Multi-task Learning - Advantages and Implementations under Computation Budget

Ayan Acharya

SWIFT Summer Intern

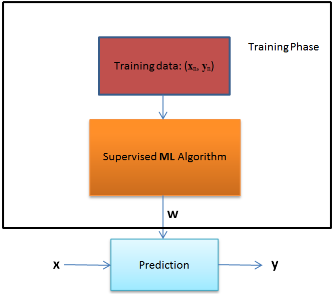
August 24, 2012

Machine Learning Applications

- Fraud detection.
- Web search ranking (Google, Yahoo!, Bing search engines).
- Speech and object recognition.
- Stock market analysis.
- Recommender Systems (Amazon, NetFlix, eBay).
- DNA sequence classification.
- Robot locomotion.
- Disease prediction (Google Flu trends).

Supervised Machine Learning Algorithm

- types of ML algorithms: supervised, unsupervised, semi-supervised, reinforcement, transductive etc.
- notation: $\mathbf{x}$: data point, $\mathbf{y}$: response variable, $\mathbf{w}$: parameters.



Multi-task Learning

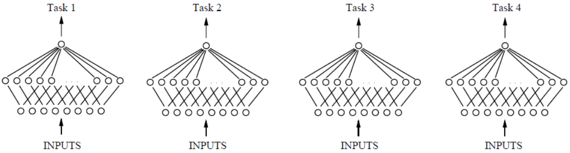


Figure : Learning Tasks Separately

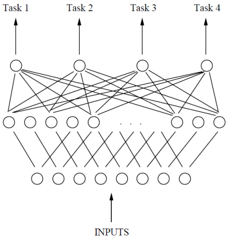
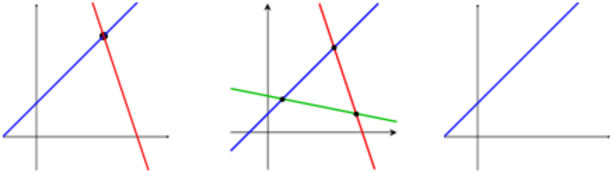


Figure : Learning Tasks Jointly

Recapulation of Linear System of Equations

- $\mathbf{y} = \boldsymbol{\beta}\mathbf{x}, \ \mathbf{y} \in \mathbb{R}^M, \mathbf{x} \in \mathbb{R}^D, \boldsymbol{\beta} \in \mathbb{R}^{M \times D}$.
- $M = D$, completely determined system – unique solution.
- $M < D$, over-determined system – no solution.
- $M > D$, under-determined system – multiple solutions.



Problems with High-dimensional Data

Living area (feet^2)	#bedrooms	Price (1000$s)
2104	3	400
1600	3	330
2400	3	369
1416	2	232
3000	4	540
⋮	⋮	⋮

- D : data dimension, N : number of measurements.
- $D >> N$ – too few measurements compared to data complexity.
- Examples: medical diagonsis data, weather prediction data.
- Compressed sensing.

Solution?

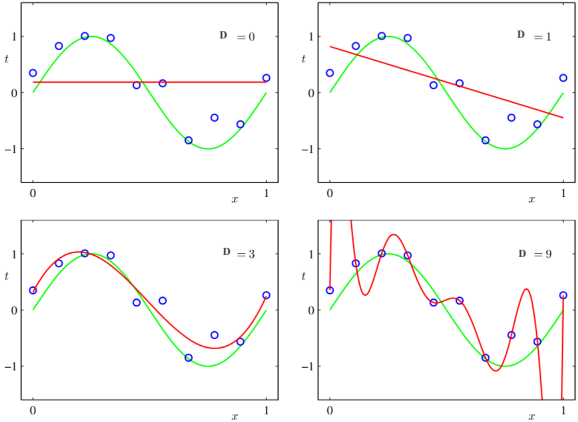
- An engineer thinks that equations are an approximation to reality.
- A physicist thinks reality is an approximation to equations.
- A mathematician doesn't care.
- We use our favorite hammer – **approximations** – limit the degree of freedom of the parameters to be learnt.

Mathematical Formulation

- using ℓ_1, ℓ_2 or ℓ_1/ℓ_q regularization.
- using graphical models.
- $\cdots$

Polynomial Curve Fitting

- $y = w_0 + \sum_{d=1}^{D} w_d x^d.$



Polynomial Curve Fitting Continued ..

Table of the coefficients $\mathbf{w}^\star$ for polynomials of various order. Observe how the typical magnitude of the coefficients increases dramatically as the order of the polynomial increases.

