MTTTS16 Learning from Multiple Sources

5 ECTS credits

Autumn 2019, University of Tampere
Lecturer: Jaakko Peltonen

Lecture 3: Transfer learning

On this lecture:

- Transfer learning

Part 1: Transfer learning

Transfer learning, motivation

- **Transfer learning:** Transferring knowledge from a familiar learning task (task A, occurring in some problem domain), or several such tasks, to a new one (task B, occurring in some related problem domain)
- Humans deal with a sequence of learning tasks over their life.
- Intuitively, learning a sequence of related tasks should be easier than learning each task in isolation.
- For example: recognizing a peach in a visual scene if you can already recognize apples and oranges
- Or learning German if one already knows Swedish and English
- The benefit comes from discovering **underlying structure** in the task domains that makes the tasks related

Following the approach from Zvika Marx, Michael T. Rosenstein, Leslie Pack Kaelbling, Transfer Learning with an Ensemble of Background Tasks, in proc. Inductive Transfer: 10 Years Later, NIPS 2005 workshop, and from Raina, R., Ng, A. Y., and Koller, D. Constructing Informative Priors using Transfer Learning. In: Proc. ICML 2006.

Transfer learning, motivation

- Not everything about the tasks is similar even if the tasks are related: Considering the tasks as identical and pooling their training data may work poorly
- In two supervised tasks, decision boundaries for task A and task B

  will not be in the same places over the feature space, even if the feature spaces (space of input features) and distributions over the inputs are the same in both tasks.
- Approach on this lecture: treat task A as defining a Bayesian prior distribution for the statistical features in task B
- In particular, if A and B are learned using probabilistic models from the same family, we treat task A as defining a prior for parameter values in task B.

Following the approach from Zvika Marx, Michael T. Rosenstein, Leslie Pack Kaelbling, Transfer Learning with an Ensemble of Background Tasks, in proc. Inductive Transfer: 10 Years Later, NIPS 2005 workshop.

Transfer learning, motivation

- Example: logistic regression model for predicting binary classes y

$$p(y=1|\boldsymbol{x})=\frac{1}{1+\exp\left(w_0+\sum_{j=1}^{d} w_j x_j\right)}$$

- Standard statistical approach to fitting the model:
    - assume an independent Gaussian prior on the weight values;

$$p(w_j)=N(w_j;\mu_j,\sigma_j)$$

    - then either find the maximum a posterior estimate: parameters maximizing the likelihood * prior

$$\sum_{i=1}^{N}\log p(y_i|\boldsymbol{x}_i)+\sum_{j=0}^{d}\log p(w_j)=\sum_{i=1}^{N}\log p(y_i|\boldsymbol{x}_i)-\sum_{j=0}^{d}(w_j-\mu_j)^2/2\sigma_j^2+const\,.$$

    - or compute the full posterior density over parameters:

$$p(w_0,\ldots,w_d|data)\sim\sum_{i=1}^{N}\log p(y_i|\boldsymbol{x}_i)-\sum_{j=0}^{d}(w_j-\mu_j)^2/2\sigma_j^2-(d/2)\log(2\pi\sigma_j)$$

Transfer learning, motivation

- Often in the priors we use $\mu_j = 0$ and some common standard deviation $\sigma_j = \sigma$ whose value is chosen by cross-validation performance (performance on validation sets left out from training). Standard deviation σ_0 of the intercept might be set to a separate larger value.
- Suppose we have data from K different logistic regression tasks.
- Suppose we want to concentrate on one of the tasks (index B). We use the other K-1 tasks to learn prior parameters μ_j and σ_j for task B.

Transfer learning, simple approach

- First simple approach: start by fitting logistic regression separately to each of the K-1 other tasks, using priors with $\mu_j = 0$ and $\sigma_j = \sigma$.
- Result: for each of the other tasks (indices k), we get the weights $\{w_j^k\}_{j=0}^d$, k=1,...,K-1.
- Estimate the prior parameters for task B, as

$$\mu_j^B = \frac{1}{K-1}\sum_{k=1}^{K-1} w_j^k \quad \text{and} \quad \sigma_j^B = \sqrt{\frac{1}{K-2}\sum_{k=1}^{K-1}(w_j^k - \mu_j^B)^2} \qquad \text{for j=0,...,d}$$

Alternative: use μ_j of each earlier task

- This is essentially maximum likelihood fitting of a normal distribution to each prior parameter, over the population of earlier tasks

