CSC 411: Lecture 2 - Linear Regression

Ethan Fetaya, James Lucas and Emad Andrews

Regression - predicting continuous outputs.
Examples:

- Future stock prices.
- Tracking - object location in the next time-step.
- Housing prices.
- Crime rates.

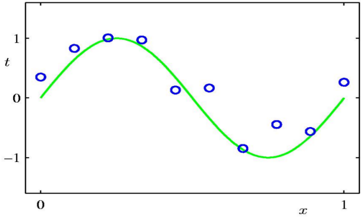
We don't just have infinite number of possible answers, we assume a simple geometry - closer is better.

We will focus on *linear* regression models.

What do I need in order to make predictions? In linear regression

- Inputs (features) x ($\mathbf{x}$ for vectors). A vector $\mathbf{x} \in \mathbb{R}^d$
- Output (dependent variable) y. $y \in \mathbb{R}$
- Training data. $(\mathbf{x}^{(1)}, y^{(1)}), ..., (\mathbf{x}^{(N)}, y^{(N)})$
- A model/hypothesis class, a family of functions that represents the relationship between x and y. $f_{\mathbf{w}}(\mathbf{x}) = w_0 + w_1 x_1 + ... w_d x_d$ for $\mathbf{w} \in \mathbb{R}^{d+1}$
- A loss function $\ell(y, \hat{y})$ that assigns a cost to each prediction. $L_2(y, \hat{y}) = (y - \hat{y})^2$, $L_1(y, \hat{y}) = |y - \hat{y}|$
- Optimization - a way to minimize the loss objective. Analytic solution, convex optimization

Linear model seems very limited, for example



is not close to linear.

In linear model we mean **linear in parameters not the inputs**!

[1]Images from Bishop

Any (fixed) transformation $\phi(x) \in \mathbb{R}^d$ we can run linear regression with features $\phi(x)$.

Example: Polynomials $w_0 + w_1 x + ... + w_d x^d$ are a linear (in w) model.

Feature engineering - design good features and feed them to a linear model.

Commonly replaced with deep models that learn the features as well.

Most common loss is $L_2(y, \hat{y}) = (y - \hat{y})^2$.

Easy to optimize (convex, analytic solution), well understood, harshly punishes large mistakes. Can be good (e.g. financial predictions) or bad (outliers).

The optimal prediction w.r.t L_2 loss is the conditional mean $\mathbb{E}[y|x]$ (show!).

Equivalent to assuming Gaussian noise (more on that later).

Another common loss is $L_1(y, \hat{y}) = |y - \hat{y}|$.

Easyish to optimize (convex), well understood, Robust to outliers.

The optimal prediction w.r.t L_2 loss is the conditional median (show!).

Equivalent to assuming Laplace noise.

You can combine both - Huber loss.

Deriving and analyzing the optimal solution:

Notation: We can include the bias into $\mathbf{x}$ by adding 1, $\mathbf{x}^{(\mathbf{i})} = [1, x_1^{(i)}, ..., x_d^{(i)}]$. Prediction is $\mathbf{x}^T \mathbf{w}$.

Target vector $\mathbf{y} = [y^{(1)}, ..., y^{(N)}]^T$.

Feature vectors $\mathbf{f}^{(j)} = [\mathbf{x}_j^{(1)}, ..., \mathbf{x}_j^{(N)}]^T$.

Design matrix $\mathbf{X}$, $\mathbf{X}_{ij} = \mathbf{x}_j^{(i)}$.

Rows correspond to data points, columns to features.

Theorem

The optimal $\mathbf{w}$ *w.r.t* L_2 *loss,* $w^* = \arg\min \sum_{i=1}^{N}(y^{(i)} - \mathbf{w}^T\mathbf{x}^{(i)})^2$ *is* $\mathbf{w}^* = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$.

Proof (sketch): Our predictions vector are $\hat{\mathbf{y}} = \mathbf{X}\mathbf{w}$ and the total loss is $L(\mathbf{w}) = ||\mathbf{y} - \hat{\mathbf{y}}||^2 = ||\mathbf{y} - \mathbf{X}\mathbf{w}||^2$.

