MTTTS16 Learning from Multiple Sources

5 ECTS credits

Autumn 2014, University of Tampere
Lecturer: Jaakko Peltonen

Lecture 5: Multitask learning with task clustering or gating

On this lecture:

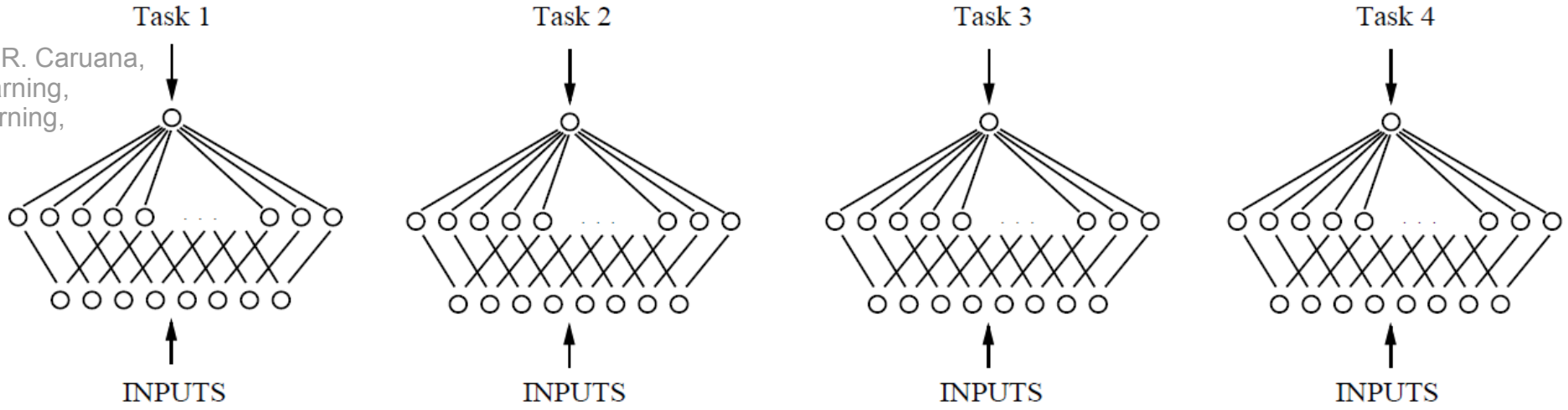
- We return to **neural network based multitask learning:** how to decide which tasks are similar and which subnetwork should handle each task
- More advanced approach to the same: learning and arbitrary number of task types (task clusters) in the case of **logistic regression**
- A general approach to the logistic regression case with more variants of how tasks can be related

Part 1: Multitask learning with task clustering or gating

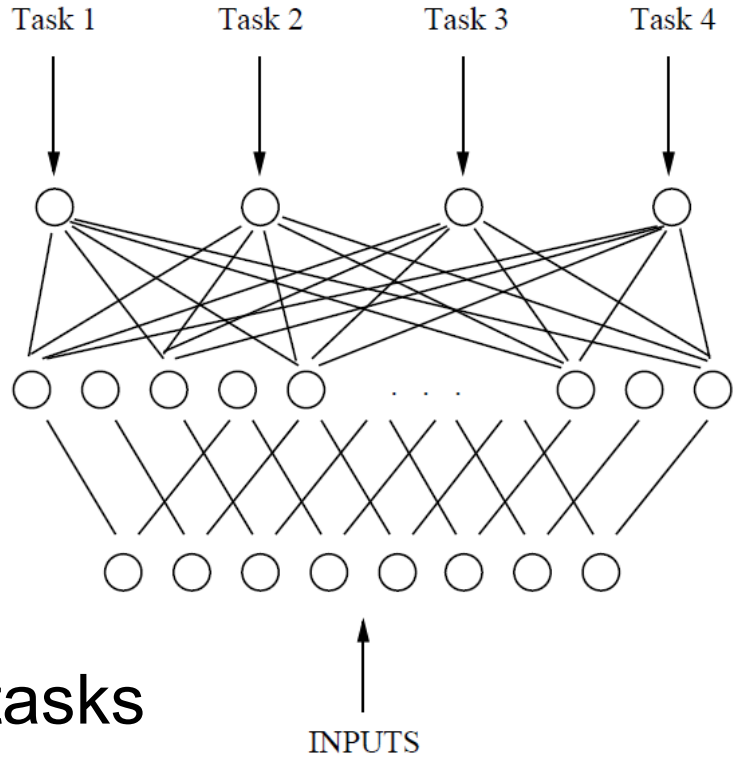
Recap: basic multitask learning in neural networks

- Single-task learning (STL): learn each task with a different ANN

Images from R. Caruana, Multitask Learning, Machine Learning, 1997



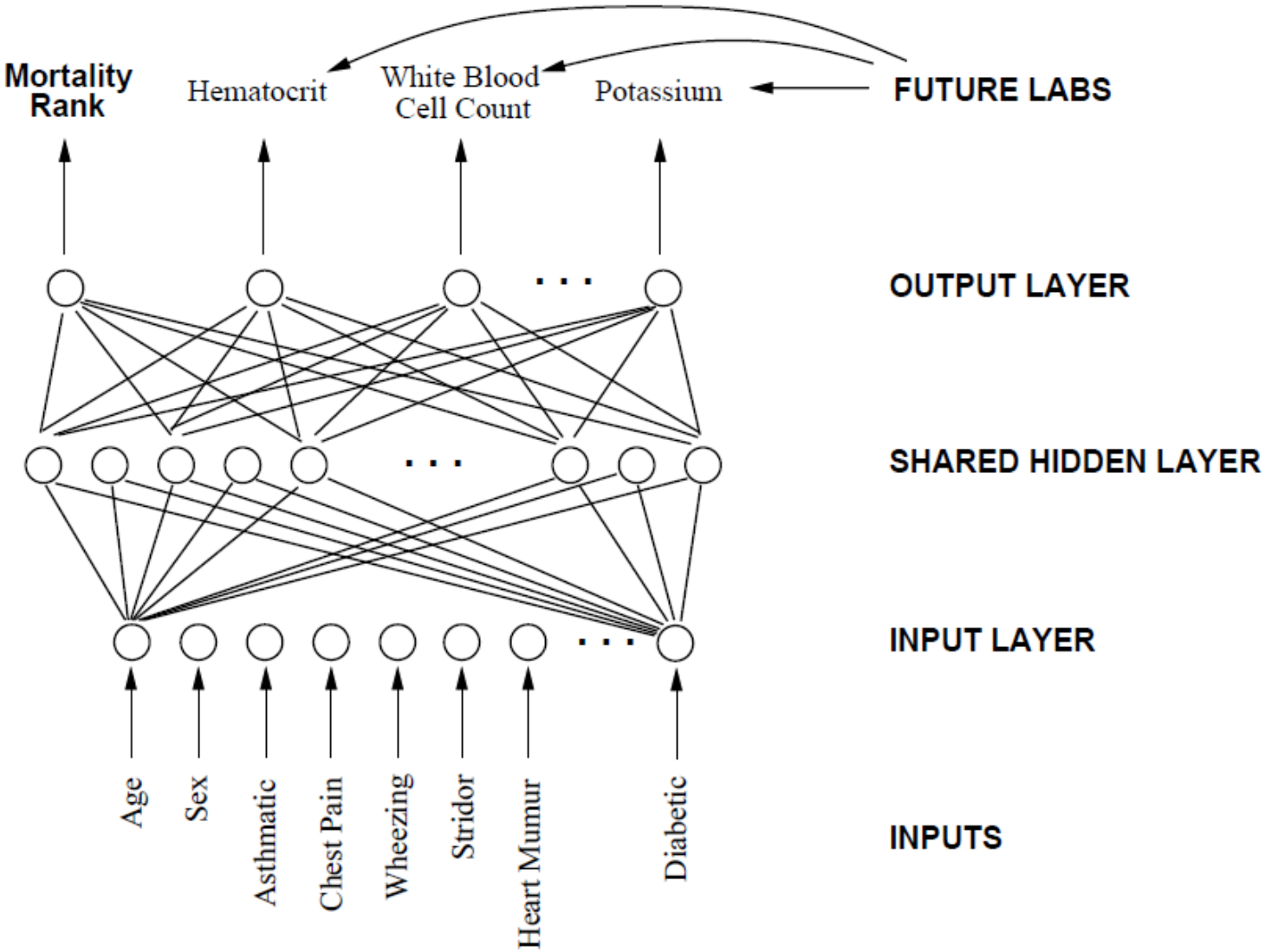
- Multi-task learning (MTL): use a shared intermediate representation: share the intermediate layers!
- Becomes a single ANN with multiple outputs in the output layer (one per task)
- Training is done in parallel for all tasks



