

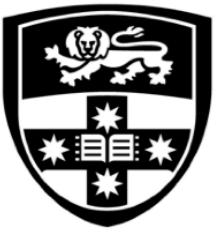
THE UNIVERSITY OF
SYDNEY

Advanced Machine Learning

(COMP 5328)

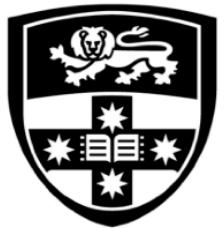
Multi-Task Learning

Tongliang Liu



THE UNIVERSITY OF
SYDNEY

Review

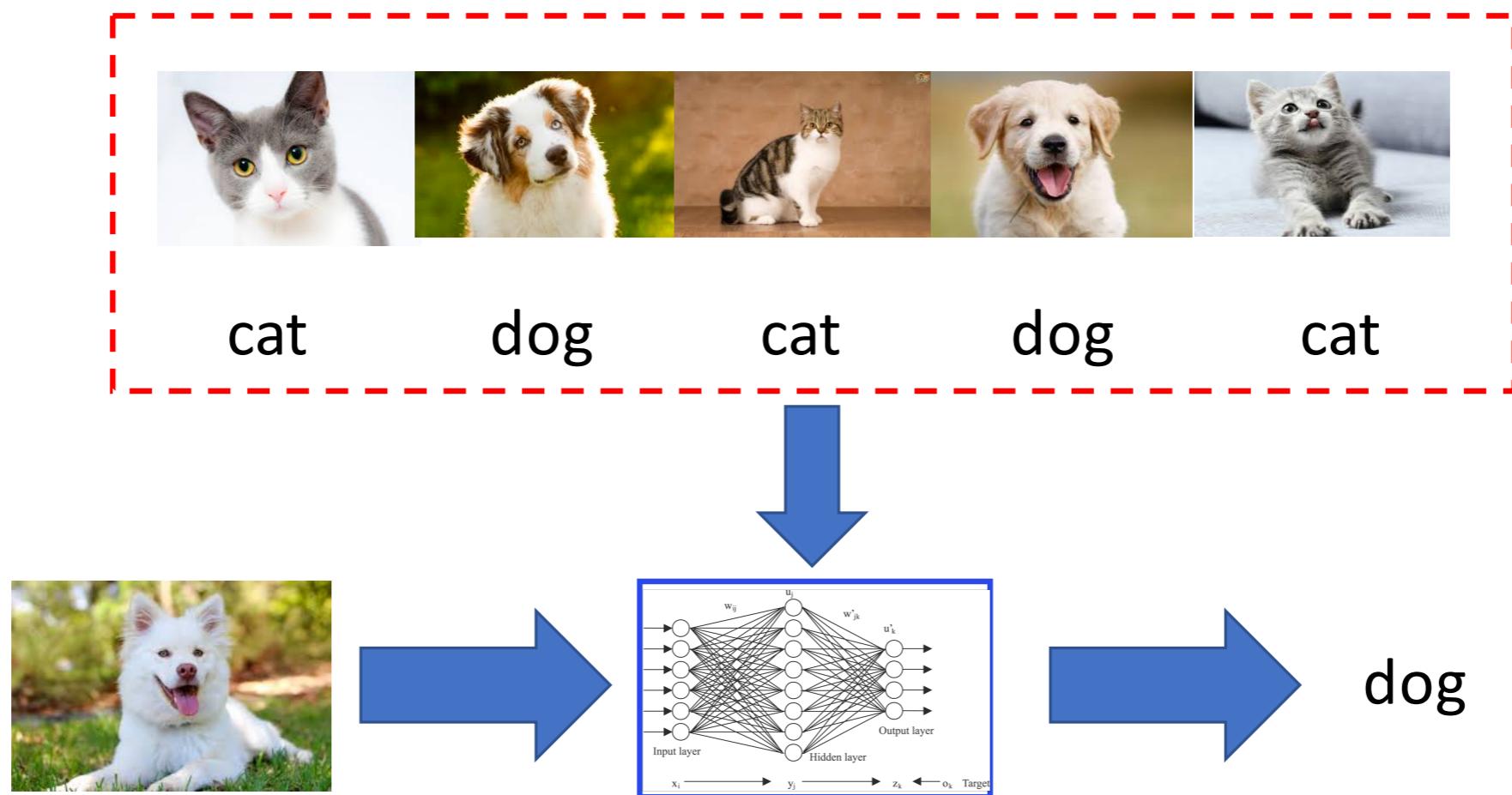


THE UNIVERSITY OF
SYDNEY

Dependence

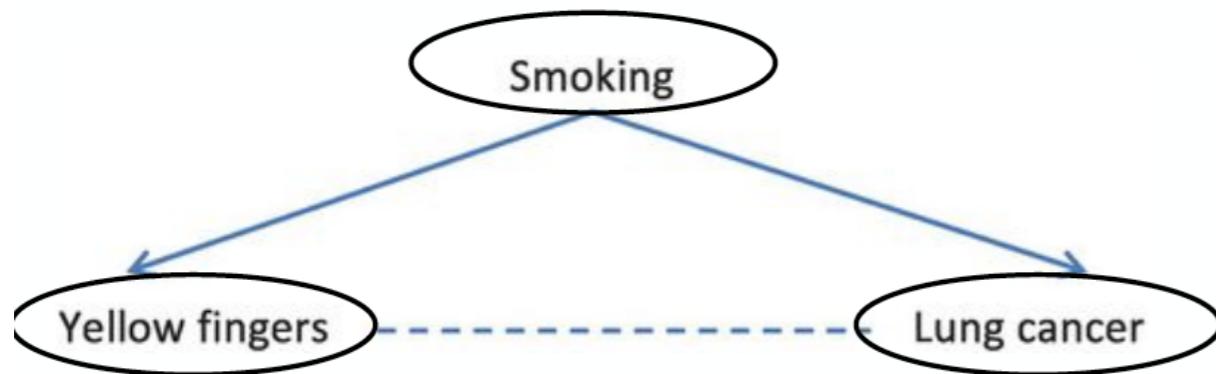


Machine learning systems are usually driven by statistical dependence.