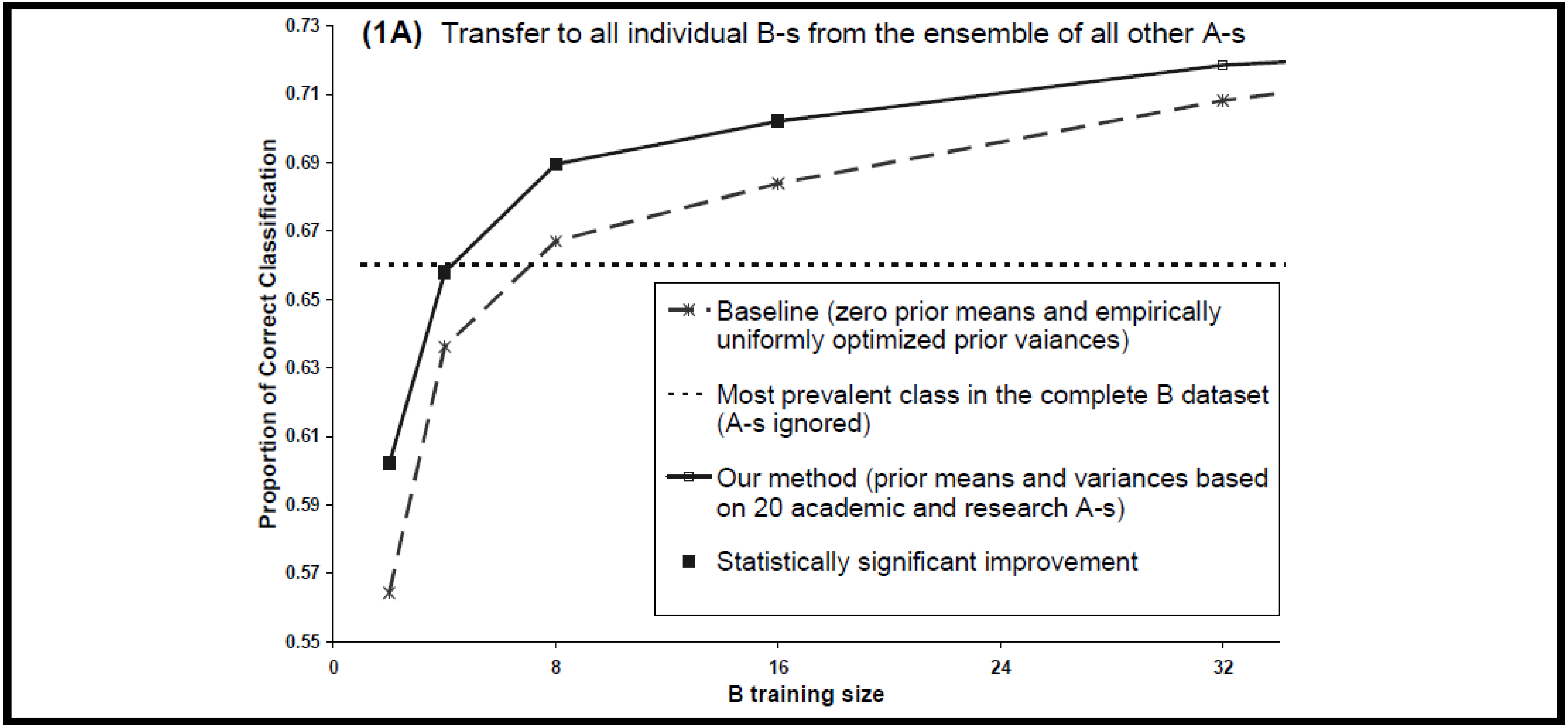
Transfer learning, example 1

- example: “Busy People”, 21 individuals make decisions of whether to accept an email invitation to a meeting.
- People come from different domains: 8 were participants in a military simulation, 13 were researchers in universities/private labs
- For each person, “background knowledge”: projects the person worked on, other people working on them, and relationships to the other people; “calendar data”: real 2-month calendar of the person
- Synthetic additional meeting invitations were generated, for each the person independently decided whether to attend based on the real calendar
- Use each person separately as “task B” and others as “tasks A”, fit logistic regression classifiers to predict accepting an invitation
- Test on new data from task B not used in training

Example and pictures from Zvika Marx, Michael T. Rosenstein, Leslie Pack Kaelbling, Transfer Learning with an Ensemble of Background Tasks, in proc. Inductive Transfer: 10 Years Later, NIPS 2005 workshop.

Transfer learning, example 1

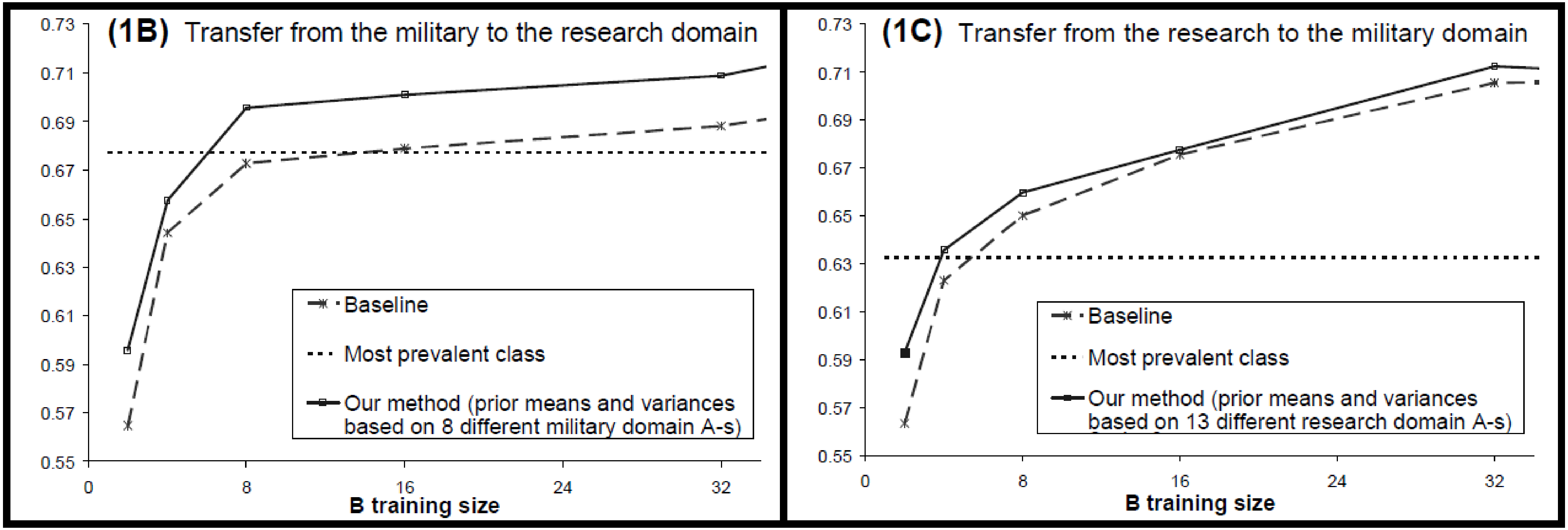
- Overall result: transfer learning improves the classifiers compared to independent learning



Example and pictures from Zvika Marx, Michael T. Rosenstein, Leslie Pack Kaelbling, Transfer Learning with an Ensemble of Background Tasks, in proc. Inductive Transfer: 10 Years Later, NIPS 2005 workshop.

Transfer learning, example 1

- We can separately examine transfer across the different types of domains (military simulation participants and researchers)
- Here, transfer from military to research (military as earlier tasks) helps more than the other way around (research as earlier tasks)



(1B) Transfer from the military to the research domain

(1C) Transfer from the research to the military domain

Example and pictures from Zvika Marx, Michael T. Rosenstein, Leslie Pack Kaelbling, Transfer Learning with an Ensemble of Background Tasks, in proc. Inductive Transfer: 10 Years Later, NIPS 2005 workshop.

Transfer learning, simple approach conclusion

- When multiple tasks are available, this simple task can already help

- The simple approach used here can be seen as an approximation of a **hierarchical Bayesian approach** in which we adopt an overall **hyperprior** over parameters, and assume that the parameter prior in each task is drawn from the hyperprior.

- The assumption is that the tasks are of the same kind (so that they were generated from a common source or taken from a common pool of tasks).

