THE UNIVERSITY OF SYDNEY

Advanced Machine Learning

(COMP 5328)

Multi-Task Learning

Tongliang Liu

THE UNIVERSITY OF
SYDNEY

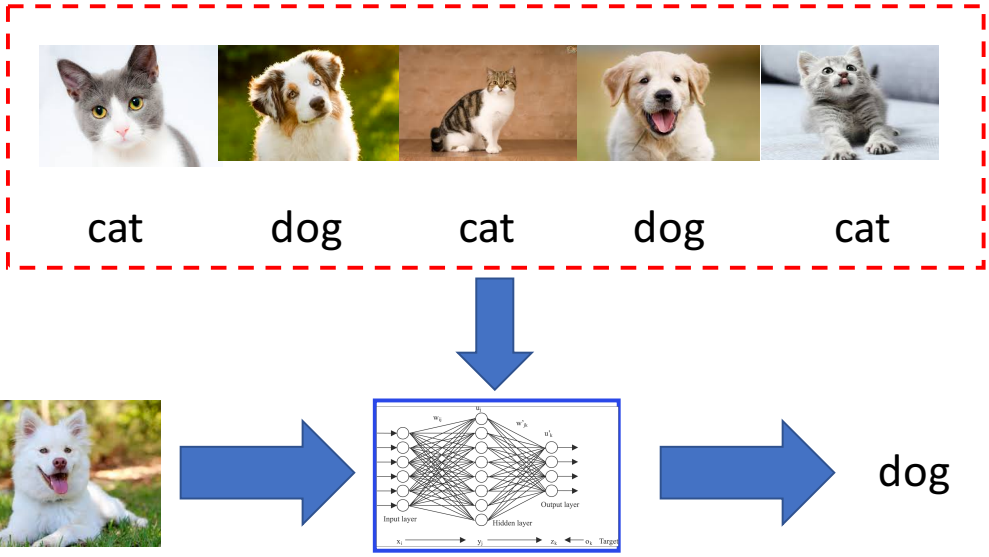
Review

THE UNIVERSITY OF
SYDNEY

Dependence

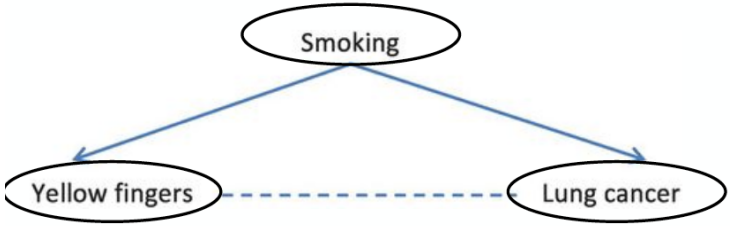
THE UNIVERSITY OF SYDNEY

Machine learning systems are usually driven by statistical dependence.

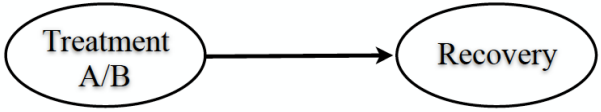


Ways to Produce Dependence

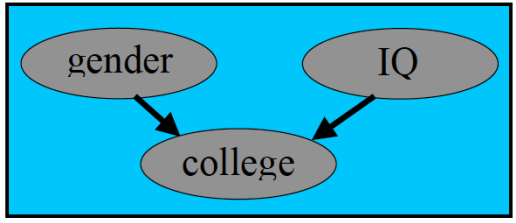
causality contains more information than dependence



Common Cause



Causal relation



Conditional dependence given common effect

common effect: college

THE UNIVERSITY OF SYDNEY

Causal thinking

Causation vs Dependence

can do prediction
but can't do intervention

THE UNIVERSITY OF SYDNEY

Causal representation

Markov Conditions

THE UNIVERSITY OF SYDNEY

- Markov conditions state that if a certain graph property holds true, then a certain statistically independence holds true.

- There are local Markov condition and global Markov condition.

