous (\( \mathbb{R}, \mathbb{R}^d, \ldots \))
- Discrete (\{0,1\}, \{1,\ldots,k\}, \ldots)
- Structured (tree, string, \ldots)
- \ldots

Kernel trick

Possibilities:

- Perceptron
- Logistic Regression
- Support Vector Machine
- Decision Tree
- Random Forest
Regression vs. Classification:

Regression

Anything:
- continuous ($\mathbb{R}$, $\mathbb{R}^d$, ...)
- discrete ({$0,1$}, {$1,\ldots,k$}, ...)
- structured (tree, string, ...)
- ...

- continuous:
  - $\mathbb{R}$, $\mathbb{R}^d$
Examples

• Voltage $\Rightarrow$ Temperature
• Processes, memory $\Rightarrow$ Power consumption
• Protein structure $\Rightarrow$ Energy
• Robot arm controls $\Rightarrow$ Torque at effector
• Location, industry, past losses $\Rightarrow$ Premium
Linear regression

Given examples \((x_i, y_i)_{i=1 \ldots n}\)

Predict \(y_{n+1}\) given a new point \(x_{n+1}\)
Linear regression

We wish to estimate $\hat{y}$ by a linear function of our data $x$:

$$\hat{y}_{n+1} = w_0 + w_1 x_{n+1,1} + w_2 x_{n+1,2}$$

$$= w^\top x_{n+1}$$

where $w$ is a parameter to be estimated and we have used the standard convention of letting the first component of $x$ be 1.
Choosing the regressor

Of the many regression fits that approximate the data, which should we choose?

Observation $y$

$$X_i = \begin{pmatrix} 1 \\ x_i \end{pmatrix}$$
LMS Algorithm
(Least Mean Squares)

In order to clarify what we mean by a good choice of $w$, we will define a cost function for how well we are doing on the training data:

$$\text{Cost} = \frac{1}{2} \sum_{i=1}^{n} (w^\top x_i - y_i)^2$$

where $X_i = \begin{pmatrix} 1 \\ x_i \end{pmatrix}$
The best choice of $w$ is the one that minimizes our cost function

$$E = \frac{1}{2} \sum_{i=1}^{n} (w^\top x_i - y_i)^2 = \sum_{i=1}^{n} E_i$$

In order to optimize this equation, we use standard gradient descent

$$w^{t+1} := w^t - \alpha \frac{\partial}{\partial w} E$$

where

$$\frac{\partial}{\partial w} E = \sum_{i=1}^{n} \frac{\partial}{\partial w} E_i$$

and

$$\frac{\partial}{\partial w} E_i = \frac{1}{2} \frac{\partial}{\partial w} (w^\top x_i - y_i)^2 = (w^\top x_i - y_i) x_i$$
The LMS algorithm is an online method that performs the following update for each new data point

$$w^{t+1} := w^t - \alpha \frac{\partial}{\partial w} E_i$$

$$= w^t + \alpha (y_i - x_i^\top w) x_i$$
LMS, Logistic regression, and Perceptron updates

• LMS

\[ w^{t+1} := w^t + \alpha(y_i - x_i^\top w)x_i \]

• Logistic Regression

\[ w^{t+1} := w^t + \alpha(y_i - f_w(x_i))x_i \]

• Perceptron

\[ w^{t+1} := w^t + \alpha(y_i - f_w(x_i))x_i \]
Ordinary Least Squares (OLS)

Observation $y$

Prediction $\hat{y}$

Error or “residual”

Cost = \[
\frac{1}{2} \sum_{i=1}^{n} (w^\top x_i - y_i)^2
\]

$X_i = \begin{pmatrix}
1 \\
x_i
\end{pmatrix}$
Minimize the sum squared error

\[ E = \frac{1}{2} \sum_{i=1}^{n} (w^\top x_i - y_i)^2 \]

\[ = \frac{1}{2} (Xw - y)^\top (Xw - y) \]

\[ = \frac{1}{2} (w^\top X^\top Xw - 2y^\top Xw + y^\top y) \]

\[ \frac{\partial}{\partial w} E = X^\top Xw - X^\top y \]

Setting the derivative equal to zero gives us the **Normal Equations**

\[ X^\top Xw = X^\top y \]

\[ w = (X^\top X)^{-1} X^\top y \]
We solved \[ \frac{\partial}{\partial w} E = X^\top (Xw - y) = 0 \]

\[ \Rightarrow \text{Residuals are orthogonal to columns of } X \]

\[ \Rightarrow \hat{y} = Xw \text{ gives the best reconstruction of } y \]

in the range of \( X \)
Subspace $S$ spanned by columns of $X$

Residual vector $y - y'$ is orthogonal to subspace $S$

$y'$ is an orthogonal projection of $y$ onto $S$
Computing the solution

We compute $w = (X^\top X)^{-1} X^\top y$.