	$D = 0$	$D = 1$	$D = 6$	$D = 9$
$w_0^\star$	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

Regularized Linear Regression

- $(\mathbf{x}_n, \mathbf{y}_n)_{n=1}^{N}$ – observed data and response value pair.
- A straw-man strategy – $\min_{\mathbf{w}} \sum_{n=1}^{N} (\mathbf{y}_n - \mathbf{w}\mathbf{x}_n)^2$.
- More advanced strategy – $\min_{\mathbf{w}} \sum_{n=1}^{N} (\mathbf{y}_n - \mathbf{w}\mathbf{x}_n)^2$ s.t. $||\mathbf{w}||_q \leq R$.
- ℓ_q norm: $||\mathbf{w}||_q = (\sum_{d=1}^{D} w_d^q)^{1/q}$.

Level Sets of Regularizers

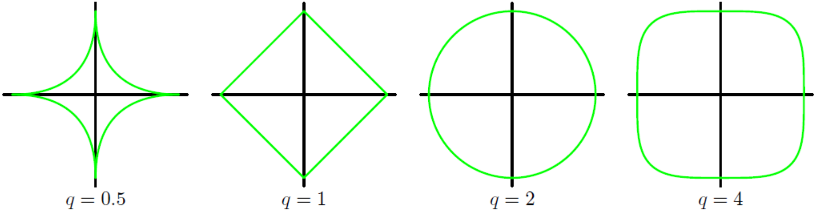
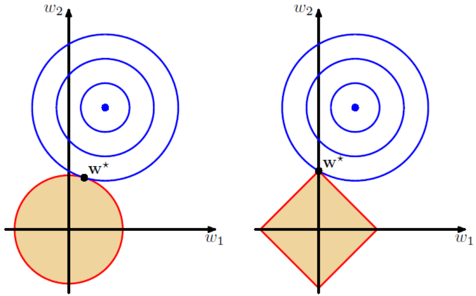


Figure : $||\mathbf{w}||_{0.5} = 1, \ ||\mathbf{w}||_1 = 1, \ ||\mathbf{w}||_2 = 1, \ ||\mathbf{w}||_4 = 1.$

Comparison of ℓ_2 and ℓ_1 Norm

- $\min_{\mathbf{w}} \sum_{n=1}^{N} (\mathbf{y}_n - \mathbf{w}\mathbf{x}_n)^2$ s.t. $||\mathbf{w}||_q \leq R$.
- $\min_{\mathbf{w}} \sum_{n=1}^{N} (\mathbf{y}_n - \mathbf{w}\mathbf{x}_n)^2 + \lambda ||\mathbf{w}||_q$.
- ℓ_1 regularization provides sparser solution.

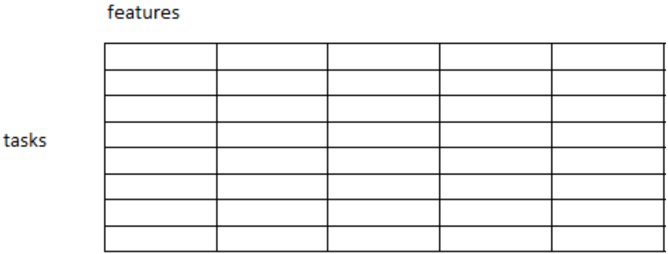


Bayesian Linear Regression

- Impose some prior belief on possible values of $\mathbf{w}$.
- Maximize the likelihood of observations with normal distribution used as prior – regularized linear regression.
- Prior on model variables acts as regularizer.

Multi-task Linear Regression

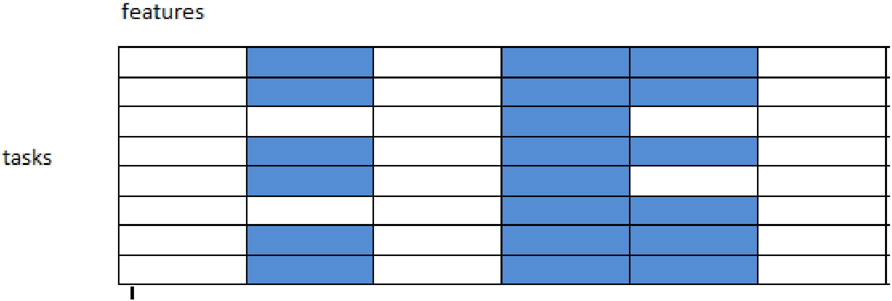
- K different linear regression problems.
- $\min_{\mathbf{w}_k} \sum_{n=1}^{N} (y_{kn} - \mathbf{w}_k \mathbf{x}_n)^2 + \lambda ||\mathbf{w}_k||_q \ \forall k.$
- Equivalent formulation: $\min_{\mathbf{w}} \sum_{k=1}^{K} \sum_{n=1}^{N} (\mathbf{y}_{kn} - \mathbf{w}_k \mathbf{x}_n)^2 + \lambda \sum_{k=1}^{K} ||\mathbf{w}_k||_q.$
- $f(\mathbf{x}) = f_1(x_1) + f_2(x_2) + f_3(x_3) + \cdots$