Example and pictures from Zvika Marx, Michael T. Rosenstein, Leslie Pack Kaelbling, Transfer Learning with an Ensemble of Background Tasks, in proc. Inductive Transfer: 10 Years Later, NIPS 2005 workshop.

Transfer learning, simple approach conclusion

- Note that in this setting, the samples available for each task are separate (different people), and each task has a different statistical structure (in particular, a different relationship between input features and the output class).

- However, in order for this kind of transfer to make sense, the feature spaces must have some kind of relationship.

- Because we transfer priors for each feature, the tasks must have
  - the same number of features, and
  - their meanings must be roughly the same
  - and in the same order.

- Distributions over features can differ between tasks, and small differences in the meaning of the feature can be tolerated.

Example and pictures from Zvika Marx, Michael T. Rosenstein, Leslie Pack Kaelbling, Transfer Learning with an Ensemble of Background Tasks, in proc. Inductive Transfer: 10 Years Later, NIPS 2005 workshop.

Transfer learning, simple approach conclusion

- The question of which tasks are useful to combine is ongoing research; negative transfer, where combining tasks hurts performance, is possible for bad choices

- Detecting cross-task and cross-domain similarities and relevances could be done for example by unsupervised mechanisms such as data clustering or modeling.

Example and pictures from Zvika Marx, Michael T. Rosenstein, Leslie Pack Kaelbling, Transfer Learning with an Ensemble of Background Tasks, in proc. Inductive Transfer: 10 Years Later, NIPS 2005 workshop.

Transfer learning, second approach, motivation

- In the previous approach, features were assumed to be independent, and transfer was done independently for each feature.
- Consider a **text classification** learning task:
  - Given a vocabulary V, each input document is represented as a "bags of words" vector telling if a word appears in the document or not: $\boldsymbol{x}=[x_1 \ldots x_{|V|}] \in \{0,1\}^{|V|}$
  - (A more complicated representation tells how many times each word appeared: $\boldsymbol{x}=[x_1 \ldots x_{|V|}] \in N_+^{|V|}$)
  - (A more complicated representation gives some real-valued weights to words, e.g. "term frequency-inverse document frequency" (TF-IDF) weights: $\boldsymbol{x}=[x_1 \ldots x_{|V|}] \in R_+^{|V|}$)
  - Each document has a binary label (e.g. "is this politics or economic news" or "is this review positive or negative")

Example and pictures from Raina, R., Ng, A. Y., and Koller, D. Constructing Informative Priors using Transfer Learning. In: Proc. ICML 2006.

Transfer learning, second approach, motivation

- We could use logistic regression to classify the documents:

$$p(y=1|\boldsymbol{x})=\frac{1}{1+\exp\left(w_0+\sum_{j=1}^{d} w_j x_j\right)}$$

where d = |V|. The classifier is again defined by the parameters $\boldsymbol{w}=[w_0 \ldots w_d]\in R^{d+1}$

- Maximize log-likelihood of parameters: $\sum_{i=1}^{N} \log p(y_i|\boldsymbol{x}_i;\boldsymbol{w})$

- Or maximize posterior probability of parameters, given a normally distributed prior for each parameter:

$$\sum_{i=1}^{N} \log p(y_i|\boldsymbol{x}_i)-\sum_{j=0}^{d} (w_j-\mu_j)^2/2\sigma_j^2+const\,.$$

- This prior assumes parameters are independent and have equal prior variance; may give poor performance

Example and pictures from Raina, R., Ng, A. Y., and Koller, D. Constructing Informative Priors using Transfer Learning. In: Proc. ICML 2006.

Transfer learning, second approach

- In text documents, features are often correlated: if the word “moon” appears, words related to “moon” like “rocket” or “astronaut” or “eclipse” or “crescent” are also likely to appear.
- Moreover, there might be underlying trends that affect which words are informative (frequent words are not necessarily the most informative)
- The “independent Gaussian priors for each feature” approach corresponds to having a multidimensional **diagonal Gaussian prior** for the whole weight vector.
- Idea: try to model feature correlations and trands by having a **full (non-diagonal) multidimensional Gaussian prior:**

  $p(\mathbf{w}) = N(\mathbf{w}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$ where $\boldsymbol{\Sigma} \in R^{d+1 \times d+1}$

Example and pictures from Raina, R., Ng, A. Y., and Koller, D. Constructing Informative Priors using Transfer Learning. In: Proc. ICML 2006.

Transfer learning, second approach

$p(\mathbf{w}) = N(\mathbf{w}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$ where $\boldsymbol{\Sigma} \in R^{d+1 \times d+1}$

- In the above equation of the prior, parameters have differing prior variances (diagonal entries), and off-diagonal entries show **dependencies between parameters**.
- Note that these are priors for the logistic regression parameters (weights), not priors for the word occurrences themselves.
- If an off-diagonal entry between “moon” and “rocket” is large, the prior says that “rocket” supports the same label as “moon”, even if we do not see this in a limited set of data.

- We will **learn the prior covariance matrix from earlier tasks** (here also called “auxiliary learning problems”), and **use it in learning the new task** (here also called “target learning problem”)
- How can we compute the prior covariance between two parameters w_1 and w_2 of the target problem, corresponding to words v_1 and v_2 in its vocabulary?

Example and pictures from Raina, R., Ng, A. Y., and Koller, D. Constructing Informative Priors using Transfer Learning. In: Proc. ICML 2006.

Transfer learning, second approach

- If we have many earlier learning tasks, we could fit logistic regression to each of them, and estimate a covariance matrix of how the parameters vary across the tasks.
- However, if we have only a few earlier tasks, we might not be able to learn the covariance matrix well. For example, what if we only had a single auxiliary task (earlier learning task) C, how can we learn a prior then?

- Idea: **create artificial copies of the earlier learning task**, using subsamples of the vocabulary (features involved in the task) and the document set (set of observed data items).
- When we fit logistic regression to each of the artificial copies, this will tell how the optimized parameters vary depending on the specific choice of the learning set and the vocabulary: **we can learn a prior from this variation**.

Example and pictures from Raina, R., Ng, A. Y., and Koller, D. Constructing Informative Priors using Transfer Learning. In: Proc. ICML 2006.