If $X^\top X$ is invertible, then $(X^\top X)^{-1} X^\top$ coincides with the pseudoinverse $X^+$ of $X$ and the solution is unique.

If $X^\top X$ is not invertible, there is no unique solution $w$.

In that case $w = X^+ y$ chooses the solution with smallest Euclidean norm.

An alternative way to deal with non-invertible $X^\top X$ is to add a small portion of the identity matrix (= Ridge regression).
Beyond lines and planes

Linear models become powerful function approximators when we consider non-linear feature transformations.

\[ X_i = \begin{pmatrix} 1 \\ x_i \\ x_i^2 \end{pmatrix} \quad \Rightarrow \quad \hat{y}_i = w_0 + w_1 x_i + w_2 x_i^2 \]

Predictions are still linear in \( X \)!
All the math is the same!
Geometric interpretation

\[ \hat{y} = w_0 + w_1 x + w_2 x^2 \]

\[ X_1 = x \]

\[ X_2 = x^2 \]
Ordinary Least Squares [summary]

Given examples \((x_i, y_i)_{i=1}^{n}\)

Let \(X_i^\top = (f_1(x_i) \ f_2(x_i) \ \ldots \ f_d(x_i))\)

For example \(X_i^\top = \begin{pmatrix} 1 & x_{i,1} & x_{i,2} & x_{i,1}^2 & x_{i,2}^2 & x_{i,1}x_{i,2} \end{pmatrix}\)

Let \(X = \begin{pmatrix} -X_1^\top & \ldots & -X_d^\top \end{pmatrix}\) \(\in\mathbb{R}^{d \times n}\), \(y = \begin{pmatrix} y_1 \\ y_2 \\ \ldots \end{pmatrix}\)

Minimize \(\|Xw - y\|_2^2\) by solving \((X^\top X)w = X^\top y\)

Predict \(\hat{y}_{n+1} = X_{n+1}^\top w\)
Probabilistic interpretation

\[ y_i | x_i \sim N(X_i^\top w, \sigma^2) \]

Likelihood
\[
L = \prod_i \exp\left(-\frac{1}{2\sigma^2}(X_i^\top w - y_i)^2\right) = \exp\left(-\frac{1}{2\sigma^2} \sum_i (X_i^\top w - y_i)^2\right)
\]

\[
\arg\max_w L = \arg\min_w E
\]
Conditional Gaussians
\( p(y|x) \)
BREAK
Outline

• Ordinary Least Squares Regression
  - Online version
  - Normal equations
  - Probabilistic interpretation
• Overfitting and Regularization
• Overview of additional topics
  - $L_1$ Regression
  - Quantile Regression
  - Generalized linear models
  - Kernel Regression and Locally Weighted Regression
Overfitting

• So the more features the better? NO!
• Carefully selected features can improve model accuracy.
• But adding too many can lead to overfitting.
• Feature selection will be discussed in a separate lecture.
Overfitting

Degree 15 polynomial
Ridge Regression (Regularization)

Effect of regularization (degree 19)

Minimize

\[
\frac{1}{2} \| Xw - y \|^2_2 + \epsilon \| w \|^2_2
\]

with \( \epsilon \) “small” by solving

\[
(X^\top X + \epsilon I)w = X^\top y
\]
Probabilistic interpretation

Likelihood
\[ y_i | x_i \sim N(X_i^\top w, \sigma^2) \]

Prior
\[ w \sim N \left( 0, \frac{\sigma^2}{\epsilon} \right) \]

Posterior
\[
P(w | X, y) = \frac{P(w, x_1, \ldots, x_n, y_1, \ldots, y_n)}{P(x_1, \ldots, x_n, y_1, \ldots, y_n)}
\propto P(w. x_1, \ldots, x, y_1, \ldots, y_n)
\propto \exp \left\{-\frac{\epsilon}{2\sigma^2} ||w||_2^2 \right\} \prod_i \exp \left\{-\frac{1}{2\sigma^2} (X_i^\top w - y_i)^2 \right\}
= \exp \left\{-\frac{1}{2\sigma^2} \left[ \epsilon ||w||_2^2 + \sum_i (X_i^\top w - y_i)^2 \right] \right\}
Outline