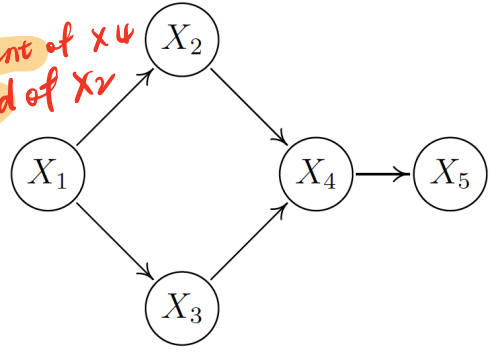
Local Markov Condition

Every variable X_i, in a directed acyclic graph, is independent of its non-descendant Y conditional on its parents, i.e., $(X_i \perp\!\!\!\perp Y \,|\, PA_i)_G$, which also implies $(X_i \perp\!\!\!\perp Y \,|\, PA_i)_p$.

X_1, X_2 are the ancestor of X_4
X_4, X_5 are the descendants of X_2
X_2 causes X_4 $\Rightarrow$ X_2 is the parent of X_4
X_4 is a child of X_2

For example:

$(X_4 \perp\!\!\!\perp X_1 \,|\, \{X_2, X_3\})_G \implies (X_4 \perp\!\!\!\perp X_1 \,|\, \{X_2, X_3\})_p.$



By the local Markov condition,
we could obtain a casual factorisation of the joint distribution as follows

$$P(X_1,...,X_5) = P(X_1)P(X_2 \mid X_1)P(X_3 \mid X_1)P(X_4 \mid X_2, X_3)P(X_5 \mid X_4)$$

Global Markov Conditions (D-Separation)

THE UNIVERSITY OF SYDNEY

element should be different

For three disjoint sets of variables $\mathbf{X}$, $\mathbf{Y}$, and $\mathbf{S}$, $\mathbf{X}$ is d-separated from $\mathbf{Y}$ conditional on $\mathbf{S}$ if and only if all paths between any member of $\mathbf{X}$ and any member of $\mathbf{Y}$ are *blocked* by $\mathbf{S}$.

path don't use direction

A path q is said to be blocked by the set $\mathbf{S}$ if

- q contains a chain $i \to m \to j$ or a fork $i \leftarrow m \to j$ such that the middle node m is in $\mathbf{S}$, or
- q contains a collider $i \to m \leftarrow j$ such that the middle node m is not in $\mathbf{S}$, and no descendant of m is in $\mathbf{S}$.

Formally, we use $(\mathbf{X} \perp\!\!\!\perp \mathbf{Y} \,|\, \mathbf{S})_G$ to denote that $\mathbf{S}$ d-separates $\mathbf{X}$ and $\mathbf{Y}$ in the DAG G, which also implies $(\mathbf{X} \perp\!\!\!\perp \mathbf{Y} \,|\, \mathbf{S})_p$.

~~Casual~~ Causal faithfulness Assumption

THE UNIVERSITY OF SYDNEY

The probability distribution may have additional conditional independence relations that are not entailed by d-separation applied to a graph. When no such extra conditional independence relations hold the distribution is said to be faithful to the graph, i.e.,

X, Y are d-seperated

$$(\mathbf{X} \perp\!\!\!\perp \mathbf{Y} \,|\, \mathbf{S})_p \implies (\mathbf{X} \perp\!\!\!\perp \mathbf{Y} \,|\, \mathbf{S})_G,$$

then the distribution p is said to be faithful to the graph.

Conditioning, Intervention, Counterfactual

THE UNIVERSITY OF SYDNEY

- **Prediction/Conditioning**
  - would the pavement be slippery if we find the sprinkler off?

with dependency information
statistical dependence

$$P(slippery \mid Sprinkler = off)$$

- **Intervention**
  - would the pavement be slippery if we turn off the sprinkler?

$$P(slippery \mid do(Sprinkler = off))$$

- **Prediction/Counterfactual reasoning**
  - would the pavement be slippery, had the sprinkler been off, given that the pavement is in fact not slippery and the sprinkler is on?

$$P(slippery_{sprinkler=off} \mid Sprinkler = on, Slippery = no)$$

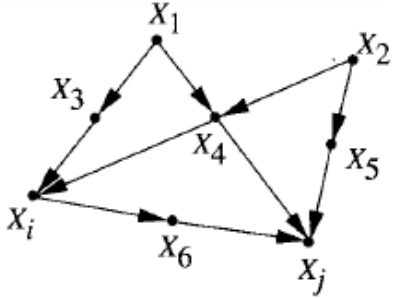
Back-Door Criterion

THE UNIVERSITY OF SYDNEY

Definition 3.3.1 (Back-Door)
A set of variables Z satisfies the back-door *criterion relative to an ordered pair of variables* (X_i, X_j) *in a DAG G if:*

(i) *no node in Z is a descendant of* X_i; *and*

(ii) *Z blocks every path between* X_i *and* X_j *that contains an arrow into* X_i.