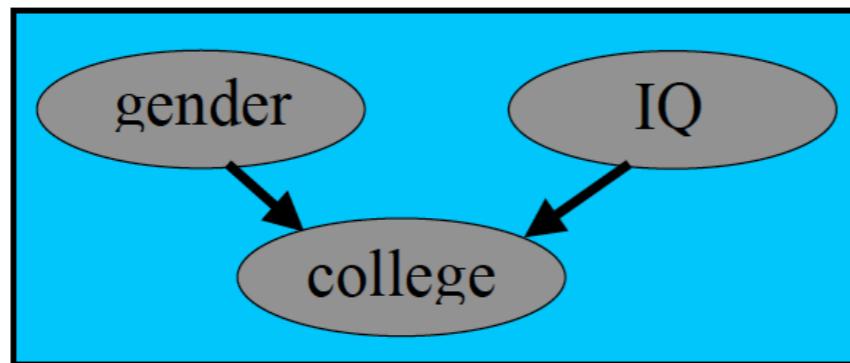
Ways to Produce Dependence



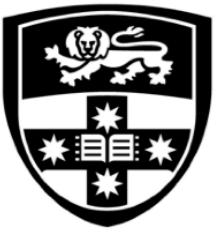
Common Cause



Causal relation



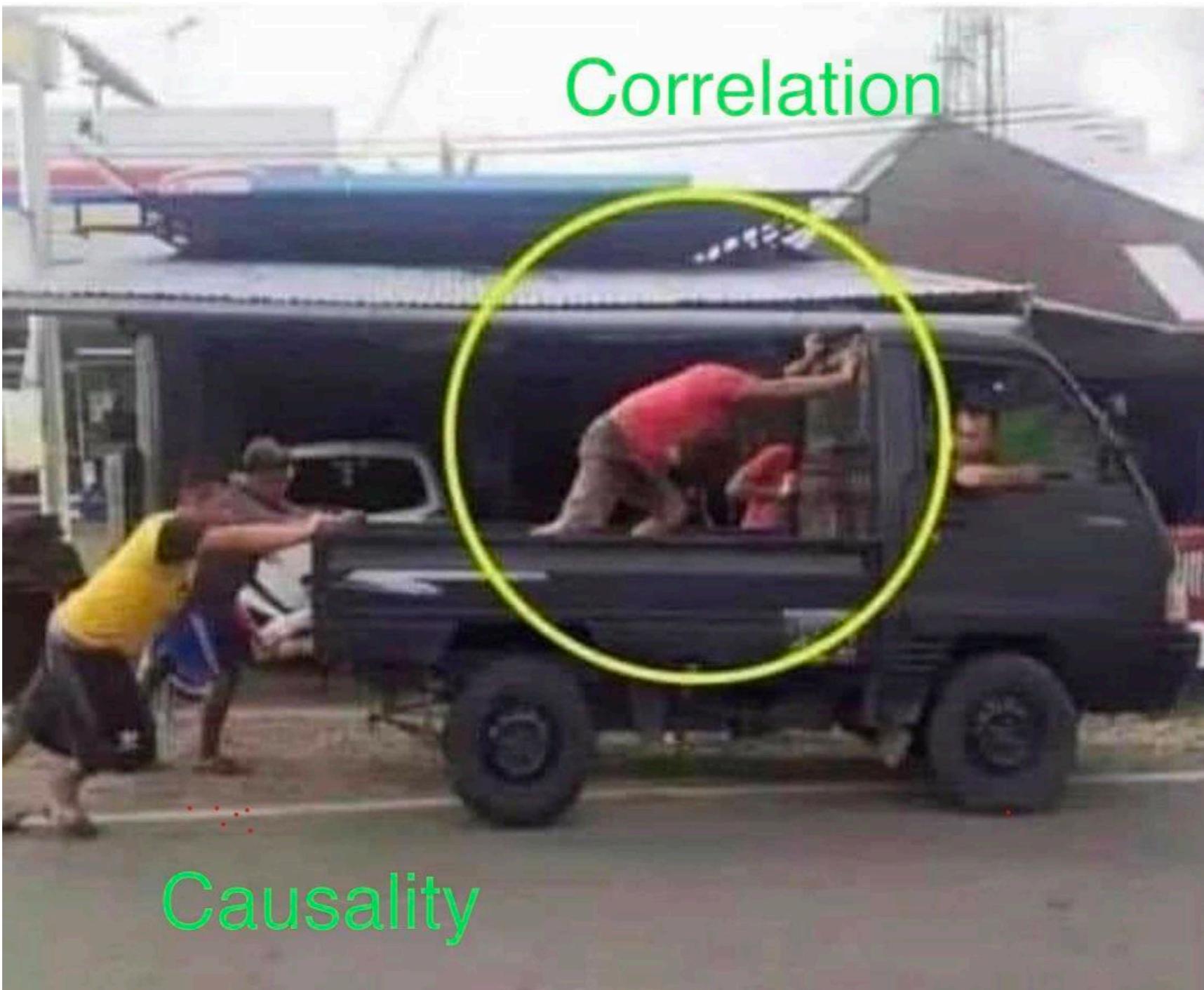
Conditional dependence
given common effect