Rewriting $L(\mathbf{w}) = ||\mathbf{y} - \mathbf{X}\mathbf{w}||^2 = (\mathbf{y} - \mathbf{X}\mathbf{w})^T(\mathbf{y} - \mathbf{X}\mathbf{w}) = \mathbf{y}^T\mathbf{y} + \mathbf{w}^T\mathbf{X}^T\mathbf{X}\mathbf{w} - 2\mathbf{w}^T\mathbf{X}^T\mathbf{y}$.

X.transpose

$\nabla L(\mathbf{w}^*) = 2\mathbf{X}^T\mathbf{X}\mathbf{w}^* - 2\mathbf{X}\mathbf{y} = 0 \Rightarrow \mathbf{X}^T\mathbf{X}\mathbf{w}^* = \mathbf{X}^T\mathbf{y}$. $\square$

If the features aren't linearly dependent $\mathbf{X}^T\mathbf{X}$ is invertible.

Never actually invert! Use linear solvers (Conjugate gradients, Cholesky decomp,...)

Some intuition: Our predictions are $\hat{\mathbf{y}} = \mathbf{X}\mathbf{w}^*$ and we have $\mathbf{X}^T\mathbf{X}\mathbf{w}^* = \mathbf{X}^T\mathbf{y}$.

Residual $r = \mathbf{y} - \hat{\mathbf{y}} = \mathbf{y} - \mathbf{X}\mathbf{w}^*$, so $\mathbf{X}^T r = 0$.

This means r is orthogonal to $\mathbf{f}^{(1)}, ..., \mathbf{f}^{(d)}$ (and zero mean).

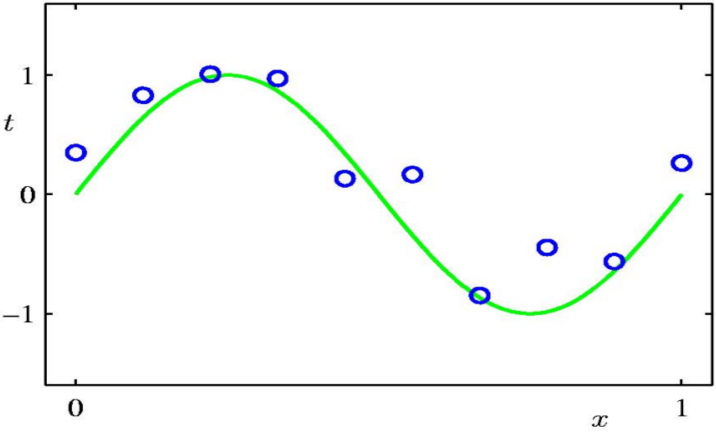
Geometrically we are projecting $\mathbf{y}$ to the subspace spun by the features.

Assume the features have zero mean $\sum_j \mathbf{f}_j^{(i)} = 0$, in this case $[\mathbf{X}^T\mathbf{X}]_{ij} = \mathrm{cov}(\mathbf{f}^{(i)}, \mathbf{f}^{(j)})$ and $[\mathbf{X}^T\mathbf{y}]_j = \mathrm{cov}(\mathbf{f}^{(j)}, \mathbf{y})$.

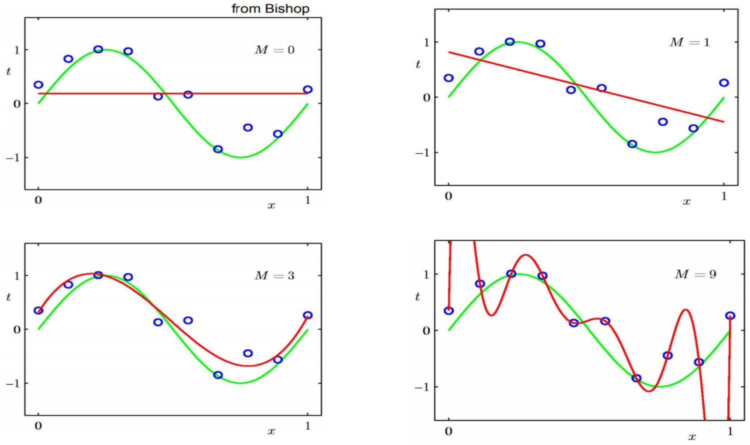
If the covariance is diagonal (data-whitening, see tutorial), $\mathrm{var}(\mathbf{f}^{(j)}) \cdot w_j = \mathrm{cov}(\mathbf{f}^{(j)}, \mathbf{y}) \Rightarrow w_j = \frac{\mathrm{cov}(\mathbf{f}^{(j)}, \mathbf{y})}{\mathrm{var}(\mathbf{f}^{(j)})}$.