- What if $Z = \{X_3, X_4\}$?
  $Z = \{X_4, X_5\}$?
  $Z = \{X_4\}$?
- What if there is a confounder?

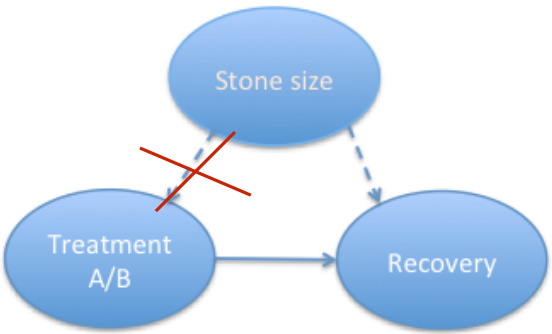
Theorem 3.3.2 (Back-Door Adjustment)
If a set of variables Z satisfies the back-door criterion relative to (X, Y), *then the causal effect of X on Y is identifiable and is given by the formula*

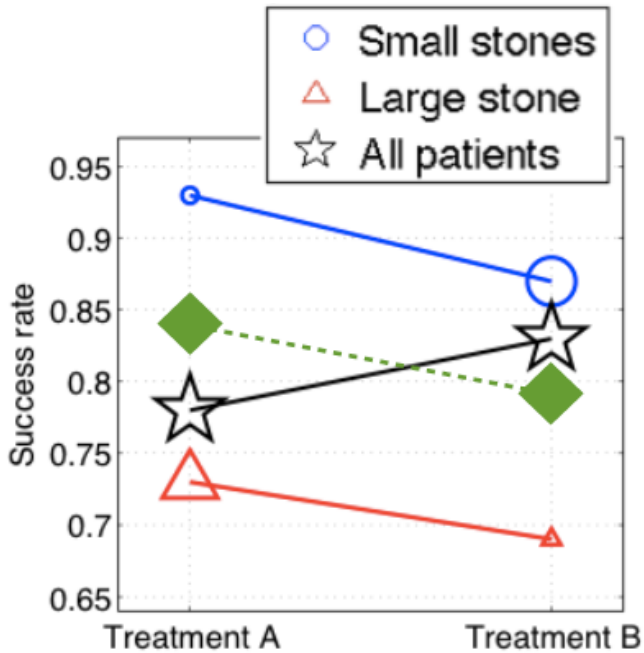
$$P(y \mid \hat{x}) = \sum_z P(y \mid x, z) P(z).$$

Example

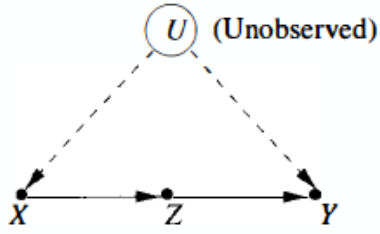
	Treatment A	Treatment B
Small Stones	*Group 1* **93% (81/87)**	*Group 2* 87% (234/270)
Large Stones	*Group 3* **73% (192/263)**	*Group 4* 69% (55/80)
Both	78% (273/350)	**83% (289/350)**

T - Treatment
R - Recovery





Front-Door Criterion



Definition 3.3.3 (Front-Door)
A set of variables Z is said to satisfy the front-door *criterion relative to an ordered pair of variables (X, Y) if:*