Basic multitask learning in neural networks, example

- Multitask setup:use the 35 lab results as extra targets, to be predicted from the 30 basic measurements

Mortality rank = patient is in 10%/20%/30%/40%/50% highest riskpatients or not



Lab results may not be available when risk is predicted, that's why they are used as extra targets instead of inputs

Images from R. Caruana, Multitask Learning, Machine Learning, 1997

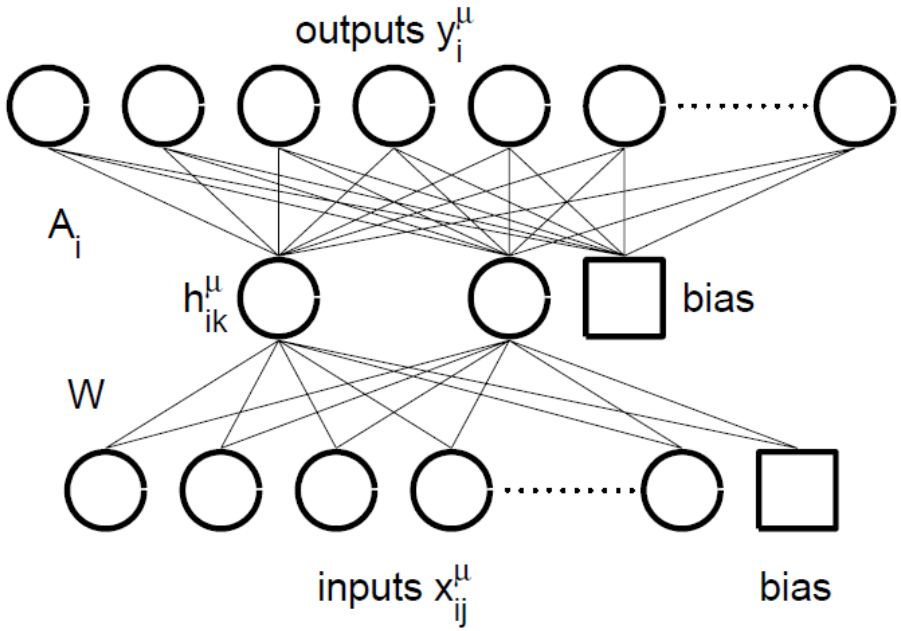
Multitask learning in neural networks, what more can we do?

- In multitask neural networks parallel tasks are modeled as multiple outputs of the same network.
- In multilevel analysis this is implemented through a mixed-effects linear model. ‘Fixed effects’ are the same for all tasks, ‘random effects’ may vary between tasks.
- Another implementation is through priors for task parameters
- Standard assumption: each task can learn equally well from any other task.
- Among many real-world tasks, some tasks will be more closely related than others (e.g. languages within a language family are more closely related than across families)

Following the approach from Bart Bakker and Tom Heskes. Task Clustering and Gating for Bayesian Multitask Learning. Journal of Machine Learning Research 4, 83-99, 2003.. Images from that paper.

Multitask learning in neural networks, what more can we do?

- A neural network can be seen as a model where observations y are generated from inputs x by an input-output function with added Gaussian noise



- Parameters of the input-output function (weights inside the network) can be learned by maximum likelihood fitting of the model to observations
- When N tasks are learned with the same network, the risk of overfitting the same parameters is an order N smaller

Following the approach from Bart Bakker and Tom Heskes. Task Clustering and Gating for Bayesian Multitask Learning. Journal of Machine Learning Research 4, 83-99, 2003.. Images from that paper.

Following the approach from Bart Bakker and Tom Heskes. Task Clustering and Gating for Bayesian Multitask Learning. Journal of Machine Learning Research 4, 83-99, 2003.. Images from that paper.

Multitask learning with task clustering

- Some tasks may be more strongly related than others: we take this into account by a form of **task clustering**.
- Notation: data of the i:th task $D_i = \{\mathbf{x}_i^\mu, y_i^\mu\}$, samples indexed by $\mu = 1 \ldots n_i$, each sample has input features x_{ik}^μ and output for this task y_i^μ . The output is assumed to have Gaussian noise with standard deviation σ .
- We will use networks with one hidden layer of neurons, with linear or nonlinear (tanh) transfer functions, and a bias term. The output layer has only linear combinations.
- Model for outputs:

g = nonlinearity

$$y_i^\mu = \sum_{j=1}^{n_{\text{hidden}}} A_{ij} h_{ij}^\mu + A_{i0} + \text{noise} , \quad h_{ij}^\mu = g\left(\sum_{k=1}^{n_{\text{input}}} W_{jk} x_{ik}^\mu + W_{j0} \right)$$

- For simplicity denote $h_{i0}^\mu = 1$ in $\mathbf{h}_i^\mu$, A_{i0} in $\mathbf{A}_i$

Multitask learning with task clustering

- Complete data of all tasks: $D = \{D_i\}$, $i = 1 \ldots N$
- Assume tasks are independent and identically distributed given the task hyperparameters.
- Assume a Gaussian prior for task-specific parameters: $\mathbf{A}_i \sim N(\mathbf{A}_i | \mathbf{m}, \Sigma)$ where $\mathbf{m}$ and Σ are hyperparameters

- Total set of hyperparameters (of all tasks): $\Lambda = \{W, \mathbf{m}, \Sigma, \sigma\}$
- Joint distribution of data and task-specific parameters:

$$P(D, A|\Lambda) = \prod_{i=1}^{N} P(D_i|\mathbf{A}_i, W, \sigma) P(\mathbf{A}_i|\mathbf{m}, \Sigma)$$

- $P(D_i|\mathbf{A}_i, W, \sigma)$ is the probability of data in task i

Following the approach from Bart Bakker and Tom Heskes. Task Clustering and Gating for Bayesian Multitask Learning. Journal of Machine Learning Research 4, 83-99, 2003.. Images from that paper.

Multitask learning with task clustering

- The distribution $P(D_i|\mathbf{A}_i, W, \sigma)$ is simply a Gaussian around the input-output function corresponding to $\mathbf{A}_i, W$ with standard deviation σ
- The distribution $P(D_i|\mathbf{A}_i, W, \sigma)$ needs known task-specific parameters $\mathbf{A}_i$. It is possible to integrate these out over the prior $P(\mathbf{A}_i|\mathbf{m}, \Sigma)$
- Result: probability of outputs in task i, given only the inputs, and hyperparameters of all tasks.

Following the approach from Bart Bakker and Tom Heskes. Task Clustering and Gating for Bayesian Multitask Learning. Journal of Machine Learning Research 4, 83-99, 2003.. Images from that paper.

Multitask learning with task clustering

Result (see paper for derivation):

$$P(D_i|\Lambda) \propto \left(|\Sigma|\sigma^{2n_i}|Q_i| \right)^{-\frac{1}{2}} \exp\left[\frac{1}{2}(\mathbf{R}_i^T Q_i^{-1}\mathbf{R}_i - S_i) \right]$$

$$Q_i = \sigma^{-2}\sum_{\mu=1}^{n_i} \mathbf{h}_i^\mu \mathbf{h}_i^{\mu T} + \Sigma^{-1}\,, \quad \mathbf{R}_i = \sigma^{-2}\sum_{\mu=1}^{n_i} y_i^\mu \mathbf{h}_i^\mu + \Sigma^{-1}\mathbf{m}\,,$$

$$S_i = \sigma^{-2}\sum_{\mu=1}^{n_i} y_i^{\mu 2} + \mathbf{m}^T\Sigma^{-1}\mathbf{m}\,.$$

Following the approach from Bart Bakker and Tom Heskes. Task Clustering and Gating for Bayesian Multitask Learning. Journal of Machine Learning Research 4, 83-99, 2003.. Images from that paper.

Following the approach from Bart Bakker and Tom Heskes. Task Clustering and Gating for Bayesian Multitask Learning. Journal of Machine Learning Research 4, 83-99, 2003.. Images from that paper.