Transfer learning, second approach

- In more detail, consider a **randomly generated (subsampled)** task based on the auxiliary task (earlier task) C. The task is generated by taking the labeled training data from C and using a **random (subsampled) vocabulary, leaving out other words in the task.**
- Suppose the vocabulary includes two words (features) v_1 and v_2 and suppose that for this randomly generated task and vocabulary, we know the optimal values of w_1 and w_2 (giving highest likelihood for new data from the task). Denote the optimal values w_1^* and w_2^*.
- The optimal values are random variables, where the randomness is over the choice of the earlier task C and the vocabulary.
- We can estimate the covariance of the optimal values, $E[\ w_1^*\ \ w_2^*\ ]$, based on the available earlier tasks.
- (Note: a priori we don't know which word favors which class, thus $E[\ w_1^*]=0$ and covariance can be computed with the simple equation above.)

Example and pictures from Raina, R., Ng, A. Y., and Koller, D. Constructing Informative Priors using Transfer Learning. In: Proc. ICML 2006.

Transfer learning, second approach

- Estimator algorithm: given a vocabulary size K, and a learning task C (for which labeled document data is available)
  - For m=1...M,
    - generate random vocabulary of size K including v_1 and v_2 .
    - for t=1,...,T,
      - generate a training set from labeled data available for this task C. Keep only the features that are in the currently generated random vocabulary.
      - learn a logistic regression classifier for the training set
    - estimate a prior for the covariance value between words v_1 and v_2 , based on results for the current random vocabulary:

      $$\boldsymbol{\mu}^{(v,t)}=(1/T)\sum_{t=1}^{T}\mathbf{w}^{(v,t)} \qquad c_{1,2}^{(v)}=(1/T)\sum_{t=1}^{T}\left(w_1^{(v,t)}-\mu_1^{(v,t)}\right)\left(w_2^{(v,t)}-\mu_2^{(v,t)}\right)$$
  - Final estimate based on all vocabularies

    $$U=(1/VT)\sum_{v,t}w_1^{(v,t)}w_2^{(v,t)}$$ sample covariance

Example and pictures from Raina, R., Ng, A. Y., and Koller, D. Constructing Informative Priors using Transfer Learning. In: Proc. ICML 2006.

Transfer learning, second approach

Technical note

- The previous algorithm has a problem, it overestimates variability; it is supposed to only estimate variance over the choice of vocabulary, but it includes variance over the choice of training set.
- For example computing variance for one word (if we used w1=w2 instead of two different words) always overestimates the variance
- Apply a bootstrap correction (Efron, 1979) to the sample covariance

Example and pictures from Raina, R., Ng, A. Y., and Koller, D. Constructing Informative Priors using Transfer Learning. In: Proc. ICML 2006.

Technical note

Transfer learning, second approach

- Estimator algorithm: given a vocabulary size K, and a learning task C (for which labeled document data is available)
  - For m=1...M,
    - generate random vocabulary of size K including v_1 and v_2 .
    - for t=1,...,T,
      - generate a training set from labeled data available for this task C. Keep only the features that are in the currently generated random vocabulary.
      - learn a logistic regression classifier for the training set
    - estimate a prior for the covariance value between words v_1 and v_2 , based on results for the current random vocabulary:

$$\boldsymbol{\mu}^{(v,t)} = (1/T) \sum_{t=1}^{T} \boldsymbol{w}^{(v,t)} \qquad c_{1,2}^{(v)} = (1/T) \sum_{t=1}^{T} (w_1^{(v,t)} - \mu_1^{(v,t)})(w_2^{(v,t)} - \mu_2^{(v,t)})$$

  - Final estimate based on all vocabularies

$$U = (1/VT) \sum_{v,t} w_1^{(v,t)} w_2^{(v,t)} \qquad \Sigma_{1,2} = U - (1/V) \sum_{v} c_{1,2}^{(v)}$$

bootstrap corrected

Example and pictures from Raina, R., Ng, A. Y., and Koller, D. Constructing Informative Priors using Transfer Learning. In: Proc. ICML 2006.

Transfer learning, second approach, example 1

- Data: documents from several newsgroups
- Target problem: classify articles between two newsgroups, “motorcycles” vs “MS-Windows”
- 9 other classification problems as auxiliary problems
- estimate covariance between the target problem's words, using the auxiliary problems
- Word pairs with highly positive covariance seem to be related to similar topics

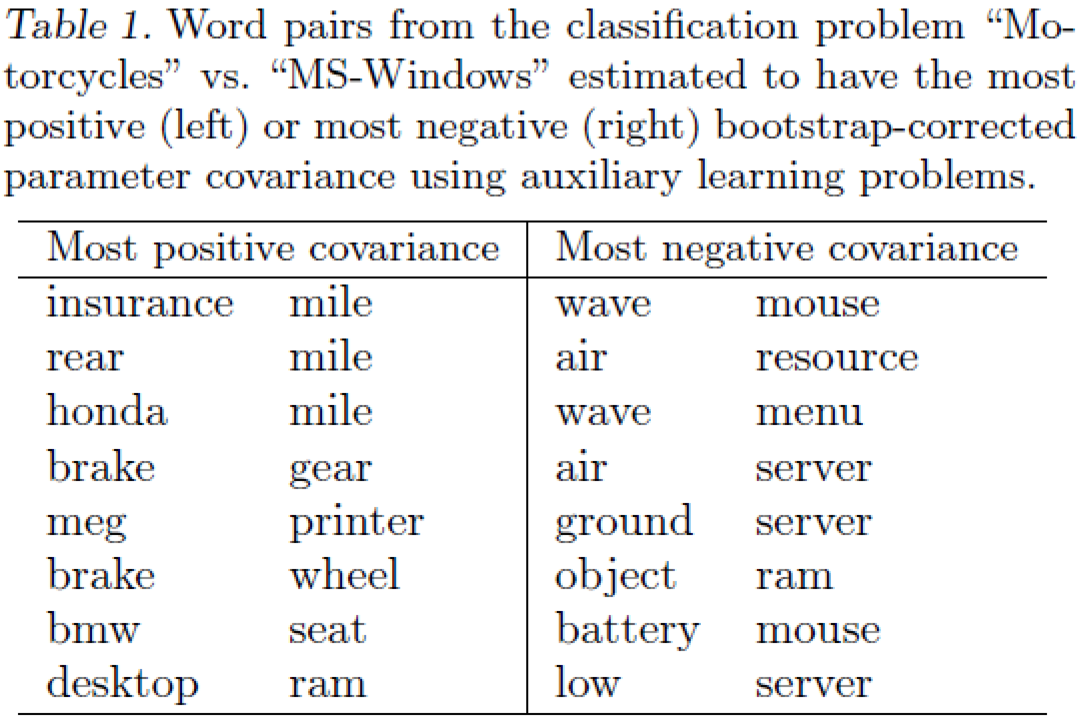


Table 1. Word pairs from the classification problem “Motorcycles” vs. “MS-Windows” estimated to have the most positive (left) or most negative (right) bootstrap-corrected parameter covariance using auxiliary learning problems.

