Source-Target Similarity Modelings for Multi-Source Transfer Gaussian Process Regression

ICML2017 accepted paper

Pengfei Wei[12], Ramon Sagarna[12], Yiping Ke[12], Yew-Soon Ong[12], Chi-Keong Goh[32]

[1]School of Computer Science and Engineering, Nanyang Technological University
[2]Rolls-Royce@Nanyang Technological University Corporate Lab
[3]Rolls-Royce Advanced Technology Centre, Singapore

Presenter: M2 Tomohiro YONEZU

2018/07/30 (Mon)

Contents

Contribution

- Topic: Proposition of new multi-source transfer GP regression model: $TC_{MS}Stack$.

- $TC_{MS}Stack$ can
  - (i) associates the similarity coefficient with the model importance.
  - (ii) reduces the computational cost by lowering the number o optimization variables.

- Show performance of existing method; $GP\text{-}TC_{MC}$

Contents

Introduction: Transfer Learning

- TL; Transfer Learning
  - The data from the Target domain (where we want to predict) is scarce.
  - But a good amount of data from another source domain is available.
  - Let's use source-data for predicting target!

- $MSTL$; Multi-source Transfer Learning
  - Multiple souce-domains are available.
  - A key issue is to capture the diverse $Source\text{-}Target$ ($S\text{-}T$) similarities.

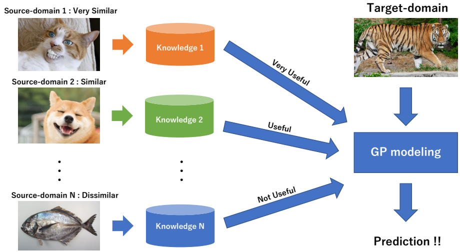


Figure: Idea of $MSTL$

Objective is Modeling to capture the diverse similarityes between different source-target domain pairs.

Contents

Problem Statement

- Definition
  - $\mathcal{D} = \mathcal{T} \cup \mathcal{S}$: Domain set
  - $\mathcal{S} = \{\mathcal{S}_i \mid 1 \leq i \leq N\}$: Set of source domains
  - $\mathcal{T}$: The target domain
  - $\boldsymbol{X}^{(\mathcal{S}_i)} \in \mathbb{R}^{n_{\mathcal{S}_i} \times d}, \boldsymbol{y}^{(\mathcal{S}_i)} \in \mathbb{R}^{n_{\mathcal{S}_i}}$: Data matrix and its labels in each $\mathcal{S}_i$
  - $\boldsymbol{X}^{(\mathcal{T}_l)} \in \mathbb{R}^{n_{\mathcal{T}_l} \times d}, \boldsymbol{y}^{(\mathcal{T}_l)} \in \mathbb{R}^{n_{\mathcal{T}_l}}$: Labeled target data matrix and its labels.
  - $\boldsymbol{X}^{(\mathcal{T}_u)} \in \mathbb{R}^{n_{\mathcal{T}_u} \times d}$: Unlabeled target data matrix.

- We use the GP model for this regression task.
  - GP model defines a Gaussian distribution over the functions, $\mathbf{f} \sim \mathcal{N}(\mu, \mathbf{K})$
  - $\mathbf{K}$ is PSD (denoted as $\mathbf{K} \succeq \mathbf{0}$)

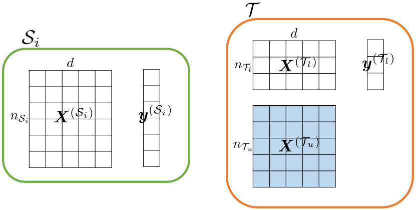


Figure: Datasets in Source-domain and Target-domain

※ $n_{\mathcal{T}_l} \ll \min(n_{\mathcal{S}_1}, \ldots, n_{\mathcal{S}_N}, n_{\mathcal{T}_u})$

MSTL regression Ideas1: $GP\text{-}TC_{MS}$

- Let $\mathbf{K}_{\mathcal{D}_i,\mathcal{D}_j}(\mathcal{D}_i\mathcal{D}_i \in \mathcal{D})$ denote a covariance matrix or points inf $\mathcal{D}_i$ and $\mathcal{D}_j$.
- $GP\text{-}TC_{MS}$ is GP model with kernel matrix specified by $S\text{-}T$ similarity parameters λ_i.