Multitask learning with task clustering

- Result can be computed in closed form (see paper for derivation):

$$P(D_i|\Lambda) \propto \left(|\Sigma|\sigma^{2n_i}|Q_i| \right)^{-\frac{1}{2}} \exp\left[\frac{1}{2}(\mathbf{R}_i^T Q_i^{-1}\mathbf{R}_i - S_i) \right]$$

note $\mathbf{h}_i^\mu$ depends on W , $\mathbf{x}_i^\mu$

$$Q_i = \sigma^{-2}\sum_{\mu=1} \mathbf{h}_i^\mu \mathbf{h}_i^{\mu T} + \Sigma^{-1}\,, \quad \mathbf{R}_i = \sigma^{-2}\sum_{\mu=1} y_i^\mu \mathbf{h}_i^\mu + \Sigma^{-1}\mathbf{m}\,,$$

$$S_i = \sigma^{-2}\sum_{\mu=1}^{n_i} y_i^{\mu 2} + \mathbf{m}^T\Sigma^{-1}\mathbf{m}\,.$$

- Since the integrated likelihood function has a closed form, it can be maximized with respect to hyperparameters $\Lambda = \{W, \mathbf{m}, \Sigma, \sigma\}$ for example by gradient descent methods
- This maximization is called **empirical Bayes** because hyperparameters are optimized to empirical values
- Optimization of the hyperparameters can be done using data of all tasks

Following the approach from Bart Bakker and Tom Heskes. Task Clustering and Gating for Bayesian Multitask Learning. Journal of Machine Learning Research 4, 83-99, 2003.. Images from that paper.

Multitask learning with task clustering

- Given the maximum likelihood hyperparameters Λ^* , **maximum a posteriori values** of task-specific parameters can be computed using the distribution $P(\mathbf{A}_i|D_i,\Lambda^*)$:

$$\tilde{\mathbf{A}}_i = \underset{\mathbf{A}_i}{\operatorname{argmax}} P(\mathbf{A}_i|D_i,\Lambda^*)$$

- The prior $\mathbf{A}_i \sim N(\mathbf{A}_i|\mathbf{m},\Sigma)$ says all tasks are equally similar a priori
- How can we use priors to allow more complicated task relationships?
- **Possibility 1: use a task-dependent prior mean.** Assume some "features $\mathbf{f}_i$ describing each task i" are known beforehand.
- Then we can set $\mathbf{A}_i \sim N(\mathbf{A}_i|\mathbf{m}_i,\Sigma)$ where $\mathbf{m}_i = M\mathbf{f}_i$ and M is some matrix. Likelihoods are same as before but now using $\mathbf{m}_i$ as the mean.

Following the approach from Bart Bakker and Tom Heskes. Task Clustering and Gating for Bayesian Multitask Learning. Journal of Machine Learning Research 4, 83-99, 2003.. Images from that paper.

Multitask learning with task clustering

- **Possibility 2: clustering of tasks.** Set a mixture distribution as the prior:

$$\mathbf{A}_i \sim \sum_{\alpha=1}^{n_{\text{cluster}}} q_\alpha N(\mathbf{m}_\alpha, \Sigma_\alpha)$$

- Now in each task the parameters can come from one of several Gaussian clusters (probability q_α to take cluster α) with their own means and covariances.
- A priori we don't know which cluster a task will come from, but a posteriori (after seeing the data) we can compute the posterior probability.
- Posterior data likelihood in task i given hyperparameters (again integrated over task-specific parameters) :

$$P(D_i|\Lambda) = \int d\mathbf{A}_i P(D_i|\mathbf{A}_i, \Lambda) \sum_{\alpha=1}^{n_{\text{cluster}}} q_\alpha P(\mathbf{A}_i|\mathbf{m}_\alpha, \Sigma_\alpha)$$

Following the approach from Bart Bakker and Tom Heskes. Task Clustering and Gating for Bayesian Multitask Learning. Journal of Machine Learning Research 4, 83-99, 2003.. Images from that paper.

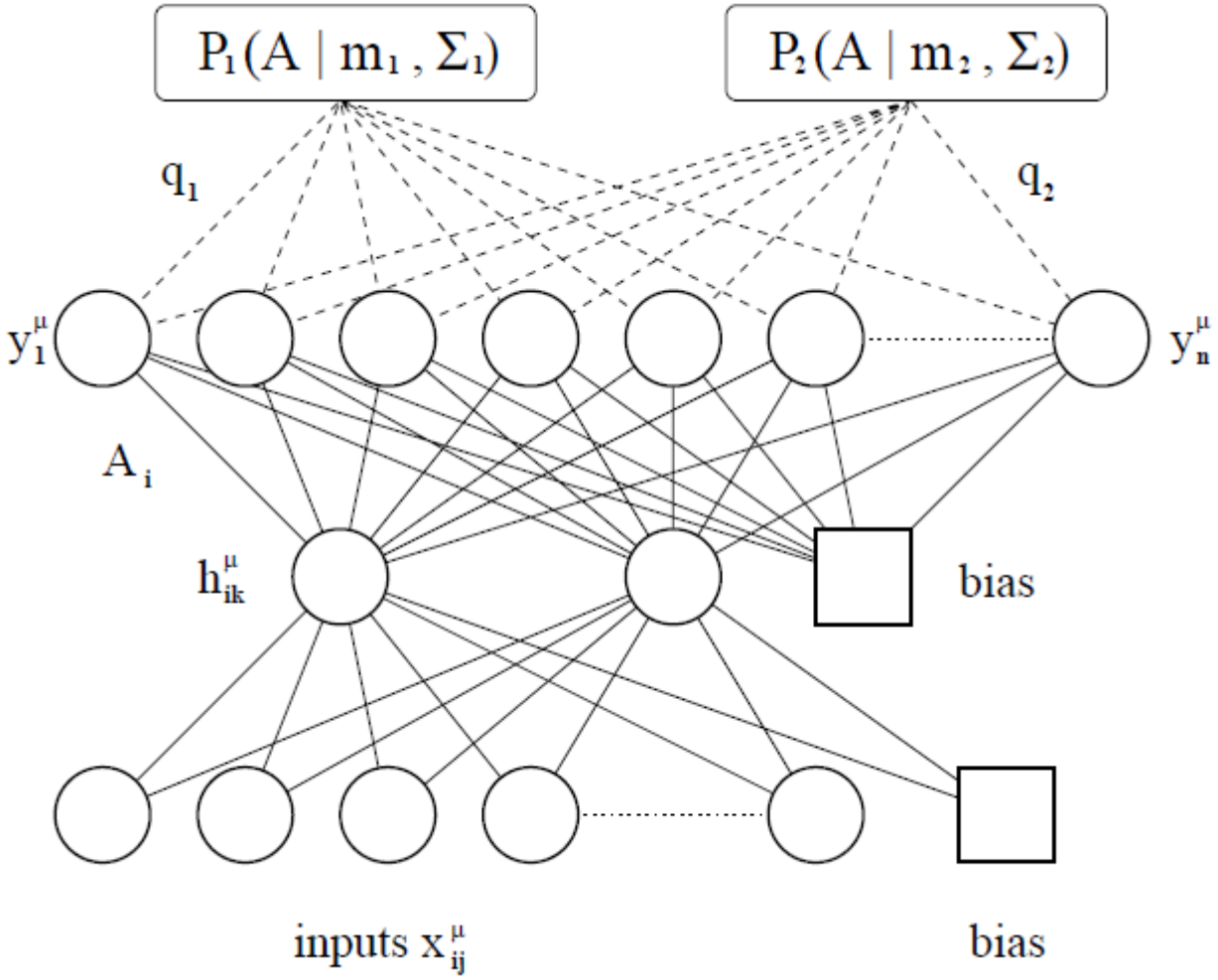
Multitask learning with task clustering

- The posterior data likelihood in task i comes from integrating over parameters $\mathbf{A}_i$ that 1) yield good probabilities for the observations and 2) have high probability in the prior under at least one of the task-clusters.
- As a result, the posterior distribution "assigns" each task to the cluster that is most compatible with the data of the task; incompatible clusters contribute little to the posterior likelihood.
- Hyperparameters now include parameters of all clusters:
  $\Lambda = \{ W, \sigma, \{ q_\alpha, \boldsymbol{m}_\alpha, \Sigma_\alpha \}_{\alpha=1}^{n_{cluster}} \}$
- The hyperparameters can be optimized to maximize the posterior likelihood, using **expectation maximization method.**
- Idea: introduce binary variables $z_{i\alpha}$ telling for each task i which cluster α it comes from. Compute in the "E-step" the distribution $P(z|D,\Lambda_n)$ of these variables given all others; in the "M-step" take the expectation of the likelihood over $P(z|D,\Lambda_n)$, optimize all other variables with respect to the expectation; repeat.