Most positive covariance		Most negative covariance	
insurance	mile	wave	mouse
rear	mile	air	resource
honda	mile	wave	menu
brake	gear	air	server
meg	printer	ground	server
brake	wheel	object	ram
bmw	seat	battery	mouse
desktop	ram	low	server

Example and pictures from Raina, R., Ng, A. Y., and Koller, D. Constructing Informative Priors using Transfer Learning. In: Proc. ICML 2006.

Transfer learning, second approach

- The previous approach estimates covariance for individual feature (word) pairs
- Covariance of word pairs that don't occur in auxiliary problems cannot be estimated like this
- Resulting matrix of estimated covariances may not be positive semidefinite.

- Suppose there is some information about the features themselves available (e.g. some feature ontology in bioinformatics; or other descriptions of the features). We can try using such descriptions to generalize across feature pairs.
- Suppose for each word pair (i,j) we have a feature vector $\mathbf{F}_{i,j}$ whose elements are features of the word pair and the current vocabulary; e.g., one feature might be “are they synonyms”.
- Idea: approximate the covariance matrix as a function of the feature vectors. $\hat{\Sigma}_{i,j}=\psi^{T}\boldsymbol{F}_{i,j}$

Example and pictures from Raina, R., Ng, A. Y., and Koller, D. Constructing Informative Priors using Transfer Learning. In: Proc. ICML 2006.

Transfer learning, second approach

- Suppose we have directly estimated the covariance for a small set G = {(i,j)} of word pairs (v_i,v_j), using the previous approach; call the result $s_{i,j}$.
- Given these “desired values” $s_{i,j}$ for G, and also the feature vectors $\mathbf{F}_{i,j}$ for each pair in G, we could try to learn a linear regression from the $\mathbf{F}_{i,j}$ to the $s_{i,j}$. Optimize: $min_{\psi} \sum_{(i,j)\in G} (s_{ij} - \psi^T \boldsymbol{F}_{i,j})^2$
- Given the linear regression solution, we could predict the covariances as $\hat{\Sigma}_{i,j} = \psi^T \boldsymbol{F}_{i,j}$.
- Technical note: However, the resulting matrix might not be positive semidefinite. It is possible to add an auxiliary variable which can be freely optimized over the set of positive semidefinite matrices; we can then optimize the regression while also minimizing difference to that matrix.

$$min_{\psi,\Sigma;\Sigma\succcurlyeq 0} \sum_{(i,j)\in G} (s_{ij} - \psi^T \boldsymbol{F}_{i,j})^2 + \lambda \sum_{i,j} (\Sigma_{ij} - \psi^T \boldsymbol{F}_{i,j})^2$$

This can be written as a semidefinite programming (SDP) problem, and solved using SDP solvers.

Example and pictures from Raina, R., Ng, A. Y., and Koller, D. Constructing Informative Priors using Transfer Learning. In: Proc. ICML 2006.

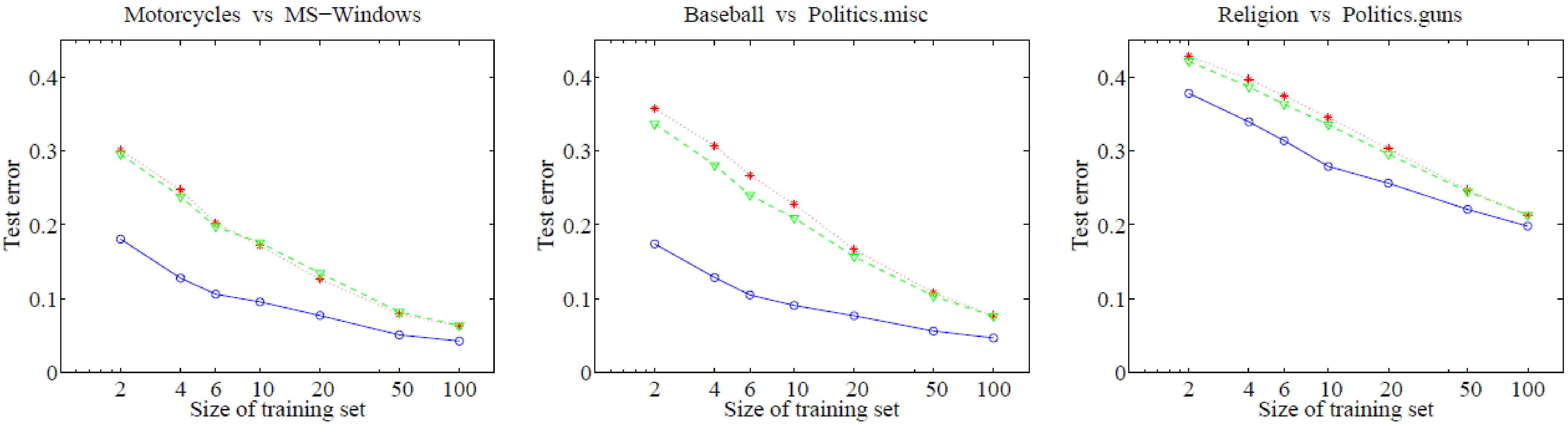
Transfer learning, second approach, example 2

- Data: documents from several newsgroups
- 10 binary classification problems, e.g. classify articles between two newsgroups, “motorcycles” vs “MS-Windows”
- For each problem, vocabulary = 250 most frequent words from the corresponding two newsgroups
- Each problem in turn treated as the target problem and the other 9 as auxiliary problems
- Covariance matrix entries generated from each auxiliary problem using the first algorithm (75% of diagonal entries, 20% of nondiagonal entries); then the second algorithm is used to estimate the covariance matrix in the target problem (technically using semidefinite programming (SDP) solvers to make sure the matrix is positive semidefinite as mentioned on the previous slide)

Example and pictures from Raina, R., Ng, A. Y., and Koller, D. Constructing Informative Priors using Transfer Learning. In: Proc. ICML 2006.