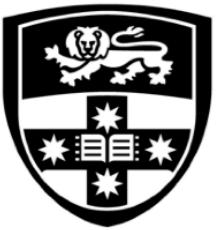


THE UNIVERSITY OF
SYDNEY

Causal thinking

Causality vs Dependence





THE UNIVERSITY OF
SYDNEY

Causal representation



THE UNIVERSITY OF
SYDNEY

Markov Conditions

- 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

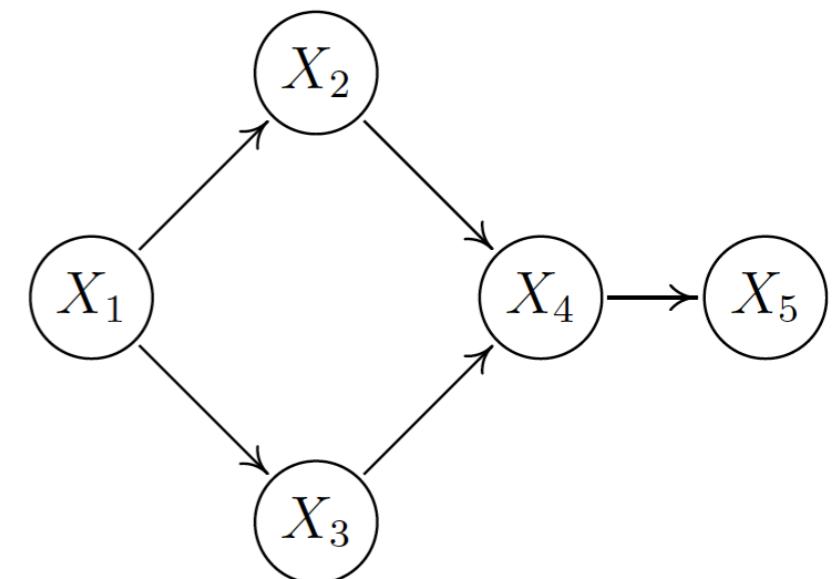
Every variable X_i , in a directed acyclic graph, is independent of its non-descendant Y conditional on its parents, i.e., $(X_i \perp\!\!\!\perp Y | PA_i)_G$, which also implies $(X_i \perp\!\!\!\perp Y | PA_i)_p$.

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 causal factorisation of the joint distribution as follows

$$P(X_1, \dots, X_5) = P(X_1)P(X_2 | X_1)P(X_3 | X_1)P(X_4 | X_2, X_3)P(X_5 | X_4)$$





Global Markov Conditions (D-Separation)

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} .

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

- q contains a chain $i \rightarrow m \rightarrow j$ or a fork $i \leftarrow m \rightarrow j$ such that the middle node m is in \mathbf{S} , or
- q contains a collider $i \rightarrow m \leftarrow j$ 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 faithfulness Assumption

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.,

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

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



Conditioning, Intervention, Counterfactual

- **Prediction/Conditioning**
 - would the pavement be slippery if we find the sprinkler off?
$$P(\text{slippery} \mid \text{Sprinkler} = \text{off})$$
- **Intervention**
 - would the pavement be slippery if we turn off the sprinkler?
$$P(\text{slippery} \mid \text{do}(\text{Sprinkler} = \text{off}))$$
- **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?
$$P(\text{slippery}_{\text{sprinkler}=\text{off}} \mid \text{Sprinkler} = \text{on}, \text{Slippery} = \text{no})$$

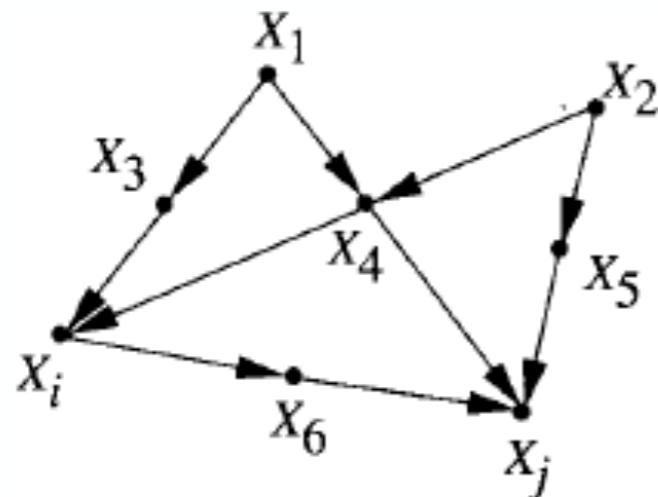


Back-Door Criterion

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?

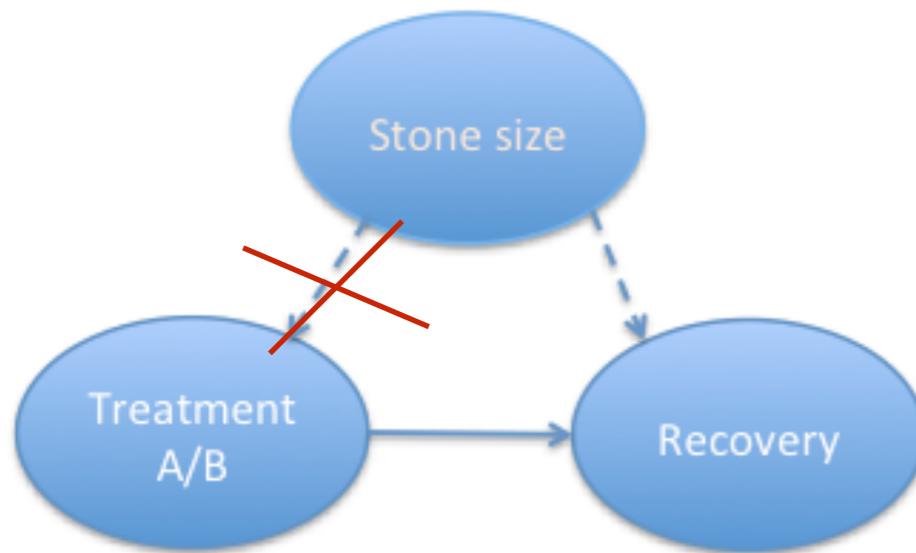
Theorem 3.3.2 (Back-Door Adjustment)