Following the approach from Bart Bakker and Tom Heskes. Task Clustering and Gating for Bayesian Multitask Learning. Journal of Machine Learning Research 4, 83-99, 2003.. Images from that paper.

Multitask learning with task clustering

- Graphical representation:



Following the approach from Bart Bakker and Tom Heskes. Task Clustering and Gating for Bayesian Multitask Learning. Journal of Machine Learning Research 4, 83-99, 2003.. Images from that paper.

Multitask learning with task clustering

- **Possibility 3: integrate the concept of "task features" and task clustering -----> task gating.**
- Idea: if we know descriptive features of each task i, we can use that information to decide which cluster each task comes from, rather than having all clusters be equally likely a priori.
- Define cluster probabilities separately for each task using $\mathbf{f}_i$:

$$q_{i\alpha} = \mathrm{e}^{\mathbf{U}_\alpha^T \mathbf{f}_i} / \sum_{\alpha'} \mathrm{e}^{\mathbf{U}_{\alpha'}^T \mathbf{f}_i}$$

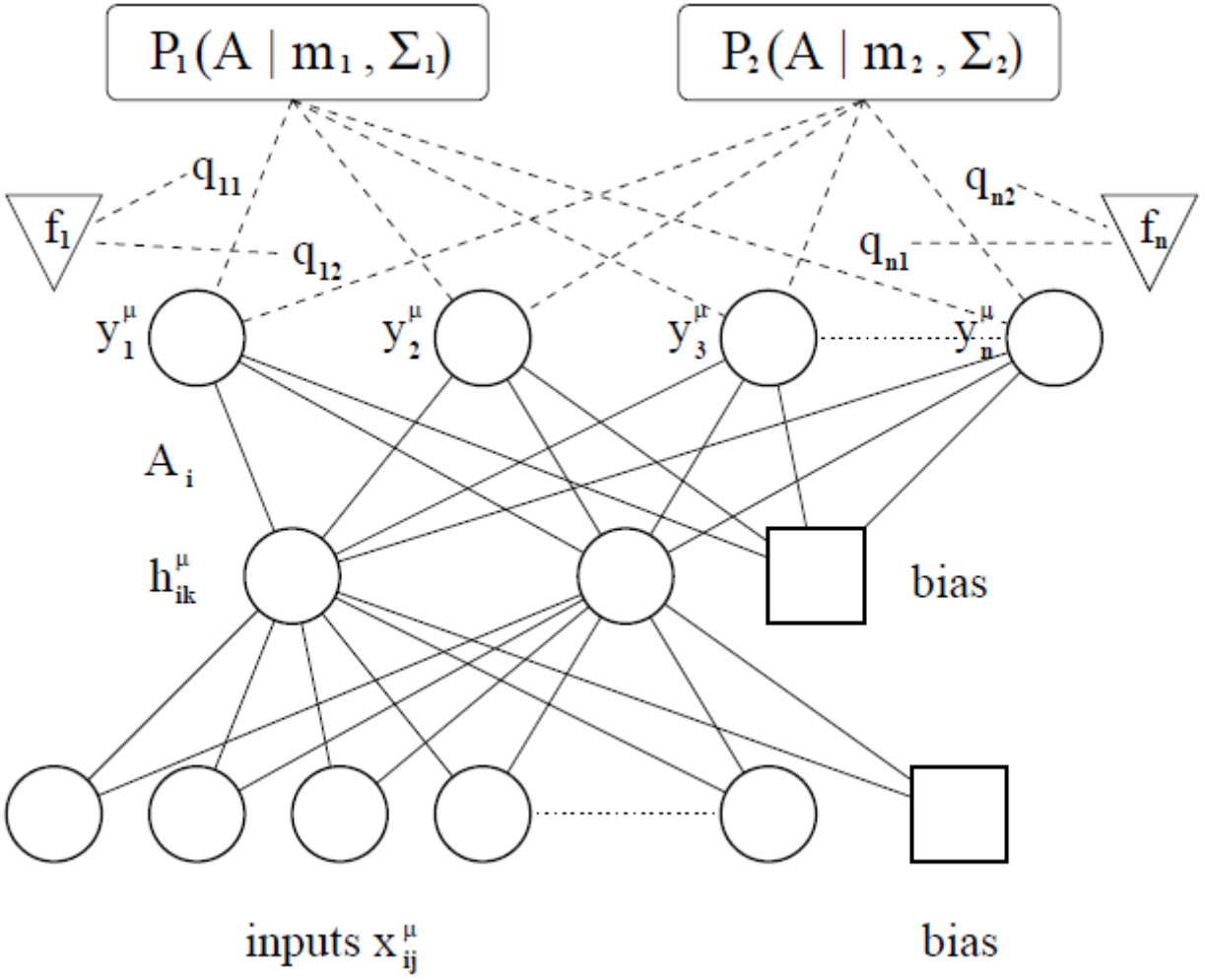
where $\mathbf{U}_\alpha$ is a hyperparameter vector for each cluster, to be learned from data

- Otherwise, follow a similar approach as in possibility 2; we again end up with an expectation maximization algorithm to learn the hyperparameters (see the paper reference for detail).

Following the approach from Bart Bakker and Tom Heskes. Task Clustering and Gating for Bayesian Multitask Learning. Journal of Machine Learning Research 4, 83-99, 2003.. Images from that paper.

Multitask learning with task clustering

- Graphical representation:



Part 2: Multitask learning with task clustering, without a fixed number of clusters

Following the approach from Xue, Y., Liao, X., Carin, L., and Krishnapuram, B. 2007. Multi-Task Learning for Classification with Dirichlet Process Priors. Journal of Machine Learning Research, 8: 35- 63. Images from that paper.

Multitask learning with nonparametric clustering

- The previous approach assumed there was a pre-specified known number of clusters that tasks could come from.
- We now consider an approach where the number of clusters it itself learned from data
- The previous approach used a neural network model with Gaussian observation noise (= regression tasks)
- We now use a logistic regression approach (= classification tasks)

Following the approach from Xue, Y., Liao, X., Carin, L., and Krishnapuram, B. 2007. Multi-Task Learning for Classification with Dirichlet Process Priors. Journal of Machine Learning Research, 8: 35- 63. Images from that paper.

Multitask learning with nonparametric clustering

- Individual tasks are here logistic regression classifier models, with task-specific parameters.
- As in the previous approach, tasks are connected by a common prior placed on those parameters
- Data of all tasks are used to learn the common prior, enabling information sharing among tasks
- Given the prior, models of individual tasks are learned independently.
- The model learned for each tasks is influenced by its own training data, and through the prior, also by data of other tasks.
- Previously the prior had a parametric form with hyperparameters, information transfer was done by learning the hyperparameters.
- Specifying a good parametric form may be hard beforehand.
- Here we learn the form of the prior from data as a nonparametric Dirichlet process.

Following the approach from Xue, Y., Liao, X., Carin, L., and Krishnapuram, B. 2007. Multi-Task Learning for Classification with Dirichlet Process Priors. Journal of Machine Learning Research, 8: 35- 63. Images from that paper.

Multitask learning with nonparametric clustering

- Multitask logistic regression setup:
- Notation: tasks m=1,...,M
  - data of each task $\mathcal{D}_m = \{(x_{m,n}, y_{m,n}) : n = 1, \cdots, N_m\}$
  - Each sample: $(x_{m,n}, y_{m,n})$
  - multidimensional input features $x_{m,n} \in \mathbb{R}^d$
  - binary output labels $y_{m,n} \in \{0, 1\}$
  - Probability of output label in task m:

$$p(y_{m,n}|w_m, x_{m,n}) = \sigma(w_m^T x_{m,n})^{y_{m,n}} [1 - \sigma(w_m^T x_{m,n})]^{1-y_{m,n}}$$

where the logistic nonlinearity is $\sigma(x) = \frac{1}{1+\exp(-x)}$

  - Task parameters: logistic regression weight vector w_m
  - We want to set a cluster-based prior for the weight vectors

Following the approach from Xue, Y., Liao, X., Carin, L., and Krishnapuram, B. 2007. Multi-Task Learning for Classification with Dirichlet Process Priors. Journal of Machine Learning Research, 8: 35- 63. Images from that paper.