$$k_*(\mathbf{x},\mathbf{x}') = \begin{cases} \lambda_i k(\mathbf{x},\mathbf{x}'), \ \mathbf{x} \in \mathbf{X}^{(\mathcal{S}_i)} \& \mathbf{x}' \in \mathbf{X}^{(\mathcal{T})} \\ \qquad\qquad or \ \mathbf{x}' \in \mathbf{X}^{(\mathcal{S}_i)} \& \mathbf{x} \in \mathbf{X}^{(\mathcal{T})} \\ k(\mathbf{x},\mathbf{x}'), \ otherwise \end{cases}$$

$$\mathbf{K}_* = \begin{bmatrix} \mathbf{K}_{\mathcal{S}_1\mathcal{S}_1} & \cdots & \mathbf{K}_{\mathcal{S}_1\mathcal{S}_N} & \lambda_1\mathbf{K}_{\mathcal{S}_1\mathcal{T}} \\ \cdots & \cdots & \cdots & \cdots \\ \mathbf{K}_{\mathcal{S}_N\mathcal{S}_1} & \cdots & \mathbf{K}_{\mathcal{S}_N\mathcal{S}_N} & \lambda_N\mathbf{K}_{\mathcal{S}_N\mathcal{T}} \\ \lambda_1\mathbf{K}_{\mathcal{T}\mathcal{S}_1} & \cdots & \lambda_N\mathbf{K}_{\mathcal{T}\mathcal{S}_N} & \mathbf{K}_{\mathcal{T}\mathcal{T}} \end{bmatrix}$$

- λ_i is expected to capture the different transfer strengths in different $S\text{-}T$ domain pairs.

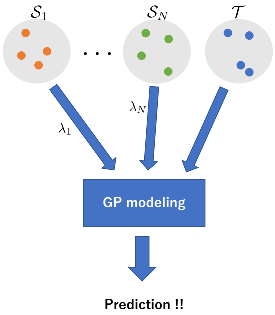


Figure: Idea of $GP\text{-}TC_{MS}$

Problem of $GP\text{-}TC_{MS}$

- Kernel matrix must be PSD; positive-semidefinite

Theorem 1

$\boldsymbol{K}_*$ is PSD for any covariance matrix $\mathbf{K}$ in the form

$$\mathbf{K} = \begin{bmatrix} \mathbf{K}_{\mathcal{S}_1\mathcal{S}_1} & \cdots & \mathbf{K}_{\mathcal{S}_1\mathcal{S}_N} & \mathbf{K}_{\mathcal{S}_1\mathcal{T}} \\ \cdots & \cdots & \cdots & \cdots \\ \mathbf{K}_{\mathcal{S}_N\mathcal{S}_1} & \cdots & \mathbf{K}_{\mathcal{S}_N\mathcal{S}_N} & \mathbf{K}_{\mathcal{S}_N\mathcal{T}} \\ \mathbf{K}_{\mathcal{T}\mathcal{S}_1} & \cdots & \mathbf{K}_{\mathcal{T}\mathcal{S}_N} & \mathbf{K}_{\mathcal{T}\mathcal{T}} \end{bmatrix}$$

if and only if $\lambda_1 = \ldots = \lambda_N$ and $|\lambda_i| \leq 1$

- This shows that $k_*(\cdot,\cdot)$ can give only one similarity coefficient for all S-T domain pairs.
- Such single similarity coefficient compromises the diverse similarities between different S-T domain pairs.

- Author also show that such single coefficient takes effects in every source on the final transfer performance.

Stacking

- "Stacking" is one of ensemble method.

- Preparation
  - $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{n}$: Dataset
  - $\mathfrak{L}_j$: Some different learners
  - $\boldsymbol{\omega}^\top = [\omega_1, \ldots, \omega_N]$: weight parameters

step1 Make prediction and create new data set $\mathcal{D}_{new}$.