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

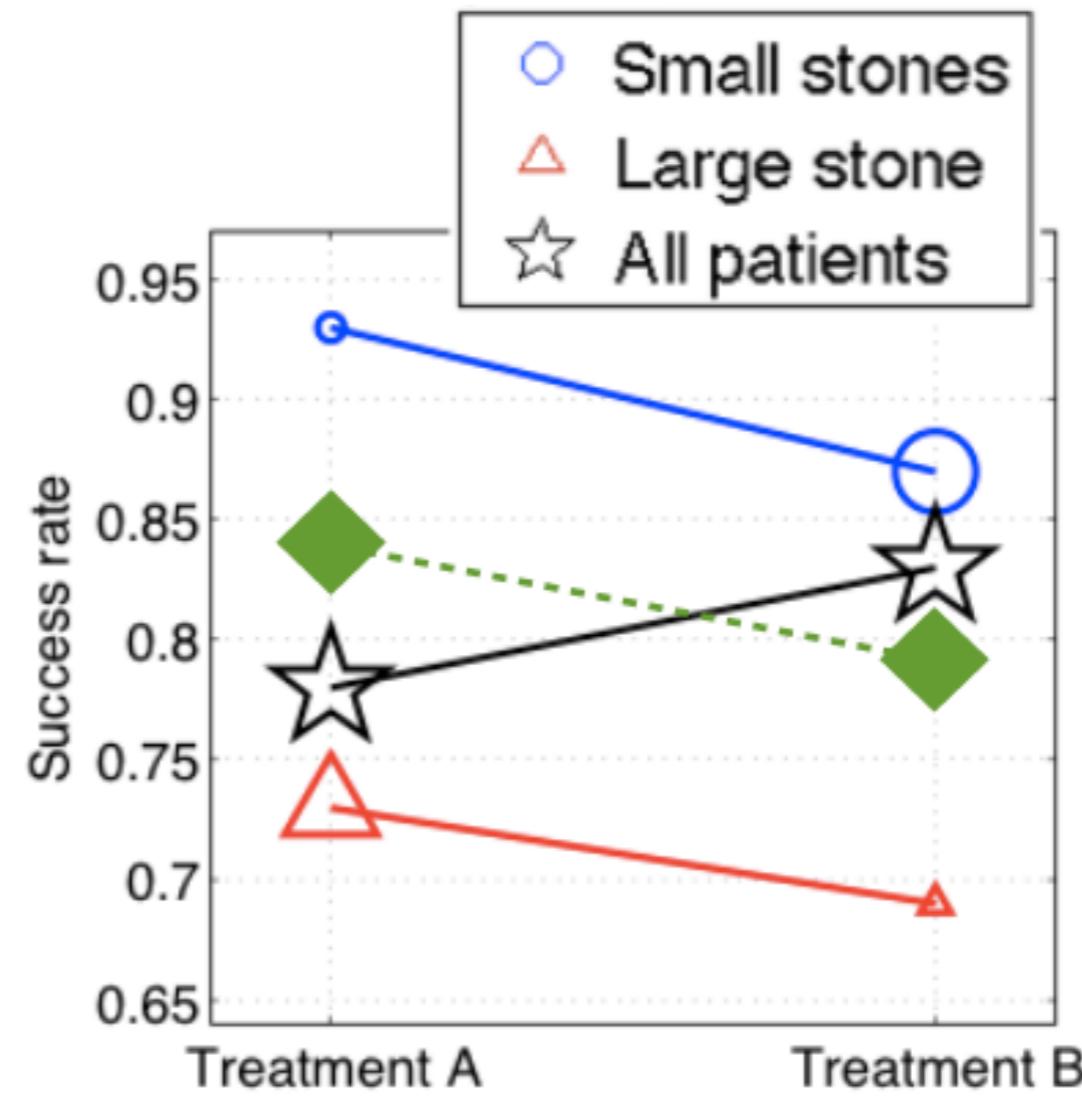
$$P(y | \hat{x}) = \sum_z P(y | x, z) P(z).$$

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)

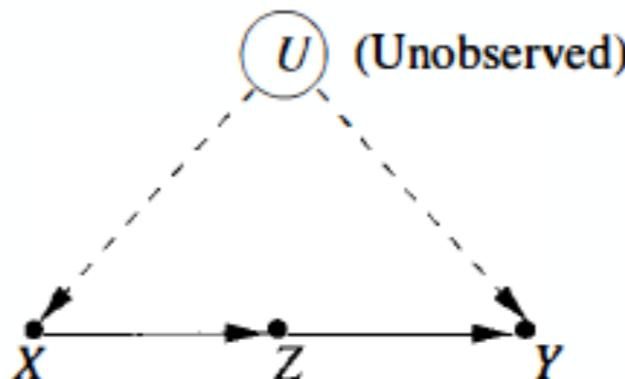


T - Treatment
R - Recovery





Front-Door Criterion



Definition 3.3.3 (Front-Door)

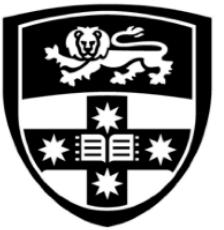
A set of variables Z is said to satisfy the front-door criterion relative to an ordered pair of variables (X, Y) if:

- (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

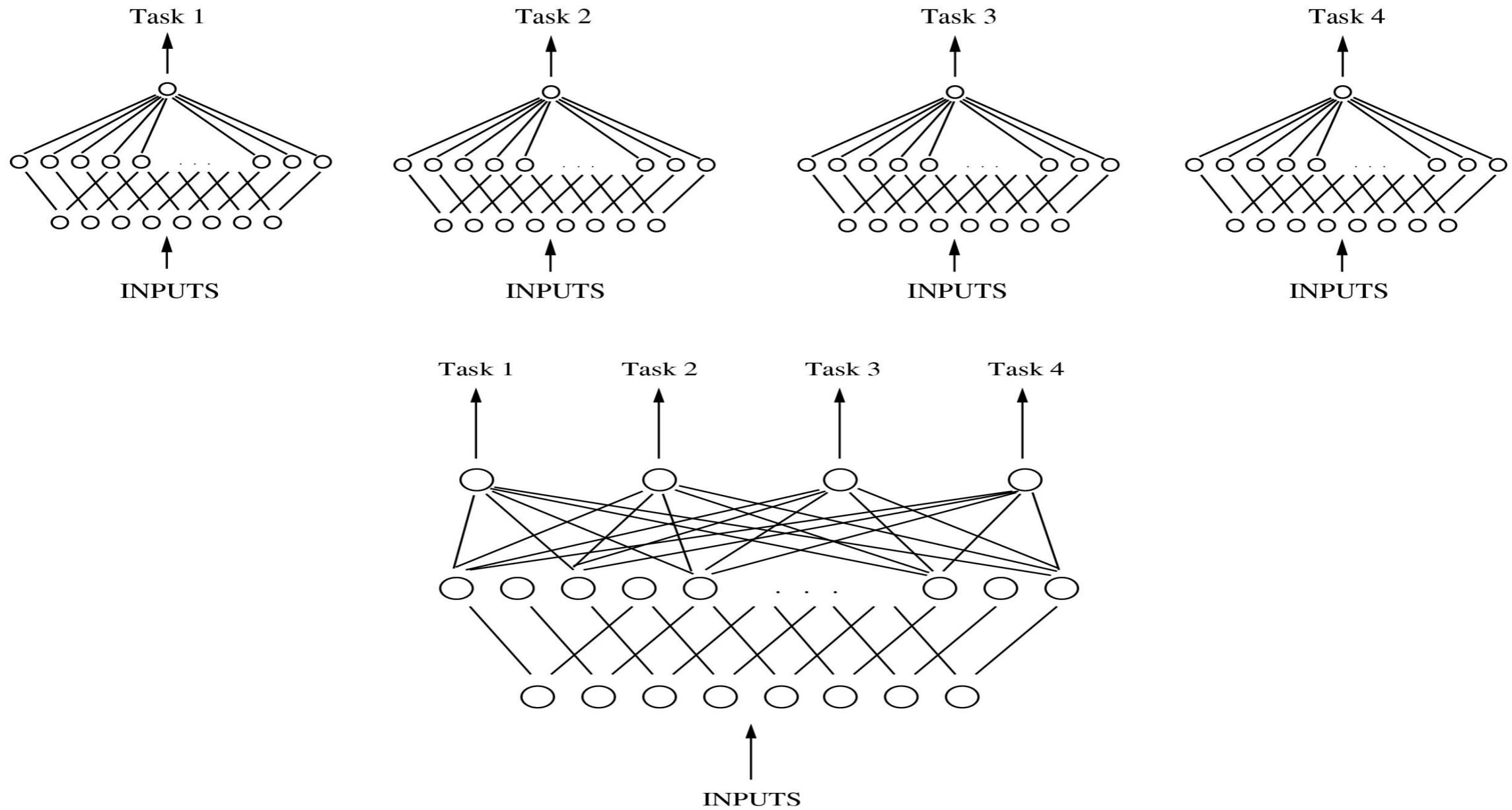
$$P(y | \hat{x}) = \sum_z P(z | x) \sum_{x'} P(y | x', z) P(x'). \quad (3.29)$$