• Ordinary Least Squares Regression
  - Online version
  - Normal equations
  - Probabilistic interpretation
• Overfitting and Regularization
• Overview of additional topics
  - L_1 Regression
  - Quantile Regression
  - Generalized linear models
  - Kernel Regression and Locally Weighted Regression
Errors in Variables
(Total Least Squares)

\[
y_i, X_i \sim N \left( \begin{pmatrix} \tilde{X}_i^\top w \\ \tilde{X}_i \end{pmatrix}, \sigma^2 I \right)
\]
Sensitivity to outliers

\[ E = \sum_{i} (x_i^\top w - y_i)^2 = \sum_{i} E_i \]

Temperature at noon

High weight given to outliers

Influence function

\[ \frac{\partial E_i}{\partial y_i} \]
L₁ Regression

\[ E' = \sum_{i} |x_i^\top w - y_i| \]

\[ = \sum_{i} E'_i \]

Lineral program

\[ \min_{w, c} \sum_i c_i \]

s.t.

\[ x_i^\top w - y_i \leq c_i \quad \forall i \]

\[ y_i - x_i^\top w \leq c_i \quad \forall i \]

[Matlab demo]
Quantile Regression

Slide courtesy of Peter Bodik
Generalized Linear Models

Probabilistic interpretation of OLS

Mean is linear in $X_i$

$$y_i | x_i \sim N(X_i^\top w, \sigma^2)$$

OLS: linearly predict the mean of a Gaussian conditional.

GLM: predict the mean of some other conditional density.

$$y_i | x_i \sim p(f(X_i^\top w))$$

May need to transform linear prediction by $f(\cdot)$ to produce a valid parameter.
Example: “Poisson regression”

Suppose data $y$ are event counts: $y \in \mathbb{N}_0$

Typical distribution for count data: Poisson

$\text{Poisson}(y|\lambda) = \frac{e^{-\lambda} \lambda^y}{y!}$  \quad \text{Mean parameter is } \lambda > 0$

Say we predict $\lambda = f(x^\top w) = \exp \{ x^\top w \}$

GLM: $y_i|x_i \sim \text{Poisson} \left( f(X_i^\top w) \right)$
Conditional Poissons
$p(y|x)$

Mean $\lambda$

$\lambda = 3$

$\lambda = 5$

$\lambda = 8$
As for OLS: optimize $\mathbf{w}$ by maximizing the likelihood of data.

Equivalently: maximize log likelihood.

Likelihood $L = \prod_i \text{Poisson} \left( y_i \mid f \left( X_i^\top \mathbf{w} \right) \right)$

Log likelihood $l = \sum_i \left( X_i^\top \mathbf{w} y_i - \exp \{ X_i^\top \mathbf{w} \} \right) + \text{const.}$

Batch gradient: $\frac{\partial l}{\partial \mathbf{w}} = \sum_i \left( y_i - \exp \{ X_i^\top \mathbf{w} \} \right) X_i$

$= \sum_i \left( y_i - f \left( X_i^\top \mathbf{w} \right) \right) X_i$

\text{“residual”}
LMS, Logistic regression, Perceptron and GLM updates

• GLM (online)

\[ w^{t+1} := w^t + \alpha (y_i - f_w(x_i))x_i \]

• LMS

\[ w^{t+1} := w^t + \alpha (y_i - x_i^\top w)x_i \]

• Logistic Regression

\[ w^{t+1} := w^t + \alpha (y_i - f_w(x_i))x_i \]

• Perceptron

\[ w^{t+1} := w^t + \alpha (y_i - f_w(x_i))x_i \]
Kernel Regression and Locally Weighted Linear Regression

• **Kernel Regression:**
  Take a very very conservative function approximator called AVERAGING. Locally weight it.

• **Locally Weighted Linear Regression:**
  Take a conservative function approximator called LINEAR REGRESSION. Locally weight it.

Slide from Paul Viola 2003
Kernel Regression

Kernel regression (sigma=1)

\[
\hat{y}(x) = \frac{\sum_i y_i k(x_i - x)}{\sum_i k(x_i - x)}
\]
Locally Weighted Linear Regression (LWR)

Kernel regression (sigma=1)

OLS cost function:

\[ E = \frac{1}{2} \sum_{i=1}^{n} (w^\top x_i - y_i)^2 \]

LWR cost function:

\[ E' = \sum_{i=1}^{n} k(x_i - x)(w^\top x_i - y_i)^2 \]

[Matlab demo]
Heteroscedasticity

#requests per minute

Time (days)

5000
What we covered

• Ordinary Least Squares Regression
  - Online version
  - Normal equations
  - Probabilistic interpretation
• Overfitting and Regularization
• Overview of additional topics
  - $L_1$ Regression
  - Quantile Regression
  - Generalized linear models
  - Kernel Regression and Locally Weighted Regression