$$
\begin{aligned}
f_{(i,j)} &= \mathfrak{L}_j(\mathbf{x}_i) \\
\mathcal{D}_{new} &= \{(\mathbf{z}_i, y_i)\}_{i=1}^{n}, \\
\mathbf{z}_i^\top &= \left[f_{(i,1)}, f_{(i,2)}, \ldots, f_{(i,N)}\right]
\end{aligned}
$$

step2 Run least square method to $\mathcal{D}_{new}$ and get $\boldsymbol{\omega}^*$

$$\boldsymbol{\omega}^* = \arg\min_{\boldsymbol{\omega}} \sum_{i=1}^{n} \left(y_i - \boldsymbol{\omega}^\top \mathbf{z}_i\right)^2$$

step3 Get final model as follows

$$f_*(\mathbf{x}) = \sum_{j=1}^{N} \omega_j^* \mathfrak{L}_j(\mathbf{x})$$

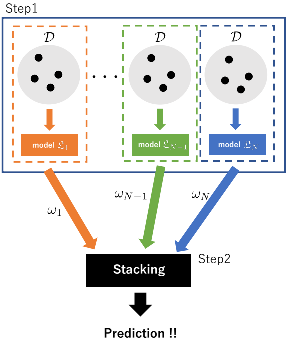


Figure: Idea of Stacking

MSTL regression Ideas2: $TC_{SS}Stack$

- "Stacking" method for Transfer learning. (Pardoe & Stone, 2010).
- Use $GP\text{-}TC_{SS}$ (GP with single-source transfer covariance) as base-model.
- $GP\text{-}TC_{SS}$ can capture $S\text{-}T$ similarity and Stacking method addes flexibility to Multi-source TL model.

step1 Train multiple $GP\text{-}TC_{SS}$ models using each $\mathcal{S}_i$ and $\mathcal{T}$.

$$\{f^{(\mathcal{S}_i,\mathcal{T})}(\cdot|\mathbf{\Omega}_i,\lambda_i)\}_{i=1}^{N}$$

step2 get final model by using Stacking method.

$$f(\mathbf{x}) = \sum_{i=1}^{N} \omega_i f^{(\mathcal{S}_i,\mathcal{T})}(\mathbf{x}|\mathbf{\Omega}_i,\lambda_i), \ \sum_{i=1}^{N}\omega_i = 1$$

where, ω_i are coefficient learned by minimizing the least square error on the target labeled data.

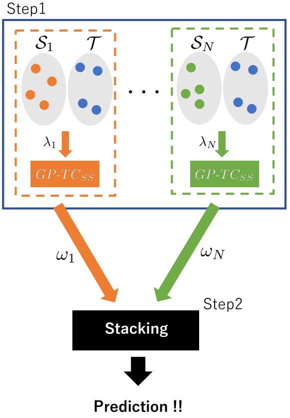


Figure: Idea of $TC_{SS}Stack$

Problem of $TC_{SS}Stack$

(i) Since each $f^{(\mathcal{S}_i,\mathcal{T})}$ is pretrained separately, inter-domain dependencies between different source domains into account aren't considered.

(ii) Both λ_i and ω_i reflect the S-T domain similarity. However, $TC_{SS}Stack$ takes them as two different variables and learns them separately.

- Intuitively, the model importance ω_i should be positively correlated with the similarity coefficientλ_i.

Contents

Proposal method: $TC_{MS}Stack$ [1/2]

- To overcome issues of $TC_{SS}Stack$, authors propose a new transfer stacking model as follows.

$$f(\mathbf{x}) = \sum_{i=1}^{N} (g(\lambda_i)/Z)\, f^{(\mathcal{S}_i,\mathcal{T})}(\mathbf{x}, \mathbf{\Omega}_i, \lambda_i).$$

  - $Z = \sum_{i=1}^{N} g(\lambda_i)$: Normalization term
  - $g(\lambda_i)$: Any function preserving the monotonicity of $|\lambda_i|$

- This also reduces the search efforts by lowering the number of free parameters to fit. (ω_i)

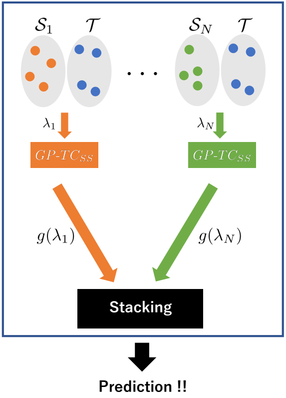


Figure: Idea of $TC_{MS}Stack$

Proposal method: $TC_{MS}Stack$ [2/2]

- We can choose relative importance $g(\cdot)$. In this paper, a simple function $g(\lambda_i) = |\lambda_i|$ is used.
- However, $|\lambda|$ is not smooth at $\lambda = 0$, so approximation below is useful.

$$|\lambda_i| \approx \alpha \log_e \left(\frac{1}{2} e^{\frac{\lambda_i}{\alpha}} + \frac{1}{2} e^{-\frac{\lambda_i}{\alpha}} \right)$$

- we get final model by minimizing the squared erros

$$\min_{\{\boldsymbol{\Omega}_i, \lambda_i\}_{i=1}^{N}} \sum_{j=1}^{n_{\mathcal{T}_l}} \left(y_j^{(\mathcal{T}_l)} - f^*(\mathbf{x}_j^{(\mathcal{T}_l)}) \right)^2$$

- This method jointly learned $f^{(\mathcal{S}_i, \mathcal{T})}$ for all the source domains.