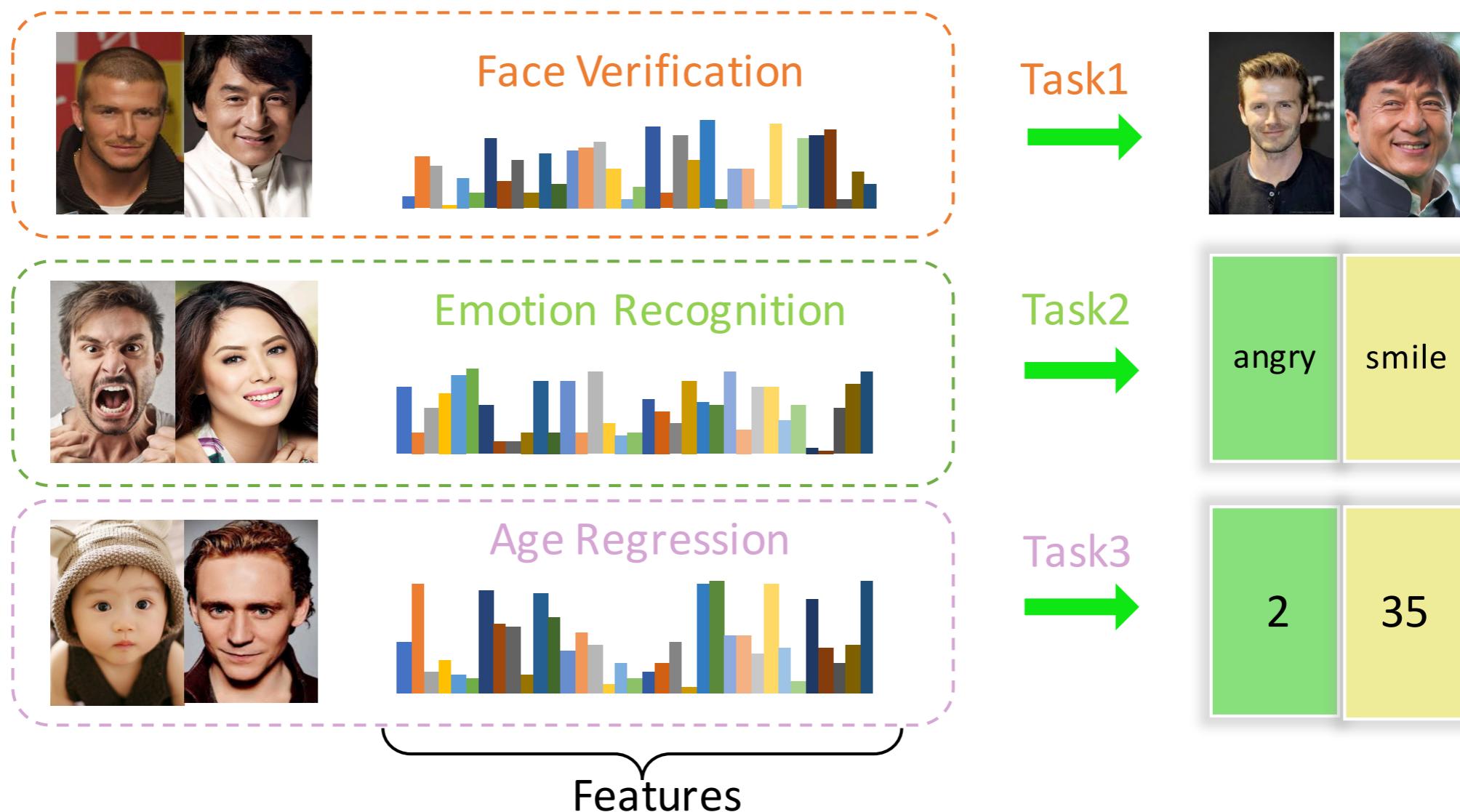
THE UNIVERSITY OF
SYDNEY

Multi-Task Learning

Motivation Examples: STL vs MTL

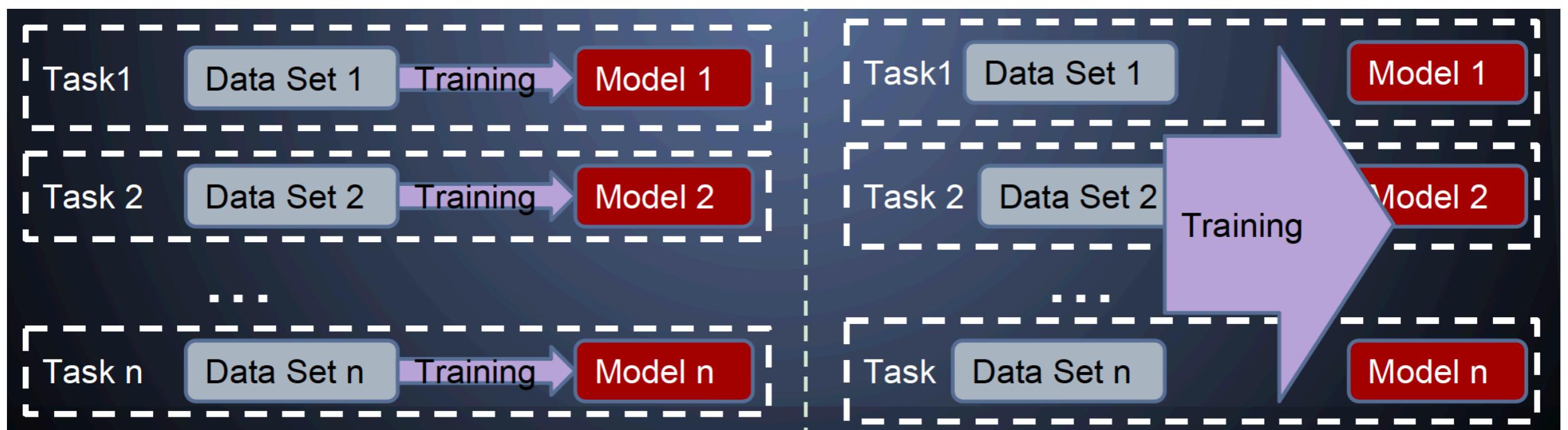


Motivation Examples: STL vs MTL



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.



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, $i = 1, \dots, m$.

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

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

MTL models

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

VS

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

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 $i = 1, \dots, m$.

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

VS

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

Which group will have a better performance?

Parameter-based MTL models

Can we improve the following MTL model?

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

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

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 rank 2 matrix:

$$S = \begin{bmatrix} 3 & 0 & 1 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Low-rank based MTL model (I):

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

Parameter-based MTL models

Low-rank based MTL model (II):

Specifically, we assume that

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

where $i = 1, \dots, m$, 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):