Transfer learning, second approach, example 2

- Example results. Blue circles = SDP learning a full covariance matrix from previous tasks, green triangles = SDP learning a diagonal matrix only from previous tasks; red stars = baseline diagonal prior
- learning from previous tasks helps, but learning a diagonal covariance matrix is not enough



Example and pictures from Raina, R., Ng, A. Y., and Koller, D. Constructing Informative Priors using Transfer Learning. In: Proc. ICML 2006.

Transfer learning, third approach

- **Estimating means** is one of the most basic statistical needs.
- We consider **estimating multiple means** in different but related populations, and discuss a multi-task regularization approach to this problem, called multi-task averaging (MTA), which has good theoretical properties
- For tasks t=1,...,T, let $\{Y_{ti}\}_{i=1}^{N_t}$ be a set of N_t samples
- Assume we have a T x T matrix A describing **task relatedness**, with zero in the diagonal entries

Following Sergey Feldman, Maya R. Gupta, and Bela A. Frigyik. Multi-Task Averaging. Advances in Neural Information Processing (NIPS), 2012.

Transfer learning, third approach

- Objective:

$$\{Y_t^*\}_{t=1}^T = \arg\min_{\{\hat{Y}_t\}_{t=1}^T} \frac{1}{T}\sum_{t=1}^T\sum_{i=1}^{N_t}\frac{(Y_{ti}-\hat{Y}_t)^2}{\sigma_t^2} + \frac{\gamma}{T^2}\sum_{r=1}^T\sum_{s=1}^T A_{rs}(\hat{Y}_r-\hat{Y}_s)^2$$

- The 1st term minimizes the sum of empirical losses. The 2nd term regularizes the estimates by penalizing their pairwise differences. The parameter γ balances the two terms.

- More general formulation:

$$\{Y_t^*\}_{t=1}^T = \arg\min_{\{\hat{Y}_t\}_{t=1}^T} \frac{1}{T}\sum_{t=1}^T\sum_{i=1}^{N_t} L(Y_{ti},\hat{Y}_t) + \gamma J\left(\{\hat{Y}_t\}_{t=1}^T\right)$$

where L is some loss function and J is some regularization function

Following Sergey Feldman, Maya R. Gupta, and Bela A. Frigyik. Multi-Task Averaging. Advances in Neural Information Processing (NIPS), 2012.

Transfer learning, third approach

- For the simpler formulation, the closed-form solution is

$$Y^* = \left(I + \frac{\gamma}{T}\Sigma L\right)^{-1} \bar{Y}$$

where $\bar{Y}$ is the vector of sample averages $\bar{Y}_t = \frac{1}{N_t}\sum_{i=1}^{N_t} Y_{ti}$ (one average for each task),

Σ is a diagonal matrix of estimated variances of the sample averages: the t:th diagonal element is $\Sigma_{tt} = \sigma_t^2 / N_t$ where σ_t^2 is the sample variance in task t and N_t is the number of samples in task t

and L is the **graph Laplacian** of A, L = D - A where $D_{ii} = \sum_j A_{ij}$

- Asymptotically (when the number of data increases to infinity) the solution **approaches the true means**
- If $\gamma \geq 0$, $0 \leq A_{rs} < \infty$ for all r,s, and $0 < \frac{\sigma_t^2}{N_t} < \infty$ for all t, then each estimate $\{Y_t^*\}$ is a convex combination of sample averages

Following Sergey Feldman, Maya R. Gupta, and Bela A. Frigyik. Multi-Task Averaging. Advances in Neural Information Processing (NIPS), 2012.

Transfer learning, third approach

Theoretical analysis

- The solution on the previous slide **requires a known task-similarity matrix A**. Often we do not know it in advance. Could we learn it from data?

- Let's analyze the learning problem theoretically. In the case of two tasks, suppose for a moment that we actually knew the true means in the tasks. It turns out it is then possible to derive the **optimal task-similarity matrix** A for learning the means from data.

Details on the next two slides.

Following Sergey Feldman, Maya R. Gupta, and Bela A. Frigyik. Multi-Task Averaging. Advances in Neural Information Processing (NIPS), 2012.

Transfer learning, third approach

Theoretical analysis

- Suppose tasks 1 has N$_1$ samples, task 2 has N$_2$ samples.

  Suppose $\{Y_{1i}\}$ are independently and identically distributed (iid) with mean μ_1 and variance σ_1^2, and $\{Y_{2i}\}$ are iid with mean $\mu_2 = \mu_1 + \Delta$ and variance σ_2^2.

  Denote the 2x2 matrix A as $A = [0\ a; a\ 0]$.

- Then the closed form solution is

$$Y_1^* = \left(\frac{T + \frac{\sigma_2^2}{N_2} a}{T + \frac{\sigma_1^2}{N_1} a + \frac{\sigma_2^2}{N_2} a} \right) \bar{Y}_1 + \left(\frac{\frac{\sigma_1^2}{N_1} a}{T + \frac{\sigma_1^2}{N_1} a + \frac{\sigma_2^2}{N_2} a} \right) \bar{Y}_2$$

- The mean squared error of the solution (mean over the underlying distribution) is

$$\mathrm{MSE}[Y_1^*] = \frac{\sigma_1^2}{N_1} \left(\frac{T^2 + 2T \frac{\sigma_2^2}{N_2} a + \frac{\sigma_1^2 \sigma_2^2}{N_1 N_2} a^2 + \frac{\sigma_2^4}{N_2^2} a^2}{(T + \frac{\sigma_1^2}{N_1} a + \frac{\sigma_2^2}{N_2} a)^2} \right) + \frac{\Delta^2 \frac{\sigma_1^4}{N_1^2} a^2}{(T + \frac{\sigma_1^2}{N_1} a + \frac{\sigma_2^2}{N_2} a)^2}$$

Following Sergey Feldman, Maya R. Gupta, and Bela A. Frigyik. Multi-Task Averaging. Advances in Neural Information Processing (NIPS), 2012.

Transfer learning, third approach

Theoretical analysis

- The mean squared error of the solution (mean over the underlying distribution) is better than the mean squared error (MSE) of the simple sample average if $\Delta^2 - \frac{\sigma_1^2}{N_1} - \frac{\sigma_2^2}{N_2} < \frac{4}{a}$
- That is, multi-task averaging helps if the task means are close relative to the tasks' sample variances
- If the difference of task means was known, the optimal task relatedness value could be computed: minimize the sum of MSEs $\mathrm{MSE}[Y_1^*] + \mathrm{MSE}[Y_2^*]$. This yields the optimal value $a^* = \frac{2}{\Delta^2}$
- In practical cases the difference is not known, but the above result can be used to **suggest estimators**.