(i) *Z intercepts all directed paths from X to Y;*

(ii) *there is no back-door path from X to Z; and*

(iii) *all back-door paths from Z to Y are blocked by X.*

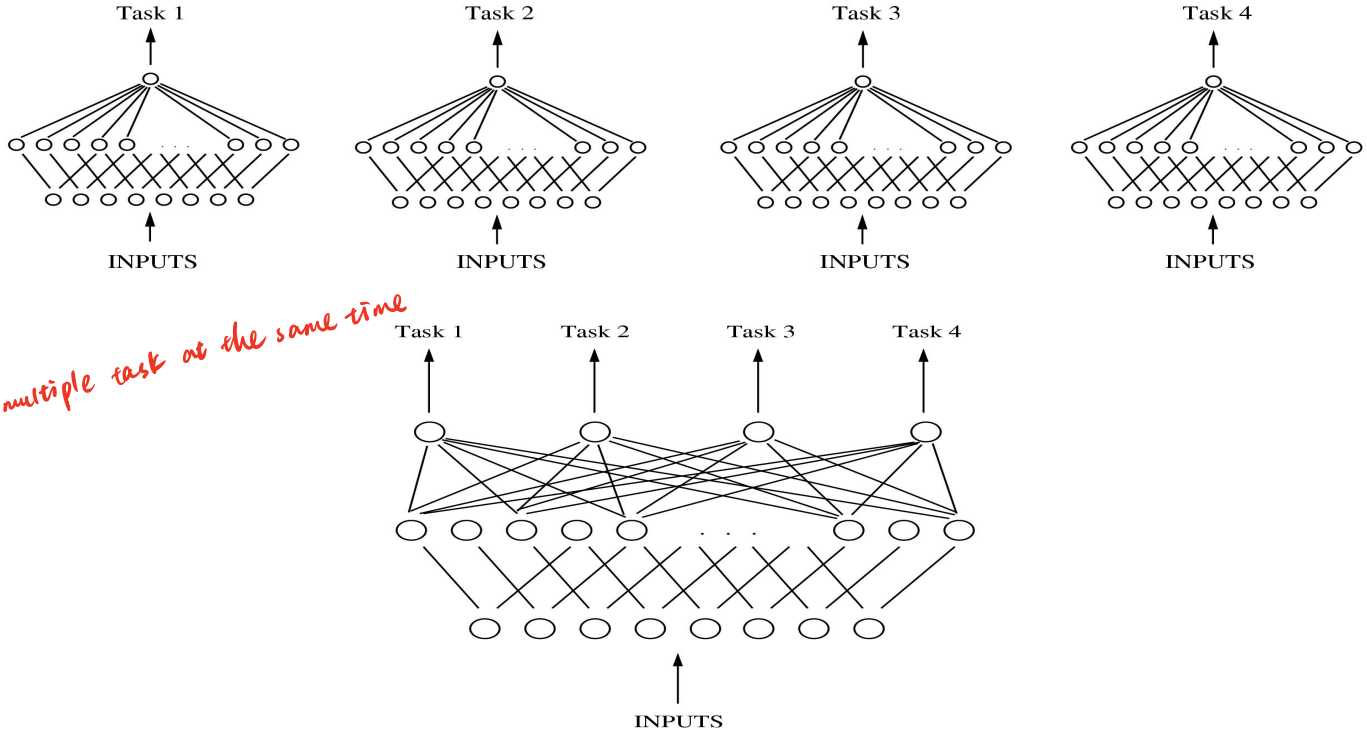
Theorem 3.3.4 (Front-Door Adjustment)
If Z satisfies the front-door criterion relative to (X, Y) and if $P(x, z) > 0$, then the causal effect of X on Y is identifiable and is given by the formula

$$P(y \mid \hat{x}) = \sum_{z} P(z \mid x) \sum_{x'} P(y \mid x', z) P(x'). \tag{3.29}$$