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

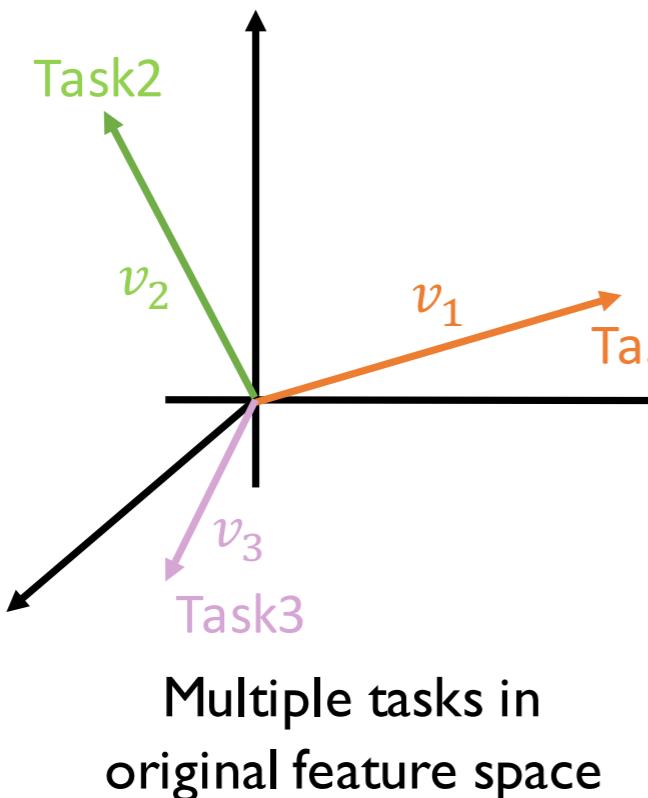
$$s.t. \quad \Theta^\top \Theta = I.$$

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

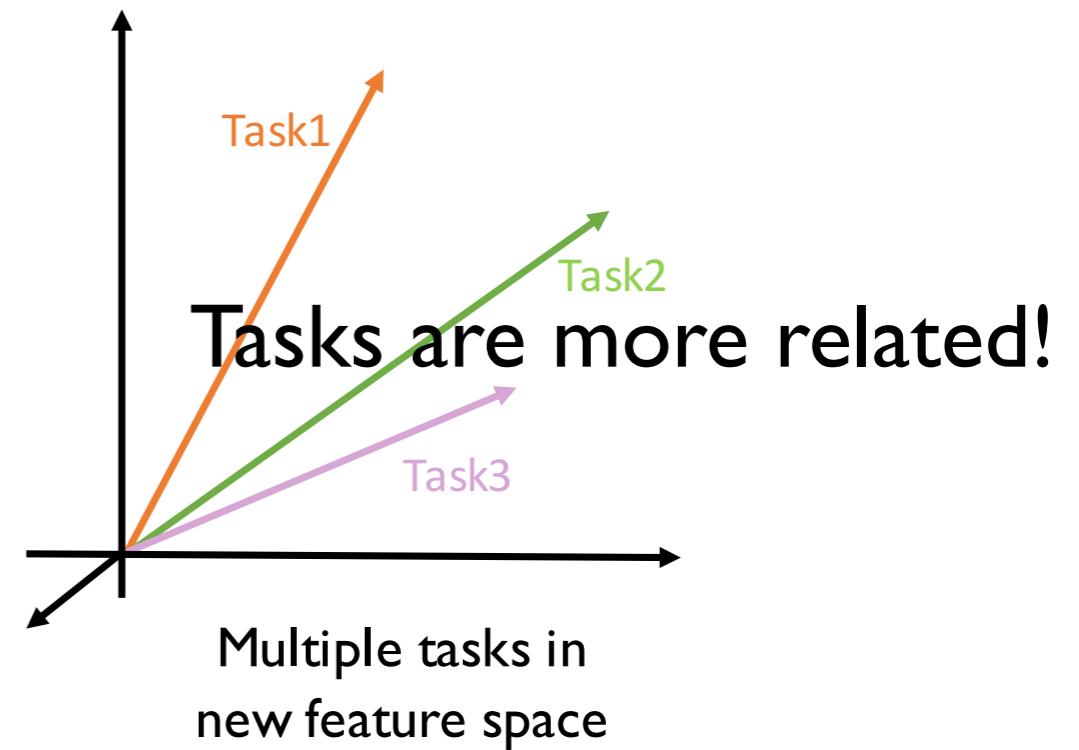
Feature-based MTL models

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



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



Feature-based MTL models

Feature-based MTL model (I):

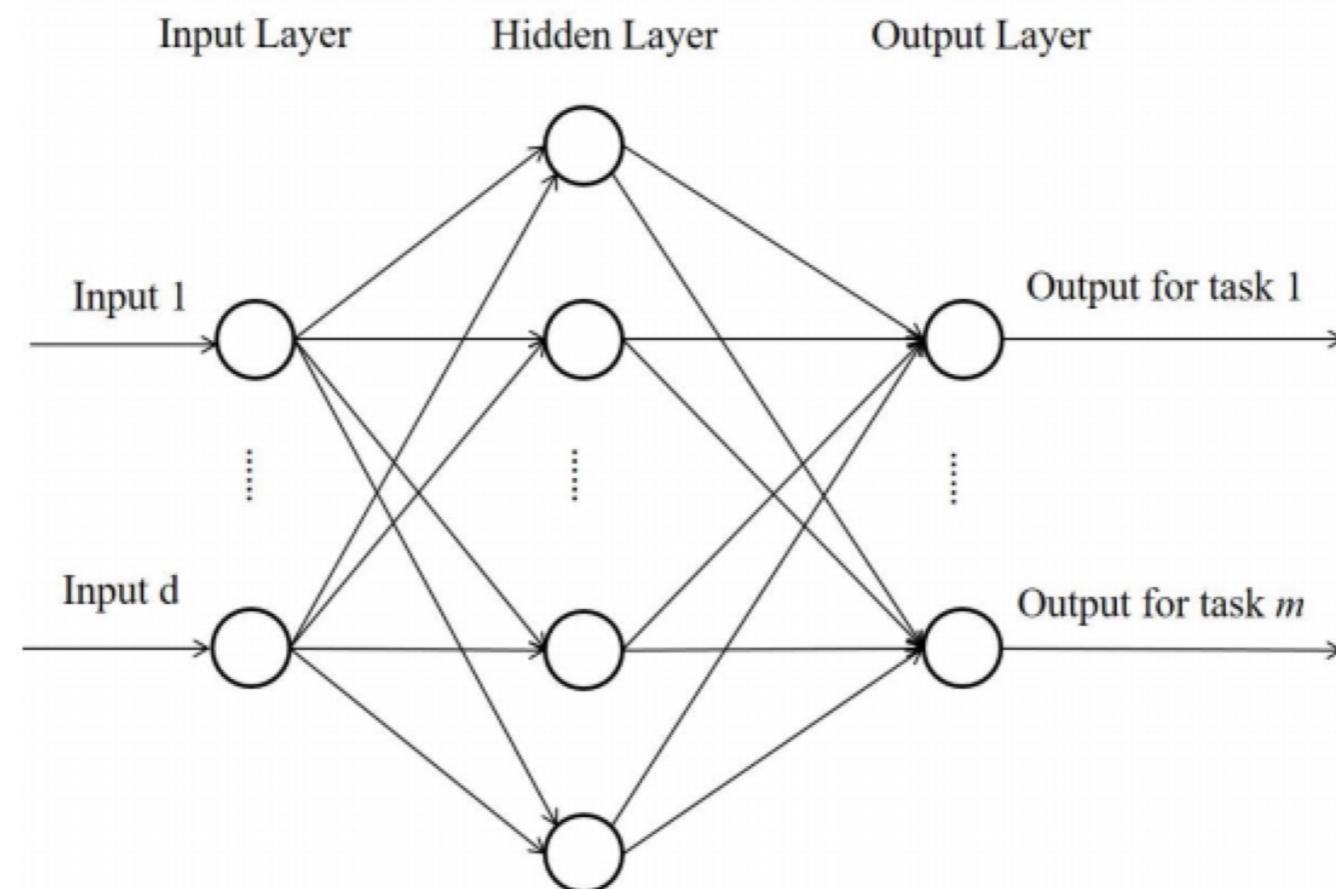
$$\min_{W, P} \frac{1}{m} \sum_{i=1}^m \frac{1}{n_i} \sum_{j=1}^{n_i} \ell(P^\top x_j^i, y_j^i, w^i) + \lambda \text{rank}(W),$$

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

Note that P is a projection matrix.