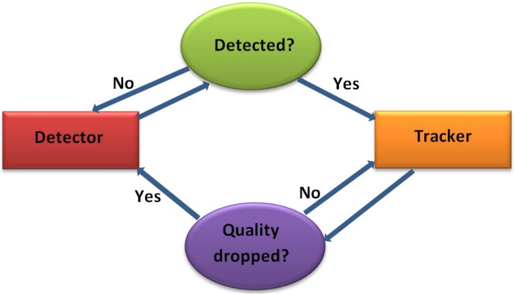
Multi-task Linear Regression Continued ..

- Alternate Formulation: $\min_{\mathbf{w}} \sum_{k=1}^{K} \sum_{n=1}^{N} (y_{kn} - \mathbf{w}_k \mathbf{x}_n)^2 + \lambda \sum_{d=1}^{D} ||\mathbf{w}^{(d)}||_q$, $\mathbf{w}^{(d)} \in \mathbb{R}^K \ \forall d$.
- ℓ_1/ℓ_q norm – Group LASSO.
- Sparsity on individual features.

Visualization of Group Sparsity

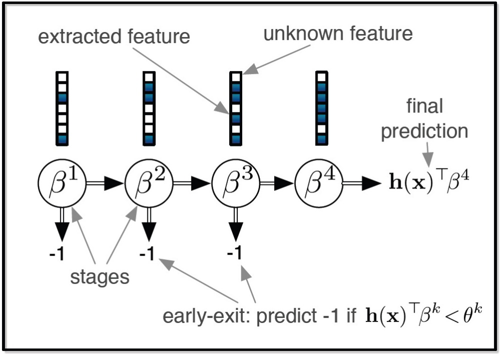


Real Time Car Detection and Tracking



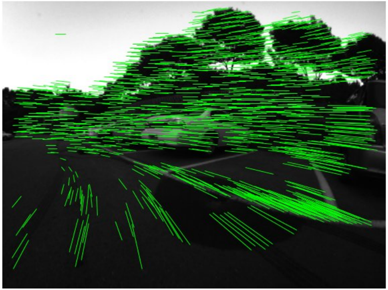
AdaBoost Classifier

- convex combination of weak learners.
- weak learners are some classifiers with error rate less than 50% (for binary classification).
- strong theoretical understanding and convergence guarantees.
- learning involves only a set of closed-form updates.
- boosting trick – $\mathbf{h}(\mathbf{x}_n) = (h_j(\mathbf{x}_n))_{j=1}^{J}$, $\mathbf{h} : \mathbb{R}^D \rightarrow \{-1, +1\}^J$ – motivation similar to kernel trick.


extracted feature
unknown feature
final prediction
β^1
β^2
β^3
β^4
$\mathbf{h}(\mathbf{x})^\top \beta^4$
-1
-1
-1
stages
early-exit: predict -1 if $\mathbf{h}(\mathbf{x})^\top \beta^k < \theta^k$

Tracker

- Corner detection based tracking,
- Computationally cheap.



An Adaptive MTL Framework

- Detect multiple types of vehicles (SUVs, cars, buses, trucks etc.), pedestrians, signals.
- Vary computation effort depending on:
  - the device
  - resources available at the prediction time
- Applications: Web Search Ranking, Real Time Object Detection.

Formulation of Adaptive MTL

- Build a predictor $H_{\beta^{(k)}}(\mathbf{x}_n) = \langle \boldsymbol{\beta}^{(k)}, \mathbf{h}(\mathbf{x}_n) \rangle \; \forall k$.
- Optimization problem:

$$\min_{\boldsymbol{\beta}} \sum_{n=1}^{N} \mathcal{L}(y_n, \max_k \{ H_{\boldsymbol{\beta}^{(k)}}(\mathbf{x}_n) \}) \text{ s.t. } c(\mathbf{q}, \boldsymbol{\beta}) \leq T, \boldsymbol{\beta} \geq \mathbf{0}. \quad (1)$$

- The regularizer:

$$c(\mathbf{q}, \boldsymbol{\beta}) = r(\boldsymbol{\beta}) + \tau(\mathbf{q}, \boldsymbol{\beta}). \quad (2)$$

- $\mathbf{q} = (q_d)_{d=1}^{D}$ – computation cost for retrieving features.
- $r(\boldsymbol{\beta})$ is an ℓ_1/ℓ_q regularizer.
- $\tau(\mathbf{q}, \boldsymbol{\beta})$ – computation cost associated with accessing the raw features from the observations.

Questions?