Good feature = large signal to noise ratio (loosely speaking).

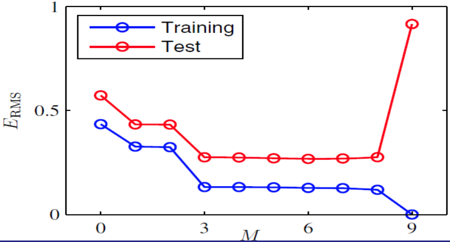
Back to our simple example - lets fit a polynomial of degree M.



Back to our simple example - lets fit a polynomial of degree M.



- Generalization = models ability to predict the held out data.
- Model with M = 1 underfits (cannot model data well).
- Model with M = 9 overfits (it models also noise).
- Not a problem if we have lots of training examples (rule-of-thumb 10×dim)
- Simple solution - model selection (validation/cross-validation)



Observation: Overfiting models term to have large norm.

	$M=0$	$M=1$	$M=6$	$M=9$
$w_0^\star$	0.19	0.82	0.31	0.35
$w_1^\star$		-1.27	7.99	232.37
$w_2^\star$			-25.43	-5321.83
$w_3^\star$			17.37	48568.31
$w_4^\star$				-231639.30
$w_5^\star$				640042.26
$w_6^\star$				-1061800.52
$w_7^\star$				1042400.18
$w_8^\star$				-557682.99
$w_9^\star$				125201.43

Solution: Regularizer $R(\mathbf{w})$ penalizing large norm,
$w^* = \arg\min_{\mathbf{w}} = L_S(\mathbf{w}) + R(\mathbf{w})$.

Commonly use $R(\mathbf{w}) = \frac{\lambda}{2}||\mathbf{w}||_2^2 = \frac{\lambda}{2}\mathbf{w}^T\mathbf{w} = \frac{\lambda}{2}\sum \mathbf{w}_j^2$

L_2 regularization $R(\mathbf{w}) = \frac{\lambda}{2}\mathbf{w}^T\mathbf{w}$

Objective $\sum_i(\mathbf{w}^T\mathbf{x}^{(i)} - y^{(i)})^2 + \frac{\lambda}{2}\mathbf{w}^T\mathbf{w}$.

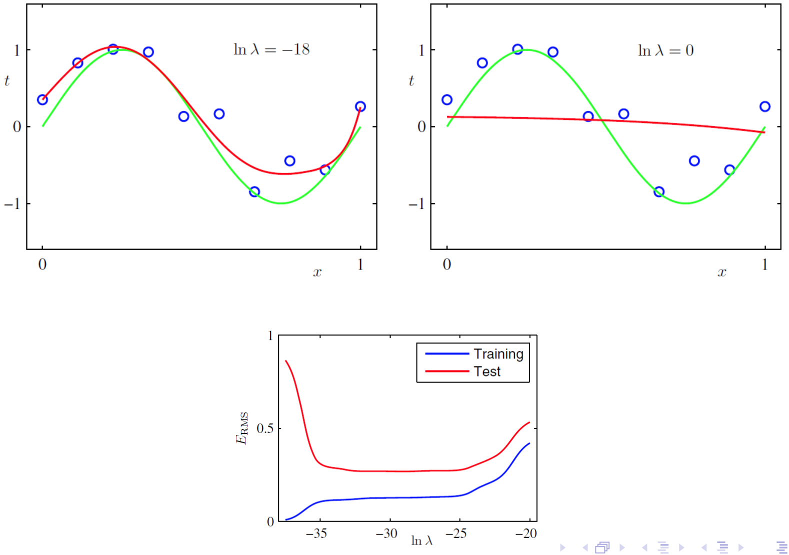
Analytic solution $\mathbf{w}^* = (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}\mathbf{y}$ (show!)