Multitask learning with nonparametric clustering

- For the weight vectors, we could specify a cluster prior for example as a mixture of a fixed number of Gaussians, and fit the hyperparameters (mixture component probabilities, means and covariances) to the data.
- However, here we do not want to specify the number of clusters beforehand, so we use a **nonparametric prior.**
- K-nearest neighbor classifier is a simple nonparametric model.
- A **Dirichlet process** is a nonparametric model for clustering: it creates a probability distribution for choices of clusters from an unknown, potentially infinite number of possible clusters.
- Here the clusters are clusters of logistic regression parameters
- Each cluster has a probability to choose this cluster: the Dirichlet process generates these probabilities.
- Each cluster generates only a single parameter value. To get different values, we take different clusters.

Following the approach from Xue, Y., Liao, X., Carin, L., and Krishnapuram, B. 2007. Multi-Task Learning for Classification with Dirichlet Process Priors. Journal of Machine Learning Research, 8: 35- 63. Images from that paper.

Multitask learning with nonparametric clustering

- A Dirichlet process can be described as a "**Chinese restaurant process**", where buffet tables are the possible clusters, and samples (tasks) are customers that decide to sit at one of the tables.
- Each customer chooses to either:
  - sit at an already occupied table: the more people at the table, the higher the probability to take that table;
  - or the customer chooses a new table, thus increasing the number of occupied tables
- Each table then generates the parameters of the task (food dish of the customer) in some way specific to that table.

Following the approach from Xue, Y., Liao, X., Carin, L., and Krishnapuram, B. 2007. Multi-Task Learning for Classification with Dirichlet Process Priors. Journal of Machine Learning Research, 8: 35- 63. Images from that paper.

Multitask learning with nonparametric clustering

- In equations: denote parameters of task m by w_m . Denote the Dirichlet process by G. Then

$$w_m | G \sim G,$$ Parameters of tasks come from a common distribution G

$$G \sim DP(\alpha, G_0)$$ G is drawn from a Dirichlet process defined by a concentration parameter alpha and a "base distribution" G0

- It is possible to integrate out G. Then the probability of w_m given known values of parameters of the other tasks $w_{-m} = \{w_1, \cdots, w_{m-1}, w_{m+1}, \cdots, w_M\}$ becomes

$$p(w_m | w_{-m}, \alpha, G_0) = \frac{\alpha}{M-1+\alpha} G_0 + \frac{1}{M-1+\alpha} \sum_{j=1, j\neq m}^{M} \delta_{w_j}$$

Choose a new parameter value from the base distribution

Choose one of the parameter values of the other tasks

where δ_{w_j} is a delta distribution concentrated around the parameter value of task j

- Note that several tasks may have the same parameter value, so the number of unique values may be less than n. of tasks

Following the approach from Xue, Y., Liao, X., Carin, L., and Krishnapuram, B. 2007. Multi-Task Learning for Classification with Dirichlet Process Priors. Journal of Machine Learning Research, 8: 35- 63. Images from that paper.

Multitask learning with nonparametric clustering

- The probability can be rewritten as

$$p(w_m|w_{-m},\alpha,G_0)=\frac{\alpha}{M-1+\alpha}G_0+\frac{1}{M-1+\alpha}\sum_{k=1}^{K}n_{-m,k}\delta_{w_k^*}$$

  where w_k^* are the unique parameter values among the tasks. The probability to choose each of the unique values *k* depends on $n_{-m,k}$, the number of other tasks that have chosen it.

- The Dirichlet process model implicitly clusters samples into groups, since a new sample prefers to take a value chosen by many other samples.
- The process can create new clusters if new values are needed to fit the data.
- The probability to create new values is governed by α , larger values yield more new clusters. If α goes to infinity each sample (task) gets its own parameter value---->single-task learning. If α goes to zero, all tasks use the same parameter ----> naive pooling of all tasks.

Following the approach from Xue, Y., Liao, X., Carin, L., and Krishnapuram, B. 2007. Multi-Task Learning for Classification with Dirichlet Process Priors. Journal of Machine Learning Research, 8: 35- 63. Images from that paper.

Multitask learning with nonparametric clustering

- Properties of the Dirichlet process:
  - The "base distribution" is an abstract entity. In this case it is uniform over the space of all possible parameter values.
  - When a task "chooses a new parameter value from the base distribution", it is like allocating a new free variable for the parameter value, other tasks may then choose to use the same variable. The actual value of the variable is then learned from the data of all tasks that chose that variable.
  - Given observed parameter values so far, the posterior of the Dirichlet process is another Dirichlet process, where the new "base distribution" has peaks at observed parameter values:

$$p(G|w_1,\ldots,w_M,\alpha,G_0) = DP(\alpha+M, \tfrac{\alpha}{M+\alpha}G_0 + \tfrac{1}{M+\alpha}\sum_{j=1}^{M}\delta_{w_j})$$

Following the approach from Xue, Y., Liao, X., Carin, L., and Krishnapuram, B. 2007. Multi-Task Learning for Classification with Dirichlet Process Priors. Journal of Machine Learning Research, 8: 35- 63. Images from that paper.

Multitask learning with nonparametric clustering

- Properties of the Dirichlet process
  - The process can also be written over the set of unique

$$G = \sum_{k=1}^{\infty} \pi_k \delta_{w_k^*} \qquad \pi_k = v_k \prod_{i=1}^{k-1} (1 - v_i) \qquad v_k \sim Be(1, \alpha)$$

A probability peak around each unique value

Probability to choose k:th unique value is essentially exponentially decreasing in k.

Stick-breaking parameters *v*
Amount of probability decrease between unique values is governed by alpha

Following the approach from Xue, Y., Liao, X., Carin, L., and Krishnapuram, B. 2007. Multi-Task Learning for Classification with Dirichlet Process Priors. Journal of Machine Learning Research, 8: 35- 63. Images from that paper.

Multitask learning with nonparametric clustering

- Full generative process of parameters and data in all tasks:

SMTL Model. Given the parameters α, μ and Σ,

1. Draw v_k from the Beta distribution $Be(1,\alpha)$ and independently draw w_k^* from the base distribution $N_d(\mu,\Sigma), k=1,\ldots,\infty$.
2. $\pi_k = v_k \prod_{i=1}^{k-1}(1-v_i), k=1,\ldots,\infty$.
3. Draw the indicators $(c_{m,1},\ldots,c_{m,\infty})$ from a multinomial distribution $M_\infty(1;\pi_1,\ldots,\pi_\infty), m=1,\ldots,M$
4. $w_m = \prod_{k=1}^{\infty}(w_k^*)^{c_{m,k}}$, or in an equivalent form $w_m = \sum_{k=1}^{\infty} c_{m,k} w_k^*, m=1,\ldots,M$.
5. Draw $y_{m,n}$ from a Binomial distribution $B(1,\sigma(w_m^T x_{m,n})), m=1,\ldots,M, n=1,\cdots,N_m$.

- Given the generative process, and given sets of observed samples $\mathcal{D}_m = \{(x_{m,n}, y_{m,n}) : n = 1,\cdots,N_m\}$ for m=1,...,M, a posterior distribution for parameters w_m given hyperparameters can be inferred by Markov chain Monte Carlo sampling or by variational inference (=posterior approximation)

Following the approach from Xue, Y., Liao, X., Carin, L., and Krishnapuram, B. 2007. Multi-Task Learning for Classification with Dirichlet Process Priors. Journal of Machine Learning Research, 8: 35- 63. Images from that paper.