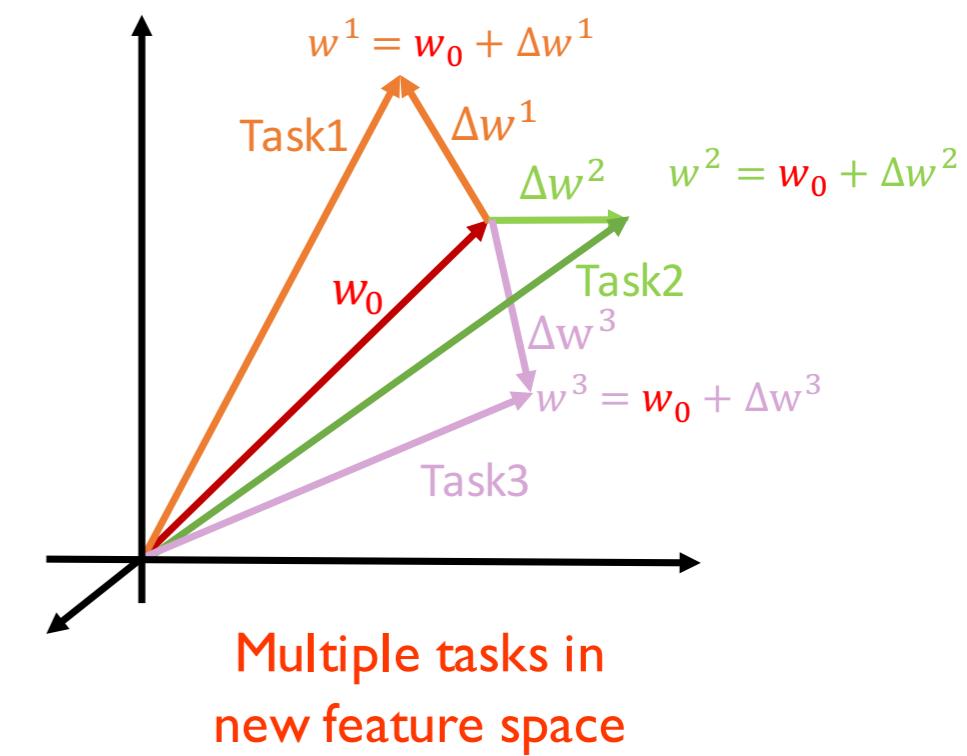
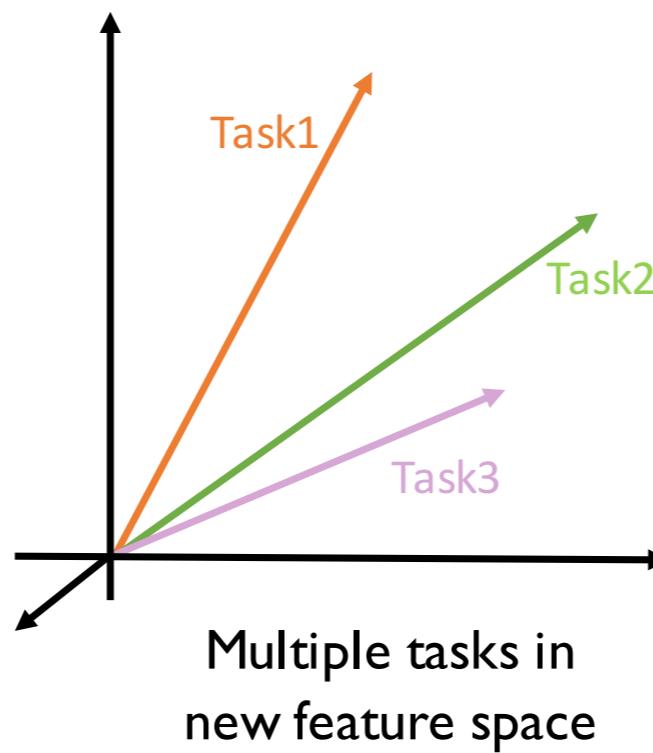
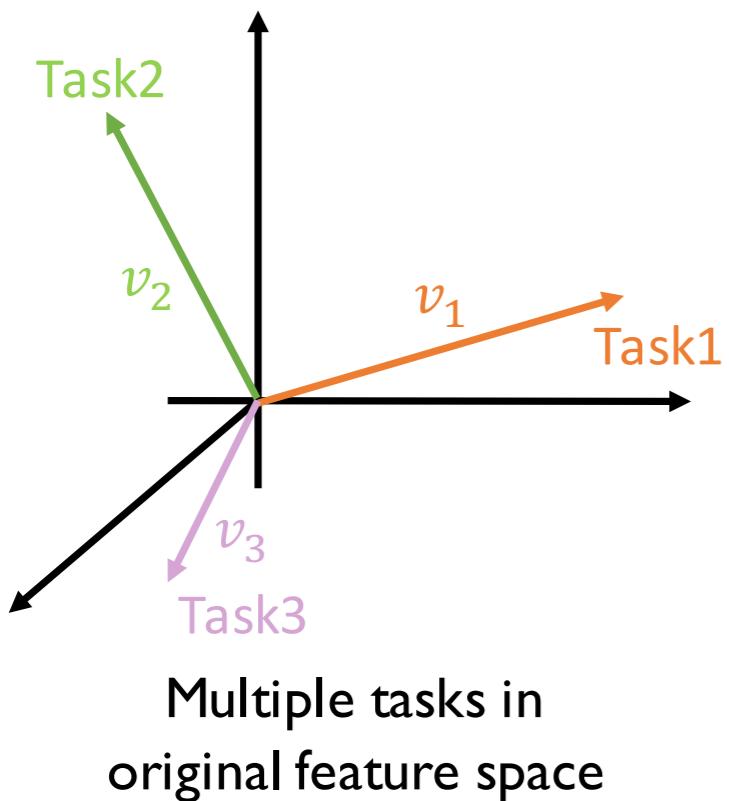
Feature-based MTL models

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



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?

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

Relationship to Transfer Learning

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/>