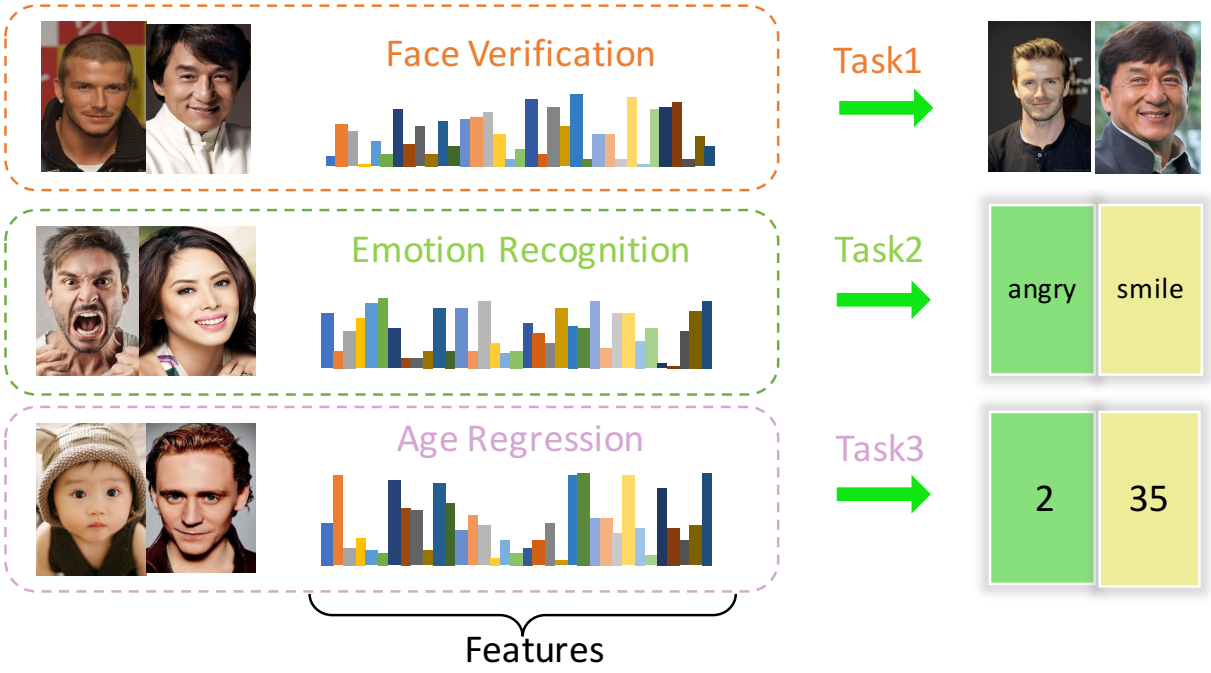
THE UNIVERSITY OF
SYDNEY

Multi-Task Learning

Motivation Examples: STL vs MTL



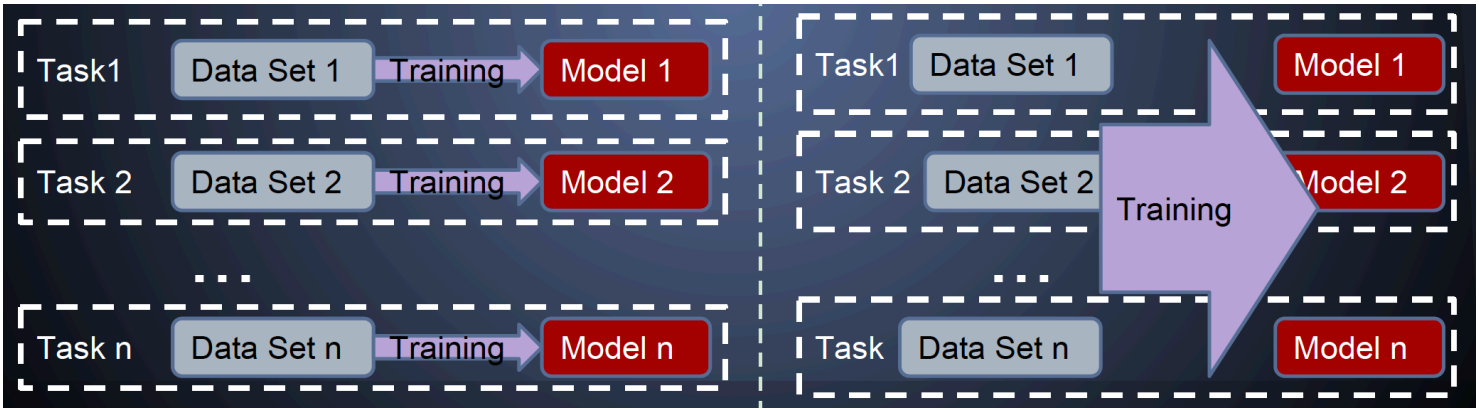
Motivation Examples: STL vs MTL



Concepts and General View

Multi-task Learning is an approach to learn multiple related problems at the same time, by exploiting the relatedness between problems.

why better performance ?



Relatedness among Tasks

Learning tasks with the aim of mutual benefit.

Assumption : Tasks are related.

Example: Spam filtering - Everybody has a slightly different distribution over spam or non-spam emails, but there is a common aspect across users.

When tasks are independent to each other, multi-task learning will have no advantage to single task learning.

Big Data Interpretation

However, in some applications, large training examples are hard to collect, such as medical image analysis.

For this data insufficient problem, Multi-Task Learning (MTL) is a good solution when there are multiple related tasks each of which has limited training samples.