Following Sergey Feldman, Maya R. Gupta, and Bela A. Frigyik. Multi-Task Averaging. Advances in Neural Information Processing (NIPS), 2012.

Transfer learning, third approach

- **Idea 1:** try to find **A** for more than one task, restrict to diagonal matrices of the form **A**=a**11**^{T}. and find "a" to minimize the sum of mean-squared errors. Set $\gamma=1$, and assume all tasks have the same variance.
- Solve
$$a^* = \arg\min_a R\left(\mu, \left(I + \frac{1}{T}\frac{\mathbf{tr}(\Sigma)}{T}L(a\mathbf{1}\mathbf{1}^T)\right)^{-1}\bar{Y}\right)$$

where $R(\mu, W\bar{Y}) = E[(W\bar{Y}-\mu)^T(W\bar{Y}-\mu)] = \mathbf{tr}(W\Sigma W^T) + \mu^T(I-W)^T(I-W)\mu$
which yields
$$a^* = \frac{2}{\frac{1}{T(T-1)}\sum_{r=1}^T\sum_{s=1}^T(\mu_r-\mu_s)^2}$$

however, that solution depends on the underlying mean values; replace with sample-based estimates to get
$$\hat{a}^* = \frac{2}{\frac{1}{T(T-1)}\sum_{r=1}^T\sum_{s=1}^T(\bar{y}_r-\bar{y}_s)^2}$$
and the corresponding estimate
$$Y^* = \left(I + \frac{1}{T}\hat{\Sigma}L(\hat{a}^*\mathbf{1}\mathbf{1}^T)\right)^{-1}\bar{Y}$$
.

Following Sergey Feldman, Maya R. Gupta, and Bela A. Frigyik. Multi-Task Averaging. Advances in Neural Information Processing (NIPS), 2012.

Transfer learning, third approach

- **Idea 2:** minimize the worst-case loss over the possible mean locations; find a minimax estimator Y^M.

$$\inf_{\hat{Y}} \sup_{\mu} R(\mu, \hat{Y}) = \sup_{\mu} R(\mu, Y^M)$$

- Consider the case of two tasks. It can be shown a minimax estimator is a Bayes estimator with respect to the "least favorable prior" and has a constant risk.
- Assume task means lie in some bounded interval $\mu_t \in [b_l, b_u]$
- Then it can be shown the minimax estimator is

$$Y^M = \left(I + \frac{2}{T(b_u - b_l)^2} \Sigma L(\mathbf{1}\mathbf{1}^T) \right)^{-1} \bar{Y},$$

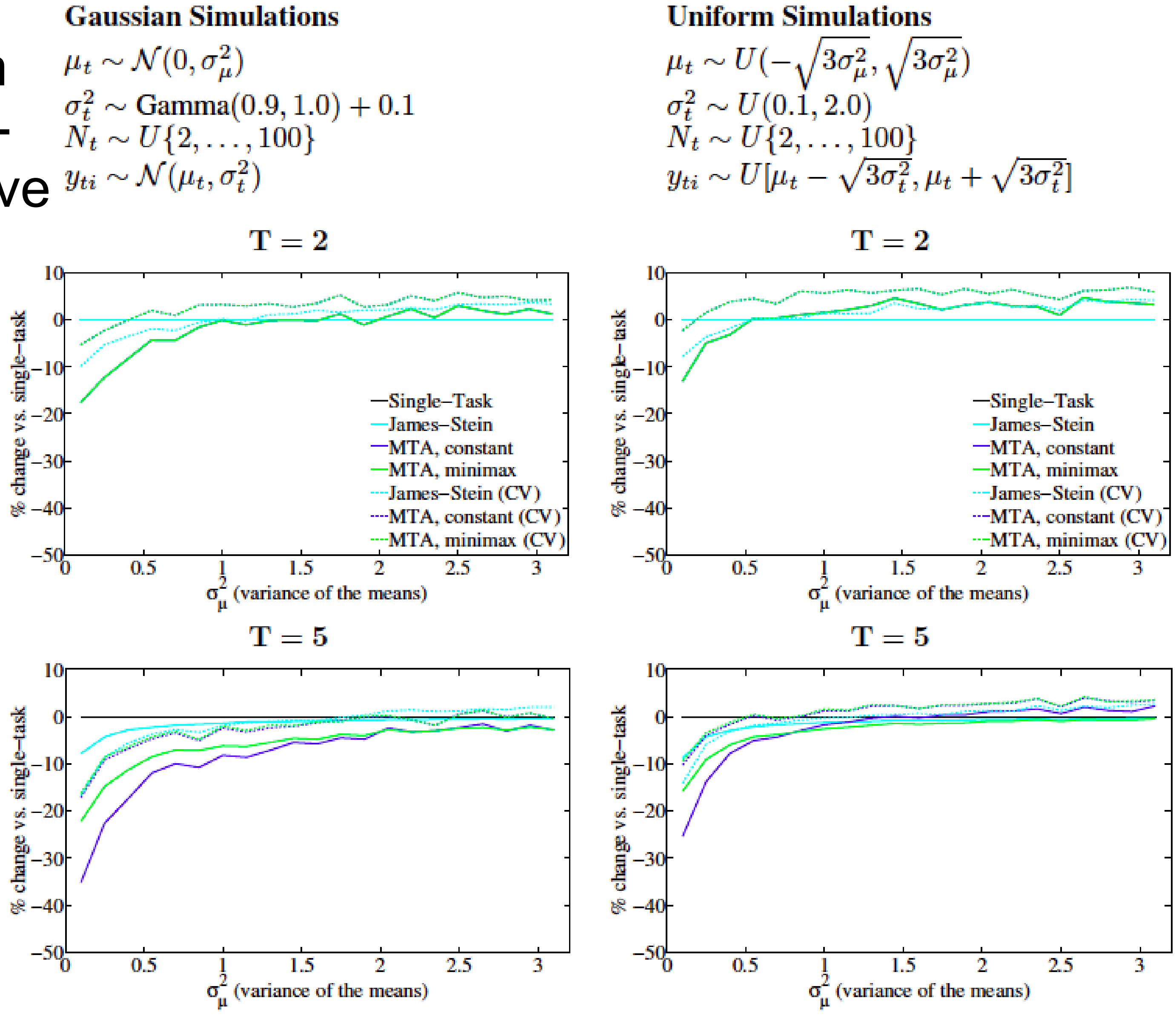
- For more tasks, one practically well working possibility is

$$\hat{b}_l = \min_t \bar{y}_t \qquad \hat{b}_u = \max_t \bar{y}_t$$

Following Sergey Feldman, Maya R. Gupta, and Bela A. Frigyik. Multi-Task Averaging. Advances in Neural Information Processing (NIPS), 2012.