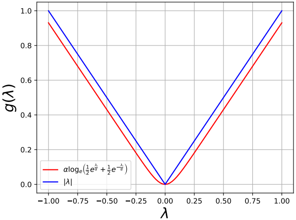


Figure: approximate of $|\lambda|$ ($\alpha = 0.1$)

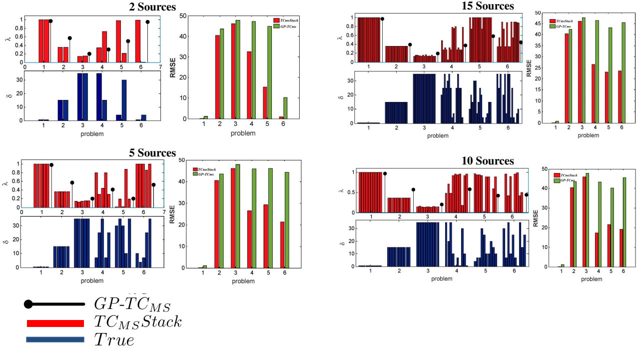
Contents

Experiment1: Synthetic dataset

- Target-domain data:
  - $f(\mathbf{x}) = \mathbf{w}_0^\top \mathbf{x} + \epsilon,\ \mathbf{w}_0 \in \mathbb{R}^{100}, \epsilon \sim \mathcal{N}(\mathbf{0}, \sigma)$: Ture function
  - $n_{\mathcal{T}_u} = 100$: 100 points from this function as target test data.
  - $n_{\mathcal{T}_l} = 20$: 20 points from this function as target train data.

- Source-domain data:
  - $g(\mathbf{x}) = (\mathbf{w}_0^\top + \delta \Delta \mathbf{w})\mathbf{x} + \epsilon$: True funciton
    - $\Delta \mathbf{w}$ is random fluction vector.
    - δ is controlling the similarith between f and g.
      (higher δ indicates lower similarity)
  - 380 points for each source-data with different δ.

Experiment1: Result



- GP-TC_{MS} can not capture different S-T similarities but $TC_{MS}Stack$ can. ($TC_{MS}Stack$ get the λ values which are strictly reverse-correlated with the δ.)
- In all problem, we can see a consistently lower RMSE for $TC_{MS}Stack$ than for GP-TC_{MS}.

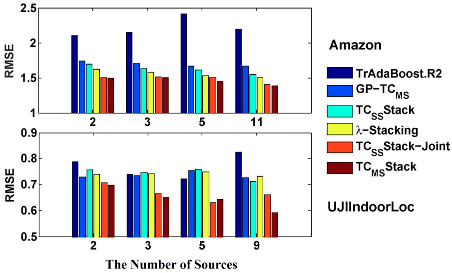
Experiment2: Real-world dataset

- Amazon dataset:
  - categorize the products into 4 top categories.
  - Products in the same category are conceptually similar.
  - Each product is taken as a domain. (select one as target)
  - label: stars

- UJIIndoorLoc dataset:
  - The building location dataset.
  - Target domain: 1st floor.
  - Source domain: other floor.
  - feature: signal strength from wire-less access points.
  - label: location

Experiment2: Result

- Several $MSTR$ approaches are compared.



- Figure above show the average RMSE.
- $TC_{MS}Stack$ is the winner among all the baselines on the two datasets, improving the transfer performance across the different amounts of source domains.

Contents

Conclusions

- Prove that, $GP\text{-}TC_{MS}$, a Gaussian process with such a transfer covariance function can only capture the same similarity coefficient for all the sources.
- Propose $TC_{MS}Stack$ that can aligns the $S\text{-}T$ similarity coefficients with the model importance and jointly learns the base models.
- Experimets show the superiority of $TC_{MS}Stack$ to other $MSTR$ methods.