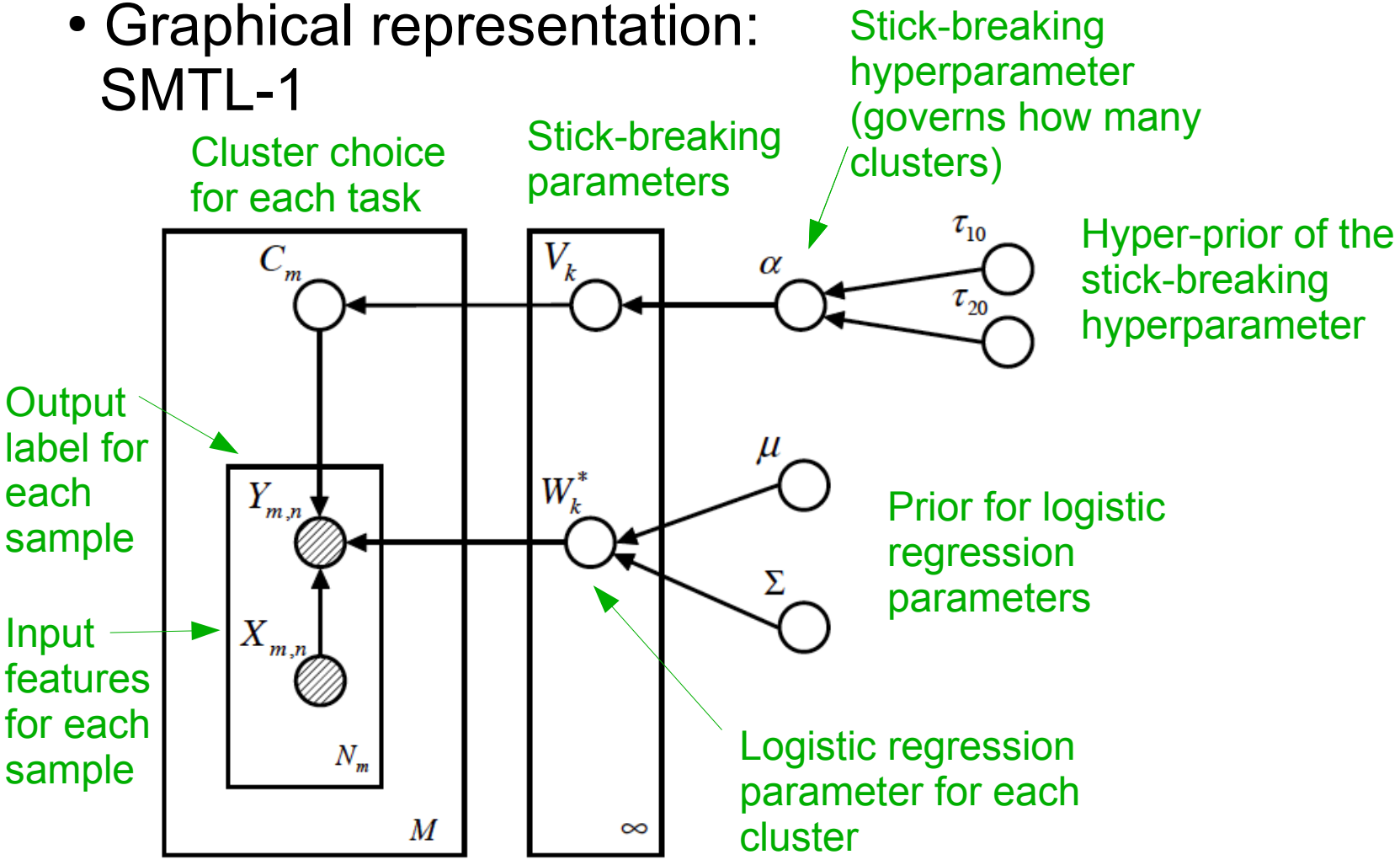
Multitask learning with nonparametric clustering

- Instead of using fixed values of hyperparameters, it is possible to integrate them over a prior of their own
- **Version 1 (SMTL-1):** integrate only alpha over a Gamma prior $Ga(\tau_{10}, \tau_{20})$ whose parameters are τ_{10}, τ_{20} . Generative process: first draw alpha from the prior, then use the previous generative process.
- **Version 1 (SMTL-2):** integrate also over a prior for μ and Σ . Draw Σ as an inverse of a diagonal matrix Λ with entries entries $\lambda_1, \cdots, \lambda_d$ on the diagonal. Draw μ from a Gaussian distribution with the same covariance multiplied by a scalar β_0.
  - Draw each of $\lambda_1, \cdots, \lambda_d$ from a Gamma distribution $Ga(\gamma_{10}, \gamma_{20})$
  - Draw μ from $N_d(\mathbf{0}, (\beta_0 \Lambda)^{-1})$
  - Draw α from $Ga(\tau_{10}, \tau_{20})$ as in version 1
  - Then use the previous generative process

Following the approach from Xue, Y., Liao, X., Carin, L., and Krishnapuram, B. 2007. Multi-Task Learning for Classification with Dirichlet Process Priors. Journal of Machine Learning Research, 8: 35- 63. Images from that paper.

Multitask learning with nonparametric clustering

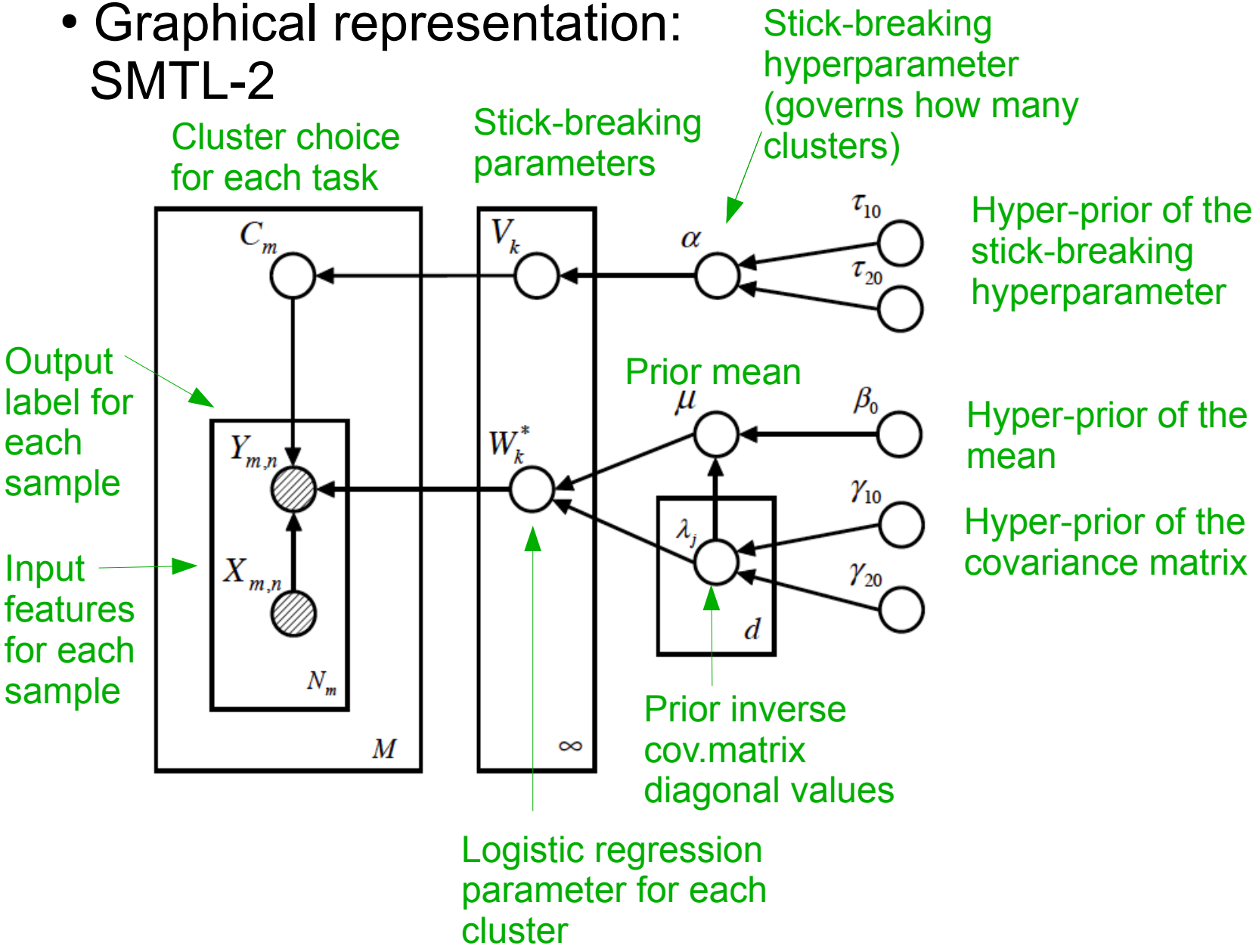
- Graphical representation: SMTL-1



Following the approach from Xue, Y., Liao, X., Carin, L., and Krishnapuram, B. 2007. Multi-Task Learning for Classification with Dirichlet Process Priors. Journal of Machine Learning Research, 8: 35- 63. Images from that paper.