Problem Setup

Given m learning tasks $\{\mathcal{T}_i\}_{i=1}^m$ where all the tasks or a subset of them are related, multi-task learning aims to help improve the learning of a model for $\mathcal{T}_i$ by using the knowledge contained in all or some of the m tasks.

The task $\mathcal{T}_i$ is accompanied by a training set $\mathcal{D}_i = \{\mathbf{x}_j^i, y_j^i\}_{j=1}^{n_i}$.

Our task is to learn hypotheses for $\{\mathcal{T}_i\}_{i=1}^m$.

MTL with Shared Knowledge

Multi-task Learning exploits the relatedness among tasks.
There are some shared knowledge across tasks.

Three questions in MTL with shared knowledge:
When to Share? What to Share? How to Share?

When to Share?

When there are relatednesses among tasks.

What to Share?

$$R_S(h) = \frac{1}{n}\sum_{i=1}^{n}\ell(X_i, Y_i, h)$$

We consider two cases: share parameters and features.

How to Share?

Parameter-based MTL and feature-Based MTL models.

MTL models

Consider the linear hypothesis function, i.e., $h(x) = w^\top x.$

We have m different but related tasks, i.e., $\{\mathcal{T}_i\}_{i=1}^m$.

Denote w^i as the hypothesis for the i^{th} task, $i = 1, \ldots, m.$

The following empirical risk minimisation algorithm learns multiple tasks simultaneously, is it superior to learning the tasks individually?

could be worse
haven't learn the relatedness yet

$$\min_{W=[w^1,\ldots,w^m]} \frac{1}{m} \sum_{i=1}^{m} \frac{1}{n_i} \sum_{j=1}^{n_i} \ell(x_j^i, y_j^i, w^i).$$

MTL models

$$\min_{W=[w^1,\ldots,w^m]} \frac{1}{m}\sum_{i=1}^{m}\frac{1}{n_i}\sum_{j=1}^{n_i}\ell(x_j^i, y_j^i, w^i).$$

$= \frac{1}{m n_1}\Sigma\ell(\cdot, w^1) + \frac{1}{m n_2}\Sigma\ell(\cdot, w^2) \ldots$

VS

$$\min_{w^i} \frac{1}{n_i}\sum_{j=1}^{n_i}\ell(x_j^i, y_j^i, w^i), i = 1, \ldots, m.$$

Which group will have a better performance?

Parameter-based MTL models

We assume the multiple tasks are related by their parameters that

$$w^i = w_0 + \Delta w^i$$

where $i = 1, \ldots, m$.

common parameters learned by making use of all examples

no relationship, $w_0 = 0$

$$\min_{w_0, \Delta W = [\Delta w^1, \ldots, \Delta w^m]} \frac{1}{m} \sum_{i=1}^{m} \frac{1}{n_i} \sum_{j=1}^{n_i} \ell(x_j^i, y_j^i, w_0 + \Delta w^i).$$

learn from all tasks

only learn from task i

VS

$$\min_{w^i} \frac{1}{n_i} \sum_{j=1}^{n_i} \ell(x_j^i, y_j^i, w^i), i = 1, \ldots, m.$$

Which group will have a better performance?

Parameter-based MTL models

Can we improve the following MTL model?

$$\min_{w_0, \Delta W = [\Delta w^1, \ldots, \Delta w^m]} \frac{1}{m} \sum_{i=1}^{m} \frac{1}{n_i} \sum_{j=1}^{n_i} \ell(x_j^i, y_j^i, w_0 + \Delta w^i).$$

regulariser

$$\min_{w_0, \Delta W = [\Delta w^1, \ldots, \Delta w^m]} \frac{1}{m} \sum_{i=1}^{m} \frac{1}{n_i} \sum_{j=1}^{n_i} \ell(x_j^i, y_j^i, w_0 + \Delta w^i) + \lambda \|\Delta W\|_F^2.$$

① push ΔW small
⇒ W_0 large
want MTL to share more tasks to better learn

The latter model is better because it pushes the multi-task learning algorithm to have stronger relatedness.

Parameter-based MTL models

Given a matrix M, the **rank** of a matrix is the maximum number of linearly independent columns.