Transfer learning, third approach, experiment

- Compare to single-task estimate, and a previous estimator (James-Stein). CV denotes versions where γ is chosen by cross-validation. Pictures show percent change in risk versus single-task, more negative is better.



Gaussian Simulations

$\mu_t \sim \mathcal{N}(0,\sigma_\mu^2)$

$\sigma_t^2 \sim \text{Gamma}(0.9,1.0)+0.1$

$N_t \sim U\{2,\ldots,100\}$

$y_{ti} \sim \mathcal{N}(\mu_t,\sigma_t^2)$

Uniform Simulations

$\mu_t \sim U(-\sqrt{3\sigma_\mu^2},\sqrt{3\sigma_\mu^2})$

$\sigma_t^2 \sim U(0.1,2.0)$

$N_t \sim U\{2,\ldots,100\}$

$y_{ti} \sim U[\mu_t-\sqrt{3\sigma_t^2},\mu_t+\sqrt{3\sigma_t^2}]$

Pictures from Sergey Feldman, Maya R. Gupta, and Bela A. Frigyik. Multi-Task Averaging. Advances in Neural Information Processing (NIPS), 2012.

Transfer learning, third approach, example 2

- Artifact Puzzles, a company that sells jigsaw puzzles online.

- Problem 1: estimate how much a random customer will spend on an order on average, if on their last order they purchased the t:th puzzle, for each of T = 77 puzzles. Data (assumed iid): amounts different customers had spent on orders after buying each of the puzzles,range 0-480.For each puzzle, 8-348 samples.

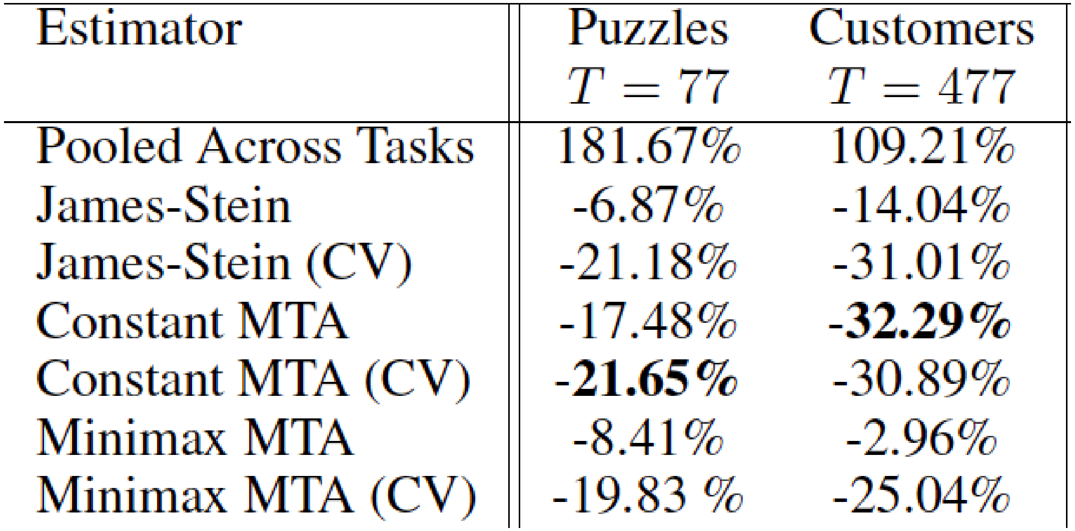
- Problem 2: estimate how much each customer will spend on an order on average, for each of the T = 477 customers that ordered at least twice. Data (assumed iid): order amounts for each of the T customers, range 15-480. 2-17 samples for each customer.

- Simulate “ground truth” by leaving out 50% of samples. Ground truth = values computed from all samples, try to estimate from the remaining 50%.

Example and pictures from Sergey Feldman, Maya R. Gupta, and Bela A. Frigyik. Multi-Task Averaging. Advances in Neural Information Processing (NIPS), 2012.

Transfer learning, third approach, example 2

- Results (change versus single-task, smaller is better):



Estimator	Puzzles $T = 77$	Customers $T = 477$
Pooled Across Tasks	181.67%	109.21%
James-Stein	-6.87%	-14.04%
James-Stein (CV)	-21.18%	-31.01%
Constant MTA	-17.48%	**-32.29%**
Constant MTA (CV)	**-21.65%**	-30.89%
Minimax MTA	-8.41%	-2.96%
Minimax MTA (CV)	-19.83 %	-25.04%

Example and pictures from Sergey Feldman, Maya R. Gupta, and Bela A. Frigyik. Multi-Task Averaging. Advances in Neural Information Processing (NIPS), 2012.

References

- Z. Marx, M.T. Rosenstein and L.P. Kaelbling, Transfer Learning with an Ensemble of Background Tasks, in: Inductive Transfer: 10 Years Later, NIPS 2005 workshop, 2005.
- Raina, R., Ng, A. Y., and Koller, D. Constructing Informative Priors using Transfer Learning. In: Proc. ICML 2006.
- W. Dai, Q. Yang, G.-R. Xue, and Y. Yu. Boosting for transfer learning. In proceedings of ICML 2007, the 24th International Conference on Machine Learning, pages 193-200, ACM, 2007.
- Sergey Feldman, Maya R. Gupta, and Bela A. Frigyik. Multi-Task Averaging. Advances in Neural Information Processing (NIPS), 2012. http://ee.washington.edu/research/guptalab/publications/FeldmanGuptaFrigyikNIPS2012.pdf
- R. Salakhutdinov, J. Tenenbaum A. Torralba; One-Shot Learning with a Hierarchical Nonparametric Bayesian Model. Proceedings of the ICML 2011 workshop on unsupervised and transfer learning. JMLR W&CP 27:195–206, 2012. http://jmlr.csail.mit.edu/proceedings/papers/v27/salakhutdinov12a/salakhutdinov12a.pdf Note: somewhat different from the previous transfer learning papers.