Multitask learning with nonparametric clustering

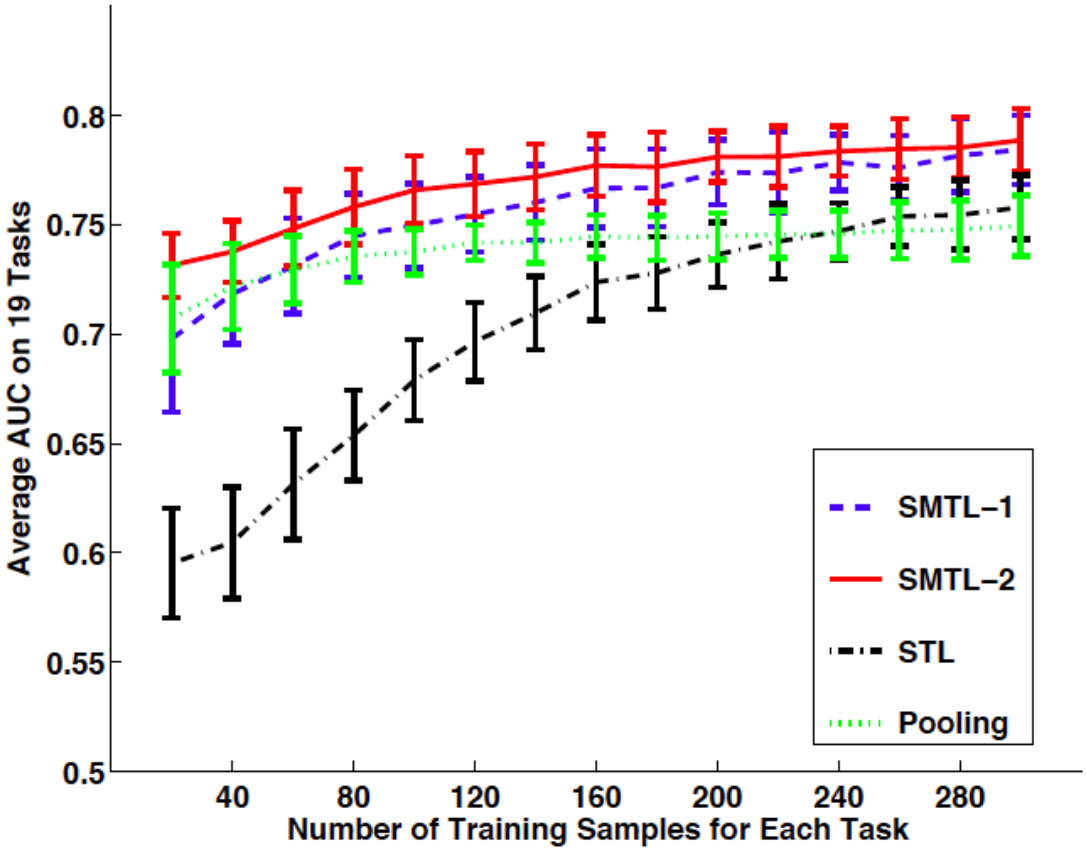
- Graphical representation: SMTL-2



Following the approach from Xue, Y., Liao, X., Carin, L., and Krishnapuram, B. 2007. Multi-Task Learning for Classification with Dirichlet Process Priors. Journal of Machine Learning Research, 8: 35- 63. Images from that paper.

Multitask learning with nonparametric clustering

- Experiment: landmine detection
- In 29 fields (= tasks), classify a piece of land as clutter or a landmine based on 9-dimensional features extracted from radar images.
- Fields 1-15 are highly foliated, 16-29 are bare earth or desert ----> likely to be about two clusters of tasks
- SMTL-1 and SMTL-2 find roughly the expected clustering



- Data available at

http://www.ee.duke.edu/lcarin/LandmineData.zip

Part 3: More variants of multitask priors

Following the approach from Zhang, J., Ghahramani, Z., and Yang, Y. 2008. Flexible Latent Variable Models for Multitask Learning. Machine Learning, 73(3):221-242.Images from that paper.

Multitask learning with various parameter priors

- We again use the setup of multitask (logistic) regression, but consider different kinds of priors than before
- Notation:
  - K tasks, each with training set
    $\mathcal{D}^{(k)} = \{(\mathbf{x}_1^{(k)}, y_1^{(k)}), \ldots, (\mathbf{x}_{n_k}^{(k)}, y_{n_k}^{(k)})\} \quad (k = 1, \ldots, K)$
  - Logistic regression predictors to be estimated:
    $\hat{f}^{(k)} \ (k = 1, \ldots, K)$
  - Vector notation for data:
    $\mathcal{D}^{(k)} = \{\mathbf{X}^{(k)}, \mathbf{y}^{(k)}\}$ where $\mathbf{X}^{(k)} = \begin{bmatrix} \mathbf{x}_1^{(k)} \\ \vdots \\ \mathbf{x}_{n_k}^{(k)} \end{bmatrix} \in \mathbb{R}^{n_k \times F}, \ \mathbf{y}^{(k)} = \begin{bmatrix} y_1^{(k)} \\ \vdots \\ y_{n_k}^{(k)} \end{bmatrix} \in \mathbb{R}^{n_k \times 1}.$
  - Task types:
    $\text{regression}: \ y_i^{(k)} \sim \text{Normal}(\langle \boldsymbol{\theta}^{(k)}, \mathbf{x}_i^{(k)} \rangle, \sigma^2)$
    $\text{classification}: \ y_i^{(k)} \sim \text{Bernoulli}(g(\langle \boldsymbol{\theta}^{(k)}, \mathbf{x}_i^{(k)} \rangle)) \quad g(t) = (1 + \exp(-t))^{-1}$
    (logistic regression) logistic function
  - Task parameters to be estimated: $\hat{\boldsymbol{\theta}}^{(k)}$

Following the approach from Zhang, J., Ghahramani, Z., and Yang, Y. 2008. Flexible Latent Variable Models for Multitask Learning. Machine Learning, 73(3):221-242.Images from that paper.

Multitask learning with various parameter priors

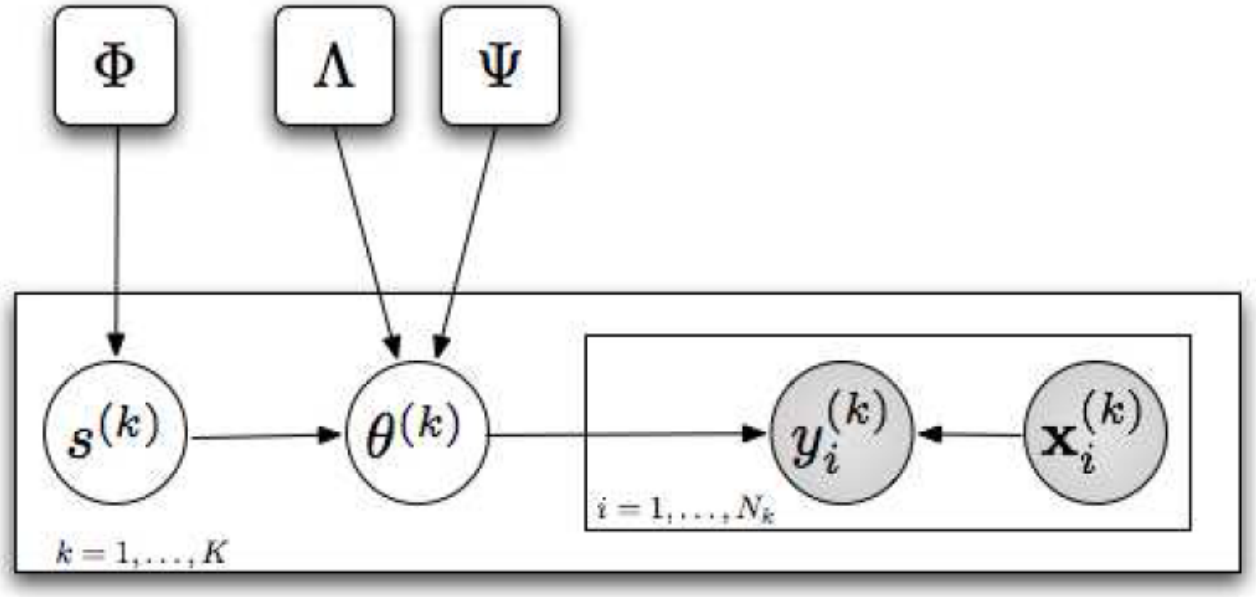
- Priors for the task parameters: of the form

$$\boldsymbol{\theta}^{(k)} = \boldsymbol{\Lambda}\mathbf{s}^{(k)} + \mathbf{e}^{(k)}$$

$$\mathbf{s}^{(1)}, \ldots, \mathbf{s}^{(K)} \sim p(\mathbf{s}^{(1)}, \ldots, \mathbf{s}^{(K)} | \boldsymbol{\Phi})$$

$$\mathbf{e}^{(k)} \sim \text{Normal}(\mathbf{0}, \boldsymbol{\Psi})$$

- Graphical model:



Following the approach from Zhang, J., Ghahramani, Z., and Yang, Y. 2008. Flexible Latent Variable Models for Multitask Learning. Machine Learning, 73(3):221-242.Images from that paper.