$a, b, c \in \mathbb{R}^d$ $\lambda_a \in R$ $\lambda_b \in R$

$c = \lambda_a \cdot a + \lambda_b \cdot b$

if there does not exist λ_a, λ_b.

c is linearly independent between a and b

A rank 2 matrix: $S = \begin{bmatrix} 3 & 0 & 1 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \end{bmatrix}$

Low-rank based MTL model (I):

with low rank

→ if w^3 is similar to w^1, w^2 use w^1, w^2 to represent w^3

learn small number of parameter

$$\min_{W=[w^1,\ldots,w^m]} \frac{1}{m} \sum_{i=1}^{m} \frac{1}{n_i} \sum_{j=1}^{n_i} \ell(x_j^i, y_j^i, w^i) + \lambda \, \mathrm{rank}(W).$$

if rank = 1

w^i are related to each other with different scale

Parameter-based MTL models

Low-rank based MTL model (II):

Specifically, we assume that

for different w^i, share the same $\Theta^\top$

$$w^i = u^i + \Theta^\top v^i,$$

where $i = 1, \ldots, m$, and $\Theta \in \mathbb{R}^{h \times d}$ is the shared low-rank subspace by multiple tasks. Then we have

$$W = U + \Theta^\top V.$$

Ando, Rie Kubota, and Tong Zhang. "A framework for learning predictive structures from multiple tasks and unlabeled data." Journal of Machine Learning Research 6.Nov (2005): 1817-1853.

Parameter-based MTL models

Low-rank based MTL model (II):

commonly shared part enlarge

$$\min_{U,V,\Theta} \frac{1}{m}\sum_{i=1}^{m}\frac{1}{n_i}\sum_{j=1}^{n_i}\ell(x_j^i, y_j^i, u^i + \Theta^\top v^i) + \lambda\|U\|_F^2,$$

$$s.t.\ \Theta^\top\Theta = I.$$

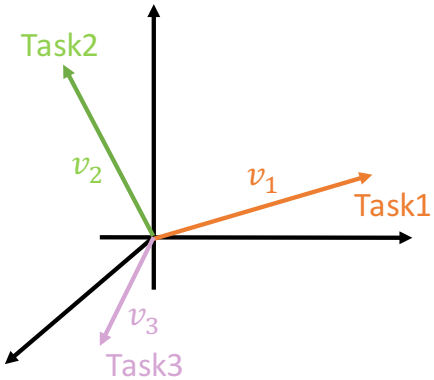
orthogonal matrix

Note that the orthogonal constraint makes the subspace non-redundant.

Feature-based MTL models

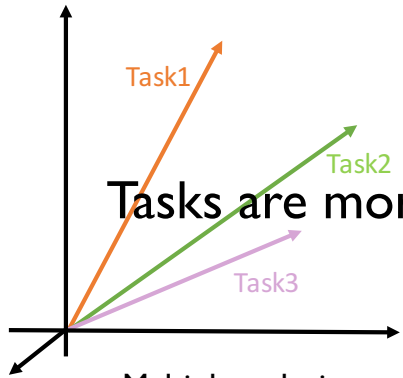
Note that the hypotheses are learned from training example.

By using $\mathcal{D}_i = \{x_j^i, y_j^i\}_{j=1}^{n_i}$ we have



Multiple tasks in original feature space

Can we map the features such that, $\mathcal{D}_i = \{P^\top x_j^i, y_j^i\}_{j=1}^{n_i}$



Multiple tasks in new feature space

Feature-based MTL models

Feature-based MTL model (I):

project to a new space

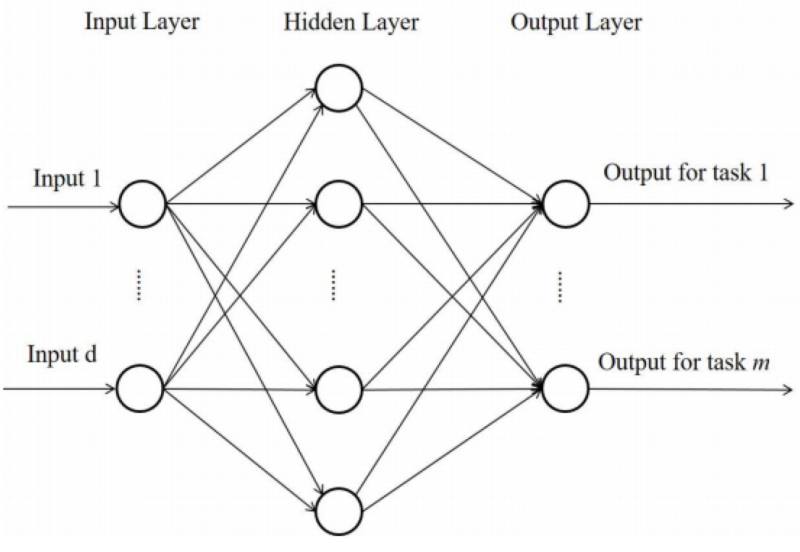
$$\min_{W,P} \frac{1}{m} \sum_{i=1}^{m} \frac{1}{n_i} \sum_{j=1}^{n_i} \ell(P^\top x_j^i, y_j^i, w^i) + \lambda \text{rank}(W),$$