Can show equivalence to Gaussian prior.

Normaly we do not regularize the bias w_0.

Use validation/cross-validation to find a good λ.

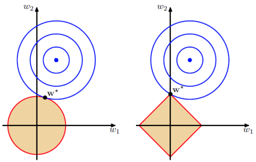
Regularization



Another common regularizer: L_1 regularization
$R(\mathbf{w}) = \lambda ||\mathbf{w}||_1 = \lambda \sum |w_i|$

Convex (SGD) but no analytic solution

Tends to induce *sparse* solutions.



Can show equivalence to Laplacian prior.

Probabilistic viewpoint: Assume $p(y^{(i)}|x^{(i)}) = \mathbf{w}^T\mathbf{x}^{(i)} + \epsilon_i$ and ϵ_i are i.i.d $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$. $p(y|x) = \mathcal{N}(\mathbf{w}^T\mathbf{x}, \sigma^2) = \frac{\exp\left(\frac{-||y-\mathbf{w}^T\mathbf{x}||^2}{2\sigma^2}\right)}{\sqrt{2\pi\sigma^2}}$.

w parametrizes a distribution. Which distribution to pick? Maximize the *likelihood* of the observation.

Log-likelihood $\log(p(\mathbf{y}^{(1)}, ..., \mathbf{y}^{(N)})|\mathbf{x}^{(1)}, ..., \mathbf{x}^{(N)}; \mathbf{w}))$
$= \log\left(\prod_{i=1}^{N} p(\mathbf{y}^{(i)}|\mathbf{x}^{(i)}; \mathbf{w})\right) = \sum_{i=1}^{N} \log\left(p(\mathbf{y}^{(i)}|\mathbf{x}^{(i)}; \mathbf{w})\right)$.

Linear Gaussian model
$\Rightarrow \log\left(p(\mathbf{y}|\mathbf{x}; \mathbf{w})\right) = \frac{-||y-\mathbf{w}^T\mathbf{x}||^2}{2\sigma^2} - 0.5\log(2\pi\sigma^2)$

maximum likelihood = minimum L_2 loss.

"When you hear hoof-beats, think of horses not zebras" *Dr. Theodore Woodward.*

ML finds a model that makes the observation likely $P(data|w)$, we want the most probable model $p(w|data)$.

Bayes formula $P(\mathbf{w}|\mathbf{y}, \mathbf{X}) = \frac{P(\mathbf{y}|\mathbf{w},\mathbf{X})p(\mathbf{w})}{p(\mathbf{y}|\mathbf{X})} \propto P(\mathbf{y}|\mathbf{w}, \mathbf{X})p(\mathbf{w})$

Need prior $p(\mathbf{w})$ - what model is more likely?

MAP=Maximum a posteriori estimator
$\mathbf{w}_{MAP} = \arg\max P(\mathbf{w}|\mathbf{y}, \mathbf{X}) = \arg\max P(\mathbf{y}|\mathbf{w}, \mathbf{X})p(\mathbf{w})$
$= \arg\max \log(P(\mathbf{y}|\mathbf{w}, \mathbf{X})) + \log(p(\mathbf{w}))$

Convenient prior (conjugate): $p(\mathbf{w}) = \mathcal{N}(0, \sigma_w^2)$

$\mathbf{w}_{map} = \arg\max \log(P(\mathbf{y}|\mathbf{w}, \mathbf{X})) + \log(p(\mathbf{w}))$
$= -\frac{||y - \mathbf{w}^T \mathbf{x}||^2}{2\sigma^2} - \frac{||\mathbf{w}||^2}{2\sigma_w^2}$

L_2 regularization = Gaussian prior.

Recap:

- Linear models benefit: Simple, fast (test time), generalize well (with regularization).
- Linear models limitations: Performance crucially depends on good features.
- Modeling questions - loss and regularizer (and features)
- L_2 loss and regularization - analytical solution, otherwise stochastic optimization (next week).
- Difficulty with multimodel distribution - discretization might work much better.