Multitask learning with various parameter priors

- Multiple variants of the priors following the same structure

$$\boldsymbol{\theta}^{(k)} = \boldsymbol{\Lambda}\mathbf{s}^{(k)} + \mathbf{e}^{(k)}$$
$$\mathbf{s}^{(1)}, \ldots, \mathbf{s}^{(K)} \sim p(\mathbf{s}^{(1)}, \ldots, \mathbf{s}^{(K)} | \boldsymbol{\Phi})$$
$$\mathbf{e}^{(k)} \sim \text{Normal}(\mathbf{0}, \boldsymbol{\Psi})$$

- Independent tasks: $\boldsymbol{\theta}^{(k)} = \mathbf{e}^{(k)} \sim \text{Normal}(\mathbf{0}, \boldsymbol{\Psi})$
  (ignores task relationships)

- Noisy tasks: $\boldsymbol{\theta}^{(k)} = \boldsymbol{\mu} + \mathbf{e}^{(k)} \sim \text{Normal}(\boldsymbol{\mu}, \boldsymbol{\Psi})$
  (task similarities come from learning the shared mean)

- Task clusters: $\mathbf{s}^{(k)} \sim \text{Multinomial}(1; p_1, p_2, \ldots, p_H)$
  (task mean depends on cluster; fixed number of clusters with probabilities p1...pH to pick each cluster)

Following the approach from Zhang, J., Ghahramani, Z., and Yang, Y. 2008. Flexible Latent Variable Models for Multitask Learning. Machine Learning, 73(3):221-242.Images from that paper.

Multitask learning with various parameter priors

- Multiple variants of the priors following the same structure

$$\boldsymbol{\theta}^{(k)} = \boldsymbol{\Lambda}\mathbf{s}^{(k)} + \mathbf{e}^{(k)}$$
$$\mathbf{s}^{(1)}, \ldots, \mathbf{s}^{(K)} \sim p(\mathbf{s}^{(1)}, \ldots, \mathbf{s}^{(K)} | \boldsymbol{\Phi})$$
$$\mathbf{e}^{(k)} \sim \text{Normal}(\mathbf{0}, \boldsymbol{\Psi})$$

- Independent tasks: $\boldsymbol{\theta}^{(k)} = \mathbf{e}^{(k)} \sim \text{Normal}(\mathbf{0}, \boldsymbol{\Psi})$
  (ignores task relationships)

- Noisy tasks: $\boldsymbol{\theta}^{(k)} = \boldsymbol{\mu} + \mathbf{e}^{(k)} \sim \text{Normal}(\boldsymbol{\mu}, \boldsymbol{\Psi})$
  (task similarities come from learning the shared mean)

- Task clusters: $\mathbf{s}^{(k)} \sim \text{Multinomial}(1; p_1, p_2, \ldots, p_H)$
  (task mean depends on cluster; fixed number of clusters with probabilities p1...pH to pick each cluster)

Following the approach from Zhang, J., Ghahramani, Z., and Yang, Y. 2008. Flexible Latent Variable Models for Multitask Learning. Machine Learning, 73(3):221-242.Images from that paper.

Multitask learning with various parameter priors

- Multiple variants of the priors following the same structure

$$\boldsymbol{\theta}^{(k)} = \boldsymbol{\Lambda}\mathbf{s}^{(k)} + \mathbf{e}^{(k)}$$
$$\mathbf{s}^{(1)}, \ldots, \mathbf{s}^{(K)} \sim p(\mathbf{s}^{(1)}, \ldots, \mathbf{s}^{(K)}|\boldsymbol{\Phi})$$
$$\mathbf{e}^{(k)} \sim \text{Normal}(\mathbf{0}, \boldsymbol{\Psi})$$

- Tasks sharing a linear subspace: $\mathbf{s}^{(k)} \sim \text{Normal}(\mathbf{0}, \mathbf{I})$ (variation of the task mean is generated along a smaller dimensional space of the now vector-valued "s" parameters, and then the projected to the final task parameter space by the Lambda matrix)
- Tasks with a sparse representation: $\mathbf{s}^{(k)} \sim \prod_{h=1}^{H} \text{Laplace}(0, 1)$ (same as before but now most comp in the linear subspace are likely to be close to zero)
- Alternative sparse representation: "s" from a Normal distribution but the colums of the projection matrix Lambda from a sparse distribution:

Following the approach from Zhang, J., Ghahramani, Z., and Yang, Y. 2008. Flexible Latent Variable Models for Multitask Learning. Machine Learning, 73(3):221-242.Images from that paper.

Multitask learning with various parameter priors

- Multiple variants of the priors following the same structure

$$\boldsymbol{\theta}^{(k)} = \boldsymbol{\Lambda}\mathbf{s}^{(k)} + \mathbf{e}^{(k)}$$
$$\mathbf{s}^{(1)}, \ldots, \mathbf{s}^{(K)} \sim p(\mathbf{s}^{(1)}, \ldots, \mathbf{s}^{(K)}|\boldsymbol{\Phi})$$
$$\mathbf{e}^{(k)} \sim \text{Normal}(\mathbf{0}, \boldsymbol{\Psi})$$

- Alternative sparse representation: “s” from a Normal distribution $\mathbf{s}^{(k)} \sim \text{Normal}(\mathbf{0}, \mathbf{I})$ but the colums of the projection matrix Lambda from a sparse distribution:

$$\boldsymbol{\lambda}_h \sim \prod_{f=1}^{F} \text{Laplace}(0, \eta)$$

- Duplicated tasks: use the Dirichlet process representation discussed before. In the current notation:

$$G \sim \text{DP}(\alpha, G_0)$$
$$\mathbf{s}^{(k)} \sim G$$

Following the approach from Zhang, J., Ghahramani, Z., and Yang, Y. 2008. Flexible Latent Variable Models for Multitask Learning. Machine Learning, 73(3):221-242.Images from that paper.

Multitask learning with various parameter priors

- Multiple variants of the priors following the same structure

$$\boldsymbol{\theta}^{(k)} = \boldsymbol{\Lambda}\mathbf{s}^{(k)} + \mathbf{e}^{(k)}$$
$$\mathbf{s}^{(1)}, \ldots, \mathbf{s}^{(K)} \sim p(\mathbf{s}^{(1)}, \ldots, \mathbf{s}^{(K)} | \boldsymbol{\Phi})$$
$$\mathbf{e}^{(k)} \sim \mathsf{Normal}(\mathbf{0}, \boldsymbol{\Psi})$$

- Evolving tasks: assume each task is related to the previous one, put e.g. some kind of Markov model for how the new parameters are related to the previous ones:

$$\mathbf{s}^{(k-1)} \longrightarrow \mathbf{s}^{(k)}$$

- For each case, an Empirical Bayes (expectation maximization style) inference algorithm is possible, see the paper for details

References

- Bart Bakker and Tom Heskes. Task Clustering and Gating for Bayesian Multitask Learning. Journal of Machine Learning Research 4, 83-99, 2003. http://jmlr.csail.mit.edu/papers/volume4/bakker03a/bakker03a.pdf

- Xue, Y., Liao, X., Carin, L., and Krishnapuram, B. 2007. Multi-Task Learning for Classification with Dirichlet Process Priors. Journal of Machine Learning Research, 8: 35- 63. http://www.jmlr.org/papers/volume8/xue07a/xue07a.pdf

- Zhang, J., Ghahramani, Z., and Yang, Y. 2008. Flexible Latent Variable Models for Multitask Learning. Machine Learning, 73(3):221-242.