$$s.t.\ PP^\top = I.$$

Note that P is a projection matrix.

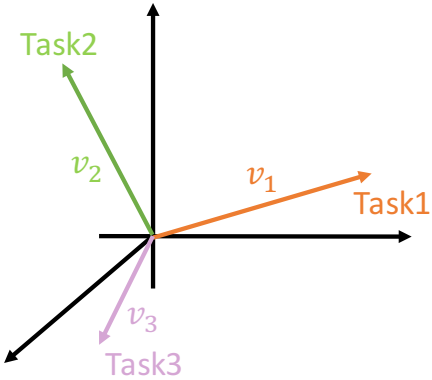
Feature-based MTL models

Feature-based MTL model (II):

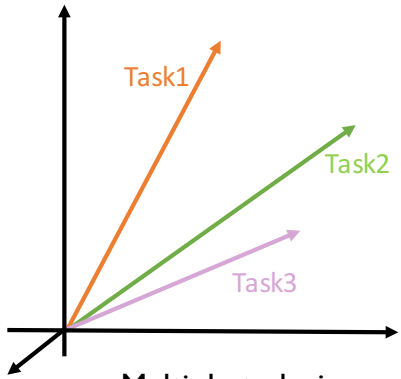


Shared Hidden nodes in a Neural Network. Note that neural network can be regarded as feature extractors.

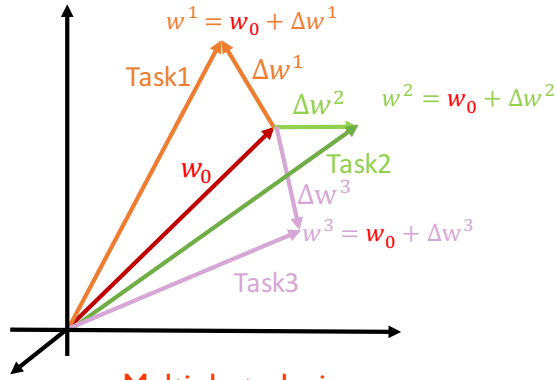
Feature- and Parameter-based MTL models



Multiple tasks in original feature space



Multiple tasks in new feature space



Multiple tasks in new feature space

Feature- and Parameter-based MTL models

$$\min_{w_0,\Delta W,P} \frac{1}{m}\sum_{i=1}^{m}\frac{1}{n_i}\sum_{j=1}^{n_i}\ell(P^\top x_j^i, y_j^i, w_0+\Delta w^i)+\lambda\|\Delta W\|_F^2,$$

$$s.t.\ PP^\top = I.$$

The above model learns the feature projection map P and the commonly shared parameter w_0 to enhance the relatedness among tasks.

Li, Ya, Xinmei Tian, Tongliang Liu, and Dacheng Tao. "Multi-Task Model and Feature Joint Learning." In IJCAI, pp. 3643-3649. 2015.

Why multi-task works?

Compared with single task learning, will all the tasks' performances be improved? NOT ALWAYS.

if one task doesn't share with other tasks

Is it possible that some task will be harmful to boost the performance? like a black sheep.

Relationship to Transfer Learning

assume : shared knowledge between { source and target domain → Transfer learning
multiple tasks → MTL

has some specific task.

In MTL, there is no distinction among different tasks and the objective is to improve the performance of all the tasks. However, in transfer learning which is to improve the performance of a target task with the help of source tasks, the target task plays a more important role than source tasks.

More about MTL

Online MTL, distributed MTL: handling large data sets.

A Survey on Mutli-Task learning: https://arxiv.org/pdf/1707.08114.pdf

An Overview of Multi-Task Learning in Deep Neural Networks: http://ruder.io/multi-task/

Matlab codes for many variants of MTL: http://jiayuzhou.github.io/MALSAR/